INTERNATIONAL FOURTH EDITION



Modern Digital and Analog Communication Systems

B.P. Lathi · Zhi Ding

OXFORD

MODERN DIGITAL AND ANALOG COMMUNICATION SYSTEMS

International Fourth Edition

B. P. Lathi
Professor Emeritus
California State University—Sacramento

Zhi Ding
Professor
University of California—Davis

New York Oxford
OXFORD UNIVERSITY PRESS
2010

Oxford University Press, Inc., publishes works that further Oxford University's objective of excellence in research, scholarship, and education.

Oxford New York
Auckland Cape Town Dar es Salaam Hong Kong Karachi
Kuala Lumpur Madrid Melbourne Mexico City Nairobi
New Delhi Shanghai Taipei Toronto

With offices in
Argentina Austria Brazil Chile Czech Republic France Greece
Guatemala Hungary Italy Japan Poland Portugal Singapore
South Korea Switzerland Thailand Turkey Ukraine Vietnam

Copyright © 1983 by CBS College Publishing; © 1989 by B. P. Lathi & Saunders College Publishing, a division of Holt, Rinehart, and Winston, Inc.; © 1995, 1998, 2010 by B. P. Lathi

Published by Oxford University Press, Inc.
198 Madison Avenue, New York, New York 10016
http://www.oup.com

Oxford is a registered trademark of Oxford University Press

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of Oxford University Press.

Library of Congress Cataloging-in-Publication Data
Lathi, B. P. (Bhagwandas Pannalal)
Modern digital and analog communication systems / B. P. Lathi, Zhi Ding.—4th ed.
p. cm.
ISBN 978-0-19-538493-2 (hardcover : alk. paper)
1. Telecommunication systems. 2. Digital communications.
3. Statistical communication theory. I. Ding, Zhi, 1962– II. Title.
TK5101.L333 2008
621.382—dc22 2008029440

Printing number: 9 8 7 6 5 4 3 2 1

Printed in the United States of America
on acid-free paper

BRIEF TABLE OF CONTENTS

Preface xvii

1 Introduction 1

2 Signals and Signal Space 20

3 Analysis and Transmission of Signals 62

4 Amplitude Modulations and Demodulations 140

5 Angle Modulation and Demodulation 202

6 Sampling and Analog-to-Digital Conversion 251

7 Principles of Digital Data Transmission 326

8 Fundamentals of Probability Theory 393

9 Random Processes and Spectral Analysis 456

10 Performance Analysis of Digital Communication Systems 506

11 Spread Spectrum Communications 614

12 Digital Communications Under Linearly Distortive Channels 666

13 Introduction to Information Theory 734

14 Error Correcting Codes 802

A Orthogonality of Some Signal Sets 873

B Cauchy-Schwarz Inequality 875

C Gram-Schmidt Orthogonalization of a Vector Set 877

D Basic Matrix Properties and Operations 880

E Miscellaneous 885

Index 889

CONTENTS

PREFACE xvii

1 INTRODUCTION 1

1.1 COMMUNICATION SYSTEMS 1

1.2 ANALOG AND DIGITAL MESSAGES 4
- 1.2.1 Noise Immunity of Digital Signals 4
- 1.2.2 Viability of Distortionless Regenerative Repeaters 5
- 1.2.3 Analog-to-Digital (A/D) Conversion 6
- 1.2.4 Pulse-Coded Modulation—A Digital Representation 7

1.3 CHANNEL EFFECT, SIGNAL-TO-NOISE RATIO, AND CAPACITY 9
- 1.3.1 Signal Bandwidth and Power 9
- 1.3.2 Channel Capacity and Data Rate 10

1.4 MODULATION AND DETECTION 11
- 1.4.1 Ease of Radiation/Transmission 11
- 1.4.2 Simultaneous Transmission of Multiple Signals—Multiplexing 12
- 1.4.3 Demodulation 13

1.5 DIGITAL SOURCE CODING AND ERROR CORRECTION CODING 13

1.6 A BRIEF HISTORICAL REVIEW OF MODERN TELECOMMUNICATIONS 15

2 SIGNALS AND SIGNAL SPACE 20

2.1 SIZE OF A SIGNAL 20

2.2 CLASSIFICATION OF SIGNALS 22
- 2.2.1 Continuous Time and Discrete Time Signals 23
- 2.2.2 Analog and Digital Signals 23
- 2.2.3 Periodic and Aperiodic Signals 24
- 2.2.4 Energy and Power Signals 25
- 2.2.5 Deterministic and Random Signals 25

2.3 UNIT IMPULSE SIGNAL 26

2.4 SIGNALS VERSUS VECTORS 28
- 2.4.1 Component of a Vector along Another Vector 28
- 2.4.2 Decomposition of a Signal and Signal Components 30
- 2.4.3 Complex Signal Space and Orthogonality 32
- 2.4.4 Energy of the Sum of Orthogonal Signals 34

2.5 CORRELATION OF SIGNALS 34
- 2.5.1 Correlation Functions 35
- 2.5.2 Autocorrelation Function 36

2.6 ORTHOGONAL SIGNAL SET 36
- 2.6.1 Orthogonal Vector Space 36
- 2.6.2 Orthogonal Signal Space 38
- 2.6.3 Parseval's Theorem 39

2.7 THE EXPONENTIAL FOURIER SERIES 39

2.8 MATLAB EXERCISES 46

3 ANALYSIS AND TRANSMISSION OF SIGNALS 62

3.1 APERIODIC SIGNAL REPRESENTATION BY FOURIER INTEGRAL 62

3.2 TRANSFORMS OF SOME USEFUL FUNCTIONS 69

3.3 SOME PROPERTIES OF THE FOURIER TRANSFORM 75
- 3.3.1 Time-Frequency Duality 76
- 3.3.2 Duality Property 77
- 3.3.3 Time-Scaling Property 79
- 3.3.4 Time-Shifting Property 81
- 3.3.5 Frequency-Shifting Property 83
- 3.3.6 Convolution Theorem 87
- 3.3.7 Time Differentiation and Time Integration 88

3.4 SIGNAL TRANSMISSION THROUGH A LINEAR SYSTEM 90
- 3.4.1 Signal Distortion during Transmission 92
- 3.4.2 Distortionless Transmission 92

3.5 IDEAL VERSUS PRACTICAL FILTERS 95

3.6 SIGNAL DISTORTION OVER A COMMUNICATION CHANNEL 97
3.6.1 Linear Distortion 97
3.6.2 Distortion Caused by Channel Nonlinearities 99
3.6.3 Distortion Caused by Multipath Effects 101
3.6.4 Fading Channels 103

3.7 SIGNAL ENERGY AND ENERGY SPECTRAL DENSITY 103
3.7.1 Parseval's Theorem 103
3.7.2 Energy Spectral Density (ESD) 104
3.7.3 Essential Bandwidth of a Signal 105
3.7.4 Energy of Modulated Signals 108
3.7.5 Time Autocorrelation Function and the Energy Spectral Density 109

3.8 SIGNAL POWER AND POWER SPECTRAL DENSITY 111
3.8.1 Power Spectral Density (PSD) 111
3.8.2 Time Autocorrelation Function of Power Signals 113
3.8.3 Input and Output Power Spectral Densities 117
3.8.4 PSD of Modulated Signals 118

3.9 NUMERICAL COMPUTATION OF FOURIER TRANSFORM: THE DFT 118

3.10 MATLAB EXERCISES 123

4 AMPLITUDE MODULATIONS AND DEMODULATIONS 140

4.1 BASEBAND VERSUS CARRIER COMMUNICATIONS 140

4.2 DOUBLE-SIDEBAND AMPLITUDE MODULATION 142

4.3 AMPLITUDE MODULATION (AM) 151

4.4 BANDWIDTH-EFFICIENT AMPLITUDE MODULATIONS 158

4.5 AMPLITUDE MODULATIONS: VESTIGIAL SIDEBAND (VSB) 167

4.6 LOCAL CARRIER SYNCHRONIZATION 170

4.7 FREQUENCY DIVISION MULTIPLEXING (FDM) 172

4.8 PHASE-LOCKED LOOP AND SOME APPLICATIONS 173

4.9 MATLAB EXERCISES 181

5 ANGLE MODULATION AND DEMODULATION 202

5.1 NONLINEAR MODULATION 202

5.2 BANDWIDTH OF ANGLE-MODULATED WAVES 209

5.3 GENERATING FM WAVES 222

5.4 DEMODULATION OF FM SIGNALS 231

5.5 EFFECTS OF NONLINEAR DISTORTION AND INTERFERENCE 234

5.6 SUPERHETERODYNE ANALOG AM/FM RECEIVERS 239

5.7 FM BROADCASTING SYSTEM 241

5.8 MATLAB EXERCISES 242

6 SAMPLING AND ANALOG-TO-DIGITAL CONVERSION 251

6.1 SAMPLING THEOREM 251
- 6.1.1 Signal Reconstruction from Uniform Samples 253
- 6.1.2 Practical Issues in Signal Sampling and Reconstruction 258
- 6.1.3 Maximum Information Rate: Two Pieces of Information per Second per Hertz 262
- 6.1.4 Nonideal Practical Sampling Analysis 263
- 6.1.5 Some Applications of the Sampling Theorem 267

6.2 PULSE CODE MODULATION (PCM) 268
- 6.2.1 Advantages of Digital Communication 270
- 6.2.2 Quantizing 271
- 6.2.3 Principle of Progressive Taxation: Nonuniform Quantization 274
- 6.2.4 Transmission Bandwidth and the Output SNR 278

6.3 DIGITAL TELEPHONY: PCM IN T1 CARRIER SYSTEMS 281

6.4 DIGITAL MULTIPLEXING 285
- 6.4.1 Signal Format 285
- 6.4.2 Asynchronous Channels and Bit Stuffing 287
- 6.4.3 Plesiochronous (almost Synchronous) Digital Hierarchy 288

6.5 DIFFERENTIAL PULSE CODE MODULATION (DPCM) 290

6.6 ADAPTIVE DIFFERENTIAL PCM (ADPCM) 294

6.7 DELTA MODULATION 295

6.8 VOCODERS AND VIDEO COMPRESSION 300
- 6.8.1 Linear Prediction Coding Vocoders 301

6.9 MATLAB EXERCISES 310

7 PRINCIPLES OF DIGITAL DATA TRANSMISSION 326

7.1 DIGITAL COMMUNICATION SYSTEMS 326
- 7.1.1 Source 326
- 7.1.2 Line Coder 327
- 7.1.3 Multiplexer 328
- 7.1.4 Regenerative Repeater 328

7.2 LINE CODING 329
- 7.2.1 PSD of Various Line Codes 330
- 7.2.2 Polar Signaling 334
- 7.2.3 Constructing a DC Null in PSD by Pulse Shaping 336
- 7.2.4 On-Off Signaling 337
- 7.2.5 Bipolar Signaling 339

7.3 PULSE SHAPING 343
- 7.3.1 Intersymbol Interferences (ISI) and Effect 343
- 7.3.2 Nyquist's First Criterion for Zero ISI 344
- 7.3.3 Controlled ISI or Partial Response Signaling 350
- 7.3.4 Example of a Duobinary Pulse 351
- 7.3.5 Pulse Relationship between Zero-ISI, Duobinary, and Modified Duobinary 352
- 7.3.6 Detection of Duobinary Signaling and Differential Encoding 353
- 7.3.7 Pulse Generation 355

7.4 SCRAMBLING 355

7.5 DIGITAL RECEIVERS AND REGENERATIVE REPEATERS 358
- 7.5.1 Equalizers 359
- 7.5.2 Timing Extraction 363
- 7.5.3 Detection Error 365

7.6 EYE DIAGRAMS: AN IMPORTANT TOOL 366

7.7 PAM: *M*-ARY BASEBAND SIGNALING FOR HIGHER DATA RATE 369

7.8 DIGITAL CARRIER SYSTEMS 372
- 7.8.1 Basic Binary Carrier Modulations 372
- 7.8.2 PSD of Digital Carrier Modulation 374
- 7.8.3 Connections between Analog and Digital Carrier Modulations 376
- 7.8.4 Demodulation 377

7.9 *M*-ARY DIGITAL CARRIER MODULATION 380

7.10 MATLAB EXERCISES 386

8 FUNDAMENTALS OF PROBABILITY THEORY 393

8.1 CONCEPT OF PROBABILITY 393

8.2 RANDOM VARIABLES 408

8.3 STATISTICAL AVERAGES (MEANS) 427

8.4 CORRELATION 436

8.5 LINEAR MEAN SQUARE ESTIMATION 440

8.6 SUM OF RANDOM VARIABLES 443

8.7 CENTRAL LIMIT THEOREM 446

9 RANDOM PROCESSES AND SPECTRAL ANALYSIS 456

9.1 FROM RANDOM VARIABLE TO RANDOM PROCESS 456

9.2 CLASSIFICATION OF RANDOM PROCESSES 461

9.3 POWER SPECTRAL DENSITY 465

9.4 MULTIPLE RANDOM PROCESSES 479

9.5 TRANSMISSION OF RANDOM PROCESSES THROUGH LINEAR SYSTEMS 480

9.6 APPLICATION: OPTIMUM FILTERING (WIENER-HOPF FILTER) 483

9.7 APPLICATION: PERFORMANCE ANALYSIS OF BASEBAND ANALOG SYSTEMS 486

9.8 APPLICATION: OPTIMUM PREEMPHASIS-DEEMPHASIS SYSTEMS 488

9.9 BANDPASS RANDOM PROCESSES 491

10 PERFORMANCE ANALYSIS OF DIGITAL COMMUNICATION SYSTEMS 506

10.1 OPTIMUM LINEAR DETECTOR FOR BINARY POLAR SIGNALING 506
- 10.1.1 Binary Threshold Detection 507
- 10.1.2 Optimum Receiver Filter—Matched Filter 508

10.2 GENERAL BINARY SIGNALING 512
- 10.2.1 Optimum Linear Receiver Analysis 512
- 10.2.2 Performance Analysis of General Binary Systems 516

10.3 COHERENT RECEIVERS FOR DIGITAL CARRIER MODULATIONS 520

10.4 SIGNAL SPACE ANALYSIS OF OPTIMUM DETECTION 525
10.4.1 Geometrical Signal Space 525
10.4.2 Signal Space and Basis Signals 527

10.5 VECTOR DECOMPOSITION OF WHITE NOISE RANDOM PROCESSES 530
10.5.1 Determining Basis Functions for a Random Process 530
10.5.2 Geometrical Representation of White Noise Processes 531
10.5.3 White Gaussian Noise 533
10.5.4 Properties of Gaussian Random Process 534

10.6 OPTIMUM RECEIVER FOR WHITE GAUSSIAN NOISE CHANNELS 536
10.6.1 Geometric Representations 536
10.6.2 Dimensionality of the Detection Signal Space 538
10.6.3 (Simplified) Signal Space and Decision Procedure 541
10.6.4 Decision Regions and Error Probability 545
10.6.5 Multiamplitude Signaling (PAM) 551
10.6.6 *M*-ary QAM Analysis 554

10.7 GENERAL EXPRESSION FOR ERROR PROBABILITY OF OPTIMUM RECEIVERS 561

10.8 EQUIVALENT SIGNAL SETS 569
10.8.1 Minimum Energy Signal Set 572
10.8.2 Simplex Signal Set 575

10.9 NONWHITE (COLORED) CHANNEL NOISE 577

10.10 OTHER USEFUL PERFORMANCE CRITERIA 578

10.11 NONCOHERENT DETECTION 581

10.12 MATLAB EXERCISES 589

11 SPREAD SPECTRUM COMMUNICATIONS 614

11.1 FREQUENCY HOPPING SPREAD SPECTRUM (FHSS) SYSTEMS 614

11.2 MULTIPLE FHSS USER SYSTEMS AND PERFORMANCE 618

11.3 APPLICATIONS OF FHSS 621

11.4 DIRECT SEQUENCE SPREAD SPECTRUM 624

11.5 RESILIENT FEATURES OF DSSS 628

11.6 CODE DIVISION MULTIPLE-ACCESS (CDMA) OF DSSS 630

11.7 MULTIUSER DETECTION (MUD) 637

11.8 MODERN PRACTICAL DSSS CDMA SYSTEMS 643
11.8.1 CDMA in Cellular Phone Networks 643
11.8.2 CDMA in the Global Positioning System (GPS) 647
11.8.3 IEEE 802.11b Standard for Wireless LAN 649

11.9 MATLAB EXERCISES 651

12 DIGITAL COMMUNICATIONS UNDER LINEARLY DISTORTIVE CHANNELS 666

12.1 LINEAR DISTORTIONS OF WIRELESS MULTIPATH CHANNELS 666

12.2 RECEIVER CHANNEL EQUALIZATION 670
12.2.1 Antialiasing Filter vs. Matched Filter 670
12.2.2 Maximum Likelihood Sequence Estimation (MLSE) 673

12.3 LINEAR *T*-SPACED EQUALIZATION (TSE) 676
12.3.1 Zero-Forcing TSE 677
12.3.2 TSE Design Based on MMSE 679

12.4 LINEAR FRACTIONALLY SPACED EQUALIZERS (FSE) 684
12.4.1 The Single-Input–Multiple-Output (SIMO) Model 684
12.4.2 FSE Designs 686

12.5 CHANNEL ESTIMATION 688

12.6 DECISION FEEDBACK EQUALIZER 689

12.7 OFDM (MULTICARRIER) COMMUNICATIONS 692
12.7.1 Principles of OFDM 692
12.7.2 OFDM Channel Noise 698
12.7.3 Zero-Padded OFDM 700
12.7.4 Cyclic Prefix Redundancy in OFDM 701
12.7.5 OFDM Equalization 701

12.8 DISCRETE MULTITONE (DMT) MODULATIONS 702

12.9 REAL-LIFE APPLICATIONS OF OFDM AND DMT 707

12.10 BLIND EQUALIZATION AND IDENTIFICATION 711

12.11 TIME-VARYING CHANNEL DISTORTIONS DUE TO MOBILITY 712

12.12 MATLAB EXERCISES 715

13 INTRODUCTION TO INFORMATION THEORY 734

13.1 MEASURE OF INFORMATION 734

13.2 SOURCE ENCODING 739

13.3 ERROR-FREE COMMUNICATION OVER A NOISY CHANNEL 745

13.4 CHANNEL CAPACITY OF A DISCRETE MEMORYLESS CHANNEL 748

13.5 CHANNEL CAPACITY OF A CONTINUOUS MEMORYLESS CHANNEL 756

13.6 PRACTICAL COMMUNICATION SYSTEMS IN LIGHT OF SHANNON'S EQUATION 773

13.7 FREQUENCY-SELECTIVE CHANNEL CAPACITY 776

13.8 MULTIPLE-INPUT–MULTIPLE-OUTPUT COMMUNICATION SYSTEMS 781
13.8.1 Capacity of MIMO Channels 781
13.8.2 Transmitter without Channel Knowledge 783
13.8.3 Transmitter with Channel Knowledge 785

13.9 MATLAB EXERCISES 789

14 ERROR CORRECTING CODES 802

14.1 OVERVIEW 802

14.2 REDUNDANCY FOR ERROR CORRECTION 803

14.3 LINEAR BLOCK CODES 806

14.4 CYCLIC CODES 813

14.5 THE EFFECTS OF ERROR CORRECTION 822

14.6 CONVOLUTIONAL CODES 827
14.6.1 Convolutional Encoder 827
14.6.2 Decoding Convolutional Codes 831

14.7 TRELLIS DIAGRAM OF BLOCK CODES 837

14.8 CODE COMBINING AND INTERLEAVING 839

14.9 SOFT DECODING 841

14.10 SOFT-OUTPUT VITERBI ALGORITHM (SOVA) 844

14.11 TURBO CODES 846

14.12 LOW-DENSITY PARITY CHECK (LDPC) CODES 854

14.13 MATLAB EXERCISES 861

A ORTHOGONALITY OF SOME SIGNAL SETS 873

A.1 ORTHOGONALITY OF THE TRIGONOMETRIC AND EXPONENTIAL SIGNAL SET 873

A.2 ORTHOGONALITY OF THE EXPONENTIAL SIGNAL SET 874

B CAUCHY-SCHWARZ INEQUALITY 875

C GRAM-SCHMIDT ORTHOGONALIZATION OF A VECTOR SET 877

D BASIC MATRIX PROPERTIES AND OPERATIONS 880

D.1 NOTATION 880

D.2 MATRIX PRODUCT AND PROPERTIES 881

D.3 IDENTITY AND DIAGONAL MATRICES 882

D.4 DETERMINANT OF SQUARE MATRICES 882

D.5 TRACE 883

D.6 EIGENDECOMPOSITION 883

D.7 SPECIAL HERMITIAN SQUARE MATRICES 884

E MISCELLANEOUS 885

E.1 L'HÔPITAL'S RULE 885

E.2 TAYLOR AND MACLAURIN SERIES 885

E.3 POWER SERIES 885

E.4 SUMS 886

E.5 COMPLEX NUMBERS 886

E.6 TRIGONOMETRIC IDENTITIES 886

E.7 INDEFINITE INTEGRALS 887

INDEX 889

PREFACE (INTERNATIONAL EDITION)

The chief objective of the fourth (international) edition is to respond to the tremendous amount of technological progress in communication systems over the decade since the third edition was published. At the same time, new software and teaching tools have also become available, making it much easier to provide solid and illustrative examples as well as more experimental opportunities for students. In this new edition, major changes are implemented to incorporate recent technological advances of telecommunications. To captivate students' attention and make it easier for students to relate the course materials to their daily experience with communication tools, we will provide relevant information on the operation and features of cellular systems, wireless local area networks (LANs), and wire-line (digital subscriber loop or DSL) internet services, among others.

Major Revision

A number of critical changes are motivated by the need to emphasize the fundamentals of digital communication systems that have permeated our daily lives. Specifically, in light of the widespread applications of new technologies such as spread spectrum and orthogonal frequency division multiplexing (OFDM), we present a new chapter (Chapter 11) on spread spectrum communications and a new chapter (Chapter 12) on frequency-selective channels and OFDM systems. As practical examples of such systems, we provide a basic introduction of current wireless communication standards including cellular systems and IEEE 802.11a/b/g/n wireless LAN systems. In addition, we summarize the latest DSL modem technologies and services. At the fundamental level, information theory and coding have also been transformed by progress in several important areas. In this edition, we include the basic principles of multiple-input–multiple-output (MIMO) technology which has begun to see broad commercial application. We also cover several notable breakthroughs in error correction coding, including soft decoding, turbo codes, and low-density parity check (LDPC) codes.

To enhance the learning experience and to give students opportunities for computer-based experimental practice, relevant MATLAB examples and exercises have been provided in chapters that can be enhanced by these hands-on experiments.

Organization

The fourth (international) edition, begins with a traditional review of signal and system fundamentals and proceeds to the core communication topics of analog modulation and digital pulse-coded modulation. We then present the fundamental tools of probability theory and random processes to be used in the design and analysis of digital communications in the rest of this text. After coverage of the fundamentals of digital communication systems, the last two

chapters provide an overview of information theory and the fundamentals of forward error correction codes.

Ideally, the subjects covered in this text should be taught in two courses: one on the basic operations of communication systems, and one on the analysis of modern communication systems under noise and other distortions. The former relies heavily on deterministic analytical tools such as Fourier series, Fourier transforms and the sampling theorem, while the latter relies on tools from probability and random processes to tackle the unpredictability of message signals and noises. Today, however, with so many competing courses, it may be difficult to squeeze into a typical electrical engineering curriculum two basic courses on communications. Some universities do require a course in probability and random processes as a prerequisite, allowing both areas to be covered reasonably well in a one-semester course. This book is designed for adoption both as a one-semester course (in which the deterministic aspects of communication systems are emphasized with little consideration of the effects of noise and interference) and for a course that deals with both the deterministic and probabilistic aspects of communication systems. The book itself is self-contained, providing all the necessary background in probabilities and random processes. However, as stated earlier, if both deterministic and probabilistic aspects of communications are to be covered in one semester, it is highly desirable for students to have a good background in probabilities.

Chapter 1 introduces a panoramic view of communication systems. All the important concepts of communication theory are explained qualitatively in a heuristic way. This attracts students to communications topics in general. With this momentum, they are motivated to study the tool of signal analysis in Chapters 2 and 3, where they are encouraged to see a signal as a vector, and to think of the Fourier spectrum as a way of representing a signal in terms of its vector components. Chapters 4 and 5 discuss amplitude (linear) and angle (nonlinear) modulations respectively. Many instructors feel that in this digital age, modulation should be deemphasized. We hold that modulation is not so much a method of communication as a basic tool of signal processing; it will always be needed, not only in the area of communication (digital or analog), but also in many other areas of electrical engineering. Hence, neglecting modulation may prove to be rather shortsighted. Chapter 6, which serves as the fundamental link between analog and digital communications, describes the process of analog-to-digital conversion (ADC). It provides details of sampling, pulse code modulation (including DPCM), delta modulation, speech coding (vocoder), image/video coding, and compression. Chapter 7 discusses the principles and techniques used in digital modulation. It introduces the concept of channel distortion and presents equalization as an effective means of distortion compensation.

Chapters 8 and 9 provide the essential background on theories of probability and random processes. These comprise the second tool required for the study of communication systems. Every attempt is made to motivate students and to maintain their interest through these chapters by providing applications to communications problems wherever possible. Chapters 10 presents the analysis of digital communication systems in the presence of noise. It contains optimum signal detection in digital communication. Chapter 11 focuses on spread spectrum communications. Chapter 12 presents various practical techniques that can be used to combat practical channel distortions. This chapter captures both channel equalization and the broadly applied technology of OFDM. Chapter 13 provides a tutorial of information theory. Finally, the principles and key practical aspects of error control coding are given in Chapter 14.

One of the goals for writing this book has been to make learning a pleasant or at least a less intimidating experience for students by presenting the subject in a clear, understandable, and logically organized manner. Every effort has been made to deliver insights—rather than just understanding—as well as heuristic explanations of theoretical results wherever possible.

Many examples are provided for further clarification of abstract results. Even partial success in achieving this stated goal would make all our efforts worthwhile.

A Whole New World

There have been a number of major technology developments since the publication of the third edition in 1998. First of all, the cellular telephone has deeply penetrated the daily lives of urban and suburban households in most developed and even developing nations. In 1998 very few students carried beepers and cell phones into the classroom. Now, nearly every college student has a cell. Second, in 1998 most of the household internet connections were linked via low speed (28.8kbit/s) voiceband modems. Today, a majority of our students are connected to cyberspace through DSL or cable services. In addition, wireless LAN has made esoteric terms such as IEEE 802.11 into household names. Most students in the classroom have had experience exploring these technologies.

Because of the vast technological advances, this new generation of students is extremely interested in learning about these new technologies and their implementation. The students are eager to understand how and where they may be able to make contributions in industry. Such strong motivation must be encouraged and taken advantage of. This new edition will enable instructors either to cover the topics themselves or to assign reading materials such that the students can acquire relevant information. The new edition achieves these goals by stressing the digital aspects of the text and by incorporating the most commonly known wire-less and wire-line digital technologies.

Course Adoption

With a combined teaching experience of over 55 years, we have taught communication classes under both quarter and semester systems in several major universities. In complementary fashion, students' personal experiences with communication systems have continuously been multiplying, from simple radio sets in the 1960s to the twenty-first century, with it easy access to wireless LAN, cellular devices, satellite radio, and home internet services. Hence, more and more students are interested in learning how familiar electronic gadgets work. With this important need and our past experiences in mind, we revised the fourth (international) edition of this text to fit well within several different curriculum configurations. In all cases, basic coverage should teach the fundamentals of analog and digital communications (Chapters 1–7).

One-Semester Course (without strong probability background)

In many existing curricula, undergraduate students are not exposed to simple probability tools until they begin to take communications. Often this occurs because the students were sent to take an introductory statistical course that is disconnected from engineering science. This text is well suited to students of such a background. The first seven chapters form a comprehensive coverage of modern digital and analog communication systems for average ECE undergraduate students. Such a course can be taught in one semester (40–45 instructional hours). Under the premise that each student has built a solid background in Fourier analysis via a prerequisite class on *signals and systems*, most of the first three chapters can be treated as a review in one week. The rest of the semester can be fully devoted to teaching Chapters 4 through 7 with partial coverage on the practical systems of Chapters 11 and 12 to enhance students' interest.

One-Semester Course (with a strong probability background)

For curricula that have strengthened the background coverage of probability theory, a much more extensive coverage of digital communications can be achieved within one semester. A rigorous probability class can be taught within the context of signal and system analysis

(cf. George R. Cooper and Clare D. McGillem, *Probabilistic Methods of Signal and System Analysis*, Oxford University Press, 1999). For this scenario, in addition to Chapters 1 through 7, Chapter 10 and part of Chapter 12 on equalization can also be taught in one semester, provided the students have a solid probability background that can limit the coverage of Chapters 8 and 9 to a few hours. Students completing this course would be well prepared to enter the telecommunications industry or to enter graduate studies.

Two-Semester Series (without a separate probability course)

The entire text can be thoroughly covered in two semesters for a curriculum that does not have any prior probability course. In other words, for a two-course series, the goal is to teach both communication systems and fundamentals of probabilities. In an era of many competing courses in the ECE curriculum, it is hard to set aside two semester courses for communications alone. On the other hand, most universities do have a probability course that is separately taught by nonengineering professors. In this scenario it would be desirable to fold probability theory into the two communication courses. Thus, for two semester courses, the coverage can be as follows:

- 1st semester: Chapters 1–7 (Signals and Communication Systems)
- 2nd semester: Chapters 8–12 (Modern Digital Communication Systems)

One-Quarter Course (with a strong probability background)

In a quarter system, students must have prior exposure to probability and statistics at a rigorous level (cf. Cooper and McGillem, *Probabilistic Methods of Signal and System Analysis*). They must also have solid knowledge of Fourier analysis. Within a quarter, the class can impart the basics of analog and digital communication systems (Chapters 3–7), and, in chapters 10 and 11, respectively, analysis of digital communication systems and spread spectrum communications.

One-Quarter Course (without a strong probability background)

In the rare case that students come in without much background in probability, it is important for them to acquire basic knowledge of communication systems. It is wise not to attempt to analyze digital communication systems. Instead, basic coverage without prior knowledge of probability can be achieved by teaching the operations of analog and digital systems (Chapters 1–7) and providing a high-level discussion of spread spectrum wireless systems (Chapter 11).

Two-Quarter Series (with basic probability background)

Unlike a one-quarter course, a two-quarter series can be well designed to teach most of the important materials on communication systems and their analysis. The entire text can be extensively taught in two quarters for a curriculum that has some preliminary coverage of Fourier analysis and probabilities. Essentially viewing Chapters 1 through 3 and Chapter 8 as partly new and partly reviews, the coverage can be as follows:

- 1st quarter: Chapters 1–9 (Communication Systems and Analysis)
- 2nd quarter: Chapters 10–14 (Digital Communication Systems)

MATLAB and Laboratory Experience

Since many universities no longer have hardware communication laboratories, MATLAB-based communication system exercises are included to enhance the learning experience. Students will be able to design systems and modify their parameters to evaluate the overall effects on the performance of communication systems through computer displays and the

measurement of bit error rates. Students will acquire first-hand knowledge on how to design and perform simulations of communication systems.

Acknowledgments

First, the authors thank all the students they have had over the years. This edition would not have been possible without the much feedback from, and discussions with, our students. The authors thank all the reviewers for providing invaluable input to improve the text. Finally, the authors also thank Professor Norman Morrison, University of Cape Town, for suggesting a new problem (P8-2.3) in this edition.

B.P. Lathi
Zhi Ding

MODERN DIGITAL AND ANALOG COMMUNICATION SYSTEMS

International Fourth Edition

1 INTRODUCTION

Over the past decade, the rapid expansion of digital communication technologies has been simply astounding. Internet, a word and concept once familiar only to technologists and the scientific community, has permeated every aspect of people's daily lives. It is quite difficult to find any individual in a modern society that has not been touched by new communication technologies ranging from cellular phones to Bluetooth. This book examines the basic principles of communication by electric signals. Before modern times, messages were carried by runners, carrier pigeons, lights, and fires. These schemes were adequate for the distances and "data rates" of the age. In most parts of the world, these modes of communication have been superseded by electrical communication systems,* which can transmit signals over much longer distances (even to distant planets and galaxies) and at the speed of light.

Electrical communication is dependable and economical; communication technologies improve productivity and energy conservation. Increasingly, business meetings are conducted through teleconferences, saving the time and energy formerly expended on travel. Ubiquitous communication allows real-time management and coordination of project participants from around the globe. E-mail is rapidly replacing the more costly and slower "snail mails." E-commerce has also drastically reduced some costs and delays associated with marketing, while customers are also much better informed about new products and product information. Traditional media outlets such as television, radio, and newspapers have been rapidly evolving in the past few years to cope with, and better utilize, the new communication and networking technologies. The goal of this textbook is to provide the fundamental technical knowledge needed by next-generation communication engineers and technologists for designing even better communication systems of the future.

1.1 COMMUNICATION SYSTEMS

Figure 1.1 presents three typical communication systems: a wire-line telephone–cellular phone connection, a TV broadcasting system, and a wireless computer network. Because of the numerous examples of communication systems in existence, it would be unwise to attempt to study the details of all kinds of communication systems in this book. Instead, the most efficient and effective way to learn about communication is by studying the major functional blocks common to practically all communication systems. This way, students are not

* With the exception of the postal service.

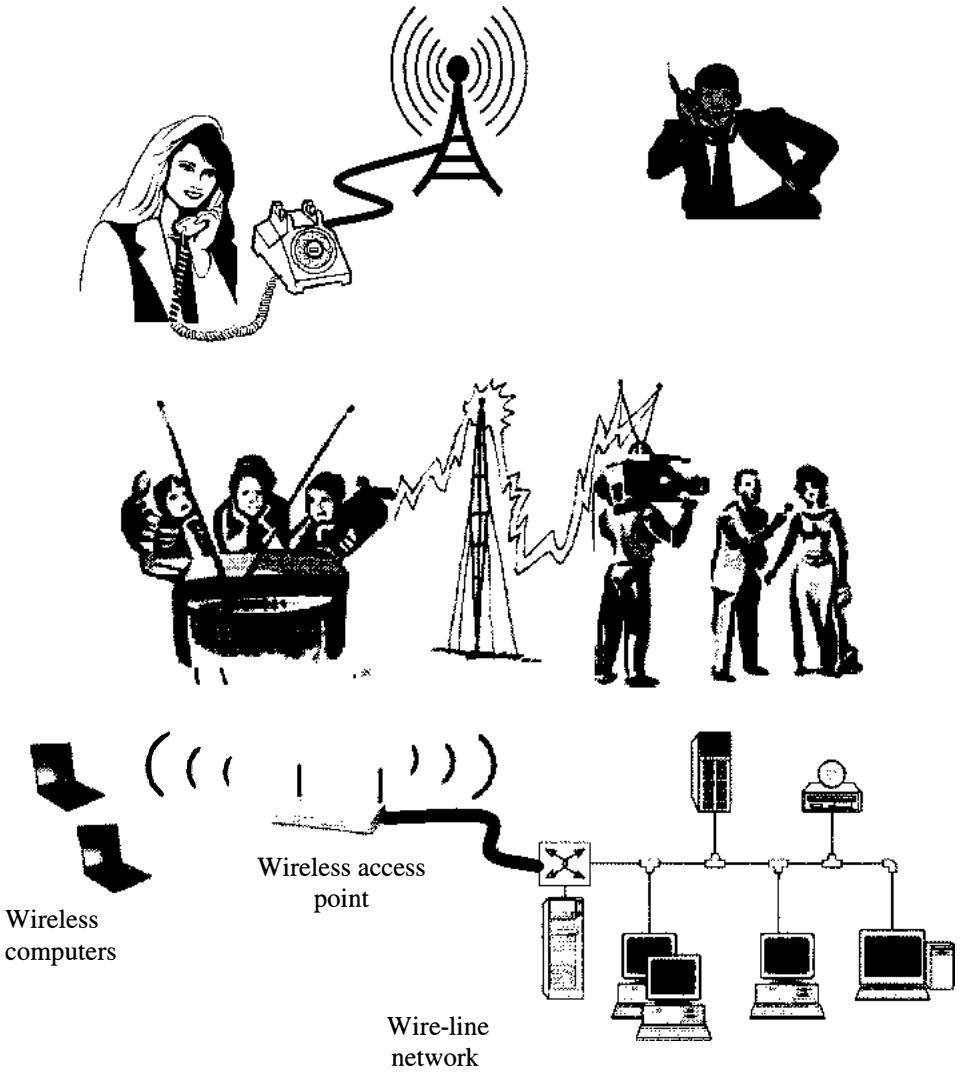


Figure 1.1 Some examples of communications systems.

merely learning the operations of those existing systems they have studied; More importantly, they can acquire the basic knowledge needed to design and analyze new systems never encountered in a textbook. To begin, it is essential to establish a typical communication system model as shown in Fig. 1.2. The key components of a communication system are as follows.

The **source** originates a message, such as a human voice, a television picture, an e-mail message, or data. If the data is nonelectric (e.g., human voice, e-mail text, television video), it must be converted by an **input transducer** into an electric waveform referred to as the **baseband signal** or **message signal** through physical devices such as a microphone, a computer keyboard, or a CCD camera.

The **transmitter** modifies the baseband signal for efficient transmission. The transmitter may consist of one or more subsystems: an A/D converter, an encoder, and a modulator. Similarly, the receiver may consist of a demodulator, a decoder, and a D/A converter.

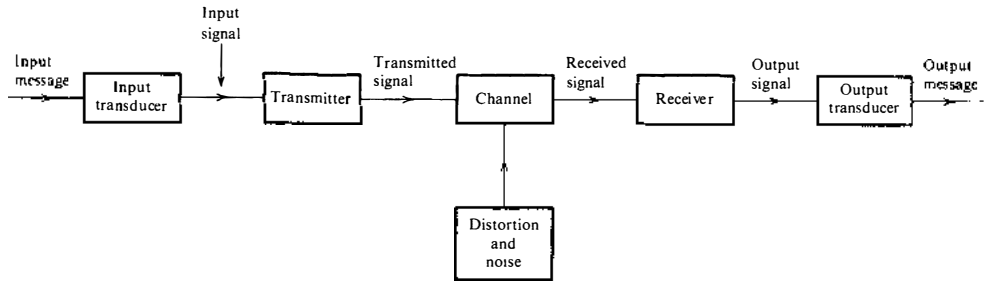


Figure 1.2 Communication system.

The **channel** is a medium of choice that can convey the electric signals at the transmitter output over a distance. A typical channel can be a pair of twisted copper wires (telephone and DSL), coaxial cable (television and internet), an optical fiber, or a radio link. Additionally, a channel can also be a point-to-point connection in a mesh of interconnected channels that form a communication network.

The **receiver** reprocesses the signal received from the channel by reversing the signal modifications made at the transmitter and removing the distortions made by the channel. The receiver output is fed to the **output transducer**, which converts the electric signal to its original form—the message.

The **destination** is the unit to which the message is communicated.

A channel is a physical medium that behaves partly like a filter that generally attenuates the signal and distorts the transmitted waveforms. The signal attenuation increases with the length of the channel, varying from a few percent for short distances to orders of magnitude in interplanetary communications. Signal waveforms are distorted because of physical phenomena such as frequency-dependent gains, multipath effects, and Doppler shift. For example, a *frequency-selective* channel causes different amounts of attenuation and phase shift to different frequency components of the signal. A square pulse is rounded or "spread out" during transmission over a low-pass channel. These types of distortion, called **linear distortion**, can be partly corrected at the receiver by an equalizer with gain and phase characteristics complementary to those of the channel. Channels may also cause **nonlinear distortion** through attenuation that varies with the signal amplitude. Such distortions can also be partly corrected by a complementary equalizer at the receiver. Channel distortions, if known, can also be precompensated by transmitters by applying channel-dependent predistortions.

In a practical environment, signals passing through communication channels not only experience channel distortions but also are corrupted along the path by undesirable interferences and disturbances lumped under the broad term **noise**. These interfering signals are random and are unpredictable from sources both external and internal. External noise includes interference signals transmitted on nearby channels, human-made noise generated by faulty contact switches of electrical equipment, automobile ignition radiation, fluorescent lights or natural noise from lightning, microwave ovens, and cellphone emissions, as well as electric storms and solar and intergalactic radiation. With proper care in system design, external noise can be minimized or even eliminated in some cases. Internal noise results from thermal motion of charged particles in conductors, random emission, and diffusion or recombination of charged carriers in electronic devices. Proper care can reduce the effect of internal noise but can never eliminate it. Noise is one of the underlying factors that limit the rate of telecommunications.

Thus in practical communication systems, the channel distorts the signal, and noise accumulates along the path. Worse yet, the signal strength decreases while the noise level remains

steady regardless of the distance from the transmitter. Thus, the signal quality is continuously worsening along the length of the channel. Amplification of the received signal to make up for the attenuation is to no avail because the noise will be amplified by the same proportion, and the quality remains, at best, unchanged.* These are the key challenges that we must face in designing modern communication systems.

1.2 ANALOG AND DIGITAL MESSAGES

Messages are digital or analog. Digital messages are ordered combinations of finite symbols or codewords. For example, printed English consists of 26 letters, 10 numbers, a space, and several punctuation marks. Thus, a text document written in English is a digital message constructed from the ASCII keyboard of 128 symbols. Human speech is also a digital message, because it is made up from a finite vocabulary in a language.† Music notes are also digital, even though the music sound itself is analog. Similarly, a Morse-coded telegraph message is a digital message constructed from a set of only **two** symbols—dash and dot. It is therefore a **binary** message, implying only two symbols. A digital message constructed with M symbols is called an M**-ary** message.

Analog messages, on the other hand, are characterized by data whose values vary over a continuous range and are defined for a continuous range of time. For example, the temperature or the atmospheric pressure of a certain location over time can vary over a continuous range and can assume an (uncountable) infinite number of possible values. A piece of music recorded by a pianist is also an analog signal. Similarly, a particular speech waveform has amplitudes that vary over a continuous range. Over a given time interval, an infinite number of possible different speech waveforms exist, in contrast to only a finite number of possible digital messages.

1.2.1 Noise Immunity of Digital Signals

It is no secret to even a casual observer that every time one looks at the latest electronic communication products, newer and better "digital technology" is replacing the old analog technology. Within the past decade, cellular phones have completed their transformation from the first-generation analog AMPS to the current second-generation (e.g., GSM, CDMA) and third-generation (e.g., WCDMA) digital offspring. More visibly in every household, digital video technology (DVD) has made the analog VHS cassette systems almost obsolete. Digital television continues the digital assault on analog video technology by driving out the last analog holdout of color television. There is every reason to ask: Why are digital technologies better? The answer has to do with both economics and quality. The case for economics is made by noting the ease of adopting versatile, powerful, and inexpensive high-speed digital microprocessors. But more importantly at the quality level, one prominent feature of digital communications is the enhanced immunity of digital signals to noise and interferences.

Digital messages are transmitted as a finite set of electrical waveforms. In other words, a digital message is generated from a finite alphabet, while each character in the alphabet can be represented by one waveform or a sequential combination of such waveforms. For example, in sending messages via Morse code, a dash can be transmitted by an electrical pulse of amplitude $A/2$ and a dot can be transmitted by a pulse of negative amplitude

* Actually, amplification may further deteriorate the signal because of additional amplifier noise.

† Here we imply the information contained in the speech rather than its details such as the pronunciation of words and varying inflections, pitch, and emphasis. The speech signal from a microphone contains all these details and is therefore an analog signal, and its information content is more than a thousand times greater than the information accessible from the written text of the same speech.

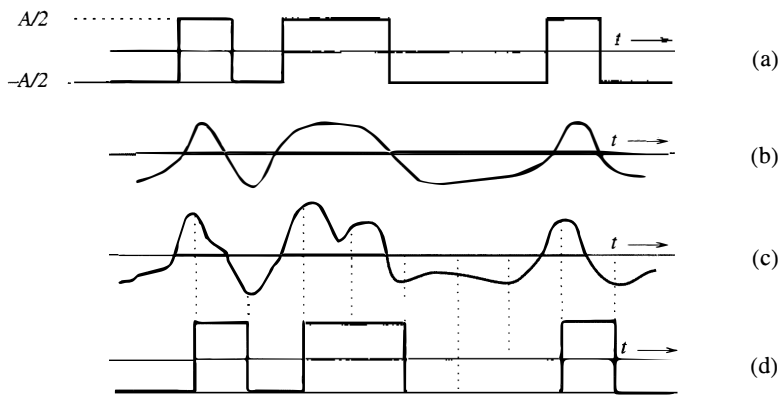


Figure 1.3 (a) Transmitted signal. (b) Received distorted signal (without noise). (c) Received distorted signal (with noise). (d) Regenerated signal (delayed).

$-A/2$ (Fig 1.3a). In an M-ary case, M distinct electrical pulses (or waveforms) are used; each of the M pulses represents one of the M possible symbols. Once transmitted, the receiver must extract the message from a distorted and noisy signal at the channel output. Message extraction is often easier from digital signals than from analog signals because the digital decision must belong to the finite-sized alphabet. Consider a binary case: two symbols are encoded as rectangular pulses of amplitudes $A/2$ and $-A/2$. The only decision at the receiver is to select between two possible pulses received; the fine details of the pulse shape are not an issue. A finite alphabet leads to noise and interference immunity. The receiver's decision can be made with reasonable certainty even if the pulses have suffered modest distortion and noise (Fig. 1.3). The digital message in Fig. 1.3a is distorted by the channel, as shown in Fig. 1.3b. Yet, if the distortion is not too large, we can recover the data without error because we need make only a simple binary decision: Is the received pulse positive or negative? Figure 1.3c shows the same data with channel distortion and noise. Here again, the data can be recovered correctly as long as the distortion and the noise are within limits. In contrast, the waveform shape itself in an analog message carries the needed information, and even a slight distortion or interference in the waveform will show up in the received signal. Clearly, a digital communication system is more rugged than an analog communication system in the sense that it can better withstand noise and distortion (as long as they are within a limit).

1.2.2 Viability of Distortionless Regenerative Repeaters

One main reason for the superior quality of digital systems over analog ones is the viability of **regenerative** repeaters and network nodes in the former. Repeater stations are placed along the communication path of a digital system at distances short enough to ensure that noise and distortion remain within a limit. This allows pulse detection with high accuracy. At each repeater station, or network node, the incoming pulses are detected such that new, "clean" pulses are retransmitted to the next repeater station or node. This process prevents the accumulation of noise and distortion along the path by cleaning the pulses at regular repeater intervals. We can thus transmit messages over longer distances with greater accuracy. There has been widespread application of distortionless regeneration by repeaters in long-haul communication systems and by nodes in a large (possibly heterogeneous) network.

For analog systems, signals and noise within the same bandwidth cannot be separated. Repeaters in analog systems are basically filters plus amplifiers and are not "regenerative."

Thus, it is impossible to avoid in-band accumulation of noise and distortion along the path. As a result, the distortion and the noise interference can accumulate over the entire transmission path as a signal traverses through the network. To compound the problem, the signal is attenuated continuously over the transmission path. Thus, with increasing distance the signal becomes weaker, whereas the distortion and the noise accumulate more. Ultimately, the signal, overwhelmed by the distortion and noise, is buried beyond recognition. Amplification is of little help, since it enhances both the signal and the noise equally. Consequently, the distance over which an analog message can be successfully received is limited by the first transmitter power. Despite these limitations, analog communication was used widely and successfully in the past for short- to medium-range communications. Nowadays, because of the advent of optical fiber communications and the dramatic cost reduction achieved in the fabrication of high-speed digital circuitry and digital storage devices, almost all new communication systems being installed are digital. But some old analog communication facilities are still in use, including those for AM and FM radio broadcasting.

1.2.3 Analog-to-Digital (A/D) Conversion

Despite the differences between analog and digital signals, a meeting ground exists between them: conversion of analog signals to digital signals (A/D conversion). A key device in electronics, the analog-to-digital (A/D) converter, enables digital communication systems to convey analog source signals such as audio and video. Generally, analog signals are continuous in time and in range; that is, they have values at every time instant, and their values can be anything within the range. On the other hand, digital signals exist only at discrete points of time, and they can take on only finite values. A/D conversion can never be 100% accurate. Since, however, human perception does not require infinite accuracy, A/D conversion can effectively capture necessary information from the analog source for digital signal transmission.

Two steps take place in A/D conversion: a continuous time signal is first *sampled* into a discrete time signal, whose continuous amplitude is then *quantized* into a discrete level signal. First, the frequency spectrum of a signal indicates relative magnitudes of various frequency components. The **sampling theorem** (Chapter 6) states that if the highest frequency in the signal spectrum is B (in hertz), the signal can be reconstructed from its discrete samples, taken uniformly at a rate not less than $2B$ samples per second. This means that to preserve the information from a continuous-time signal, we need transmit only its samples (Fig. 1.4).

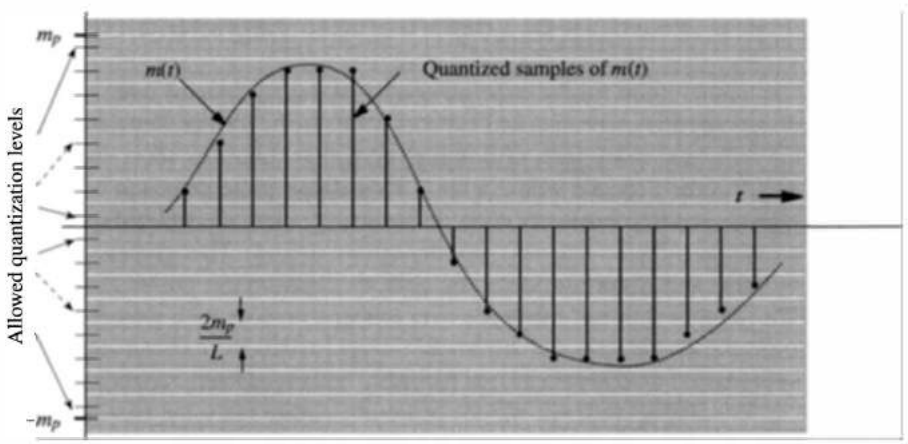


Figure 1.4 Analog-to-digital conversion of a signal.

However, the sample values are still not digital because they lie in a continuous dynamic range. Here, the second step of **quantization** comes to rescue. Through quantization, each sample is approximated, or "rounded off," to the nearest quantized level, as shown in Fig. 1.4. As human perception has only limited accuracy, quantization with sufficient granularity does not compromise the signal quality. If amplitudes of the message signal $m(t)$ lie in the range $(-m_p, m_p)$, the quantizer partitions the signal range into L intervals. Each sample amplitude is approximated by the midpoint of the interval in which the sample value falls. Each sample is now represented by one of the L numbers. The information is thus digitized. Hence, after the two steps of sampling and quantizing, the analog-to-digital (A/D) conversion is completed.

The quantized signal is an approximation of the original one. We can improve the accuracy of the quantized signal to any desired level by increasing the number of levels L. For intelligibility of voice signals, for example, $L = 8$ or 16 is sufficient. For commercial use, $L = 32$ is a minimum, and for telephone communication, $L = 128$ or 256 is commonly used.

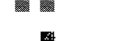
A typical distorted binary signal with noise acquired over the channel is shown in Fig. 1.3. If A is sufficiently large in comparison to typical noise amplitudes, the receiver can still correctly distinguish between the two pulses. The pulse amplitude is typically 5 to 10 times the rms noise amplitude. For such a high signal-to-noise ratio (SNR) the probability of error at the receiver is less than 10^{-6}; that is, on the average, the receiver will make fewer than one error per million pulses. The effect of random channel noise and distortion is thus practically eliminated. Hence, when analog signals are transmitted by digital means, some error, or uncertainty, in the received signal can be caused by quantization, in addition to channel noise and interferences. By increasing L, we can reduce to any desired amount the uncertainty, or error, caused by quantization. At the same time, because of the use of regenerative repeaters, we can transmit signals over a much longer distance than would have been possible for the analog signal. As will be seen later in this text, the price for all these benefits of digital communication is paid in terms of increased processing complexity and bandwidth of transmission.

1.2.4 Pulse-Coded Modulation—A Digital Representation

Once the A/D conversion is over, the original analog message is represented by a sequence of samples, each of which takes on one of the L preset quantization levels. The transmission of this quantized sequence is the task of digital communication systems. For this reason, signal waveforms must be used to represent the quantized sample sequence in the transmission process. Similarly, a digital storage device also would need to represent the samples as signal waveforms. *Pulse-coded modulation* (PCM) is a very simple and yet common mechanism for this purpose.

First, one information *bit* refers to one *binary digit* of **1** or **0**. The idea of PCM is to represent each quantized sample by an ordered combination of two basic pulses: $p_1(t)$ representing **1** and $p_0(t)$ representing **0**. Because each of the L possible sample values can be written as a bit string of length $\log_2 L$, each sample can therefore also be mapped into a short pulse sequence that represents the binary sequence of bits. For example, if $L = 16$, then, each quantized level can be described uniquely by 4 bits. If we use two basic pulses, $p_1(t) = A/2$ and $p_0(t) = -A/2$. A sequence of four such pulses gives $2 \times 2 \times 2 \times 2 = 16$ distinct patterns, as shown in Fig. 1.5. We can assign one pattern to each of the 16 quantized values to be transmitted. Each quantized sample is now coded into a sequence of four binary pulses. This is the principle of PCM transmission, where signaling is carried out by means of only two basic pulses (or symbols).

Figure 1.5 Example of PCM encoding.

Digit	Binary equivalent	Pulse code waveform
0	**0000**	
1	**0001**	
2	**0010**	
3	**0011**	
4	**0100**	
5	**0101**	
6	**0110**	
7	**0111**	
8	**1000**	
9	**1001**	
10	**1010**	
11	**1011**	
12	**1100**	
13	**1101**	
14	**1110**	
15	**1111**	

The binary case is of great practical importance because of its simplicity and ease of detection. Much of today's digital communication is binary.*

Although PCM was invented by P. M. Rainey in 1926 and rediscovered by A. H. Reeves in 1939, it was not until the early 1960s that the Bell System installed the first communication link using PCM for digital voice transmission. The cost and size of vacuum tube circuits were the chief impediments to the use of PCM in the early days before the discovery of semiconductor devices. It was the transistor that made PCM practicable.

From all these discussions on PCM, we arrive at a rather interesting (and to certain extent not obvious) conclusion—that every possible communication can be carried on with a minimum of two symbols. Thus, merely by using a proper sequence of a wink of the eye, one can convey any message, be it a conversation, a book, a movie, or an opera. Every possible detail (such as various shades of colors of the objects and tones of the voice, etc.) that is reproducible on a movie screen or on the high-definition color television can be conveyed with no less accuracy, merely by winks of an eye.†

* An intermediate case exists where we use four basic pulses (quaternary pulses) of amplitudes $\pm A/2$ and $\pm 3A/2$. A sequence of two quaternary pulses can form $4 \times 4 = 16$ distinct levels of values.

† Of course, to convey the information in a movie or a television program in real time, the winking would have to be at an inhumanly high speed. For example, the HDTV signal is represented by 19 million bits (winks) per second.

1.3 CHANNEL EFFECT, SIGNAL-TO-NOISE RATIO, AND CAPACITY

In designing communication systems, it is important to understand and analyze important factors such as the channel and signal characteristics, the relative noise strength, the maximum number of bits that can be sent over a channel per second, and, ultimately, the signal quality.

1.3.1 Signal Bandwidth and Power

In a given (digital) communication system, the fundamental parameters and physical limitations that control the rate and quality are the channel bandwidth B and the signal power P_S. Their precise and quantitative relationships will be discussed in later chapters. Here we shall demonstrate these relationships qualitatively.

The **bandwidth** of a channel is the range of frequencies that it can transmit with reasonable fidelity. For example, if a channel can transmit with reasonable fidelity a signal whose frequency components vary from 0 Hz (dc) up to a maximum of 5000 Hz (5 kHz), the channel bandwidth B is 5 kHz. Likewise, each signal also has a bandwidth that measures the maximum range of its frequency components.

The faster a signal changes, the higher its maximum frequency is, and the larger its bandwidth is. Signals rich in content that changes quickly (such as those for battle scenes in a video) have larger bandwidth than signals that are dull and vary slowly (such as those for a daytime soap opera or a video of sleeping animals). A signal can be successfully sent over a channel if the channel bandwidth exceeds the signal bandwidth.

To understand the role of B, consider the possibility of increasing the speed of information transmission by compressing the signal in time. Compressing a signal in time by a factor of 2 allows it to be transmitted in half the time, and the transmission speed (rate) doubles. Time compression by a factor of 2, however, causes the signal to "wiggle" twice as fast, implying that the frequencies of its components are doubled. Many people have had firsthand experience of this effect when playing a piece of audiotape twice as fast, making the voices of normal people sound like the high-pitched speech of cartoon characters. Now, to transmit this compressed signal without distortion, the channel bandwidth must also be doubled. Thus, the rate of information transmission that a channel can successfully carry is directly proportional to B. More generally, if a channel of bandwidth B can transmit N pulses per second, then to transmit KN pulses per second by means of the same technology, we need a channel of bandwidth KB. To reiterate, the number of pulses per second that can be transmitted over a channel is directly proportional to its bandwidth B.

The **signal power** P_s plays a dual role in information transmission. First, P_s is related to the quality of transmission. Increasing P_s strengthens the signal pulse and diminishes the effect of channel noise and interference. In fact, the quality of either analog or digital communication systems varies with the signal-to-noise ratio (SNR). In any event, a certain minimum SNR at the receiver is necessary for successful communication. Thus, a larger signal power P_s allows the system to maintain a minimum SNR over a longer distance, thereby enabling successful communication over a longer span.

The second role of the signal power is less obvious, although equally important. From the information theory point of view, the channel bandwidth B and the signal power P_s are, to some extent, exchangeable; that is, to maintain a given rate and accuracy of information transmission, we can trade P_S for B, and vice versa. Thus, one may use less B if one is willing

to increase P_s, or one may reduce P_s if one is given bigger B. The rigorous proof of this will be provided in Chapter 13.

In short, the two primary communication resources are the bandwidth and the transmitted power. In a given communication channel, one resource may be more valuable than the other, and the communication scheme should be designed accordingly. A typical telephone channel, for example, has a limited bandwidth (3 kHz), but the power is less restrictive. On the other hand, in space vehicles, huge bandwidth is available but the power is severely limited. Hence, the communication solutions in the two cases are radically different.

1.3.2 Channel Capacity and Data Rate

Channel bandwidth limits the bandwidth of signals that can successfully pass through, whereas signal SNR at the receiver determines the recoverability of the transmitted signals. Higher SNR means that the transmitted signal pulse can use more signal levels, thereby carrying more bits with each pulse transmission. Higher bandwidth B also means that one can transmit more pulses (faster variation) over the channel. Hence, SNR and bandwidth B can both affect the underlying channel "throughput." The peak throughput that can be reliably carried by a channel is defined as the channel capacity.

One of the most commonly encountered channels is known as the additive white Gaussian noise (AWGN) channel. The AWGN channel model assumes no channel distortions except for the additive white Gaussian noise and its finite bandwidth B. This ideal model captures application cases with distortionless channels and provides a performance upper bound for more general distortive channels. The band-limited AWGN channel capacity was dramatically highlighted by Shannon's equation,

$$C = B\log_2(1 + \text{SNR}) \quad \text{bit/s} \tag{1.1}$$

Here the channel capacity C is the upper bound on the rate of information transmission per second. In other words, C is the maximum number of bits that can be transmitted per second with a probability of error arbitrarily close to zero; that is, the transmission is as accurate as one desires. The capacity only points out this *possibility*, however; it does not specify how it is to be realized. Moreover, it is impossible to transmit at a rate higher than this without incurring errors. Shannon's equation clearly brings out the limitation on the rate of communication imposed by B and SNR. If there is no noise on the channel (assuming SNR $= \infty$), then the capacity C would be ∞, and communication rate could be arbitrarily high. We could then transmit any amount of information in the world over one noiseless channel. This can be readily verified. If noise were zero, there would be no uncertainty in the received pulse amplitude, and the receiver would be able to detect any pulse amplitude without error. The minimum pulse amplitude separation can be arbitrarily small, and for any given pulse, we have an infinite number of fine levels available. We can assign one level to every possible message. Because an infinite number of levels are available, it is possible to assign one level to any conceivable message. Cataloging such a code may not be practical, but that is beside the point. Rather, the point is that if the noise is zero, communication ceases to be a problem, at least theoretically. Implementation of such a scheme would be difficult because of the requirement of generation and detection of pulses of precise amplitudes. Such practical difficulties would then set a limit on the rate of communication. It should be remembered that Shannon's result, which represents the upper limit on the rate of communication over a channel, would be achievable only with a system of monstrous and impractical complexity, and with a time delay in reception approaching infinity. Practical systems operate at rates below the Shannon rate.

In conclusion, Shannon's capacity equation demonstrates qualitatively the basic role played by B and SNR in limiting the performance of a communication system. These two parameters then represent the ultimate limitation on the rate of communication. The possibility of resource exchange between these two basic parameters is also demonstrated by the Shannon equation.

As a practical example of trading SNR for bandwidth B, consider the scenario in which we meet a soft-spoken man who speaks a little bit too fast for us to fully understand. This means that as listeners, our bandwidth B is too low and therefore, the capacity C is not high enough to accommodate the rapidly spoken sentences. However, if the man can speak louder (increasing power and hence the SNR), we are likely to understand him much better without changing anything else. This example illustrates the concept of resource exchange between SNR and B. Note, however, that this is not a one-to-one trade. Doubling the speaker volume allows the speaker to talk a little faster, but not twice as fast. This unequal trade effect is fully captured by Shannon's equation [Eq. (1.1)], where doubling the SNR cannot always compensate the loss of B by 50%.

1.4 MODULATION AND DETECTION

Analog signals generated by the message sources or digital signals generated through A/D conversion of analog signals are often referred to as baseband signals because they typically are low pass in nature. Baseband signals may be directly transmitted over a suitable channel (e.g., telephone, fax). However, depending on the channel and signal frequency domain characteristics, baseband signals produced by various information sources are not always suitable for direct transmission over a given channel. When signal and channel frequency bands do not match exactly, channels cannot be moved. Hence, messages must be moved to the right channel frequency bandwidth. Message signals must therefore be further modified to facilitate transmission. In this conversion process, known as **modulation**, the baseband signal is used to modify (i.e., modulate), some parameter of a radio-frequency (RF) *carrier* signal.

A **carrier** is a sinusoid of high frequency. Through modulation, one of the carrier sinusoidal parameters—such as amplitude, frequency, or phase—is varied in proportion to the baseband signal $m(t)$. Accordingly, we have amplitude modulation (AM), frequency modulation (FM), or phase modulation (PM). Figure 1.6 shows a baseband signal $m(t)$ and the corresponding AM and FM waveforms. In AM, the carrier amplitude varies in proportion to $m(t)$, and in FM, the carrier frequency varies in proportion $m(t)$. To reconstruct the baseband signal at the receiver, the modulated signal must pass through a reversal process called **demodulation**.

As mentioned earlier, modulation is used to facilitate transmission. Some of the important reasons for modulation are given next.

1.4.1 Ease of Radiation/Transmission

For efficient radiation of electromagnetic energy, the radiating antenna should be on the order of a fraction or more of the wavelength of the driving signal. For many baseband signals, the wavelengths are too large for reasonable antenna dimensions. For example, the power in a speech signal is concentrated at frequencies in the range of 100 to 3000 Hz. The corresponding wavelength is 100 to 3000 km. This long wavelength would necessitate an impractically large antenna. Instead, by modulating a high-frequency carrier, we effectively translate the signal spectrum to the neighborhood of the carrier frequency that corresponds to a much smaller wavelength. For example, a 10 MHz carrier has a wavelength of only 30 m, and its transmission can be achieved with an antenna size on the order of 3 m. In this respect, modulation is like

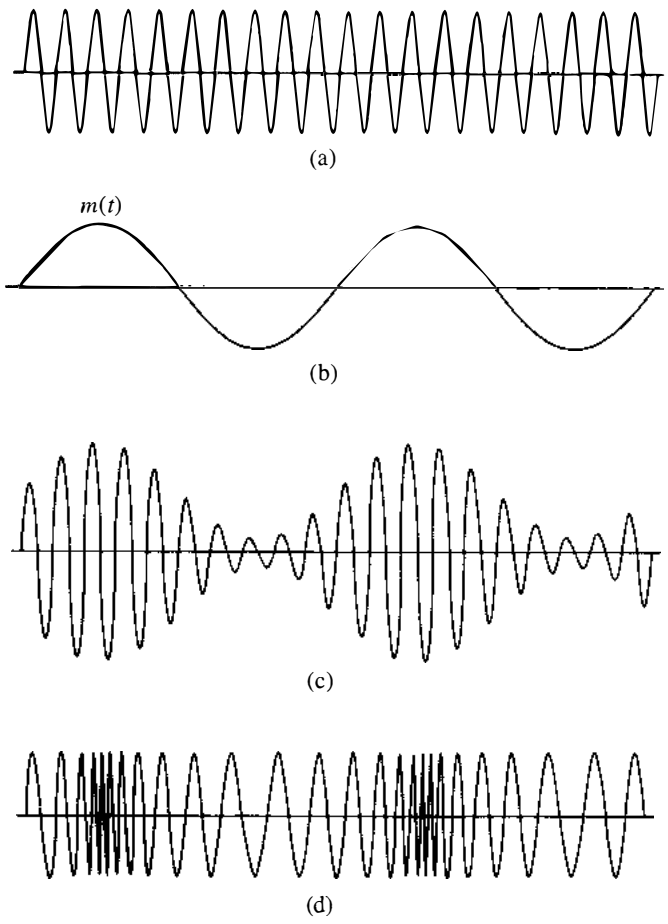


Figure 1.6 Modulation: (a) carrier; (b) modulating (baseband) signal; (c) amplitude-modulated wave; (d) frequency-modulated wave.

letting the baseband signal hitch a ride on a high-frequency sinusoid (carrier). The carrier and the baseband signal may also be compared to a stone and a piece of paper. If we wish to throw a piece of paper, it cannot go too far by itself. But if it is wrapped around a stone (a carrier), it can be thrown over a longer distance.

1.4.2 Simultaneous Transmission of Multiple Signals—Multiplexing

Modulation also allows multiple signals to be transmitted at the same time in the same geographical area without direct mutual interference. This case in point is simply demonstrated by considering the output of multiple television stations carried by the same cable (or over the air) to people's television receivers. Without modulation, multiple video signals will all be interfering with one another because all baseband video signals effectively have the same bandwidth. Thus, cable TV or broadcast TV without modulation would be limited to one station at a time in a given location—a highly wasteful protocol because the channel bandwidth is many times larger than that of the signal.

One way to solve this problem is to use modulation. We can use various TV stations to modulate different carrier frequencies, thus translating each signal to a different frequency

range. If the various carriers are chosen sufficiently far apart in frequency, the spectra of the modulated signals (known as TV channels) will not overlap and thus will not interfere with each other. At the receiver (TV set), a tunable bandpass filter can select the desired station or TV channel for viewing. This method of transmitting several signals simultaneously, over nonoverlapping frequency bands, is known as **frequency division multiplexing (FDM)**. A similar approach is also used in AM and FM radio broadcasting. Here the bandwidth of the channel is shared by various signals without any overlapping.

Another method of multiplexing several signals is known as **time division multiplexing (TDM)**. This method is suitable when a signal is in the form of a pulse train (as in PCM). When the pulses are made narrower, the spaces left between pulses of one user signal are used for pulses from other signals. Thus, in effect, the transmission time is shared by a number of signals by interleaving the pulse trains of various signals in a specified order. At the receiver, the pulse trains corresponding to various signals are separated.

1.4.3 Demodulation

Once multiple modulated signals have arrived at the receiver, the desired signal must be detected and recovered into its original baseband form. Note that because of FDM, the first stage of a demodulator typically requires a tunable bandpass filter so that the receiver can select the modulated signal at a predetermined frequency band specified by the transmission station or channel. Once a particular modulated signal has been isolated, the demodulator will then need to convert the carrier variation of amplitude, frequency, or phase, back into the baseband signal voltage.

For the three basic modulation schemes of AM, FM, and PM, the corresponding demodulators must be designed such that the detector output voltage varies in proportion to the input modulated signal's amplitude, frequency, and phase, respectively. Once circuits with such response characteristics have been implemented, the demodulators can downconvert the modulated (RF) signals back into the baseband signals that represent the original source message, be it audio, video, or data.

1.5 DIGITAL SOURCE CODING AND ERROR CORRECTION CODING

As stated earlier, SNR and bandwidth are two factors that determine the performance of a given communication. Unlike analog communication systems, digital systems often adopt aggressive measures to lower the source data rate and to fight against channel noise. In particular, *source coding* is applied to generate the fewest bits possible for a given message without sacrificing its detection accuracy. On the other hand, to combat errors that arise from noise and interferences, *redundancy* needs to be introduced systematically at the transmitter, such that the receivers can rely on the redundancy to correct errors caused by channel distortion and noise. This process is known as error correction coding by the transmitter and decoding by the receiver.

Source coding and error correction coding are two successive stages in a digital communication system that work in a see-saw battle. On one hand, the job of source coding is to remove as much redundancy from the message as possible to shorten the digital message sequence that requires transmission. Source coding aims to use as little bandwidth as possible without considering channel noise and interference. On the other hand, error correction coding intentionally introduces redundancy intelligently, such that if errors occur upon detection, the redundancy can help correct the most likely errors.

Randomness, Redundancy, and Source Coding

To understand source coding, it is important to first discuss the role of *randomness* in communications. As noted earlier, channel noise is a major factor limiting communication performance because it is random and cannot be removed by prediction. On other other hand, randomness is also closely associated with the desired signals in communications. Indeed, randomness is the essence of communication. Randomness means unpredictability, or uncertainty, of a source message. If a source had no unpredictability, like a friend who always wants to repeat the same story on "how I was abducted by an alien," then the information would be known beforehand and would contain no information. Similarly, if a person winks, it conveys some information in a given context. But if a person winks continuously with the regularity of a clock, the winks convey no information. In short, a predictable signal is not random and is fully redundant. Thus, a message contains information only if it is unpredictable. Higher predictability means higher redundancy and, consequently, less information. Conversely, more unpredictable or less likely random signals contain more information.

Source coding reduces redundancy based on the predictability of the message source. The objective of source coding is to use codes that are as short as possible to represent the source signal. Shorter codes are more efficient because they require less time to transmit at a given data rate. Hence, source coding should remove signal redundancy while encoding and transmitting the unpredictable, random part of the signal. The more predictable messages contain more redundancy and require shorter codes, while messages that are less likely contain more information and should be encoded with longer codes. By assigning more likely messages with shorter source codes and less likely messages with longer source codes, one obtains more efficient source coding. Consider the Morse code, for example. In this code, various combinations of dashes and dots (code words) are assigned to each letter. To minimize transmission time, shorter code words are assigned to more frequently occurring (more probable) letters (such as e, t, and a) and longer code words are assigned to rarely occurring (less probable) letters (such as x, q, and z). Thus, on average, messages in English would tend to follow a known letter distribution, thereby leading to shorter code sequences that can be quickly transmitted. This explains why Morse code is a good source code.

It will be shown in Chapter 13 that for digital signals, the overall transmission time is minimized if a message (or symbol) of probability P is assigned a code word with a length proportional to $\log(1/P)$. Hence, from an engineering point of view, the information of a message with probability P is proportional to $\log(1/P)$. This is known as entropy (source) coding.

Error Correction Coding

Error correction coding also plays an important role in communication. While source coding removes redundancy, error correction codes add redundancy. The systematic introduction of redundancy supports reliable communication.[4] Because of redundancy, if certain bits are in error due to noise or interference, other related bits may help them recover, allowing us to decode a message accurately despite errors in the received signal. All languages are redundant. For example, English is about 50% redundant; that is, on the average, we may throw out half the letters or words without losing the meaning of a given message. This also means that in any English message, the speaker or the writer has free choice over half the letters or words, on the average. The remaining half is determined by the statistical structure of the language. If all the redundancy of English were removed, it would take about half the time to transmit a telegram or telephone conversation. If an error occurred at the receiver, however, it would be rather difficult to make sense out of the received message. The redundancy in a message, therefore, plays a useful role in combating channel noises and interferences.

It may appear paradoxical that in source coding we would remove redundancy, only to add more redundancy at the subsequent error correction coding. To explain why this is sensible, consider the removal of all redundancy in English through source coding. This would shorten the message by 50% (for bandwidth saving). However, for error correction, we may restore some systematic redundancy, except that this well-designed redundancy is only half as long as what was removed by source coding while still providing the same amount of error protection. It is therefore clear that a good combination of source coding and error correction coding can remove inefficient redundancy without sacrificing error correction. In fact, a very popular problem in this field is the persistent pursuit of *joint source-channel coding* that can maximally remove signal redundancy without losing error correction.

How redundancy can enable error correction can be seen with an example: to transmit samples with $L = 16$ quantizing levels, we may use a group of four binary pulses, as shown in Fig. 1.5. In this coding scheme, no redundancy exists. If an error occurs in the reception of even one of the pulses, the receiver will produce a wrong value. Here we may use redundancy to eliminate the effect of possible errors caused by channel noise or imperfections. Thus, if we add to each code word one more pulse of such polarity as to make the number of positive pulses even, we have a code that can detect a single error in any place. Thus, to the code word **0001** we add a fifth pulse, of positive polarity, to make a new code word, **00011**. Now the number of positive pulses is 2 (even). If a single error occurs in any position, this parity will be violated. The receiver knows that an error has been made and can request retransmission of the message. This is a very simple coding scheme. It can only detect an error; it cannot locate or correct it. Moreover, it cannot detect an even number of errors. By introducing more redundancy, it is possible not only to detect but also to correct errors. For example, for $L = 16$, it can be shown that properly adding three pulses will not only detect but also correct a single error occurring at any location. Details on the subject of error correcting codes will be discussed in Chapter 14.

1.6 A BRIEF HISTORICAL REVIEW OF MODERN TELECOMMUNICATIONS

Telecommunications (literally: communications at a distance) are always critical to human society. Even in ancient times, governments and military units relied heavily on telecommunications to gather information and to issue orders. The first type was with messengers on foot or on horseback; but the need to convey a short message over a large distance (such as one warning a city of approaching raiders) led to the use of fire and smoke signals. Using signal mirrors to reflect sunlight (heliography), was another effective way of telecommunication. Its first recorded use was in ancient Greece. Signal mirrors were also mentioned in Marco Polo's account of his trip to the Far East.[1] These ancient *visual* communication technologies are, amazingly enough, digital. Fires and smoke in different configurations would form different codewords. On hills or mountains near Greek cities there were also special personnel for such communications, forming a chain of regenerative repeaters. In fact, fire and smoke signal platforms still dot the Great Wall of China. More interestingly, reflectors or lenses, equivalent to the amplifiers and antennas we use today, were used to directionally guide the light farther.

Naturally, these early *visual* communication systems were very tedious to set up and could transmit only several bits of information per hour. A much faster visual communication system was developed just over two centuries ago. In 1793 Claude Chappe of France invented and performed a series of experiments on the concept of "semaphore telegraph." His system was a series of signaling devices called semaphores, which were mounted on towers, typically spaced

10 km apart. (A semaphore looked like a large human figure with signal flags in both hands.) A receiving semaphore operator would transcribe visually, often with the aid of a telescope, and then relay the message from his tower to the next, and so on. This visual telegraph became the government telecommunication system in France and spread to other countries, including the United States. The semaphore telegraph was eventually eclipsed by electric telegraphy. Today, only a few remaining streets and landmarks with the name "Telegraph Hill" remind us of the place of this system in history. Still, visual communications (via Aldis lamps, ship flags, and heliographs) remained an important part of maritime communications well into the twentieth century.

These early telecommunication systems are optical systems based on visual receivers. Thus, they can cover only line-of-sight distance, and human operators are required to decode the signals. An important event that changed the history of telecommunication occurred in 1820, when Hans Christian Oersted of Denmark discovered the interaction between electricity and magnetism.[2] **Michael Faraday** made the next crucial discovery, which **changed the history of both electricity and telecommunications,** when he found that electric current can be induced on a conductor by a changing magnetic field. Thus, electricity generation became possible by magnetic field motion. Moreover, the transmission of electric signals became possible by varying an electromagnetic field to induce current change in a distant circuit. The amazing aspect of Faraday's discovery on current induction is that it provides the foundation for wireless telecommunication over distances without line-of-sight, and more importantly, it shows how to generate electricity as an energy source to power such systems. The invention of the electric telegraph soon followed, and the world entered the modern electric telecommunication era.

Modern communication systems have come a long way from their infancy. Since it would be difficult to detail all the historical events that mark the recent development of telecommunication, we shall instead use Table 1.1 to chronicle some of the most notable events in the development of modern communication systems. Since our focus is on electrical telecommunication, we shall refrain from reviewing the equally long history of optical (fiber) communications.

It is remarkable that all the early telecommunication systems are symbol-based digital systems. It was not until Alexander Graham Bell's invention of the telephone system that analog **live signals** were transmitted. Live signals can be instantly heard or seen by the receiving users. The Bell invention that marks the beginning of a new (analog communication) era is therefore a major milestone in the history of telecommunications. Figure 1.7 shows a copy of an illustration from Bell's groundbreaking 1876 telephone patent. Scientific historians often hail this invention as the *most valuable* patent ever issued in history.

The invention of telephone systems also marks the beginning of the analog communication era and live signal transmission. On an exciting but separate path, wireless communication began in 1887, when Heinrich Hertz first demonstrated a way to detect the presence of electromagnetic waves. French scientist Edouard Branly, English physicist Oliver Lodge, and Russian inventor Alexander Popov all made important contributions to the development of radio receivers. Another important contributor to this area was the Croatian-born genius Nikola Tesla. Building upon earlier experiments and inventions, Italian scientist and inventor Guglielmo Marconi developed a wireless telegraphy system in 1895 for which he shared the Nobel Prize in Physics in 1909. Marconi's wireless telegraphy marked a historical event of commercial wireless communications. Soon, the marriage of the inventions of Bell and Marconi allowed analog audio signals to go wireless, thanks to amplitude modulation (AM) technology. Quality music transmission via FM radio

TABLE 1.1
Important Events of the Past Two Centuries of Telecommunications

Year	Major Events
1820	**First experiment of electric current causing magnetism (by Hans C. Oersted)**
1831	**Discovery of induced current from electromagnetic radiation (by Michael Faraday)**
1830–32	Birth of telegraph (credited to Joseph Henry and Pavel Schilling)
1837	Invention of Morse code by Samuel F. B. Morse
1864	Theory of electromagnetic waves developed by James C. Maxwell
1866	First transatlantic telegraph cable in operation
1876	**Invention of telephone by Alexander G. Bell**
1878	First telephone exchange in New Haven, Connecticut
1887	**Detection of electromagnetic waves by Heinrich Hertz**
1896	**Wireless telegraphy (radio telegraphy) patented by Guglielmo Marconi**
1901	First transatlantic radio telegraph transmission by Marconi
1906	**First amplitude modulation radio broadcasting (by Reginald A. Fessenden)**
1907	Regular transatlantic radio telegraph service
1915	First transcontinental telephone service
1920	First commercial AM radio stations
1921	Mobile radio adopted by Detroit Police Department
1925	First television system demonstration (by Charles F. Jenkins)
1928	First television station W3XK in the United States
1935	**First FM radio demonstration (by Edwin H. Armstrong)**
1941	NTSC black and white television standard
	First commercial FM radio service
1947	**Cellular concept first proposed at Bell Labs**
1948	**First major information theory paper published by Claude E. Shannon**
	Invention of transistor by William Shockley, Walter Brattain, and John Bardeen
1949	The construction of Golay code for 3 (or fewer) bit error correction
1950	Hamming codes constructed for simple error corrections
1953	NTSC color television standard
1958	**Integrated circuit proposed by Jack Kilby (Texas Instruments)**
1960	**Construction of the powerful Reed-Solomon error correcting codes**
1962	First computer telephone modem developed: Bell Dataphone 103A (300 bit/s)
1962	Low-density parity check error correcting codes proposed by Robert G. Gallager
1968–9	First error correction encoders on board NASA space missions (Pioneer IX and Mariner VI)
1971	**First wireless computer network: AlohaNet**
1973	First portable cellular telephone demonstration to the U.S. Federal Communications Commission, by Motorola
1978	First mobile cellular trial by AT&T
1984	First handheld (analog) AMPS cellular phone service by Motorola
1989	Development of DSL modems for high-speed computer connections
1991	First (digital) GSM cellular service launched (Finland)
	First wireless local area network (LAN) developed (AT&T-NCR)
1993	Digital ATSC standard established
1993	Turbo codes proposed by Berrou, Glavieux, and Thitimajshima
1996	First commercial CDMA (IS-95) cellular service launched
	First HDTV broadcasting
1997	IEEE 802.11(b) wireless LAN standard
1998	Large-scope commercial ADSL deployment
1999	IEEE 802.11a wireless LAN standard
2000	First 3G cellular service launched
2003	IEEE 802.11g wireless LAN standard

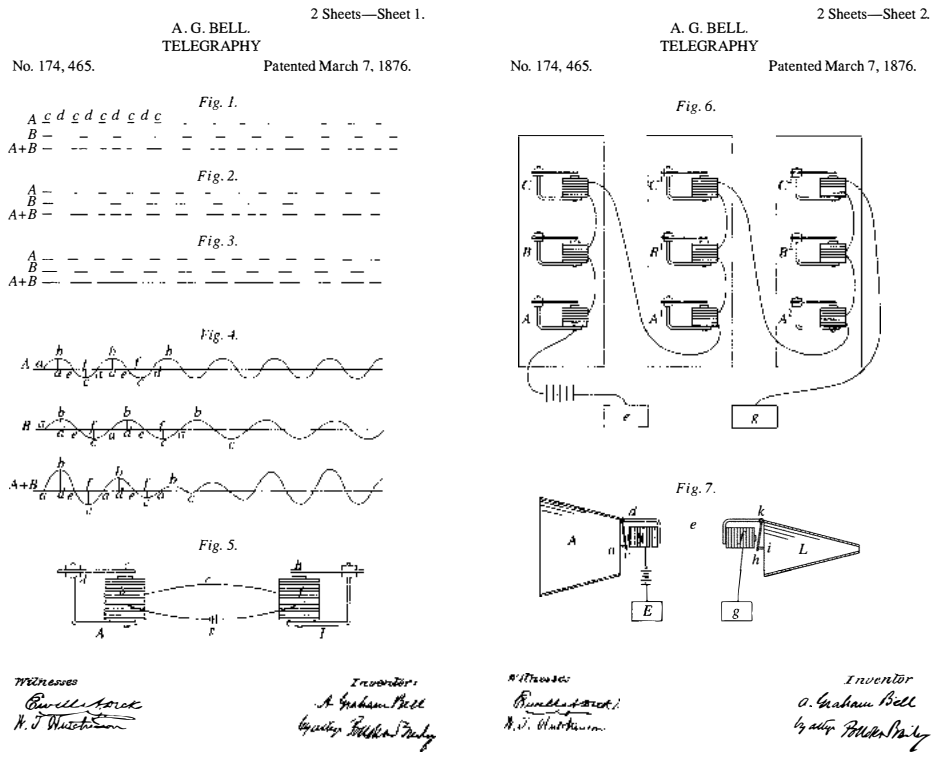

2 Sheets—Sheet 1.
A. G. BELL.
TELEGRAPHY
No. 174,465. Patented March 7, 1876.
Fig. 1.
Fig. 2.
Fig. 3.
Fig. 4.
Fig. 5.
Witnesses
Inventor:
A. Graham Bell
2 Sheets—Sheet 2.
A. G. BELL.
TELEGRAPHY
No. 174,465. Patented March 7, 1876.
Fig. 6.
Fig. 7.
Witnesses:
Inventor
A. Graham Bell

Figure 1.7 Illustration from Bell's U.S. Patent No. 174,465 issued March 7, 1876. (From the U.S. Patent and Trademark Office.)

broadcast was first demonstrated by American inventor Major Edwin H. Armstrong. Armstrong's FM demonstration in 1935 took place at an IEEE meeting in New York's Empire State Building.

A historic year for both communications and electronics was 1948, the year that witnessed the rebirth of digital communications and the invention of semiconductor transistors. The rebirth of digital communications is owing to the originality and brilliance of Claude E. Shannon, widely known as the father of modern digital communication and information theory. In two seminal articles published in 1948, he first established the fundamental concept of channel capacity and its relation to information transmission rate. Deriving the channel capacity of several important models, Shannon[3] proved that as long as the information is transmitted through a channel at a rate below the channel capacity, error-free communications can be possible. Given noisy channels, Shannon showed the existence of good codes that can make the probability of transmission error arbitrarily small. This noisy channel coding theorem gave rise to the modern field of error correcting codes. Coincidentally, the invention of the first transistor in the same year (by Bill Shockley, Walter Brattain, and John Bardeen) paved the way to the design and implementation of more compact, more powerful, and less noisy circuits to put Shannon's theorems into practical use. The launch of Mariner IX Mars orbiter in March of 1971 was the first NASA mission officially equipped with error correcting codes, which reliably transmitted photos taken from Mars.

Today, we are in an era of digital and multimedia communications, marked by the widespread applications of computer networking and cellular phones. The first telephone modem for home computer connection to a mainframe was developed by AT&T Bell Labs in 1962. It uses an acoustic coupler to interface with a regular telephone handset. The acoustic coupler converts the local computer data into audible tones and uses the regular telephone microphone

to transmit the tones over telephone lines. The coupler receives the mainframe computer data via the telephone headphone and converts them into bits for the local computer terminal, typically at rates below 300 bit/s. Rapid advances in integrated circuits (first credited to Jack Kilby in 1958) and digital communication technology dramatically increased the link rate to 56 kbit/s by the 1990s. By 2000, wireless local area network (WLAN) modems were developed to connect computers at speed up to 11 Mbit/s. These commercial WLAN modems, the size of a credit card, were first standardized as IEEE 802.11b.

Technological advances also dramatically reshaped the cellular systems. While the cellular concept was developed in 1947 at Bell Labs, commercial systems were not available until 1983. The "mobile" phones of the 1980s were bulky and expensive, mainly used for business. The world's first cellular phone, developed by Motorola in 1983 and known as DynaTAC 8000X, weighed 28 ounces, earning the nickname of "brick" and costing $3995. These analog phones are basically two-way FM radios for voice only. Today, a cellphone is truly a multimedia, multifunctional device that is useful not only for voice communication but also can send and receive e-mail, access websites, and display videos. Cellular devices are now very small, weighing no more than a few ounces. Unlike in the past, cellular phones are now for the masses. In fact, Europe now has more cellphones than people. In Africa, 13% of the adult population now owns a cellular phone.

Throughout history, the progress of human civilization has been inseparable from technological advances in telecommunications. Telecommunications played a key role in almost every major historical event. It is not an exaggeration to state that telecommunications helped shape the very world we live in today and will continue to define our future. It is therefore the authors' hope that this text can help stimulate the interest of many students in telecommunication technologies. By providing the fundamental principles of modern digital and analog communication systems, the authors hope to provide a solid foundation for the training of future generations of communication scientists and engineers.

REFERENCES

1. M. G. Murray, "Aimable Air/Sea Rescue Signal Mirrors," *The Bent of Tau Beta Pi*, pp. 29–32, Fall 2004.
2. B. Bunch and A. Hellemans, Eds., *The History of Science and Technology: A Browser's Guide to the Great Discoveries, Inventions, and the People Who Made Them from the Dawn of Time to Today*, Houghton Mifflin, Boston, 2004.
3. C. E. Shannon, "A Mathematical Theory of Communications," *Bell Syst. Tech. J.* part I: pp. 379–423; part II: 623–656, July 1948.
4. S. Lin and D. J. Costello Jr., *Error Control Coding*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 2004.

2 SIGNALS AND SIGNAL SPACE

In this chapter we discuss certain basic signal concepts. Signals are processed by systems. We shall start with explaining the terms *signals* and *systems*.

Signals

A signal, as the term implies, is a set of information or data. Examples include a telephone or a television signal, the monthly sales figures of a corporation, and closing stock prices (e.g., in the United States, the Dow Jones averages). In all these examples, the signals are functions of the independent variable *time*. This is not always the case, however. When an electrical charge is distributed over a surface, for instance, the signal is the charge density, a function of *space* rather than time. In this book we deal almost exclusively with signals that are functions of time. The discussion, however, applies equally well to other independent variables.

Systems

Signals may be processed further by **systems**, which may modify them or extract additional information from them. For example, an antiaircraft missile launcher may want to know the future location of a hostile moving target, which is being tracked by radar. Since the radar signal gives the past location and velocity of the target, by properly processing the radar signal (the input), one can approximately estimate the future location of the target. Thus, a system is an entity that *processes* a set of signals (**inputs**) to yield another set of signals (**outputs**). A system may be made up of physical components, as in electrical, mechanical, or hydraulic systems (hardware realization), or it may be an algorithm that computes an output from an input signal (software realization).

2.1 SIZE OF A SIGNAL

Signal Energy

The size of any entity is a quantity that indicates its strength. Generally speaking, a signal varies with time. To set a standard quantity that measures signal strength, we normally view a signal $g(t)$ as a voltage across a one-ohm resistor. We define **signal energy** E_g of the signal $g(t)$ as the energy that the voltage $g(t)$ dissipates on the resistor. More formally, we define E_g

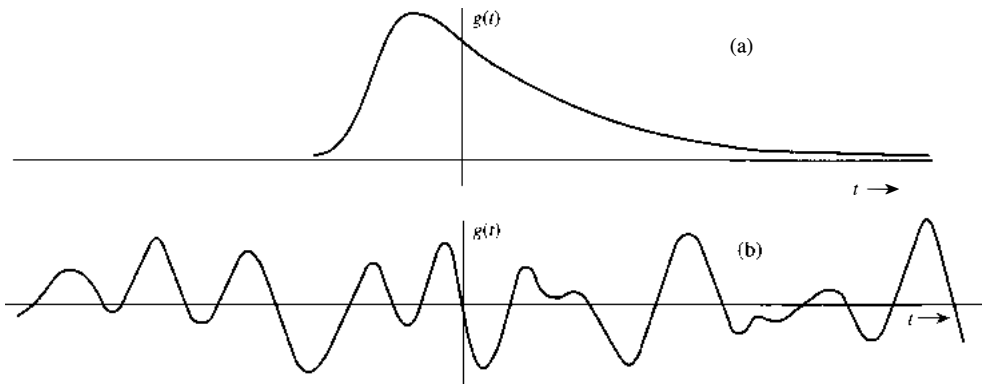


Figure 2.1 Examples of signals. (a) Signal with finite energy. (b) Signal with finite power.

(for a real signal) as

$$E_g = \int_{-\infty}^{\infty} g^2(t)\,dt \tag{2.1}$$

This definition can be generalized to a complex-valued signal $g(t)$ as

$$E_g = \int_{-\infty}^{\infty} |g(t)|^2\,dt \tag{2.2}$$

Signal Power

To be a meaningful measure of signal size, the signal energy must be finite. A necessary condition for energy to be finite is that the signal amplitude goes to zero as $|t|$ approaches infinity (Fig. 2.1a). Otherwise the integral in Eq. (2.1) will not converge.

If the amplitude of $g(t)$ does not go to zero as $|t|$ approaches infinity (Fig. 2.1b), the signal energy is infinite. A more meaningful measure of the signal size in such a case would be the time average of the energy (if it exists), which is the average power P_g defined (for a real signal) by

$$P_g = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} g^2(t)\,dt \tag{2.3}$$

We can generalize this definition for a complex signal $g(t)$ as

$$P_g = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} |g(t)|^2\,dt \tag{2.4}$$

Observe that the signal power P_g is the time average (mean) of the signal amplitude square, that is, the **mean square** value of $g(t)$. Indeed, the square root of P_g is the familiar **rms** (root mean square) value of $g(t)$.

The mean of an entity averaged over a large time interval approaching infinity exists if the entity either is periodic or has a statistical regularity. If such a condition is not satisfied, an average may not exist. For instance, a ramp signal $g(t) = t$ increases indefinitely as $|t| \to \infty$, and neither the energy, nor the power exists for this signal.

Units of Signal Energy and Power

The standard units of signal energy and power are the joule and the watt. However, in practice, it is often customary to use logarithmic scales to describe signal power. This notation saves

the trouble of dealing with many decimal places when signal power is large or small. As a convention, a signal with average power of P watts can be said to have power of

$$[10 \cdot \log_{10} P]\,\text{dBw} \qquad \text{or} \qquad [30 + 10 \cdot \log_{10} P]\,\text{dBm}$$

For example, -30 dBm represents signal power of 10^{-6} W in normal decimal scale.

Example 2.1 Determine the suitable measures of the signals in Fig. 2.2.

The signal in Fig. 2.2a approaches 0 as $|t| \to \infty$. Therefore, the suitable measure for this signal is its energy E_g, given by

$$E_g = \int_{-\infty}^{\infty} g^2(t)\,dt = \int_{-1}^{0} (2)^2\,dt + \int_{0}^{\infty} 4e^{-t}\,dt = 4 + 4 = 8$$

The signal in Fig. 2.2b does not approach 0 as $|t| \to \infty$. However, it is periodic, and therefore its power exists. We can use Eq. (2.3) to determine its power. For periodic signals, we can simplify the procedure by observing that a periodic signal repeats regularly each period (2 seconds in this case). Therefore, averaging $g^2(t)$ over an infinitely large interval is equivalent to averaging it over one period (2 seconds in this case). Thus

$$P_g = \frac{1}{2}\int_{-1}^{1} g^2(t)\,dt = \frac{1}{2}\int_{-1}^{1} t^2\,dt = \frac{1}{3}$$

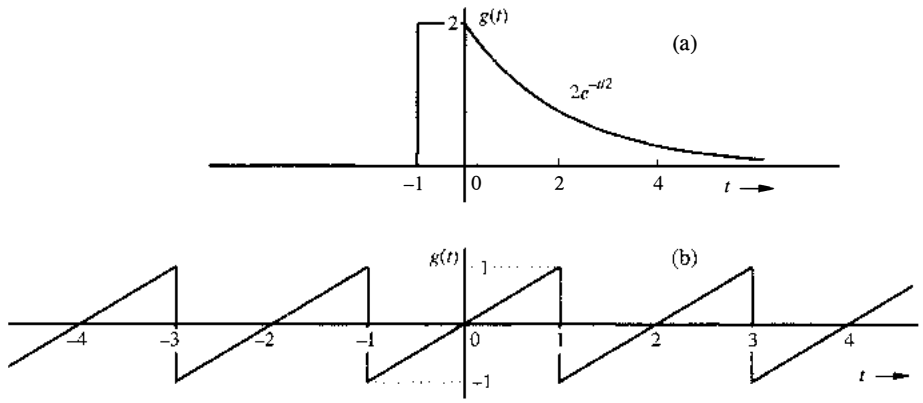


Figure 2.2 Signal for Example 2.1.

Recall that the signal power is the square of its rms value. Therefore, the rms value of this signal is $1/\sqrt{3}$.

2.2 CLASSIFICATION OF SIGNALS

There are various classes of signals. Here we shall consider only the following pairs of classes, which are suitable for the scope of this book.

1. Continuous time and discrete time signals
2. Analog and digital signals

3. Periodic and aperiodic signals
4. Energy and power signals
5. Deterministic and probabilistic signals

2.2.1 Continuous Time and Discrete Time Signals

A signal that is specified for every value of time t (Fig. 2.3a) is a **continuous time signal**, and a signal that is specified only at discrete points of $t = nT$ (Fig. 2.3b) is a **discrete time signal**. Audio and video recordings are continuous time signals, whereas the quarterly gross domestic product (GDP), monthly sales of a corporation, and stock market daily averages are discrete time signals.

2.2.2 Analog and Digital Signals

One should not confuse analog signals with continuous time signals. The two concepts are not the same. This is also true of the concepts of discrete time and digital. A signal whose amplitude can take on any value in a continuous range is an **analog signal**. This means that

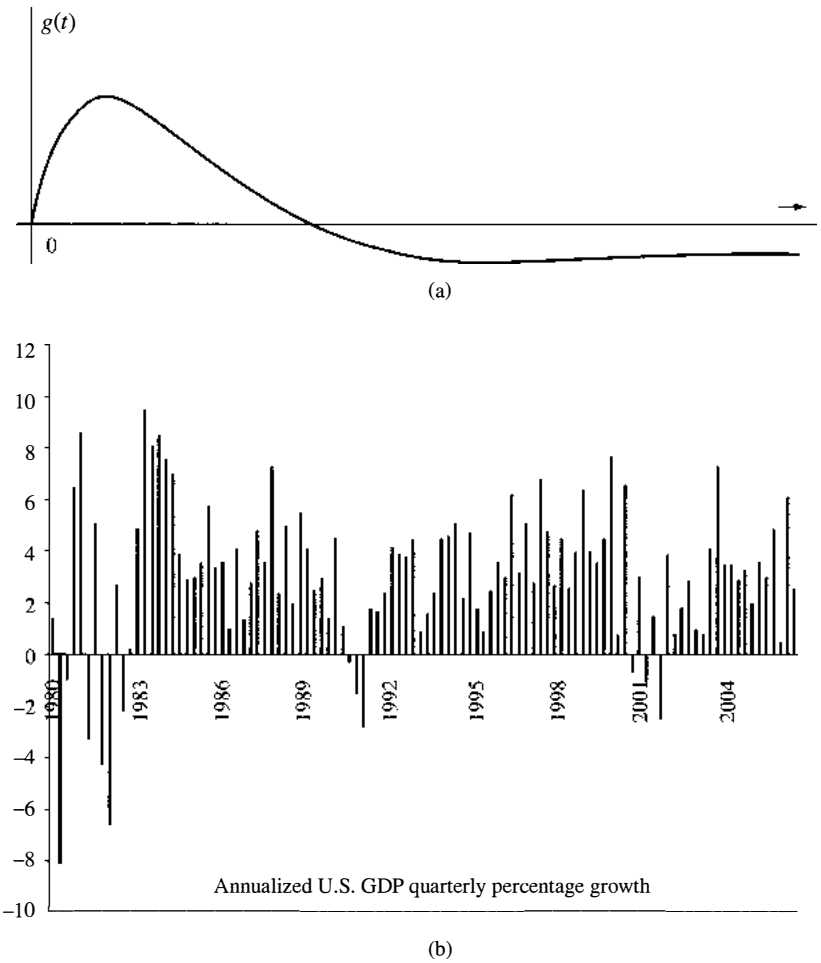


Figure 2.3
(a) Continuous time signal.
(b) Discrete time signals.

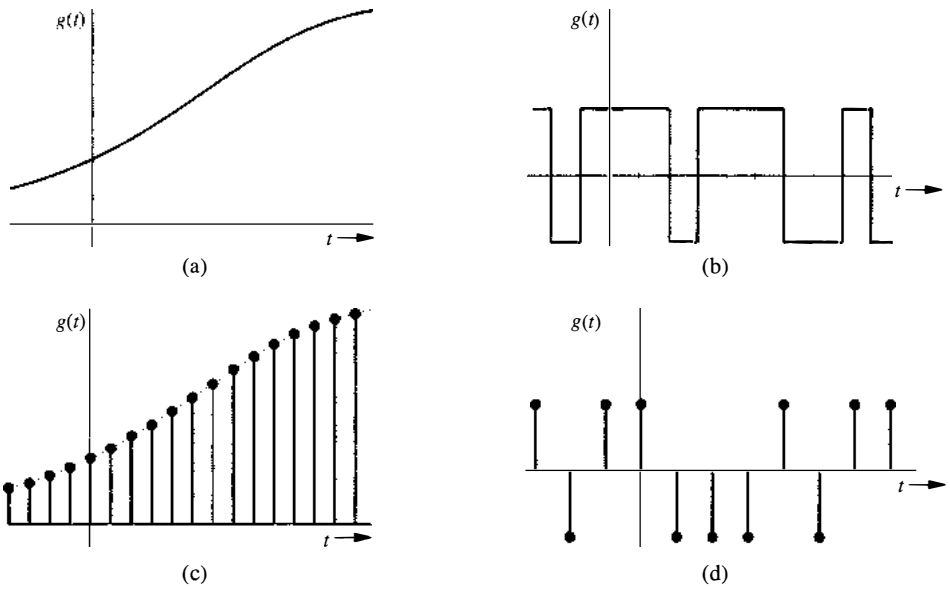


Figure 2.4 Examples of signals: (a) analog and continuous time, (b) digital and continuous time, (c) analog and discrete time, (d) digital and discrete time.

an analog signal amplitude can take on an (uncountably) infinite number of values. A **digital signal**, on the other hand, is one whose amplitude can take on only a finite number of values. Signals associated with a digital computer are digital because they take on only two values (binary signals). For a signal to qualify as digital, the number of values need not be restricted to two. It can be any finite number. A digital signal whose amplitudes can take on M values is an M-**ary** signal of which binary ($M = 2$) is a special case. The terms "continuous time" and "discrete time" qualify the nature of signal along the time (horizontal) axis. The terms "analog" and "digital," on the other hand, describe the nature of the signal amplitude (vertical) axis. Figure 2.4 shows examples of signals of various types. It is clear that analog is not necessarily continuous time, whereas digital need not be discrete time. Figure 2.4c shows an example of an analog but discrete time signal. An analog signal can be converted into a digital signal (via analog-to-digital, or A/D, conversion) through quantization (rounding off), as explained in Chapter 6.

2.2.3 Periodic and Aperiodic Signals

A signal $g(t)$ is said to be **periodic** if there exists a positive constant T_0 such that

$$g(t) = g(t + T_0) \quad \text{for all } t \tag{2.5}$$

The **smallest** value of T_0 that satisfies the periodicity condition of Eq. (2.5) is the **period** of $g(t)$. The signal in Fig. 2.2b is a periodic signal with period of 2. Naturally, a signal is **aperiodic** if it is not periodic. The signal in Fig. 2.2a is aperiodic.

By definition, a periodic signal $g(t)$ remains unchanged when time-shifted by one period. This means that a periodic signal must start at $t = -\infty$ because if it starts at some finite instant, say, $t = 0$, the time-shifted signal $g(t + T_0)$ will start at $t = -T_0$ and $g(t + T_0)$ would not be the same as $g(t)$. Therefore, a **periodic signal, by definition, must start from $-\infty$ and continue forever**, as shown in Fig. 2.5. Observe that a periodic signal shifted by an integral multiple of T_0 remains unchanged. Therefore, $g(t)$ may be considered to be a periodic signal

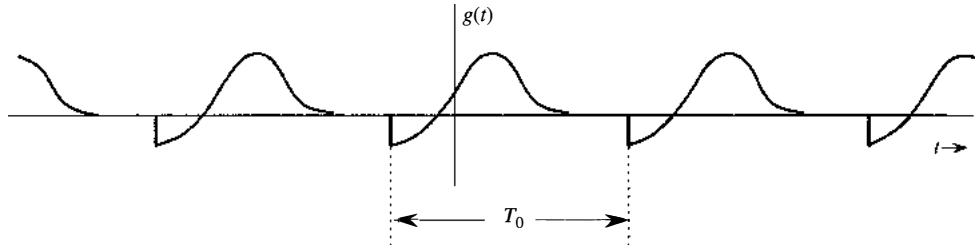


Figure 2.5 A periodic signal of period T_0.

with period mT_0, where m is any integer. However, by definition, the period is the smallest interval that satisfies periodicity condition of Eq. (2.5). Therefore, T_0 is the period.

2.2.4 Energy and Power Signals

A signal with finite energy is an **energy signal**, and a signal with finite power is a **power signal**. In other words, a signal $g(t)$ is an energy signal if

$$\int_{-\infty}^{\infty} |g(t)|^2 \, dt < \infty \tag{2.6}$$

Similarly, a signal with a finite and nonzero power (mean square value) is a power signal. In other words, a signal is a power signal if

$$0 < \lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} |g(t)|^2 \, dt < \infty \tag{2.7}$$

The signals in Fig. 2.2a and 2.2b are examples of energy and power signals, respectively. Observe that power is time average of the energy. Since the averaging is over an infinitely large interval, a signal with finite energy has zero power, and a signal with finite power has infinite energy. Therefore, a signal cannot be both an energy and a power signal. If it is one, it cannot be the other. On the other hand, some signals with infinite power are neither energy nor power signals. The ramp signal is one example.

Comments

Every signal observed in real life is an energy signal. A power signal, on the other hand, must have an infinite duration. Otherwise its power, which is its average energy (averaged over infinitely large interval) will not approach a (nonzero) limit. Obviously it is impossible to generate a true power signal in practice because such a signal would have infinite duration and infinite energy.

Also, because of periodic repetition, periodic signals for which the area under $|g(t)|^2$ over one period is finite are power signals; however, not all power signals are periodic.

2.2.5 Deterministic and Random Signals

A signal whose physical description is known completely, either in a mathematical form or a graphical form is a **deterministic signal**. A signal that is known only in terms of probabilistic description, such as mean value, mean square value, and distributions, rather than its full mathematical or graphical description is a **random signal**. Most of the noise signals encountered in practice are random signals. All message signals are random signals because, as will be shown



Figure 2.6 A unit impulse and its approximation.

later, a signal, to convey information, must have some uncertainty (randomness) about it. The treatment of random signals will be discussed in later chapters.

2.3 UNIT IMPULSE SIGNAL

The unit impulse function $\delta(t)$ is one of the most important functions in the study of signals and systems. Its definition and application provide much convenience that is not permissible in pure mathematics.

The unit impulse function $\delta(t)$ was first defined by P. A. M. Dirac (hence often known as the "Dirac delta") as

$$\delta(t) = 0, \quad t \neq 0 \tag{2.8}$$

$$\int_{-\infty}^{\infty} \delta(t)\, dt = 1 \tag{2.9}$$

We can visualize an impulse as a tall, narrow rectangular pulse of unit area, as shown in Fig. 2.6. The width of this rectangular pulse is a very small value ϵ; its height is a very large value $1/\epsilon$ in the limit as $\epsilon \to 0$. The unit impulse therefore can be regarded as a rectangular pulse with a width that has become infinitesimally small, a height that has become infinitely large, and an overall area that remains constant at unity.* Thus, $\delta(t) = 0$ everywhere except at $t = 0$, where it is, strictly speaking, undefined. For this reason, a unit impulse is graphically represented by the spearlike symbol in Fig. 2.6a.

Multiplication of a Function by an Impulse

Let us now consider what happens when we multiply the unit impulse $\delta(t)$ by a function $\phi(t)$ that is known to be continuous at $t = 0$. Since the impulse exists only at $t = 0$, and the value of $\phi(t)$ at $t = 0$ is $\phi(0)$, we obtain

$$\phi(t)\delta(t) = \phi(0)\delta(t) \tag{2.10a}$$

Similarly, if $\phi(t)$ is multiplied by an impulse $\delta(t - T)$ (an impulse located at $t = T$), then

$$\phi(t)\delta(t - T) = \phi(T)\delta(t - T) \tag{2.10b}$$

provided $\phi(t)$ is defined at $t = T$.

* The impulse function can also be approximated by other pulses, such as a positive triangle, an exponential pulse, or a Gaussian pulse.

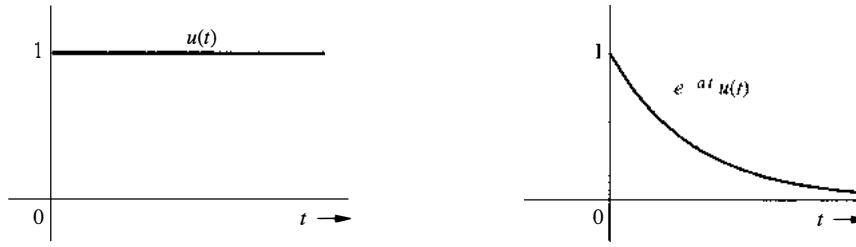


Figure 2.7 (a) Unit step function $u(t)$. (b) Causal exponential $e^{-at}u(t)$.

The Sampling Property of the Unit Impulse Function

From Eq. (2.10) it follows that

$$\int_{-\infty}^{\infty} \phi(t)\delta(t-T)\,dt = \phi(T)\int_{-\infty}^{\infty} \delta(t-T)\,dt = \phi(T) \tag{2.11a}$$

provided $\phi(t)$ is continuous at $t = T$. This result means that *the area under the product of a function with an impulse $\delta(t)$ is equal to the value of that function at the instant where the unit impulse is located.* This very important and useful property is known as the **sampling** (or **sifting**) **property** of the unit impulse.

Depending on the value of T and the integration limit, the impulse function may or may not be within the integration limit. Thus, it follows that

$$\int_a^b \phi(t)\delta(t-T)\,dt = \phi(T)\int_a^b \delta(t-T)\,dt = \begin{cases} \phi(T) & a \le T < b \\ 0 & T < a \le b, \text{ or } T \ge b > a \end{cases} \tag{2.11b}$$

The Unit Step Function $u(t)$

Another familiar and useful function is the **unit step function** $u(t)$, often encountered in circuit analysis and defined by Fig. 2.7a:

$$u(t) = \begin{cases} 1 & t \ge 0 \\ 0 & t < 0 \end{cases} \tag{2.12}$$

If we want a signal to start at $t = 0$ (so that it has a value of zero for $t < 0$), we need only multiply the signal by $u(t)$. A signal that starts after $t = 0$ is called a **causal signal**. In other words, $g(t)$ is a causal signal if

$$g(t) = 0 \qquad t < 0$$

The signal e^{-at} represents an exponential that starts at $t = -\infty$. If we want this signal to start at $t = 0$ (the causal form), it can be described as $e^{-at}u(t)$ (Fig. 2.7b). From Fig. 2.6b, we observe that the area from $-\infty$ to t under the limiting form of $\delta(t)$ is zero if $t < 0$ and unity if $t \ge 0$. Consequently,

$$\int_{-\infty}^{t} \delta(\tau)\,d\tau = \begin{cases} 0, & t < 0 \\ 1, & t \ge 0 \end{cases}$$
$$= u(t) \tag{2.13a}$$

From this result it follows that

$$\frac{du}{dt} = \delta(t) \tag{2.13b}$$

2.4 SIGNALS VERSUS VECTORS

There is a strong connection between signals and vectors. Signals that are defined for only a finite number of time instants (say N) can be written as vectors (of dimension N). Thus, consider a signal $g(t)$ defined over a closed time interval $[a, b]$. Let we pick N points uniformly on the time interval $[a, b]$ such that

$$t_1 = a, \quad t_2 = a + \epsilon, \quad t_3 = a + 2\epsilon, \quad t_N = a + (N-1)\epsilon = b, \qquad \epsilon = \frac{b-a}{N-1}$$

Then we can write a signal vector $\mathbf{g}$ as an N-dimensional vector

$$\mathbf{g} = \begin{bmatrix} g(t_1) & g(t_2) & \cdots & g(t_N) \end{bmatrix}$$

As the number of time instants N increases, the sampled signal vector $\mathbf{g}$ will grow. Eventually, as $N \to \infty$, the signal values will form a vector $\mathbf{g}$ of infinitely long dimension. Because $\epsilon \to 0$, the signal vector $\mathbf{g}$ will transform into the continuous-time signal $g(t)$ defined over the interval $[a, b]$. In other words,

$$\lim_{N \to \infty} \mathbf{g} = g(t) \quad t \in [a, b]$$

This relationship clearly shows that continuous time signals are straightforward generalizations of finite dimension vectors. Thus, basic definitions and operations in a vector space can be applied to continuous time signals as well. We now highlight this connection between the finite dimension vector space and the continuous time signal space.

We shall denote all vectors by boldface type. For example, $\mathbf{x}$ is a certain vector with magnitude or length $||\mathbf{x}||$. A vector has magnitude and direction. In a vector space, we can define the inner (dot or scalar) product of two real-valued vectors $\mathbf{g}$ and $\mathbf{x}$ as

$$< \mathbf{g}, \mathbf{x} >= ||\mathbf{g}|| \cdot ||\mathbf{x}|| \cos\theta \tag{2.14}$$

where θ is the angle between vectors $\mathbf{g}$ and $\mathbf{x}$. By using this definition, we can express $||\mathbf{x}||$, the length (norm) of a vector $\mathbf{x}$ as

$$||\mathbf{x}||^2 =< \mathbf{x}, \mathbf{x} > \tag{2.15}$$

This defines a normed vector space.

2.4.1 Component of a Vector along Another Vector

Consider two vectors $\mathbf{g}$ and $\mathbf{x}$, as shown in Fig. 2.8. Let the component of $\mathbf{g}$ along $\mathbf{x}$ be $c\mathbf{x}$. Geometrically the component of $\mathbf{g}$ along $\mathbf{x}$ is the projection of $\mathbf{g}$ on $\mathbf{x}$, and is obtained by drawing a perpendicular from the tip of $\mathbf{g}$ on the vector $\mathbf{x}$, as shown in Fig. 2.8. What is the mathematical significance of a component of a vector along another vector? As seen from

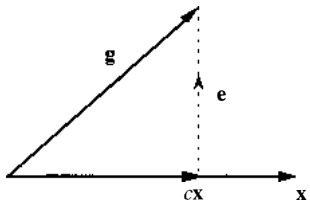
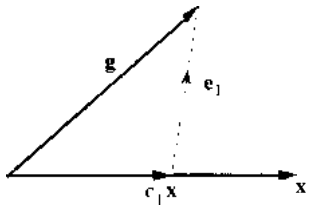
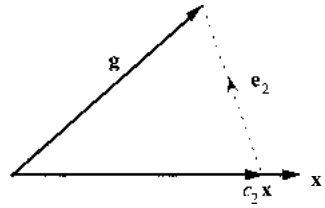


Figure 2.8 Component (projection) of a vector along another vector.



(a)



(b)

Figure 2.9 Approximations of a vector in terms of another vector.

Fig. 2.8, the vector **g** can be expressed in terms of vector **x** as

$$\mathbf{g} = c\mathbf{x} + \mathbf{e} \tag{2.16}$$

However, this does not describe a unique way to decompose **g** in terms of **x** and **e**. Figure 2.9 shows two of the infinite other possibilities. From Fig. 2.9a and b, we have

$$\mathbf{g} = c_1\mathbf{x} + \mathbf{e}_1 = c_2\mathbf{x} + \mathbf{e}_2 \tag{2.17}$$

The question is: Which is the "best" decomposition? The concept of optimality depends on what we wish to accomplish by decomposing **g** into two components.

In each of these three representations, **g** is given in terms of **x** plus another vector called the **error vector**. If our goal is to approximate **g** by $c\mathbf{x}$ (Fig. 2.8),

$$\mathbf{g} \simeq \hat{\mathbf{g}} = c\mathbf{x} \tag{2.18}$$

then the error in this approximation is the (difference) vector $\mathbf{e} = \mathbf{g} - c\mathbf{x}$. Similarly, the errors in approximations of Fig. 2.9a and b are $\mathbf{e}_1$ and $\mathbf{e}_2$, respectively. The approximation in Fig. 2.8 is unique because its error vector is the shortest (with the smallest magnitude or norm). We can now define mathematically the component (or projection) of a vector **g** along vector **x** to be $c\mathbf{x}$, where c is chosen to minimize the magnitude of the error vector $\mathbf{e} = \mathbf{g} - c\mathbf{x}$.

Geometrically, the magnitude of the component of **g** along **x** is $||\mathbf{g}|| \cos\theta$, which is also equal to $c||\mathbf{x}||$. Therefore

$$c||\mathbf{x}|| = ||\mathbf{g}|| \cos\theta$$

Based on the definition of inner product between two vectors, multiplying both sides by $||\mathbf{x}||$ yields

$$c||\mathbf{x}||^2 = ||\mathbf{g}||\,||\mathbf{x}|| \cos\theta = <\mathbf{g}, \mathbf{x}>$$

and

$$c = \frac{<\mathbf{g}, \mathbf{x}>}{<\mathbf{x}, \mathbf{x}>} = \frac{1}{||\mathbf{x}||^2} <\mathbf{g}, \mathbf{x}> \tag{2.19}$$

From Fig. 2.8, it is apparent that when **g** and **x** are perpendicular, or orthogonal, then **g** has a zero component along **x**; consequently, $c = 0$. Keeping an eye on Eq. (2.19), we therefore define **g** and **x** to be **orthogonal** if the inner (scalar or dot) product of the two vectors is zero, that is, if

$$< \mathbf{g}, \mathbf{x} >= 0 \tag{2.20}$$

2.4.2 Decomposition of a Signal and Signal Components

The concepts of vector component and orthogonality can be directly extended to continuous time signals. Consider the problem of approximating a real signal $g(t)$ in terms of another real signal $x(t)$ over an interval $[t_1, t_2]$:

$$g(t) \simeq cx(t) \qquad t_1 \leq t \leq t_2 \tag{2.21}$$

The error $e(t)$ in this approximation is

$$e(t) = \begin{cases} g(t) - cx(t) & t_1 \leq t \leq t_2 \\ 0 & \text{otherwise} \end{cases} \tag{2.22}$$

For "best approximation," we need to minimize the error signal, that is, minimize its norm. Minimum signal norm corresponds to minimum energy E_e over the interval $[t_1, t_2]$ given by

$$\begin{aligned} E_e &= \int_{t_1}^{t_2} e^2(t)\, dt \\ &= \int_{t_1}^{t_2} [g(t) - cx(t)]^2\, dt \end{aligned}$$

Note that the right-hand side is a definite integral with t as the dummy variable. Hence E_e is a function of the parameter c (not t), and E_e is minimum for some choice of c. To minimize E_e, a necessary condition is

$$\frac{dE_e}{dc} = 0 \tag{2.23}$$

or

$$\frac{d}{dc}\left[\int_{t_1}^{t_2} [g(t) - cx(t)]^2\, dt\right] = 0$$

Expanding the squared term inside the integral, we obtain

$$\frac{d}{dc}\left[\int_{t_1}^{t_2} g^2(t)\, dt\right] - \frac{d}{dc}\left[2c\int_{t_1}^{t_2} g(t)x(t)\, dt\right] + \frac{d}{dc}\left[c^2\int_{t_1}^{t_2} x^2(t)\, dt\right] = 0$$

from which we obtain

$$-2\int_{t_1}^{t_2} g(t)x(t)\, dt + 2c\int_{t_1}^{t_2} x^2(t)\, dt = 0$$

and

$$c = \frac{\int_{t_1}^{t_2} g(t)x(t)\,dt}{\int_{t_1}^{t_2} x^2(t)\,dt} = \frac{1}{E_x}\int_{t_1}^{t_2} g(t)x(t)\,dt \tag{2.24}$$

To summarize our discussion, if a signal $g(t)$ is approximated by another signal $x(t)$ as

$$g(t) \simeq cx(t)$$

then the optimum value of c that minimizes the energy of the error signal in this approximation is given by Eq. (2.24).

Taking our cue from vectors, we say that a signal $g(t)$ contains a component $cx(t)$, where c is given by Eq. (2.24). As in vector space, $cx(t)$ is the projection of $g(t)$ on $x(t)$. Consistent with the vector space terminolog, we say that if the component of a signal $g(t)$ of the form $x(t)$ is zero (i.e., $c = 0$), the signals $g(t)$ and $x(t)$ are orthogonal over the interval $[t_1, t_2]$. In other words, with respect to real-valued signals, two signals $x(t)$ and $g(t)$ are orthogonal when there is zero contribution from one signal to the other (i.e., $c = 0$). Thus, $x(t)$ and $g(t)$ are orthogonal if and only if

$$\int_{t_1}^{t_2} g(t)x(t)\,dt = 0 \tag{2.25}$$

Based on the illustrations of vectors in Fig. 2.9, we can say that two signals are orthogonal if and only if their inner product is zero. This relationship indicates that the integral of Eq. (2.25) is closely related to the concept of an inner product between vectors.

Indeed, the standard definition of the inner product of two N-dimensional vectors $\mathbf{g}$ and $\mathbf{x}$

$$< \mathbf{g}, \mathbf{x} >= \sum_{i=1}^{N} g_i x_i$$

is almost identical in form to the integration of Eq. (2.25). We therefore define the inner product of two (real-valued) signals $g(t)$ and $x(t)$, both defined over a time interval $[t_1, t_2]$, as

$$< g(t), x(t) >= \int_{t_1}^{t_2} g(t)x(t)\,dt \tag{2.26}$$

Recall from algebraic geometry that the square of a vector length $||\mathbf{x}||^2$ is equal to $< \mathbf{x}, \mathbf{x} >$. Keeping this concept in mind and continuing our analogy with vector analysis, we define the the norm of a signal $g(t)$ as

$$||g(t)|| = \sqrt{< g(t), g(t) >} \tag{2.27}$$

which is the square root of the signal energy in the time interval. It is therefore clear that the norm of a signal is analogous to the length of a finite dimensional vector. More generally, signals may not be merely defined over a continuous segment $[t_1, t_2]$.*

* Indeed, the signal space under consideration may be over a set of time segments represented simply by Θ. For such a more general space of signals, the inner product is defined as an integral over the time domain Θ. For

Example 2.2 For the square signal $g(t)$ shown in Fig. 2.10 find the component in $g(t)$ of the form of $\sin t$. In other words, approximate $g(t)$ in terms of $\sin t$:

$$g(t) \simeq c \sin t \qquad 0 \leq t \leq 2\pi$$

so that the energy of the error signal is minimum.

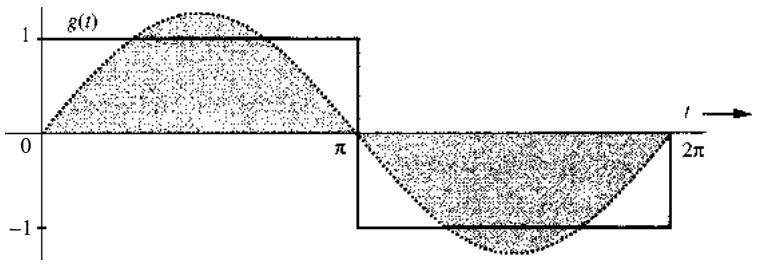


Figure 2.10 Approximation of square signal in terms of a single sinusoid.

In this case

$$x(t) = \sin t \qquad \text{and} \qquad E_x = \int_0^{2\pi} \sin^2(t)\, dt = \pi$$

From Eq. (2.24), we find

$$c = \frac{1}{\pi}\int_0^{2\pi} g(t) \sin t\, dt = \frac{1}{\pi}\left[\int_0^{\pi} \sin t\, dt + \int_{\pi}^{2\pi} (-\sin t)\, dt\right] = \frac{4}{\pi} \tag{2.29}$$

Therefore

$$g(t) \simeq \frac{4}{\pi} \sin t \tag{2.30}$$

represents the best approximation of $g(t)$ by the function $\sin t$, which will minimize the error signal energy. This sinusoidal component of $g(t)$ is shown shaded in Fig. 2.10. As in vector space, we say that the square function $g(t)$ shown in Fig. 2.10 has a component of signal $\sin t$ with magnitude of $4/\pi$.

2.4.3 Complex Signal Space and Orthogonality

So far we have restricted our discussions to real functions of t. To generalize the results to complex functions of t, consider again the problem of approximating a function $g(t)$ by a

---

complex valued signals, the inner product is modified into

$$< g(t),\, x(t) >= \int_{\Theta} g(t)x^*(t)\, dt \tag{2.28}$$

Given the inner product definition, the signal norm $||g(t)|| = \sqrt{< g(t),\, g(t) >}$ and the signal space can be defined for any time domain signal.

function $x(t)$ over an interval $(t_1 \leq t \leq t_2)$

$$g(t) \simeq cx(t) \tag{2.31}$$

where $g(t)$ and $x(t)$ are complex functions of t. In general, both the coefficient c and the error

$$e(t) = g(t) - cx(t) \tag{2.32}$$

are complex. Recall that the energy E_x of the complex signal $x(t)$ over an interval $[t_1, t_2]$ is

$$E_x = \int_{t_1}^{t_2} |x(t)|^2 \, dt$$

For the best approximation, we need to choose c that minimizes E_e, the energy of the error signal $e(t)$ given by

$$E_e = \int_{t_1}^{t_2} |g(t) - cx(t)|^2 \, dt \tag{2.33}$$

Recall also that

$$|u + v|^2 = (u + v)(u^* + v^*) = |u|^2 + |v|^2 + u^*v + uv^* \tag{2.34}$$

Using this result, we can, after some manipulation, express the integral E_e in Eq. (2.33) as

$$E_e = \int_{t_1}^{t_2} |g(t)|^2 \, dt - \left| \frac{1}{\sqrt{E_x}} \int_{t_1}^{t_2} g(t)x^*(t) \, dt \right|^2 + \left| c\sqrt{E_x} - \frac{1}{\sqrt{E_x}} \int_{t_1}^{t_2} g(t)x^*(t) \, dt \right|^2$$

Since the first two terms on the right-hand side are independent of c, it is clear that E_e is minimized by choosing c such that the third term is zero. This yields the optimum coefficient

$$c = \frac{1}{E_x} \int_{t_1}^{t_2} g(t)x^*(t) \, dt \tag{2.35}$$

In light of the foregoing result, we need to redefine orthogonality for the complex case as follows: complex functions (signals) $x_1(t)$ and $x_2(t)$ are orthogonal over an interval $(t \leq t_1 \leq t_2)$ as long as

$$\int_{t_1}^{t_2} x_1(t)x_2^*(t) \, dt = 0 \qquad \text{or} \qquad \int_{t_1}^{t_2} x_1^*(t)x_2(t) \, dt = 0 \tag{2.36}$$

In fact, either equality suffices. This is a general definition of orthogonality, which reduces to Eq. (2.25) when the functions are real.

Similarly, the definition of inner product for complex signals over a time domain Θ can be modified:

$$< g(t), \, x(t) >= \int_{\{t:t\in\Theta\}} g(t)x^*(t) \, dt \tag{2.37}$$

Consequently, the norm of a signal $g(t)$ is simply

$$||g(t)|| = \left[\int_{\{t:t\in\Theta\}} |g(t)|^2 \, dt \right]^{1/2} \tag{2.38}$$

2.4.4 Energy of the Sum of Orthogonal Signals

We know that the geometric length (or magnitude) of the sum of two orthogonal vectors is equal to the sum of the magnitude squares of the two vectors. Thus, if vectors $\mathbf{x}$ and $\mathbf{y}$ are orthogonal, and if $\mathbf{z} = \mathbf{x} + \mathbf{y}$, then

$$||\mathbf{z}||^2 = ||\mathbf{x}||^2 + ||\mathbf{y}||^2$$

We have a similar result for signals. The energy of the sum of two orthogonal signals is equal to the sum of the energies of the two signals. Thus, if signals $x(t)$ and $y(t)$ are orthogonal over an interval $[t_1, t_2]$, and if $z(t) = x(t) + y(t)$, then

$$E_z = E_x + E_y \tag{2.39}$$

We now prove this result for complex signals of which real signals are a special case. From Eq. (2.34) it follows that

$$\begin{aligned}\int_{t_1}^{t_2} |x(t) + y(t)|^2\, dt &= \int_{t_1}^{t_2} |x(t)|^2\, dt + \int_{t_1}^{t_2} |y(t)|^2\, dt + \int_{t_1}^{t_2} x(t)y^*(t)\, dt + \int_{t_1}^{t_2} x^*(t)y(t)\, dt \\ &= \int_{t_1}^{t_2} |x(t)|^2\, dt + \int_{t_1}^{t_2} |y(t)|^2\, dt \end{aligned} \tag{2.40}$$

The last equality follows because, as a result of orthogonality, the two integrals of the cross products $x(t)y^*(t)$ and $x^*(t)y(t)$ are zero. This result can be extended to sum of any number of mutually orthogonal signals.

2.5 CORRELATION OF SIGNALS

By defining the inner product and the norm of signals, we paved the foundation for signal comparison. Here again, we can benefit by drawing parallels to the familiar vector space. Two vectors $\mathbf{g}$ and $\mathbf{x}$ are similar if $\mathbf{g}$ has a large component along $\mathbf{x}$. In other words, if c in Eq. (2.19) is large, the vectors $\mathbf{g}$ and $\mathbf{x}$ are similar. We could consider c to be a quantitative measure of similarity between $\mathbf{g}$ and $\mathbf{x}$. Such a measure, however, would be defective because it varies with the norms (or lengths) of $\mathbf{g}$ and $\mathbf{x}$. To be fair, the amount of similarity between $\mathbf{g}$ and $\mathbf{x}$ should be independent of the lengths of $\mathbf{g}$ and $\mathbf{x}$. If we double the length of $\mathbf{g}$, for example, the amount of similarity between $\mathbf{g}$ and $\mathbf{x}$ should not change. From Eq. (2.19), however, we see that doubling $\mathbf{g}$ doubles the value of c (whereas doubling $\mathbf{x}$ halves the value of c). The similarity measure based on signal correlation is clearly faulty. Similarity between two vectors is indicated by the angle θ between the vectors. The smaller the θ, the larger the similarity, and vice versa. The amount of similarity can therefore be conveniently measured by $\cos\theta$. The larger the $\cos\theta$, the larger the similarity between the two vectors. Thus, a suitable measure would be $\rho = \cos\theta$, which is given by

$$\rho = \cos\theta = \frac{< \mathbf{g}, \mathbf{x} >}{||\mathbf{g}||\,||\mathbf{x}||} \tag{2.41}$$

We can readily verify that this measure is independent of the lengths of $\mathbf{g}$ and $\mathbf{x}$. This similarity measure ρ is known as the **correlation coefficient**. Observe that

$$-1 \leq \rho \leq 1 \tag{2.42}$$

Thus, the magnitude of ρ is never greater than unity. If the two vectors are aligned, the similarity is maximum ($\rho = 1$). Two vectors aligned in opposite directions have maximum dissimilarity ($\rho = -1$). If the two vectors are orthogonal, the similarity is zero.

We use the same argument in defining a similarity index (the correlation coefficient) for signals. For convenience, we shall consider the signals over the entire time interval from $-\infty$ to ∞. To establish a similarity index independent of energies (sizes) of $g(t)$ and $x(t)$, we must normalize c by normalizing the two signals to have unit energies. Thus, the appropriate similarity index ρ analogous to Eq. (2.41) is given by

$$\rho = \frac{1}{\sqrt{E_g E_x}} \int_{-\infty}^{\infty} g(t)x(t)\,dt \tag{2.43}$$

Observe that multiplying either $g(t)$ or $x(t)$ by any constant has no effect on this index. Thus, it is independent of the size (energies) of $g(t)$ and $x(t)$. Using the Cauchy-Schwarz inequality (proved in Appendix B),[†] one can show that the magnitude of ρ is never greater than 1:

$$-1 \leq \rho \leq 1 \tag{2.44}$$

2.5.1 Correlation Functions

We should revisit the application of correlation to signal detection in a radar unit, where a signal pulse is transmitted to detect a suspected target. By detecting the presence or absence of the reflected pulse, we confirm the presence or absence of the target. By measuring the time delay between the transmitted and received (reflected) pulse, we determine the distance of the target. Let the transmitted and the reflected pulses be denoted by $g(t)$ and $z(t)$, respectively. If we were to use Eq. (2.43) directly to measure the correlation coefficient ρ, we would obtain

$$\rho = \frac{1}{\sqrt{E_g E_z}} \int_{-\infty}^{\infty} z(t)g^*(t)\,dt = 0 \tag{2.45}$$

Thus, the correlation is zero because the pulses are disjoint (nonoverlapping in time). The integral in Eq. (2.45) will yield zero even when the pulses are identical but with relative time shift. To avoid this difficulty, we compare the received pulse $z(t)$ with the transmitted pulse $g(t)$ shifted by τ. If for some value of τ, there is a strong correlation, we not only detect the presence of the pulse but we also detect the relative time shift of $z(t)$ with respect to $g(t)$. For this reason, instead of using the integral on the right-hand side, we use the modified integral $\psi_{gz}(\tau)$, the **cross-correlation** function of two complex signals $g(t)$ and $z(t)$, defined by

$$\psi_{gz}(\tau) \equiv \int_{-\infty}^{\infty} z(t)g^*(t-\tau)\,dt = \int_{-\infty}^{\infty} z(t+\tau)g^*(t)\,dt \tag{2.46}$$

Therefore, $\psi_{gz}(\tau)$ is an indication of similarity (correlation) of $g(t)$ with $z(t)$ advanced (left-shifted) by τ seconds.

[†] The Cauchy-Schwarz inequality states that for two real energy signals $g(t)$ and $x(t)$, $\left(\int_{-\infty}^{\infty} g(t)x(t)\,dt\right)^2 \leq E_g E_x$ with equality if and only if $x(t) = Kg(t)$, where K is an arbitrary constant. There is similar inequality for complex signals.

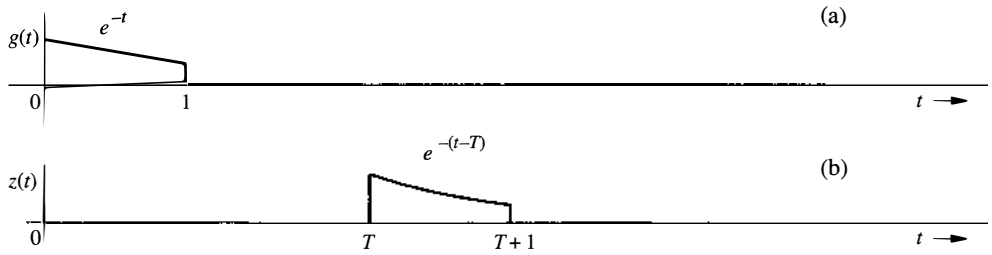


Figure 2.11 Physical explanation of the autocorrelation function.

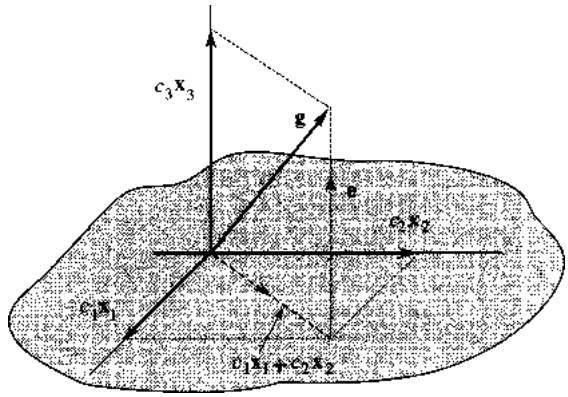


Figure 2.12 Representation of a vector in three-dimensional space.

2.5.2 Autocorrelation Function

As shown in Fig. 2.11, correlation of a signal with itself is called the **autocorrelation**. The autocorrelation function $\psi_g(\tau)$ of a real signal $g(t)$ is defined as

$$\psi_g(\tau) \equiv \int_{-\infty}^{\infty} g(t)g(t+\tau)\,dt \tag{2.47}$$

It measures the similarity of the signal $g(t)$ with its own displaced version. In Chapter 3, we shall show that the autocorrelation function provides valuable spectral information about the signal.

2.6 ORTHOGONAL SIGNAL SET

In this section we show a way of representing a signal as a sum of orthogonal set of signals. In effect, the signals in this orthogonal set form a basis for the specific signal space. Here again we can benefit from the insight gained from a similar problem in vectors. We know that a vector can be represented as a sum of orthogonal vectors, which form the coordinate system of a vector space. The problem in signals is analogous, and the results for signals are parallel to those for vectors. For this reason, let us review the case of vector representation.

2.6.1 Orthogonal Vector Space

Consider a multidimensional Cartesian vector space described by three mutually orthogonal vectors $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$, as shown in Fig. 2.12 for the special case of three-dimensional vector space. First, we shall seek to approximate a three-dimensional vector $\mathbf{g}$ in terms of two

orthogonal vectors $\mathbf{x}_1$ and $\mathbf{x}_2$:

$$\mathbf{g} \simeq c_1\mathbf{x}_1 + c_2\mathbf{x}_2$$

The error $\mathbf{e}$ in this approximation is

$$\mathbf{e} = \mathbf{g} - (c_1\mathbf{x}_1 + c_2\mathbf{x}_2)$$

or equivalently,

$$\mathbf{g} = c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \mathbf{e}$$

In accordance with our earlier geometrical argument, it is clear from Fig. 2.12 that the length of error vector $\mathbf{e}$ is minimum when it is perpendicular to the $(\mathbf{x}_1, \mathbf{x}_2)$ plane, and when $c_1\mathbf{x}_1$ and $c_2\mathbf{x}_2$ are the projections (components) of $\mathbf{g}$ on $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively. Therefore, the constants c_1 and c_2 are given by formula in Eq. (2.19).

Now let us determine the best approximation to $\mathbf{g}$ in terms of all the three mutually orthogonal vectors $\mathbf{x}_1, \mathbf{x}_2$, and $\mathbf{x}_3$:

$$\mathbf{g} \simeq c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + c_3\mathbf{x}_3 \tag{2.48}$$

Figure 2.12 shows that a unique choice of c_1, c_2, and c_3 exists, for which (2.48) is no longer an approximation but an equality:

$$\mathbf{g} = c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + c_3\mathbf{x}_3$$

In this case, $c_1\mathbf{x}_1, c_2\mathbf{x}_2$, and $c_3\mathbf{x}_3$ are the projections (components) of $\mathbf{g}$ on $\mathbf{x}_1, \mathbf{x}_2$, and $\mathbf{x}_3$, respectively. Note that the approximation error $\mathbf{e}$ is now zero when $\mathbf{g}$ is approximated in terms of three mutually orthogonal vectors: $\mathbf{x}_1, \mathbf{x}_2$, and $\mathbf{x}_3$. This is because $\mathbf{g}$ is a three-dimensional vector, and the vectors $\mathbf{x}_1, \mathbf{x}_2$, and $\mathbf{x}_3$ represent a *complete set* of orthogonal vectors in three-dimensional space. Completeness here means that it is impossible in this space to find any other vector $\mathbf{x}_4$, which is orthogonal to all the three vectors $\mathbf{x}_1, \mathbf{x}_2$, and $\mathbf{x}_3$. Any vector in this space can therefore be represented (with zero error) in terms of these three vectors. Such vectors are known as **basis** vectors, and the set of vector is known as a **complete orthogonal basis** of this vector space. If a set of vectors $\{\mathbf{x}_i\}$ is not complete, then the approximation error will generally not be zero. For example, in the three-dimensional case just discussed earlier, it is generally not possible to represent a vector $\mathbf{g}$ in terms of only two basis vectors without an error.

The choice of basis vectors is not unique. In fact, each set of basis vectors corresponds to a particular choice of coordinate system. Thus, a three-dimensional vector $\mathbf{g}$ may be represented in many different ways depending on the coordinate system used.

To summarize, if a set of vectors $\{\mathbf{x}_i\}$ is mutually orthogonal, that is, if

$$< \mathbf{x}_m, \mathbf{x}_n > = \begin{cases} 0 & m \neq n \\ |\mathbf{x}_m|^2 & m = n \end{cases}$$

and if this basis set is complete, a vector $\mathbf{g}$ in this space can be expressed as

$$\mathbf{g} = c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + c_3\mathbf{x}_3 \tag{2.49}$$

where the constants c_i are given by

$$c_i = \frac{< \mathbf{g},\ \mathbf{x}_i >}{< \mathbf{x}_i,\ \mathbf{x}_i >} \tag{2.50a}$$

$$= \frac{1}{||\mathbf{x}_i||^2} < \mathbf{g},\ \mathbf{x}_i > \qquad i = 1,\ 2,\ 3 \tag{2.50a}$$

2.6.2 Orthogonal Signal Space

We continue with our signal approximation problem, using clues and insights developed for vector approximation. As before, we define orthogonality of a signal set $x_1(t)$, $x_2(t)$, ... $x_N(t)$ over a time domain Θ (may be an interval $[t_1, t_2]$) as

$$\int_{t\in\Theta} x_m(t)x_n^*(t)\,dt = \begin{cases} 0 & m \neq n \\ E_n & m = n \end{cases} \tag{2.51}$$

If all signal energies are equal $E_n = 1$, then the set is *normalized* and is called an **orthonormal set**. An orthogonal set can always be normalized by dividing $x_n(t)$ by $\sqrt{E_n}$ for all n. Now, consider the problem of approximating a signal $g(t)$ over the Θ by a set of N mutually orthogonal signals $x_1(t), x_2(t), \ldots, x_N(t)$:

$$g(t) \simeq c_1x_1(t) + c_2x_2(t) + \cdots + c_Nx_N(t) \tag{2.52a}$$

$$= \sum_{n=1}^{N} c_nx_n(t) \qquad t \in \Theta \tag{2.52b}$$

It can be shown that E_e, the energy of the error signal $e(t)$ in this approximation, is minimized if we choose

$$c_n = \frac{\displaystyle\int_{t\in\Theta} g(t)x_n^*(t)\,dt}{\displaystyle\int_{t\in\Theta} |x_n(t)|^2\,dt}$$

$$= \frac{1}{E_n}\int_{\Theta} g(t)x_n^*(t)\,dt \qquad n = 1, 2, \ldots, N \tag{2.53}$$

Moreover, if the orthogonal set is **complete**, then the error energy $E_e \to 0$, and the representation in (2.52) is no longer an approximation, but an equality. More precisely, let the N-term approximation error be defined by

$$e_N(t) = g(t) - c_1x_1(t) + c_2x_2(t) + \cdots + c_Nx_N(t) = g(t) - \sum_{n=1}^{N} c_nx_n(t) \qquad t \in \Theta \tag{2.54}$$

If the orthogonal basis is **complete**, then the error signal energy converges to zero; that is,

$$\lim_{N\to\infty} \int_{t\in\Theta} |e_N(t)|^2 dt = 0 \tag{2.55}$$

In a strictly mathematical sense, however, a signal may not converge to zero even though its energy does. This is because a signal may be nonzero at some isolated points.* Still, for all practical purposes, signals are continuous for all t, and the equality (2.55) states that the error signal has zero energy as $N \to \infty$. Thus, for $N \to \infty$, the equality (2.52) can be loosely written as

$$\begin{aligned} g(t) &= c_1 x_1(t) + c_2 x_2(t) + \cdots + c_n x_n(t) + \cdots \\ &= \sum_{n=1}^{\infty} c_n x_n(t) \qquad t \in \Theta \end{aligned} \tag{2.56}$$

where the coefficients c_n are given by Eq. (2.53). Because the error signal energy approaches zero, it follows that the energy of $g(t)$ is now equal to the sum of the energies of its orthogonal components.

The series on the right-hand side of Eq. (2.56) is called the **generalized Fourier series** of $g(t)$ with respect to the set $\{x_n(t)\}$. When the set $\{x_n(t)\}$ is such that the error energy $E_N \to 0$ as $N \to \infty$ for every member of some particular signal class, we say that the set $\{x_n(t)\}$ is complete on $\{t : \Theta\}$ for that class of $g(t)$, and the set $\{x_n(t)\}$ is called a set of **basis functions** or **basis signals**. In particular, the class of (finite) energy signals over Θ is denoted as $L^2\{\Theta\}$. Unless otherwise mentioned, in the future we shall consider only the class of energy signals.

2.6.3 Parseval's Theorem

Recall that the energy of the sum of orthogonal signals is equal to the sum of their energies. Therefore, the energy of the right-hand side of Eq. (2.56) is the sum of the energies of the individual orthogonal components. The energy of a component $c_n x_n(t)$ is $c_n^2 E_n$. Equating the energies of the two sides of Eq. (2.56) yields

$$\begin{aligned} E_g &= c_1^2 E_1 + c_2^2 E_2 + c_3^2 E_3 + \cdots \\ &= \sum_n c_n^2 E_n \end{aligned} \tag{2.57}$$

This important result goes by the name of **Parseval's theorem**. Recall that the signal energy (area under the squared value of a signal) is analogous to the square of the length of a vector in the vector-signal analogy. In vector space we know that the square of the length of a vector is equal to the sum of the squares of the lengths of its orthogonal components. Parseval's theorem [Eq. (2.57)] is the statement of this fact as applied to signals.

2.7 THE EXPONENTIAL FOURIER SERIES

We noted earlier that orthogonal signal representation is NOT unique. While the traditional trigonometric Fourier series allows a good representation of all periodic signals, here we provide an orthogonal representation of periodic signals that is **equivalent** but has a simpler form.

* Known as a measure-zero set.

First of all, it is clear that the set of exponentials $e^{jn\omega_0 t}$ $(n = 0, \pm 1, \pm 2, \ldots)$ is orthogonal over any interval of duration $T_0 = 2\pi/\omega_0$, that is,

$$\int_{T_0} e^{jm\omega_0 t}(e^{jn\omega_0 t})^* \, dt = \int_{T_0} e^{j(m-n)\omega_0 t} \, dt = \begin{cases} 0 & m \neq n \\ T_0 & m = n \end{cases} \tag{2.58}$$

Moreover, this set is a complete set.[1,2] From Eqs. (2.53) and (2.56), it follows that a signal $g(t)$ can be expressed over an interval of duration T_0 second(s) as an exponential Fourier series

$$\begin{aligned} g(t) &= \sum_{n=-\infty}^{\infty} D_n e^{jn\omega_0 t} \\ &= \sum_{n=-\infty}^{\infty} D_n e^{jn2\pi f_0 t} \end{aligned} \tag{2.59}$$

where [see Eq. (2.53)]

$$D_n = \frac{1}{T_0} \int_{T_0} g(t) e^{-jn2\pi f_0 t} \, dt \tag{2.60}$$

The exponential Fourier series in Eq. (2.59) consists of components of the form $e^{jn2\pi f_0 t}$ with n varying from $-\infty$ to ∞. It is periodic with period T_0.

Example 2.3 Find the exponential Fourier series for the signal in Fig. 2.13b.

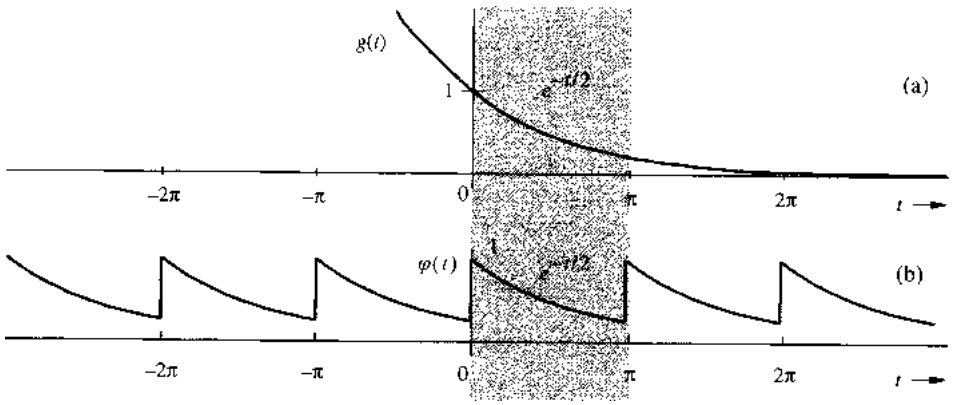


Figure 2.13 A periodic signal.

In this case, $T_0 = \pi$, $2\pi f_0 = 2\pi/T_0 = 2$, and

$$\varphi(t) = \sum_{n=-\infty}^{\infty} D_n e^{j2nt}$$

where

$$\begin{aligned} D_n &= \frac{1}{T_0}\int_{T_0} \varphi(t)e^{-j2nt}\,dt \\ &= \frac{1}{\pi}\int_0^{\pi} e^{-t/2}\,e^{-j2nt}\,dt \\ &= \frac{1}{\pi}\int_0^{\pi} e^{-(\frac{1}{2}+j2n)t}\,dt \\ &= \frac{-1}{\pi\left(\frac{1}{2}+j2n\right)} e^{-(\frac{1}{2}+j2n)t}\Bigg|_0^{\pi} \\ &= \frac{0.504}{1+j4n} \end{aligned} \tag{2.61}$$

and

$$\varphi(t) = 0.504\sum_{n=-\infty}^{\infty} \frac{1}{1+j4n}e^{j2nt} \tag{2.62a}$$

$$\begin{aligned} = 0.504\Bigg[1 &+ \frac{1}{1+j4}e^{j2t} + \frac{1}{1+j8}e^{j4t} + \frac{1}{1+j12}e^{j6t} + \cdots \\ &+ \frac{1}{1-j4}e^{-j2t} + \frac{1}{1-j8}e^{-j4t} + \frac{1}{1-j12}e^{-j6t} + \cdots\Bigg] \end{aligned} \tag{2.62b}$$

Observe that the coefficients D_n are complex. Moreover, D_n and D_{-n} are conjugates, as expected.

Exponential Fourier Spectra

In exponential spectra, we plot coefficients D_n as a function of ω. But since D_n is complex in general, we need two plots: the real and the imaginary parts of D_n or the amplitude (magnitude) and the angle of D_n. We prefer the latter because of its close connection to the amplitudes and phases of corresponding components of the trigonometric Fourier series. We therefore plot $|D_n|$ versus ω and $\angle D_n$ versus ω. This requires that the coefficients D_n be expressed in polar form as $|D_n|e^{j\angle D_n}$.

For a real periodic signal, the twin coefficients D_n and D_{-n} are conjugates,

$$|D_n| = |D_{-n}| \tag{2.63a}$$

$$\angle D_n = \theta_n \quad \text{and} \quad \angle D_{-n} = -\theta_n \tag{2.63b}$$

Thus,

$$D_n = |D_n|e^{j\theta_n} \quad \text{and} \quad D_{-n} = |D_n|e^{-j\theta_n} \tag{2.64}$$

Note that $|D_n|$ are the amplitudes (magnitudes) and $\angle D_n$ are the angles of various exponential components. From Eq. (2.63) it follows that the amplitude spectrum ($|D_n|$ vs. f) is an even function of ω and the angle spectrum ($\angle D_n$ vs. f) is an odd function of f when $g(t)$ is a real signal.

For the series in Example 2.3, for instance,

$$D_0 = 0.504$$

$$D_1 = \frac{0.504}{1+j4} = 0.122e^{-j75.96^\circ} \Longrightarrow |D_1| = 0.122,\ \angle D_1 = -75.96^\circ$$

$$D_{-1} = \frac{0.504}{1-j4} = 0.122e^{j75.96^\circ} \Longrightarrow |D_{-1}| = 0.122,\ \angle D_{-1} = 75.96^\circ$$

and

$$D_2 = \frac{0.504}{1+j8} = 0.0625e^{-j82.87^\circ} \Longrightarrow |D_2| = 0.0625,\ \angle D_2 = -82.87^\circ$$

$$D_{-2} = \frac{0.504}{1-j8} = 0.0625e^{j82.87^\circ} \Longrightarrow |D_{-2}| = 0.0625,\ \angle D_{-2} = 82.87^\circ$$

and so on. Note that D_n and D_{-n} are conjugates, as expected [see Eq. (2.63b)].

Figure 2.14 shows the frequency spectra (amplitude and angle) of the exponential Fourier series for the periodic signal $\varphi(t)$ in Fig. 2.13b.

We notice some interesting features of these spectra. First, the spectra exist for positive as well as negative values of f (the frequency). Second, the amplitude spectrum is an even function of f and the angle spectrum is an odd function of f. Equations (2.63) show the symmetric characteristics of the amplitude and phase of D_n.

What Does Negative Frequency Mean?

The existence of the spectrum at negative frequencies is somewhat disturbing to some people because by definition, the frequency (number of repetitions per second) is a positive quantity. How do we interpret a negative frequency f_0? We can use a trigonometric identity to express a sinusoid of a negative frequency $-f_0$ by borrowing $\omega_0 = 2\pi f_0$, as

$$\cos(-\omega_0 t + \theta) = \cos(\omega_0 t - \theta)$$

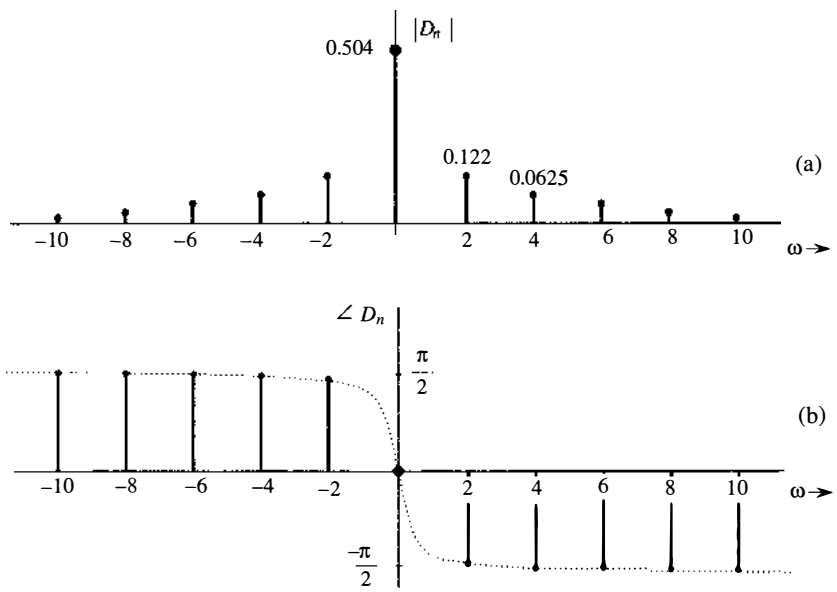


Figure 2.14 Exponential Fourier spectra for the signal in Fig. 2.13a.

This clearly shows that the angular frequency of a sinusoid cos $(-\omega_0 t + \theta)$ is $|\omega_0|$, which is a positive quantity. The commonsense statement that a frequency must be positive comes from the traditional notion that frequency is associated with a real-valued sinusoid (such as a sine or a cosine). In reality, the concept of frequency for a real-valued sinusoid describes only the rate of the sinusoidal variation without addressing the direction of the variation. This is because real-valued sinusoidal signals do NOT contain information on the direction of its variation.

The concept of negative frequency is meaningful **only** when we are considering complex sinusoids for which the rate and the *direction* of variation are meaningful. Observe that

$$e^{\pm j\omega_0 t} = \cos \omega_0 t \pm j \sin \omega_0 t$$

This relationship clearly shows that either positive or negative ω leads to periodic variation of the same rate. However, the resulting complex signals are NOT the same. Because $|e^{\pm j\omega_0 t}| = 1$, both $e^{+j\omega_0 t}$ and $e^{-j\omega_0 t}$ are unit length complex variables that can be shown on the complex plane. We illustrate the two exponential sinusoids as unit length complex variables that vary with time t in Fig. 2.15. Thus, the rotation rate for both exponentials $e^{\pm j\omega_0 t}$ is $|\omega_0|$. It is clear that for positive frequency, the exponential sinusoid rotates counterclockwise while for negative frequency, the exponential sinusoid rotates clockwise. This illustrates the actual meaning of negative frequency.

There exists a good analogy between positive/negative frequency and positive/negative velocity. Just as people are reluctant to use *negative* velocity in describing a moving object, they are equally unwilling to accept the notion of "negative" frequency. However, once we understand that negative velocity simply refers to both the negative direction and the actual speed of a moving object, negative velocity makes perfect sense. Likewise, negative frequency does NOT describe the rate of periodic variation of a sine or a cosine. It describe the direction of rotation of a unit length exponential sinusoid and its rate of revolution.

Another way of looking at the situation is to say that *exponential spectra are a graphical representation of coefficients D_n as a function of f. Existence of the spectrum at $f = -nf_0$ merely indicates that an exponential component $e^{-jn2\pi f_0 t}$ exists in the series.* We know from Euler's identity

$$\cos(\omega t + \theta) = \frac{e^{j\theta}}{2} \exp(j\omega t) + \frac{e^{-j\theta}}{2} \exp(-j\omega t)$$

that a sinusoid of frequency $n\omega_0$ can be expressed in terms of a pair of exponentials $e^{jn\omega_0 t}$ and $e^{-jn\omega_0 t}$. That both sine and cosine consist of positive and negative frequency exponential sinusoidal components clearly indicates that we are NOT at all able to describe the *direction* of their periodic variations. Indeed, both sine and cosine functions of frequency ω_0 consist of two equal-size exponential sinusoids of frequency $\pm\omega_0$. Thus, the frequency of sine or cosine is the absolute value of its two component frequencies and denotes only the rate of the sinusoidal variations.

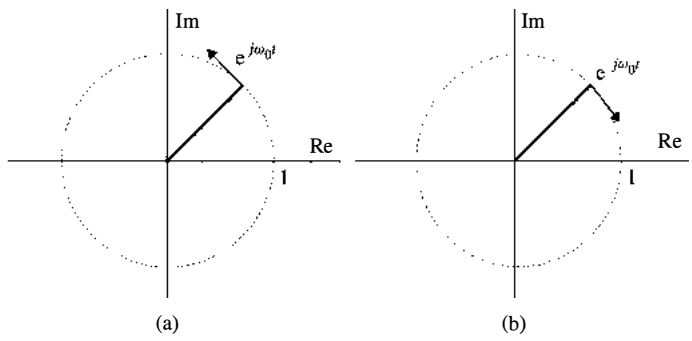
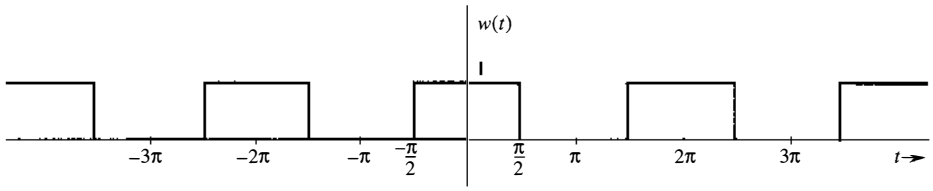


Figure 2.15 Unit length complex variable with positive frequency (rotating counter-clockwise) versus unit length complex variable with negative frequency (rotating clockwise).

Example 2.4 Find the exponential Fourier series for the periodic square wave $w(t)$ shown in Fig. 2.16.



(a)

Figure 2.16 A square pulse periodic signal.

$$w(t) = \sum_{n=-\infty}^{\infty} D_n e^{jn2\pi f_0 t}$$

where

$$D_0 = \frac{1}{T_0}\int_{T_0} w(t)\,dt = \frac{1}{2}$$

$$D_n = \frac{1}{T_0}\int_{T_0} w(t)e^{-jn2\pi f_0 t}\,dt, \qquad n \neq 0$$

$$= \frac{1}{T_0}\int_{-T_0/4}^{T_0/4} e^{-jn2\pi f_0 t}\,dt$$

$$= \frac{1}{-jn2\pi f_0 T_0}\left[e^{-jn2\pi f_0 T_0/4} - e^{jn2\pi f_0 T_0/4}\right]$$

$$= \frac{2}{n2\pi f_0 T_0}\sin\left(\frac{n2\pi f_0 T_0}{4}\right) = \frac{1}{n\pi}\sin\left(\frac{n\pi}{2}\right)$$

In this case D_n is real. Consequently, we can do without the phase or angle plot if we plot D_n vs. f instead of the amplitude spectrum ($|D_n|$ vs. f) as shown in Fig. 2.17.

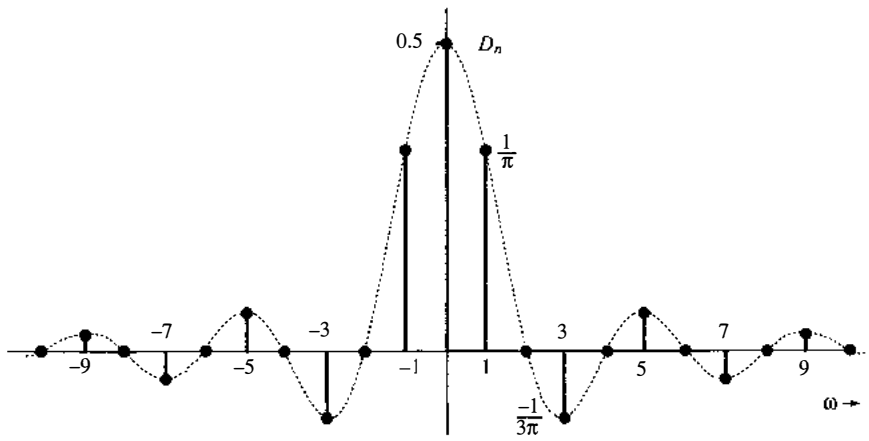


Figure 2.17 Exponential Fourier spectrum of the square pulse periodic signal.

Example 2.5 Find the exponential Fourier series and sketch the corresponding spectra for the impulse train $\delta_{T_0}(t)$ shown in Fig. 2.18a.

The exponential Fourier series is given by

$$\delta_{T_0}(t) = \sum_{n=-\infty}^{\infty} D_n e^{jn2\pi f_0 t} \qquad f_0 = \frac{1}{T_0} \tag{2.65}$$

where

$$D_n = \frac{1}{T_0} \int_{T_0} \delta_{T_0}(t) e^{-jn2\pi f_0 t}\, dt$$

Choosing the interval of integration $(\frac{-T_0}{2}, \frac{T_0}{2})$ and recognizing that over this interval $\delta_{T_0}(t) = \delta(t)$, were have

$$D_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} \delta(t) e^{-jn2\pi f_0 t}\, dt$$

In this integral, the impulse is located at $t = 0$. From the sampling property of the impulse function, the integral on the right-hand side is the value of $e^{-jn2\pi f_0 t}$ at $t = 0$ (where the impulse is located). Therefore

$$D_n = \frac{1}{T_0} \tag{2.66}$$

and

$$\delta_{T_0}(t) = \frac{1}{T_0} \sum_{n=-\infty}^{\infty} e^{jn2\pi f_0 t} \qquad f_0 = \frac{1}{T_0} \tag{2.67}$$

Equation (2.67) shows that the exponential spectrum is uniform ($D_n = 1/T_0$) for all the frequencies, as shown in Fig. 2.18b. The spectrum, being real, requires only the amplitude plot. All phases are zero.

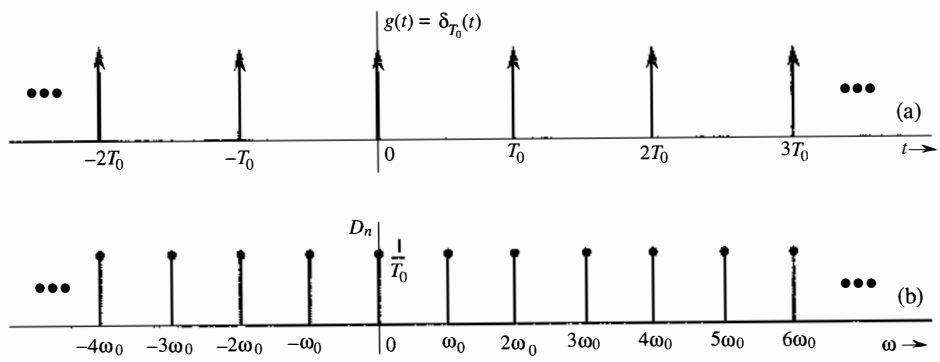


Figure 2.18 Impulse train and its exponential Fourier spectra.

Parseval's Theorem in the Fourier Series

A periodic signal $g(t)$ is a power signal, and every term in its Fourier series is also a power signal. The power P_g of $g(t)$ is equal to the power of its Fourier series. Because the Fourier series consists of terms that are mutually orthogonal over one period, the power of the Fourier series is equal to the sum of the powers of its Fourier components. This follows from Parseval's theorem.

Thus, for the exponential Fourier series

$$g(t) = D_0 + \sum_{n=-\infty,\, n\neq 0}^{\infty} D_n e^{jn\omega_0 t}$$

the power is given by (see Prob. 2.1-7)

$$P_g = \sum_{n=-\infty}^{\infty} |D_n|^2 \tag{2.68a}$$

For a real $g(t)$, $|D_{-n}| = |D_n|$. Therefore

$$P_g = {D_0}^2 + 2\sum_{n=1}^{\infty} |D_n|^2 \tag{2.68b}$$

Comment: Parseval's theorem occurs in many different forms, such as in Eqs. (2.57) and Eq. (2.68a). Yet another form is found in the next chapter for nonperiodic signals. Although these forms appear to be different, they all state the same principle: that is, the square of the length of a vector equals the sum of the squares of its orthogonal components. The first form [Eq. (2.57)] applies to energy signals, and the second [Eq. (2.68a)] applies to periodic signals represented by the exponential Fourier series.

Some Other Examples of Orthogonal Signal Sets

The signal representation by Fourier series shows that signals are vectors in every sense. Just as a vector can be represented as a sum of its components in a variety of ways, depending upon the choice of a coordinate system, a signal can be represented as a sum of its components in a variety of ways. Just as we have vector coordinate systems formed by mutually orthogonal vectors (rectangular, cylindrical, spherical, etc.), we also have signal coordinate systems, basis signals, formed by a variety of sets of mutually orthogonal signals. There exist a large number of orthogonal signal sets that can be used as basis signals for generalized Fourier series. Some well-known signal sets are trigonometric (sinusoid) functions, exponential functions, Walsh functions, Bessel functions, Legendre polynomials, Laguerre functions, Jacobi polynomials, Hermite polynomials, and Chebyshev polynomials. The functions that concern us most in this book are the exponential sets discussed next in the chapter.

2.8 MATLAB EXERCISES

In this section, we provide some basic MATLAB exercises to illustrate the process of signal generation, signal operations, and Fourier series analysis.

Basic Signals and Signal Graphing

Basic functions can be defined by using MATLAB's m-files. We gave three MATLAB programs that implement three basic functions when a time vector `t` is provided:

- `ustep.m` implements the unit step function $u(t)$
- `rect.m` implements the standard rectangular function $rect(t)$
- `triangl.m` implements standard triangle function $\Delta(t)$

```
% (file name:  ustep.m)
%  The unit step function is a function of time 't'.
%  Usage  y = ustep(t)
%
%  ustep(t) = 0    if t < 0
%  ustep(t) = 1,   if t >= 1
%
%  t - must be real-valued and can be a vector or a matrix
%
function y=ustep(t)
   y = (t>=0);
end
```

```
% (file name: rect.m)
%  The rectangular function is a function of time 't'.
%
%  Usage  y = rect(t)
%  t - must be real-valued and can be a vector or a matrix
%
%  rect(t) = 1,    if |t| < 0.5
%  rect(t) = 0,    if |t| > 0.5
%
function y=rect(t)
   y =(sign(t+0.5)-sign(t-0.5) >0);
end
```

```
% (file name: triangl.m)
%  The triangle function is a function of time 't'.
%
%  triangl(t) = 1-|t|, if  |t| < 1
%  triangl(t) = 0, if      |t| > 1
%
%  Usage  y = triangl(t)
%  t - must be real-valued and can be a vector or a matrix
%
```

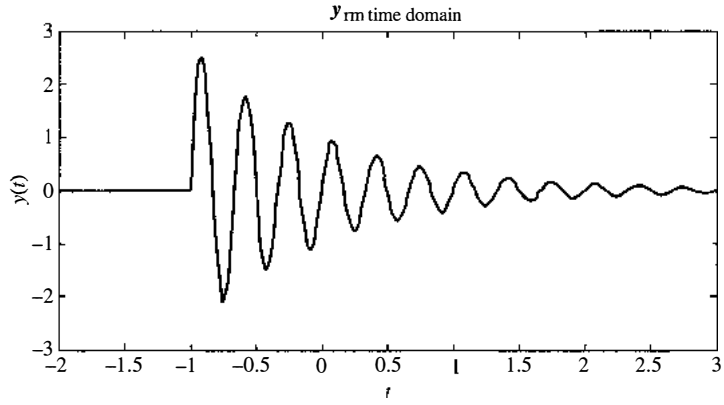


Figure 2.19 Graphing a signal.

```
function y=triangl(t)
    y = (1-abs(t)).*(t>=-1).*(t<1);
end
```

We now show how to use MATLAB to generate a simple signal plot through an example. `siggraf.m` is provided. In this example, we construct and plot a signal

$$y(t) = \exp(-t)\sin(6\pi t)u(t+1)$$

The resulting graph shown in Fig. 2.19.

```
% (file name: siggraf.m)
%  To graph a signal, the first step is to determine
%  the x-axis and the y-axis to plot
%  We can first decide the length of x-axis to plot
   t=[-2:0.01:3];      % "t" is from -2 to 3 in 0.01 increment
% Then evaluate the signal over the range of "t" to plot
   y=exp(-t).*sin(10*pi*t).*ustep(t+1);
   figure(1);  fig1=plot(t,y);     % plot t vs y in figure 1
   set(fig1,'Linewidth',2);        % choose a wider line-width
   xlabel('\it t');         % use italic 't' to label x-axis
   ylabel('\{\bf y\}(\{\it t\})');      % use boldface 'y'
                                          to label y-axis
   title('\{\bf y\}\_\{\rm time domain\}');  % can use subscript
```

Periodic Signals and Signal Power

Periodic signals can be generated by first determining the signal values in one period before repeating the same signal vector multiple times.

In the following MATLAB program `PfuncEx.m`, we generate a periodic signal and observe its behavior over $2M$ periods. The period of this example is $T = 6$. The program also evaluates the average signal power which is stored as a variable `y_power` and signal energy in one period which is stored in variable `y_energyT`.

```
% (file name:  PfuncEx.m)
% This example generates a periodic signal, plots the signal
% and evaluates the average signal power in y_power and signal
% energy in 1 period T:    y_energyT
    echo off;clear;clf;
% To generate a periodic signal g_T(t),
% we can first decide the signal within the period of 'T' for g(t)
    Dt=0.002;   % Time interval (to sample the signal)
    T=6;        % period=T
    M=3;        % To generate 2M periods of the signal
    t=[0:Dt:T-Dt];  %"t" goes for one period [0, T] in Dt increment
% Then evaluate the signal over the range of "T"
    y=exp(-abs(t)/2).*sin(2*pi*t).*(ustep(t)-ustep(t-4));
% Multiple periods can now be generated.
    time=[];
    y_periodic=[];
for i=-M:M-1,
    time=[time i*T+t];
    y_periodic=[y_periodic y];
end
    figure(1); fy=plot(time,y_periodic);
    set(fy,'Linewidth',2);xlabel('{\it t}');
    echo on
%   Compute average power
    y_power=sum(y_periodic*y_periodic')*Dt/(max(time)-min(time))
%   Compute signal energy in 1 period T
    y_energyT=sum(y.*conj(y))*Dt
```

The program generates a periodic signal as shown in Fig. 2.20 and numerical answers:

```
y_power =
      0.0813

y_energyT =
      0.4878
```

Signal Correlation

The MATLAB program can implement directly the concept of signal correlation introduced in Section 2.5. In the next computer example, we provide a program, `sign_cor.m`, that evaluates the signal correlation coefficients between $x(t)$ and signals $g_1(t)$, $g_2(t)$, ... $g_5(t)$. The program first generates Fig. 2.21, which illustrates the six signals in the time domain.

```
% (file name: sign_cor.m)
clear
%   To generate 6 signals x(t), g_1(t), ...  g_5(t);
%   of this Example
%   we can first decide the signal within the period of 'T' for g(t)
```

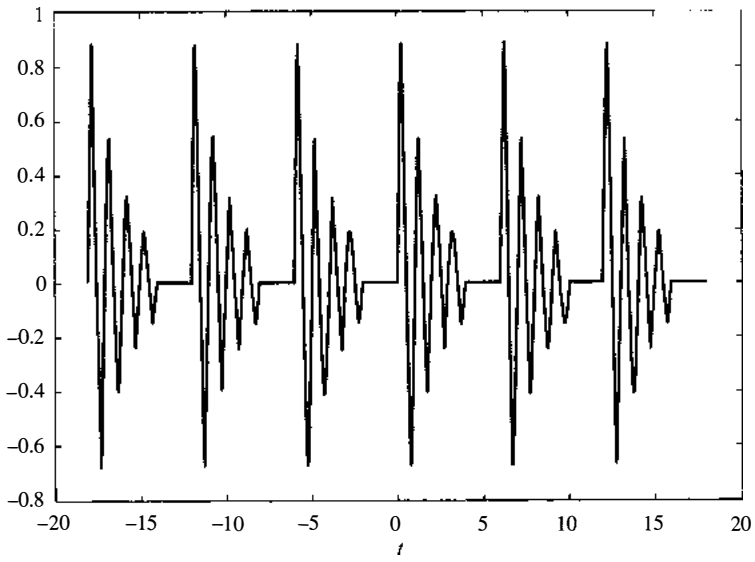


Figure 2.20 Generating a periodic signal.

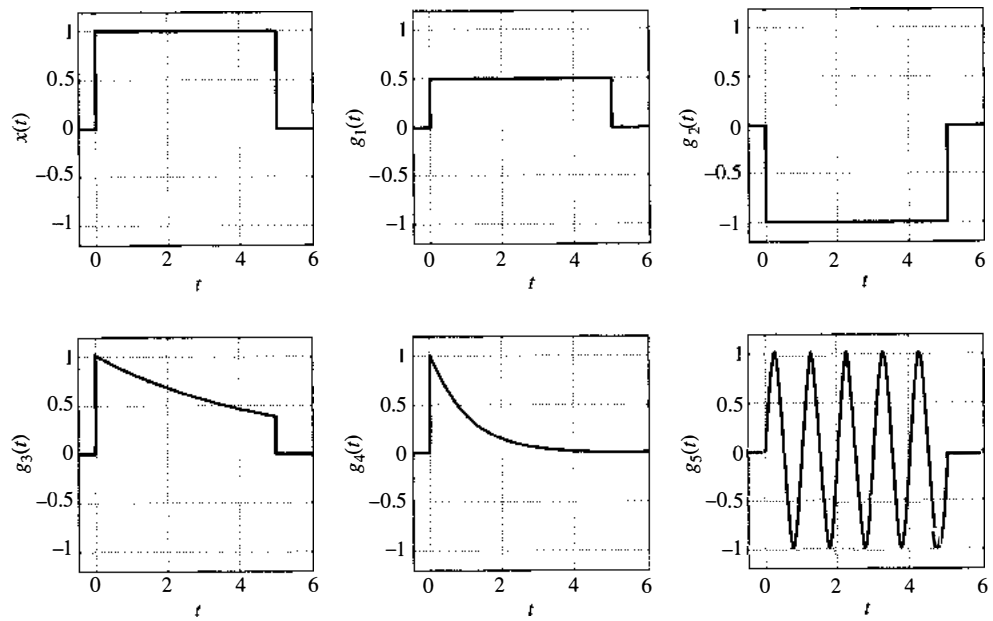


Figure 2.21 Six simple signals.

```
    Dt=0.01;     % time increment Dt
    T=6.0;       % time duration = T
    t=[-1:Dt:T];     %"t" goes between [-1, T] in Dt increment
% Then evaluate the signal over the range of "t" to plot
    x=ustep(t)-ustep(t-5);
    g1=0.5*(ustep(t)-ustep(t-5));
    g2=-(ustep(t)-ustep(t-5));
    g3=exp(-t/5).*(ustep(t)-ustep(t-5));
    g4=exp(-t).*(ustep(t)-ustep(t-5));
    g5=sin(2*pi*t).*(ustep(t)-ustep(t-5));
```

```
subplot(231); sig1=plot(t,x,'k');
xlabel('\it t'); ylabel('{\it x}({\it t})');    % Label axis
set(sig1,'Linewidth',2);            % change linewidth
axis([-.5 6 -1.2 1.2]); grid            % set plot range
subplot(232); sig2=plot(t,g1,'k');
xlabel('\it t'); ylabel('{\it g}_1({\it t})');
set(sig2,'Linewidth',2);
axis([-.5 6 -1.2 1.2]); grid
subplot(233); sig3=plot(t,g2,'k');
xlabel('\it t'); ylabel('{\it g}_2({\it t})');
set(sig3,'Linewidth',2);
axis([-.5 6 -1.2 1.2]); grid
subplot(234); sig4=plot(t,g3,'k');
xlabel('\it t'); ylabel('{\it g}_3({\it t})');
set(sig4,'Linewidth',2);
axis([-.5 6 -1.2 1.2]); grid
subplot(235); sig5=plot(t,g4,'k');
xlabel('\it t'); ylabel('{\it g}_4({\it t})');
set(sig5,'Linewidth',2);grid
axis([-.5 6 -1.2 1.2]);
subplot(236); sig6=plot(t,g5,'k');
xlabel('\it t'); ylabel('{\it g}_5({\it t})');
set(sig6,'Linewidth',2);grid
axis([-.5 6 -1.2 1.2]);

% Computing signal energies
E0=sum(x.*conj(x))*Dt;
E1=sum(g1.*conj(g1))*Dt;
E2=sum(g2.*conj(g2))*Dt;
E3=sum(g3.*conj(g3))*Dt;
E4=sum(g4.*conj(g4))*Dt;
E5=sum(g5.*conj(g5))*Dt;

c0=sum(x.*conj(x))*Dt/(sqrt(E0*E0))
c1=sum(x.*conj(g1))*Dt/(sqrt(E0*E1))
c2=sum(x.*conj(g2))*Dt/(sqrt(E0*E2))
c3=sum(x.*conj(g3))*Dt/(sqrt(E0*E3))
c4=sum(x.*conj(g4))*Dt/(sqrt(E0*E4))
c5=sum(x.*conj(g5))*Dt/(sqrt(E0*E5))
```

The six correlation coefficients are obtained from the program as

```
c0 =
     1

c1 =
     1

c2 =
    -1
```

```
c3 =
    0.9614

c4 =
    0.6282

c5 =
   8.6748e-17
```

Numerical Computation of Coefficients D_n

There are several ways to numerically compute the Fourier series coefficients D_n. We will use MATLAB to show how to use numerical integration in the evaluation of Fourier series.

To carry out a direct numerical integration of Eq. (2.60), the first step is to define the symbolic expression of the signal $g(t)$ under analysis. We use the triangle function $\Delta(t)$ in the following example.

```
% (funct_tri.m)
% A standard triangle function of base -1 to 1
    function y = funct_tri(t)
% Usage  y = func_tri(t)
%    t = input variable i
y=((t>-1)-(t>1)).*(1-abs(t));
```

Once the file `funct_tri.m` defines the function $y = g(t)$, we can directly carry out the necessary integration of Eq. (2.60) for a finite number of Fourier series coefficients $\{D_n, n = -N, \ldots, -1, 0, 1, \ldots, N\}$. We provide the following MATLAB program called `FSexample.m` to evaluate the Fourier series of $\Delta(t/2)$ with period $[a, b]$ $(a = -2, b = 2)$. In this example, $N = 11$ is selected. Executing this short program in MATLAB will generate Fig. 2.22 with both amplitude and angle of D_n.

```
% (file name:  FSexp_a.m)
%   This example shows how to numerically evaluate
%   the exponential Fourier series coefficients Dn
%   directly.
%   The user needs to define a symbolic function
%   g(t).  In this example, g(t)=funct_tri(t).
echo off; clear; clf;
    j=sqrt(-1); % Define j for complex algebra
    b=2; a=-2;  % Determine one signal period
    tol=1.e-5;  % Set integration error tolerance
    T=b-a;      % length of the period
    N=11;       % Number of FS coefficients
            % on each side of zero frequency
    Fi=[-N:N]*2*pi/T;   % Set frequency range
```

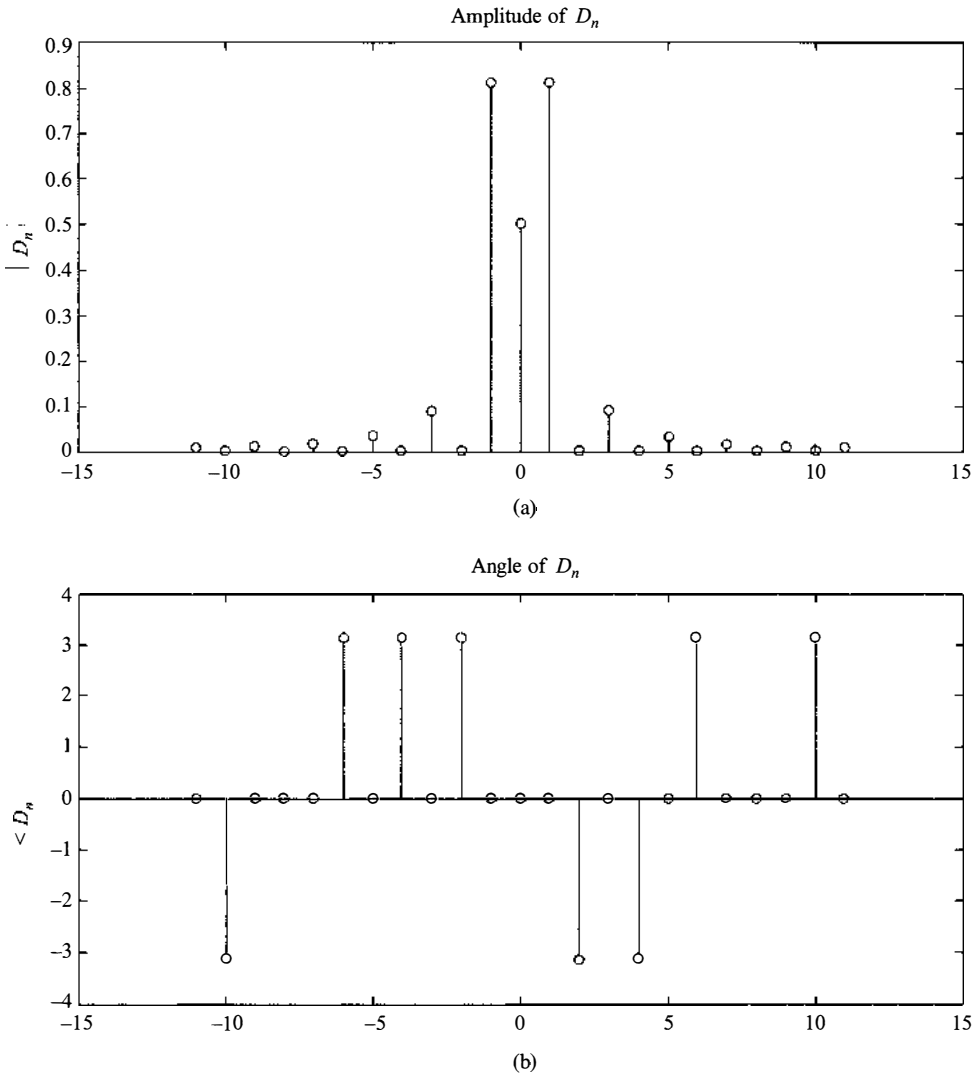


Figure 2.22 Exponential Fourier series coefficients of a repeated $\Delta(t/2)$ with period $T = 4$.

```
% now calculate D_0 and store it in D(N+1);
    Func= @(t) funct_tri(t/2);
    D(N+1)=1/T*quad(Func,a,b,tol);  % Using quad.m integration
for i=1:N
% Calculate Dn for n=1,...,N (stored in D(N+2) ... D(2N+1)
    Func= @(t) exp(-j*2*pi*t*i/T).*funct_tri(t/2);
    D(i+N+1)=quad(Func,a,b,tol);
% Calculate Dn for n=-N,...,-1 (stored in D(1) ... D(N)
    Func= @(t) exp(j*2*pi*t*(N+1-i)/T).*func_tri(t/2);
    D(i)= quad(Func,a,b,tol);
end
figure(1);
subplot(211);s1=stem([-N:N],abs(D));
set(s1,'Linewidth',2); ylabel('|{\it D}_{\it n}|');
```

```
title('Amplitude of {\it D}_{\it n}')
subplot(212);s2=stem([-N:N],angle(D));
set(s2,'Linewidth',2); ylabel('<{\it D}_{\it n}');
title('Angle of {\it D}_{\it n}');
```

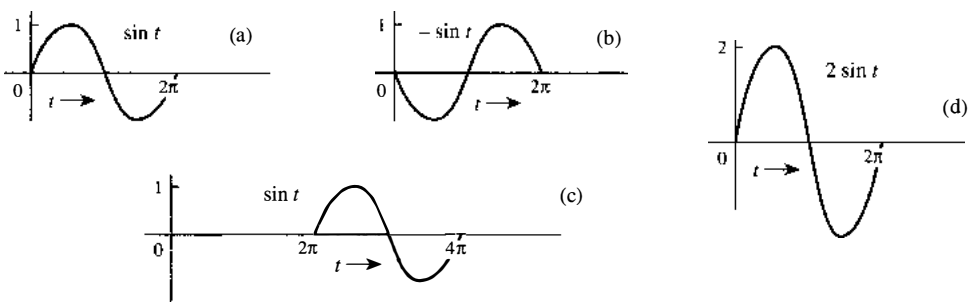
REFERENCES

1. P. L. Walker, *The Theory of Fourier Series and Integrals*, Wiley-Interscience, New York, 1986.
2. R. V. Churchill, and J. W. Brown, *Fourier Series and Boundary Value Problems*, 3rd ed., McGraw-Hill, New York, 1978.

PROBLEMS

2.1-1 Find the energies of the signals shown in Fig. P2.1-1. Comment on the effect on energy of sign change, time shift, or doubling of the signal. What is the effect on the energy if the signal is multiplied by k?

Figure P.2.1-1



2.1-2 **(a)** Find E_x and E_y, the energies of the signals $x(t)$ and $y(t)$ shown in Fig. P2.1-2a. Sketch the signals $x(t) + y(t)$ and $x(t) - y(t)$ and show that the energy of either of these two signals is equal to $E_x + E_y$. Repeat the procedure for signal pair in Fig. P2.1-2b.

(b) Now repeat the procedure for signal pair in Fig. P2.1-2c. Are the energies of the signals $x(t) + y(t)$ and $x(t) - y(t)$ identical in this case?

2.1-3 Find the power of a sinusoid $C \cos(\omega_0 t + \theta)$.

2.1-4 Show that if $\omega_1 = \omega_2$, the power of $g(t) = C_1 \cos(\omega_1 t + \theta_1) + C_2 \cos(\omega_2 t + \theta_2)$ is $[C_1{}^2 + C_2{}^2 + 2C_1C_2 \cos(\theta_1 - \theta_2)]/2$, which is not equal to $(C_1{}^2 + C_2{}^2)/2$.

2.1-5 Find the power of the periodic signal $g(t)$ shown in Fig. P2.1-5. Find also the powers and the rms values of **(a)** $-g(t)$ **(b)** $2g(t)$ **(c)** $cg(t)$. Comment.

2.1-6 Find the power and the rms value for the signals in **(a)** Fig. P2-1-6a; **(b)** Fig. 2.16; **(c)** Fig. P2-1-6b; **(d)** Fig. P2.7-4a; **(e)** Fig. P2.7-4c.

Figure P.2.1-2

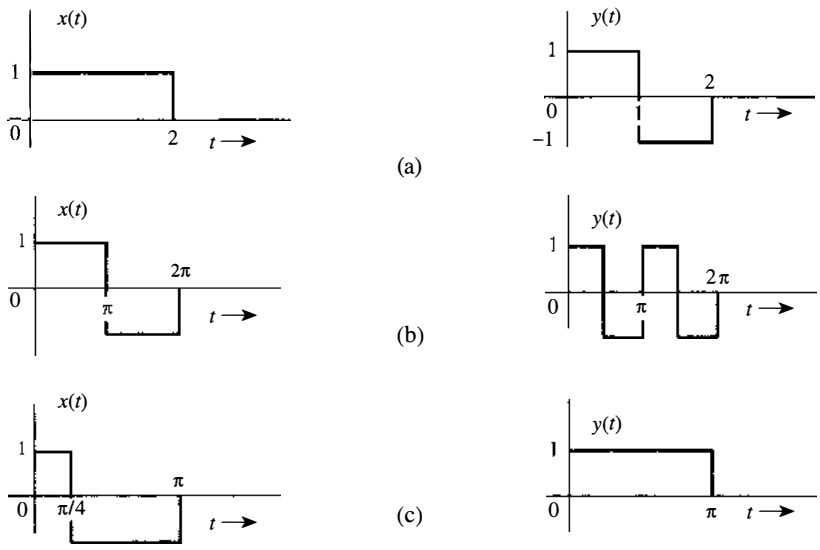


Figure P.2.1-5

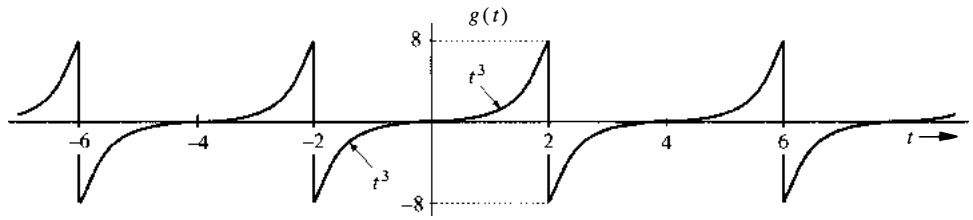
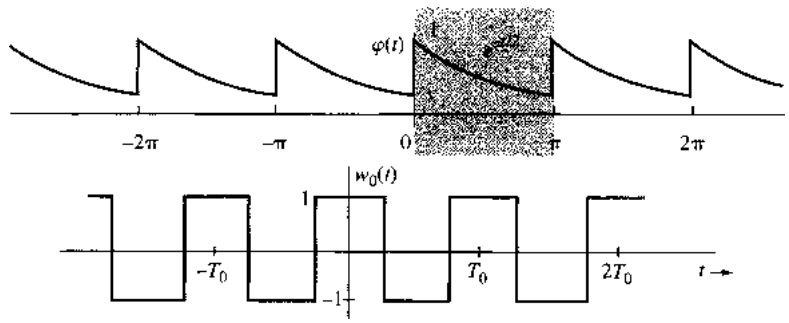


Figure P.2.1-6



2.1-7 Show that the power of a signal $g(t)$ given by

$$g(t) = \sum_{k=m}^{n} D_k e^{j\omega_k t} \qquad \omega_i \neq \omega_k \text{ for all } i \neq k$$

is (Parseval's theorem)

$$P_g = \sum_{k=m}^{n} |D_k|^2$$

2.1-8 Determine the power and the rms value for each of the following signals:

(a) $10\cos\left(100t+\frac{\pi}{3}\right)$

(b) $10\cos\left(100t+\frac{\pi}{3}\right)+16\sin\left(150t+\frac{\pi}{5}\right)$

(c) $(10+2\sin 3t)\cos 10t$

(d) $10\cos 5t\cos 10t$

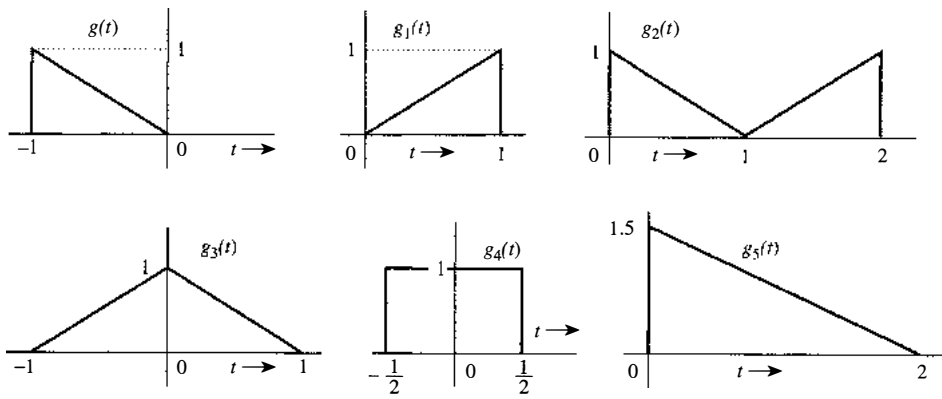
(e) $10\sin 5t\cos 10t$

(f) $e^{j\alpha t}\cos\omega_0 t$

2.2-1 Show that an exponential e^{-at} starting at $-\infty$ is neither an energy nor a power signal for any real value of a. However, if a is imaginary, it is a power signal with power $P_g = 1$ regardless of the value of a.

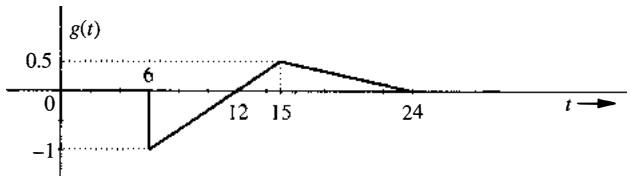
2.3-1 In Fig. P2.3-1, the signal $g_1(t) = g(-t)$. Express signals $g_2(t)$, $g_3(t)$, $g_4(t)$, and $g_5(t)$ in terms of signals $g(t)$, $g_1(t)$, and their time-shifted, time-scaled, or time-inverted versions. For instance, $g_2(t) = g(t-T)+g_1(t-T)$ for some suitable value of T. Similarly, both $g_3(t)$ and $g_4(t)$ can be expressed as $g(t-T)+g(t-T)$ for some suitable value of T. In addition, $g_5(t)$ can be expressed as $g(t)$ time-shifted, time-scaled, and then multiplied by a constant.

Figure P.2.3-1



2.3-2 For the signal $g(t)$ shown in Fig. P2.3-2, sketch the following signals: **(a)** $g(-t)$; **(b)** $g(t+6)$; **(c)** $g(3t)$; **(d)** $g(6-t)$.

Figure P.2.3-2

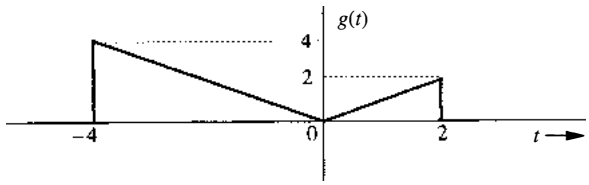


2.3-3 For the signal $g(t)$ shown in Fig. P2.3-3, sketch **(a)** $g(t-4)$; **(b)** $g(t/1.5)$; **(c)** $g(2t-4)$; **(d)** $g(2-t)$.

Hint: Recall that replacing t with $t-T$ delays the signal by T. Thus, $g(2t-4)$ is $g(2t)$ with t replaced by $t-2$. Similarly, $g(2-t)$ is $g(-t)$ with t replaced by $t-2$.

2.3-4 For an energy signal $g(t)$ with energy E_g, show that the energy of any one of the signals $-g(t)$, $g(-t)$, and $g(t-T)$ is E_g. Show also that the energy of $g(at)$ as well as $g(at-b)$ is E_g/a. This shows that time inversion and time shifting do not affect signal energy. On the other

Figure P.2.3-3



hand, time compression of a signal by a factor a reduces the energy by the factor a. What is the effect on signal energy if the signal is **(a)** time-expanded by a factor a ($a > 1$) and **(b)** multiplied by a constant a?

2.3-5 Simplify the following expressions:

(a) $\left(\dfrac{\tan t}{2t^2+1}\right)\delta(t)$

(b) $\left(\dfrac{j\omega-3}{\omega^2+9}\right)\delta(\omega)$

(c) $\left[e^{-t}\cos(3t-\pi/3)\right]\delta(t+\pi)$

(d) $\left(\dfrac{\sin \pi(t+2)}{t^2-4}\right)\delta(t-1)$

(e) $\left(\dfrac{\cos(\pi t)}{t+2}\right)\delta(2t+3)$

(f) $\left(\dfrac{\sin k\omega}{\omega}\right)\delta(\omega)$

Hint: Use Eq. (2.10b). For part **(f)** use L'Hospital's rule.

2.3-6 Evaluate the following integrals:

(a) $\int_{-\infty}^{\infty} g(\tau)\delta(t-\tau)\,d\tau$

(b) $\int_{-\infty}^{\infty} \delta(\tau)g(t-\tau)\,d\tau$

(c) $\int_{-\infty}^{\infty} \delta(t)e^{-j\omega t}\,dt$

(d) $\int_{-\infty}^{1} \delta(t-2)\sin \pi t\,dt$

(e) $\int_{-2}^{\infty} \delta(3+t)e^{-t}\,dt$

(f) $\int_{-2}^{2} (t^3+4)\delta(1-t)\,dt$

(g) $\int_{-\infty}^{\infty} g(2-t)\delta(3-t)\,dt$

(h) $\int_{-\infty}^{\infty} e^{(x-1)}\cos\frac{\pi}{2}(x-5)\delta(2x-3)\,dx$

Hint: $\delta(x)$ is located at $x = 0$. For example, $\delta(1-t)$ is located at $1-t=0$; that is, at $t=1$, and so on.

2.3-7 Prove that

$$\delta(at) = \frac{1}{|a|}\delta(t)$$

Hence show that

$$\delta(\omega) = \frac{1}{2\pi}\delta(f) \qquad \text{where} \quad \omega = 2\pi f$$

Hint: Show that

$$\int_{-\infty}^{\infty} \phi(t)\delta(at)\,dt = \frac{1}{|a|}\phi(0)$$

2.4-1 Derive Eq. (2.19) in an alternate way by observing that $\mathbf{e} = (\mathbf{g} - c\,\mathbf{x})$, and

$$|\mathbf{e}|^2 = (\mathbf{g}-c\,\mathbf{x})\cdot(\mathbf{g}-c\,\mathbf{x}) = |\mathbf{g}|^2 + c^2|\mathbf{x}|^2 - 2c\mathbf{g}\cdot\mathbf{x}$$

To minimize $|\mathbf{e}|^2$, equate its derivative with respect to c to zero.

2.4-2 For the signals $g(t)$ and $x(t)$ shown in Fig. P2.4-2, find the component of the form $x(t)$ contained in $g(t)$. In other words, find the optimum value of c in the approximation $g(t) \approx cx(t)$ so that the error signal energy is minimum. What is the resulting error signal energy?

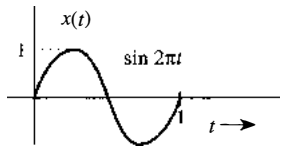
Figure P.2.4-2



2.4-3 For the signals $g(t)$ and $x(t)$ shown in Fig. P2.4-2, find the component of the form $g(t)$ contained in $x(t)$. In other words, find the optimum value of c in the approximation $x(t) \approx cg(t)$ so that the error signal energy is minimum. What is the resulting error signal energy?

2.4-4 Repeat Prob. 2.4-2 if $x(t)$ is a sinusoid pulse shown in Fig. P2.4-4.

Figure P.2.4-4



2.4-5 The Energies of the two energy signals $x(t)$ and $y(t)$ are E_x and E_y, respectively.

(a) If $x(t)$ and $y(t)$ are orthogonal, then show that the energy of the signal $x(t) + y(t)$ is identical to the energy of the signal $x(t) - y(t)$, and is given by $E_x + E_y$.

(b) if $x(t)$ and $y(t)$ are orthogonal, find the energies of signals $c_1x(t) + c_2y(t)$ and $c_1x(t) - c_2y(t)$.

(c) We define E_{xy}, the cross-energy of the two energy signals $x(t)$ and $y(t)$, as

$$E_{xy} = \int_{-\infty}^{\infty} x(t)y^*(t)\,dt$$

If $z(t) = x(t) \pm y(t)$, then show that

$$E_z = E_x + E_y \pm (E_{xy} + E_{yx})$$

2.4-6 Let $x_1(t)$ and $x_2(t)$ be two unit energy signals orthogonal over an interval from $t = t_1$ to t_2. Signals $x_1(t)$ and $x_2(t)$ are unit energy, orthogonal signals; we can represent them by two unit length, orthogonal vectors $(\mathbf{x}_1, \mathbf{x}_2)$. Consider a signal $g(t)$ where

$$g(t) = c_1x_1(t) + c_2x_2(t) \qquad t_1 \le t \le t_2$$

This signal can be represented as a vector $\mathbf{g}$ by a point (c_1, c_2) in the $x_1 - x_2$ plane.

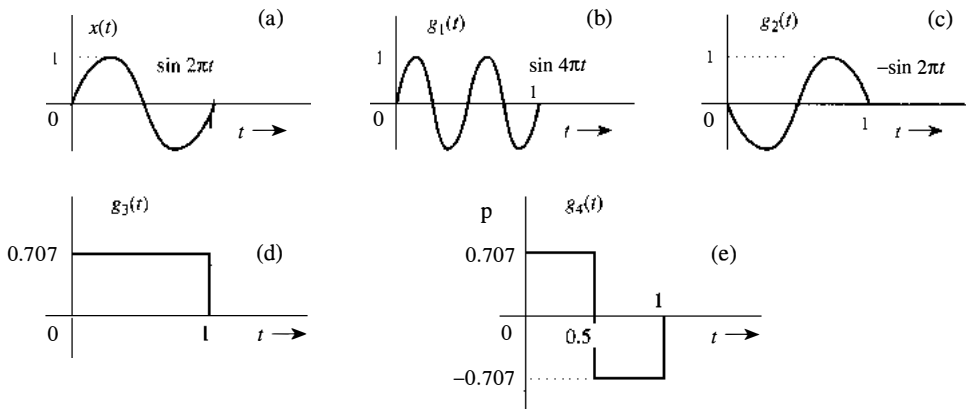
(a) Determine the vector representation of the following six signals in this two-dimensional vector space:

(i) $g_1(t) = 2x_1(t) - x_2(t)$
(ii) $g_2(t) = -x_1(t) + 2x_2(t)$
(iii) $g_3(t) = -x_2(t)$
(iv) $g_4(t) = x_1(t) + 2x_2(t)$
(v) $g_5(t) = 2x_1(t) + x_2(t)$
(vi) $g_6(t) = 3x_1(t)$

(b) Point out pairs of mutually orthogonal vectors among these six vectors. Verify that the pairs of signals corresponding to these orthogonal vectors are also orthogonal.

2.5-1 Find the correlation coefficient c_n of signal $x(t)$ and each of the four pulses $g_1(t)$, $g_2(t)$, $g_3(t)$, and $g_4(t)$ shown in Fig. P2.5-1. To provide maximum margin against the noise along the transmission path, which pair of pulses would you select for a binary communication?

Figure P.2.5-1



2.7-1 **(a)** Sketch the signal $g(t) = t^2$ and find the exponential Fourier series to represent $g(t)$ over the interval $(-1, 1)$. Sketch the Fourier series $\varphi(t)$ for all values of t.

(b) Verify Parseval's theorem [Eq. (2.68a)] for this case, given that

$$\sum_{n=1}^{\infty} \frac{1}{n^4} = \frac{\pi^4}{90}$$

2.7-2 **(a)** Sketch the signal $g(t) = t$ and find the exponential Fourier series to represent $g(t)$ over the interval $(-\pi, \pi)$. Sketch the Fourier series $\varphi(t)$ for all values of t.

(b) Verify Parseval's theorem [Eq. (2.68a)] for this case, given that

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

2.7-3 If a periodic signal satisfies certain symmetry conditions, the evaluation of the Fourier series coefficients is somewhat simplified.

(a) Show that if $g(t) = g(-t)$ (even symmetry), then the coefficients of the exponential Fourier series are real.

(b) Show that if $g(t) = -g(-t)$ (odd symmetry), the coefficients of the exponential Fourier series are imaginary.

(c) Show that in each case, the Fourier coefficients can be evaluated by integrating the periodic signal over the half-cycle only. This is because the entire information of one cycle is implicit in a half-cycle owing to symmetry.

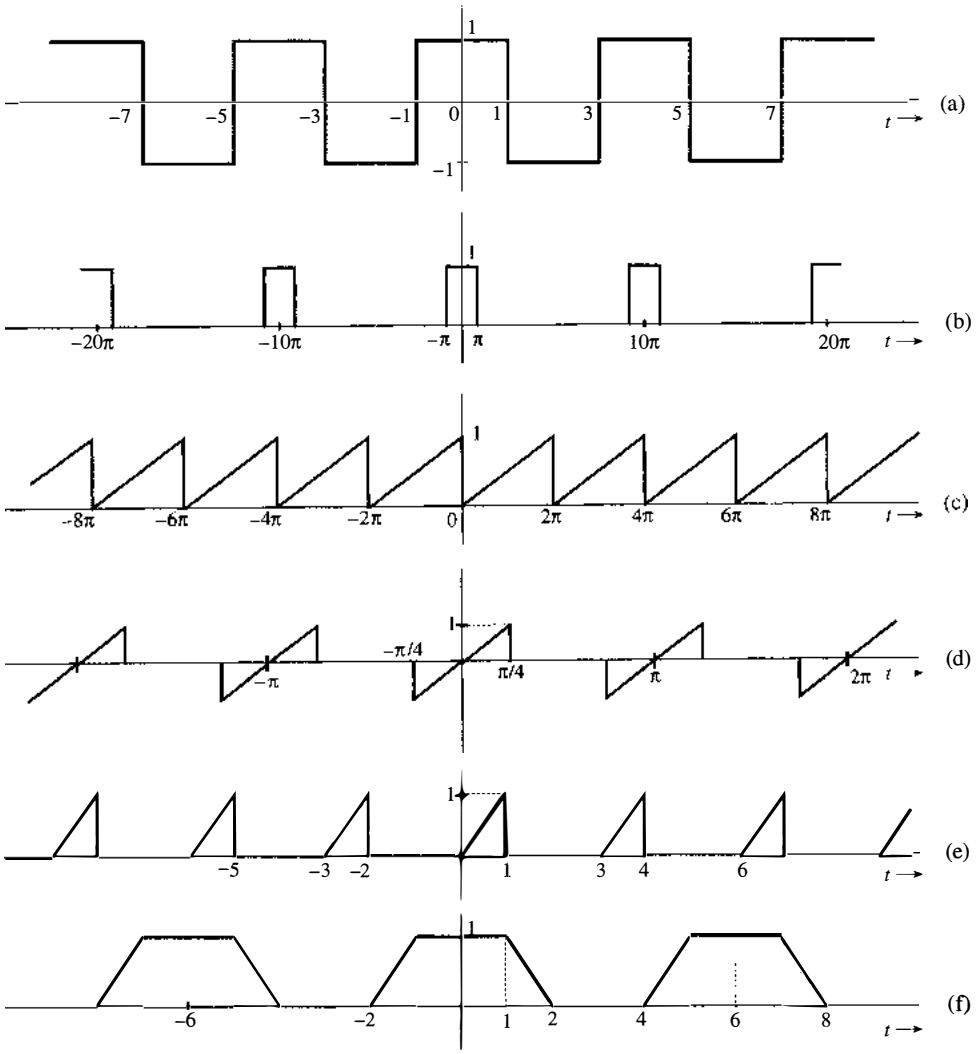
Hint: If $g_e(t)$ and $g_o(t)$ are even and odd functions, respectively, of t, then (assuming no impulse or its derivative at the origin),

$$\int_{-a}^{a} g_e(t)\,dt = \int_{0}^{2a} g_e(t)\,dt \quad \text{and} \quad \int_{-a}^{a} g_o(t)\,dt = 0$$

Also, the product of an even and an odd function is an odd function, the product of two odd functions is an even function, and the product of two even functions is an even function.

2.7-4 For each of the periodic signals shown in Fig. P2.7-4, find the exponential Fourier series and sketch the amplitude and phase spectra. Note any symmetric property.

Figure P.2.7-4



2.7-5 **(a)** Show that an arbitrary function $g(t)$ can be expressed as a sum of an even function $g_e(t)$ and an odd function $g_o(t)$:

$$g(t) = g_e(t) + g_o(t)$$

$$\textit{Hint}: \quad g(t) = \underbrace{\frac{1}{2}\big[g(t) + g(-t)\big]}_{g_e(t)} + \underbrace{\frac{1}{2}\big[g(t) - g(-t)\big]}_{g_o(t)}$$

(b) Determine the odd and even components of the following functions: **(i)** $u(t)$; **(ii)** $e^{-at}u(t)$; **(iii)** e^{jt}.

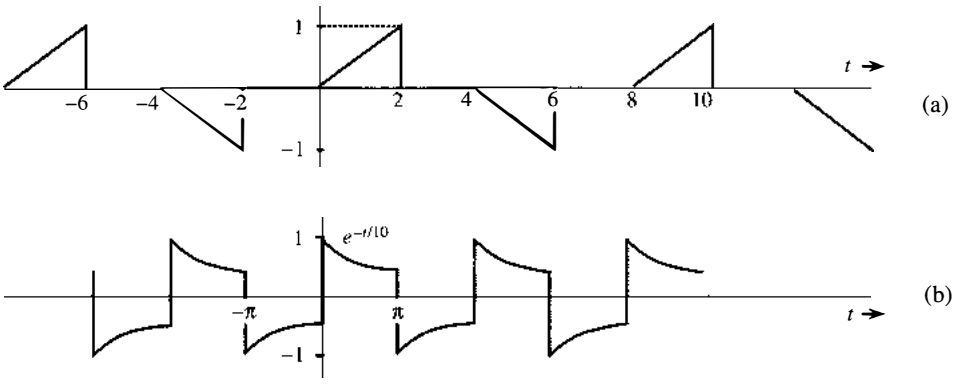
2.7-6 **(a)** If the two halves of one period of a periodic signal are of identical shape except that one is the negative of the other, the periodic signal is said to have a **half-wave symmetry**. If a periodic signal $g(t)$ with a period T_0 satisfies the half-wave symmetry condition, then

$$g\left(t - \frac{T_0}{2}\right) = -g(t)$$

In this case, show that all the even-numbered harmonics (coefficients) vanish.

(b) Use this result to find the Fourier series for the periodic signals in Fig. P2.7-6.

Figure P.2.7-6



2.8-1 A periodic signal $g(t)$ is expressed by the following Fourier series:

$$g(t) = 3\sin t + \cos\left(3t - \frac{2\pi}{3}\right) + 2\cos\left(8t + \frac{\pi}{3}\right)$$

(a) By applying Euler's identities on the signal $g(t)$ directly, write the exponential Fourier series for $g(t)$.

(b) By applying Euler's identities on the signal $g(t)$ directly, sketch the exponential Fourier series spectra.

3 ANALYSIS AND TRANSMISSION OF SIGNALS

Electrical engineers instinctively think of signals in terms of their frequency spectra and think of systems in terms of their frequency responses. Even teenagers know about audio signals having a bandwidth of 20 kHz and good-quality loud speakers responding up to 20 kHz. This is basically thinking in the frequency domain. In the last chapter we discussed spectral representation of periodic signals (Fourier series). In this chapter we extend this spectral representation to aperiodic signals.

3.1 APERIODIC SIGNAL REPRESENTATION BY FOURIER INTEGRAL

Applying a limiting process, we now show that an aperiodic signal can be expressed as a continuous sum (integral) of everlasting exponentials. To represent an aperiodic signal $g(t)$ such as the one shown in Fig. 3.1a by everlasting exponential signals, let us construct a new periodic signal $g_{T_0}(t)$ formed by repeating the signal $g(t)$ every T_0 seconds, as shown in Fig. 3.1b. The period T_0 is made long enough to avoid overlap between the repeating pulses. The periodic signal $g_{T_0}(t)$ can be represented by an exponential Fourier series. If we let $T_0 \to \infty$, the pulses in the periodic signal repeat after an infinite interval, and therefore

$$\lim_{T_0 \to \infty} g_{T_0}(t) = g(t)$$

Thus, the Fourier series representing $g_{T_0}(t)$ will also represent $g(t)$ in the limit $T_0 \to \infty$. The exponential Fourier series for $g_{T_0}(t)$ is given by

$$g_{T_0}(t) = \sum_{n=-\infty}^{\infty} D_n e^{jn\omega_0 t} \tag{3.1}$$

in which

$$D_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} g_{T_0}(t) e^{-jn\omega_0 t}\, dt \tag{3.2a}$$

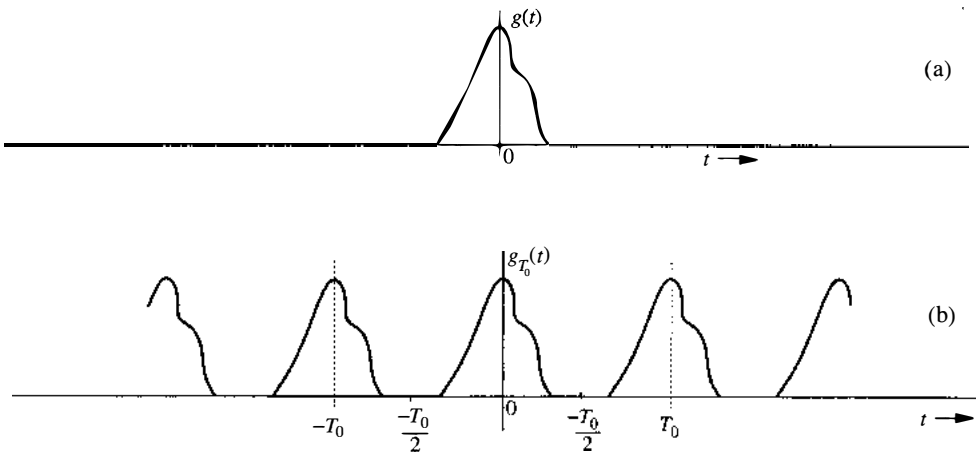


Figure 3.1 Construction of a periodic signal by periodic extension of $g(t)$.

and

$$\omega_0 = \frac{2\pi}{T_0} = 2\pi f_0 \tag{3.2b}$$

Observe that integrating $g_{T_0}(t)$ over $(-T_0/2, T_0/2)$ is the same as integrating $g(t)$ over $(-\infty, \infty)$. Therefore, Eq. (3.2a) can be expressed as

$$D_n = \frac{1}{T_0}\int_{-\infty}^{\infty} g(t)e^{-jn\omega_0 t}\,dt \tag{3.2c}$$

$$= \frac{1}{T_0}\int_{-\infty}^{\infty} g(t)e^{-j2\pi nf_0 t}\,dt$$

It is interesting to see how the nature of the spectrum changes as T_0 increases. To understand this fascinating behavior, let us define $G(f)$, a continuous function of ω, as

$$G(f) = \int_{-\infty}^{\infty} g(t)e^{-j\omega t}\,dt \tag{3.3}$$

$$= \int_{-\infty}^{\infty} g(t)e^{-j2\pi ft}\,dt \tag{3.4}$$

in which $\omega = 2\pi f$. A glance at Eqs. (3.2c) and (3.3) shows that

$$D_n = \frac{1}{T_0}G(nf_0) \tag{3.5}$$

This in turn shows that the Fourier coefficients D_n are $(1/T_0$ times) the samples of $G(f)$ uniformly spaced at intervals of f_0 Hz, as shown in Fig. 3.2a.*

Therefore, $(1/T_0)G(f)$ is the envelope for the coefficients D_n. We now let $T_0 \to \infty$ by doubling T_0 repeatedly. Doubling T_0 halves the fundamental frequency f_0, so that there are now twice as many components (samples) in the spectrum. However, by doubling T_0, the envelope $(1/T_0)G(f)$ is halved, as shown in Fig. 3.2b. If we continue this process of doubling T_0 repeatedly, the spectrum progressively becomes denser while its magnitude becomes smaller.

* For the sake of simplicity we assume D_n and therefore $G(f)$ in Fig. 3.2 to be real. The argument, however, is also valid for complex D_n [or $G(f)$].

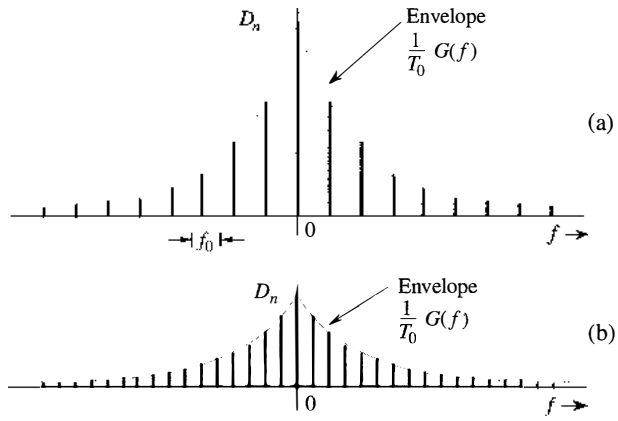


Figure 3.2 Change in the Fourier spectrum when the period T_0 in Fig. 3.1 is doubled.

Note, however, that the relative shape of the envelope remains the same [proportional to $G(f)$ in Eq. (3.3)]. In the limit as $T_0 \to \infty, f_0 \to 0$ and $D_n \to 0$. This means that the spectrum is so dense that the spectral components are spaced at zero (infinitesimal) interval. At the same time, the amplitude of each component is zero (infinitesimal). We have *nothing of everything, yet we have something!* This sounds like *Alice in Wonderland*, but as we shall see, these are the classic characteristics of a very familiar phenomenon.*

Substitution of Eq. (3.5) in Eq. (3.1) yields

$$g_{T_0}(t) = \sum_{n=-\infty}^{\infty} \frac{G(nf_0)}{T_0} e^{jn2\pi f_0 t} \tag{3.6}$$

As $T_0 \to \infty, f_0 = 1/T_0$ becomes infinitesimal ($f_0 \to 0$). Because of this, we shall replace f_0 by a more appropriate notation, Δf. In terms of this new notation, Eq. (3.2b) becomes

$$\Delta f = \frac{1}{T_0}$$

and Eq. (3.6) becomes

$$g_{T_0}(t) = \sum_{n=-\infty}^{\infty} [G(n\Delta f)\Delta f] e^{(j2\pi n\Delta f)t} \tag{3.7a}$$

Equation (3.7a) shows that $g_{T_0}(t)$ can be expressed as a sum of everlasting exponentials of frequencies $0, \pm\Delta f, \pm 2\Delta f, \pm 3\Delta f, \cdots$ (the Fourier series). The amount of the component of frequency $n\Delta f$ is $[G(n\Delta f)\Delta f]$. In the limit as $T_0 \to \infty$, $\Delta f \to 0$ and $g_{T_0}(t) \to g(t)$. Therefore,

$$g(t) = \lim_{T_0 \to \infty} g_{T_0}(t) = \lim_{\Delta f \to 0} \sum_{n=-\infty}^{\infty} G(n\Delta f) e^{(j2\pi n\Delta f)t} \Delta f \tag{3.7b}$$

The sum on the right-hand side of Eq. (3.7b) can be viewed as the area under the function $G(f)e^{j2\pi ft}$, as shown in Fig. 3.3. Therefore,

* You may consider this as an irrefutable proof of the proposition that 0% ownership of everything is better than 100% ownership of nothing!

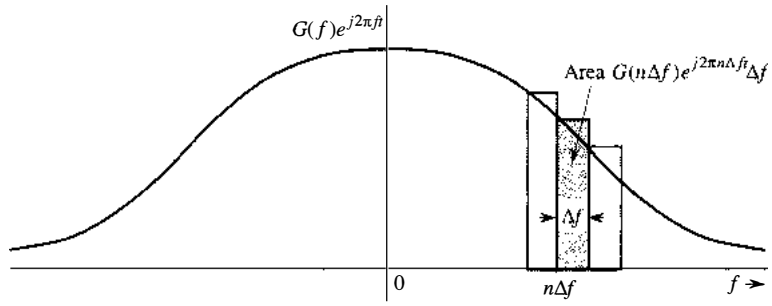


Figure 3.3 The Fourier series becomes the Fourier integral in the limit as $T_0 \to \infty$.

$$g(t) = \int_{-\infty}^{\infty} G(f)e^{j2\pi ft}df \tag{3.8}$$

The integral on the right-hand side is called the **Fourier integral**. We have now succeeded in representing an aperiodic signal $g(t)$ by a Fourier integral* (rather than a Fourier series). This integral is basically a Fourier series (in the limit) with fundamental frequency $\Delta f \to 0$, as seen from Eq. (3.7b). The amount of the exponential $e^{j2\pi n\Delta ft}$ is $G(n\Delta f)\Delta f$. Thus, the function $G(f)$ given by Eq. (3.3) acts as a spectral function.

We call $G(f)$ the **direct** Fourier transform of $g(t)$, and $g(t)$ the **inverse** Fourier transform of $G(f)$. The same information is conveyed by the statement that $g(t)$ and $G(f)$ are a Fourier transform pair. Symbolically, this is expressed as

$$G(f) = \mathcal{F}[g(t)] \qquad \text{and} \qquad g(t) = \mathcal{F}^{-1}[G(f)]$$

or

$$g(t) \Longleftrightarrow G(f)$$

To recapitulate,

$$G(f) = \int_{-\infty}^{\infty} g(t)e^{-j\omega t}\,dt \tag{3.9a}$$

and

$$g(t) = \int_{-\infty}^{\infty} G(f)e^{j\omega t}\,df \tag{3.9b}$$

where $\omega = 2\pi f$.

It is helpful to keep in mind that the Fourier integral in Eq. (3.9b) is of the nature of a Fourier series with fundamental frequency Δf approaching zero [Eq. (3.7b)]. Therefore, most of the discussion and properties of Fourier series apply to the Fourier transform as well. We can plot the spectrum $G(f)$ as a function of f. Since $G(f)$ is complex, we have both amplitude and angle (or phase) spectra:

$$G(f) = |G(f)|e^{j\theta_g(f)}$$

in which $|G(f)|$ is the amplitude and $\theta_g(f)$ is the angle (or phase) of $G(f)$. From Eq. (3.9a),

$$G(-f) = \int_{-\infty}^{\infty} g(t)e^{j2\pi ft}\,dt$$

* This should not be considered as a rigorous proof of Eq. (3.8). The situation is not as simple as we have made it appear.[1]

f versus ω

Traditionally, we often use two equivalent notations of angular frequency ω and frequency f interchangeably in representing signals in the frequency domain. There is no conceptual difference between the use of angular frequency ω (in unit of radians per second) and frequency f (in units of hertz, Hz). Because of their direct relationship, we can simply substitute $\omega = 2\pi f$ into $G(f)$ to arrive at the Fourier transform relationship in the ω-domain:

$$\mathcal{F}[g(t)] = \int_{-\infty}^{\infty} g(t)e^{-j\omega t}dt \tag{3.10}$$

Because of the additional 2π factor in the variable ω used by Eq. (3.10), the inverse transform as a function of ω requires an extra division by 2π. Therefore, the notation of f is slightly favored in practice when one is writing Fourier transforms. For this reason, we shall, for the most part, denote the Fourier transform of signals as functions of $G(f)$ in this book. On the other hand, the notation of angular frequency ω can also offer some convenience in dealing with sinusoids. Thus, in later chapters, whenever it is *convenient and nonconfusing,* we shall use the two equivalent notations interchangeably.

Conjugate Symmetry Property

From Eq. (3.9a), it follows that if $g(t)$ is a real function of t, then $G(f)$ and $G(-f)$ are complex conjugates, that is,*

$$G(-f) = G^*(f) \tag{3.11}$$

Therefore,

$$|G(-f)| = |G(f)| \tag{3.12a}$$

$$\theta_g(-f) = -\theta_g(f) \tag{3.12b}$$

Thus, for real $g(t)$, the amplitude spectrum $|G(f)|$ is an even function, and the phase spectrum $\theta_g(f)$ is an odd function of f. This property (the **conjugate symmetry property**) is valid only for real $g(t)$. These results were derived for the Fourier spectrum of a periodic signal in Chapter 2 and should come as no surprise. *The transform $G(f)$ is the frequency domain specification of $g(t)$.*

Example 3.1 Find the Fourier transform of $e^{-at}u(t)$.

By definition [Eq. (3.9a)],

$$G(f) = \int_{-\infty}^{\infty} e^{-at}u(t)e^{-j2\pi ft}\,dt = \int_{0}^{\infty} e^{-(a+j2\pi f)t}\,dt = \frac{-1}{a+j2\pi f}e^{-(a+j2\pi f)t}\Big|_0^{\infty}$$

But $|e^{-j2\pi ft}| = 1$. Therefore, as $t \to \infty$, $e^{-(a+j2\pi f)t} = e^{-at}e^{-j2\pi ft} = 0$ if $a > 0$. Therefore,

$$G(f) = \frac{1}{a+j\omega} \qquad a > 0 \tag{3.13a}$$

---

* *Hermitian symmetry* is the term used to describe complex functions that satisfy Eq. (3.11)

where $\omega = 2\pi f$. Expressing $a + j\omega$ in the polar form as $\sqrt{a^2 + \omega^2}\, e^{j \tan^{-1}(\frac{\omega}{a})}$, we obtain

$$G(f) = \frac{1}{\sqrt{a^2 + (2\pi f)^2}} e^{-j \tan^{-1}(\frac{2\pi f}{a})} \tag{3.13ba}$$

Therefore,

$$|G(f)| = \frac{1}{\sqrt{a^2 + (2\pi f)^2}} \quad \text{and} \quad \theta_g(f) = -\tan^{-1}\left(\frac{2\pi f}{a}\right)$$

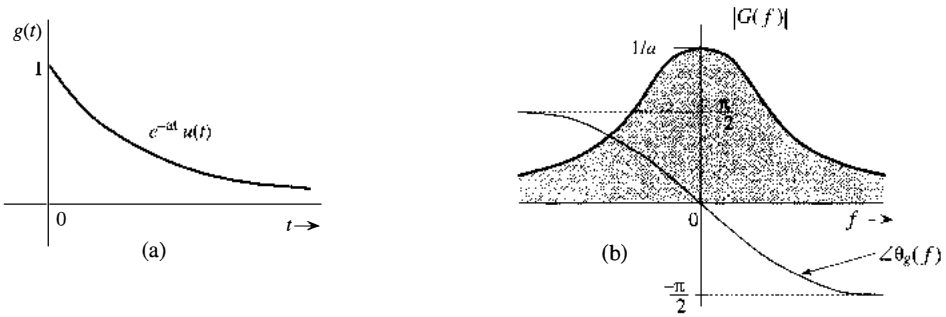


Figure 3.4 $e^{-at}u(t)$ and its Fourier spectra.

The amplitude spectrum $|G(f)|$ and the phase spectrum $\theta_g(f)$ are shown in Fig. 3.4b. Observe that $|G(f)|$ is an even function of f, and $\theta_g(f)$ is an odd function of f, as expected.

Existence of the Fourier Transform

In Example 3.1 we observed that when $a < 0$, the Fourier integral for $e^{-at}u(t)$ does not converge. Hence, the Fourier transform for $e^{-at}u(t)$ does not exist if $a < 0$ (growing exponentially). Clearly, not all signals are Fourier transformable. The existence of the Fourier transform is assured for any $g(t)$ satisfying the Dirichlet conditions, the first of which is*

$$\int_{-\infty}^{\infty} |g(t)|\, dt < \infty \tag{3.14}$$

To show this, recall that $|e^{-j2\pi ft}| = 1$. Hence, from Eq. (3.9a) we obtain

$$|G(f)| \leq \int_{-\infty}^{\infty} |g(t)|\, dt$$

This shows that the existence of the Fourier transform is assured if condition (3.14) is satisfied. Otherwise, there is no guarantee. We have seen in Example 3.1 that for an exponentially growing signal (which violates this condition) the Fourier transform does not exist. Although this condition is sufficient, it is not necessary for the existence of the Fourier transform of a signal.

* The remaining Dirichlet conditions are as follows: In any finite interval, $g(t)$ may have only a finite number of maxima and minima and a finite number of finite discontinuities. When these conditions are satisfied, the Fourier integral on the right-hand side of Eq. (3.9b) converges to $g(t)$ at all points where $g(t)$ is continuous and converges to the average of the right-hand and left-hand limits of $g(t)$ at points where $g(t)$ is discontinuous.

For example, the signal $(\sin at)/t$, violates condition (3.14), but does have a Fourier transform. Any signal that can be generated in practice satisfies the Dirichlet conditions and therefore has a Fourier transform. Thus, the physical existence of a signal is a sufficient condition for the existence of its transform.

Linearity of the Fourier Transform (Superposition Theorem)

The Fourier transform is linear; that is, if

$$g_1(t) \Longleftrightarrow G_1(f) \qquad \text{and} \qquad g_2(t) \Longleftrightarrow G_2(f)$$

then for all constants a_1 and a_2, we have

$$a_1 g_1(t) + a_2 g_2(t) \Longleftrightarrow a_1 G_1(f) + a_2 G_2(f) \tag{3.15}$$

The proof is simple and follows directly from Eq. (3.9a). This theorem simply states that linear combinations of signals in the time domain correspond to linear combinations of their Fourier transforms in the frequency domain. This result can be extended to any finite number of terms as

$$\sum_k a_k g_k(t) \Longleftrightarrow \sum_k a_k G_k(f)$$

for any constants $\{a_k\}$ and signals $\{g_k(t)\}$.

Physical Appreciation of the Fourier Transform

To understand any aspect of the Fourier transform, we should remember that Fourier representation is a way of expressing a signal in terms of everlasting sinusoids, or exponentials. The Fourier spectrum of a signal indicates the relative amplitudes and phases of the sinusoids that are required to synthesize that signal. A periodic signal's Fourier spectrum has finite amplitudes and exists at discrete frequencies (f and its multiples). Such a spectrum is easy to visualize, but the spectrum of an aperiodic signal is not easy to visualize because it has a continuous spectrum that exists at every frequency. The continuous spectrum concept can be appreciated by considering an analogous, more tangible phenomenon. One familiar example of a continuous distribution is the loading of a beam. Consider a beam loaded with weights $D_1, D_2, D_3, \ldots, D_n$ units at the uniformly spaced points $x_1, x_2, \ldots, x_n$, as shown in Fig. 3.5a. The total load W_T on the beam is given by the sum of these loads at each of the n points:

$$W_T = \sum_{i=1}^{n} D_i$$

Consider now the case of a continuously loaded beam, as shown in Fig. 3.5b. In this case, although there appears to be a load at every point, the load at any one point is zero. This does

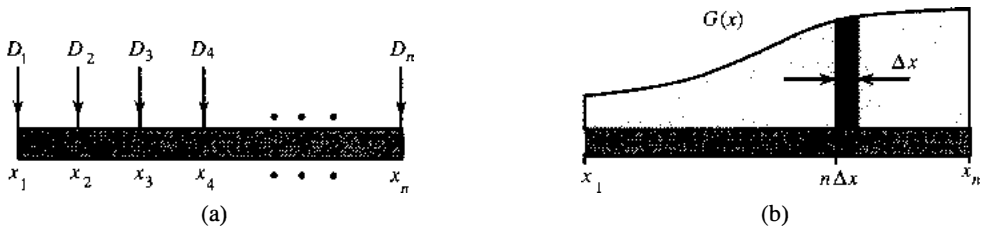


Figure 3.5 Analogy for Fourier transform.

not mean that there is no load on the beam. A meaningful measure of load in this situation is not the load at a point, but rather the loading density per unit length at that point. Let $G(x)$ be the loading density per unit length of beam. This means that the load over a beam length Δx ($\Delta x \to 0$) at some point x is $G(x)\Delta x$. To find the total load on the beam, we divide the beam into segments of interval Δx ($\Delta x \to 0$). The load over the nth such segment of length Δx is $[G(n\Delta x)]\,\Delta x$. The total load W_T is given by

$$W_T = \lim_{\Delta x \to 0} \sum_{x_1}^{x_n} G(n\Delta x)\,\Delta x$$

$$= \int_{x_1}^{x_n} G(x)\,dx$$

In the case of discrete loading (Fig. 3.5a), the load exists only at the n discrete points. At other points there is no load. On the other hand, in the continuously loaded case, the load exists at every point, but at any specific point x the load is zero. The load over a small interval Δx, however, is $[G(n\Delta x)]\,\Delta x$ (Fig. 3.5b). Thus, even though the load at a point x is zero, the relative load at that point is $G(x)$.

An exactly analogous situation exists in the case of a signal spectrum. When $g(t)$ is periodic, the spectrum is discrete, and $g(t)$ can be expressed as a sum of discrete exponentials with finite amplitudes:

$$g(t) = \sum_n D_n e^{j2\pi n f_0 t}$$

For an aperiodic signal, the spectrum becomes continuous; that is, the spectrum exists for every value of f, but the amplitude of each component in the spectrum is zero. The meaningful measure here is not the amplitude of a component of some frequency but the spectral density per unit bandwidth. From Eq. (3.7b) it is clear that $g(t)$ is synthesized by adding exponentials of the form $e^{j2\pi n\Delta f t}$, in which the contribution by any one exponential component is zero. But the contribution by exponentials in an infinitesimal band Δf located at $f = n\Delta f$ is $G(n\Delta f)\Delta f$, and the addition of all these components yields $g(t)$ in the integral form:

$$g(t) = \lim_{\Delta f \to 0} \sum_{n=-\infty}^{\infty} G(n\Delta f) e^{(jn2\pi f)t}\,\Delta f = \int_{-\infty}^{\infty} G(f) e^{j2\pi f t}\,df$$

The contribution by components within the band df is $G(f)\,df$, in which df is the bandwidth in hertz. Clearly $G(f)$ is the **spectral density** per unit bandwidth (in hertz). This also means that even if the amplitude of any one component is zero, the relative amount of a component of frequency f is $G(f)$. Although $G(f)$ is a spectral density, in practice it is customarily called the **spectrum** of $g(t)$ rather than the spectral density of $g(t)$. Deferring to this convention, we shall call $G(f)$ the Fourier spectrum (or Fourier transform) of $g(t)$.

3.2 TRANSFORMS OF SOME USEFUL FUNCTIONS

For convenience, we now introduce a compact notation for some useful functions such as rectangular, triangular, and interpolation functions.

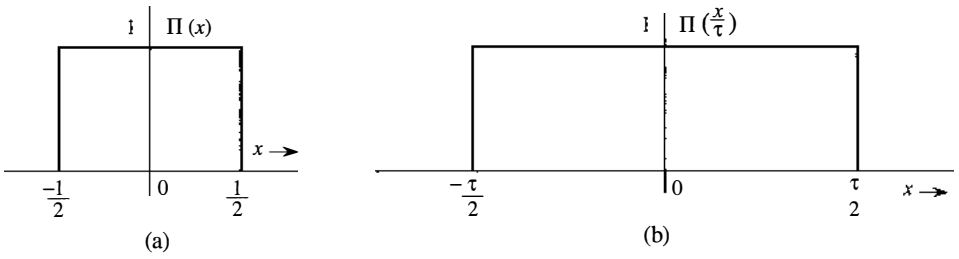


Figure 3.6 Rectangular pulse.

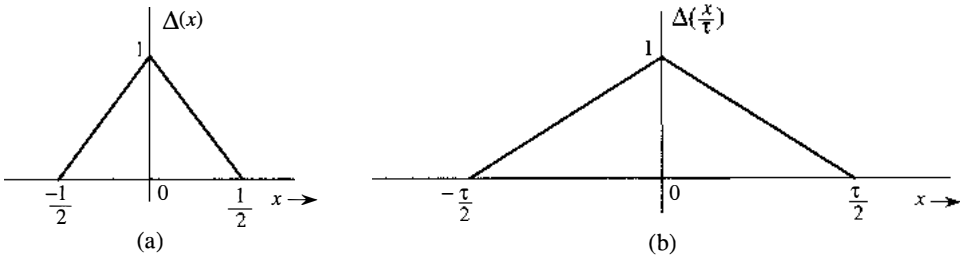


Figure 3.7 Triangular pulse.

Unit Rectangular Function

We use the pictorial notation $\Pi(x)$ for a rectangular pulse of unit height and unit width, centered at the origin, as shown in Fig. 3.6a:

$$\Pi(x) = \begin{cases} 1 & |x| \leq \frac{1}{2} \\ 0.5 & |x| = \frac{1}{2} \\ 0 & |x| > \frac{1}{2} \end{cases} \tag{3.16}$$

Notice that the rectangular pulse in Fig. 3.6b is the unit rectangular pulse $\Pi(x)$ expanded by a factor τ and therefore can be expressed as $\Pi(x/\tau)$. Observe that the denominator τ in $\Pi(x/\tau)$ indicates the width of the pulse.

Unit Triangular Function

We use the pictorial notation $\Delta(x)$ for a triangular pulse of unit height and unit width, centered at the origin, as shown in Fig. 3.7a:

$$\Delta(x) = \begin{cases} 1 - 2|t| & |x| < \frac{1}{2} \\ 0 & |x| > \frac{1}{2} \end{cases} \tag{3.17}$$

Observe that the pulse in Fig. 3.7b is $\Delta(x/\tau)$. Observe that here, as for the rectangular pulse, the denominator τ in $\Delta(x/\tau)$ indicates the pulse width.

Sinc Function sinc(x)

The function $\sin x/x$ is the "sine over argument" function denoted by sinc (x).*

* sinc (x) is also denoted by Sa (x) in the literature. Some authors define sinc (x) as

$$\text{sinc}\,(x) = \frac{\sin \pi x}{\pi x}$$

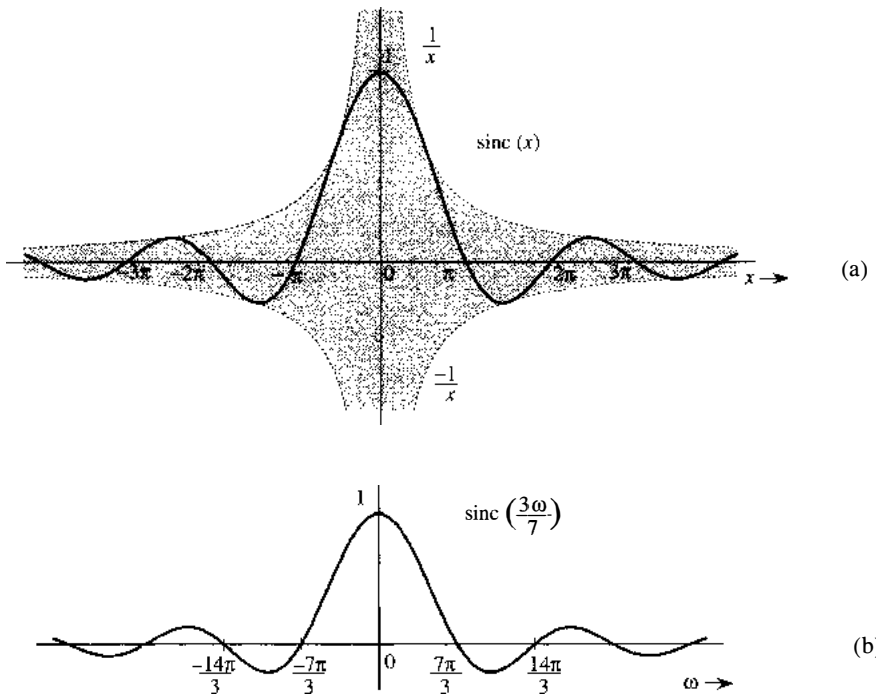


Figure 3.8
Sinc pulse.

This function plays an important role in signal processing. We define

$$\text{sinc}\,(x) = \frac{\sin x}{x} \tag{3.18}$$

Inspection of Eq. (3.18) shows that

1. sinc (x) is an even function of x.
2. sinc $(x) = 0$ when $\sin x = 0$ except at $x = 0$, where it is indeterminate. This means that sinc $(x) = 0$ for $t = \pm\pi, \pm 2\pi, \pm 3\pi, \ldots$.
3. Using L'Hôspital's rule, we find sinc $(0) = 1$.
4. sinc (x) is the product of an oscillating signal $\sin x$ (of period 2π) and a monotonically decreasing function $1/x$. Therefore, sinc (x) exhibits sinusoidal oscillations of period 2π, with amplitude decreasing continuously as $1/x$.
5. In summary, sinc (x) is an even oscillating function with decreasing amplitude. It has a unit peak at $x = 0$ and zero crossings at integer multiples of π.

Figure 3.8a shows sinc (x). Observe that sinc $(x) = 0$ for values of x that are positive and negative integral multiples of π. Figure 3.8b shows sinc $(3\omega/7)$. The argument $3\omega/7 = \pi$ when $\omega = 7\pi/3$ or $f = 7/6$. Therefore, the first zero of this function occurs at $\omega = 7\pi/3$ $(f = 7/6)$.

Example 3.2 Find the Fourier transform of $g(t) = \Pi(t/\tau)$ (Fig. 3.9a).

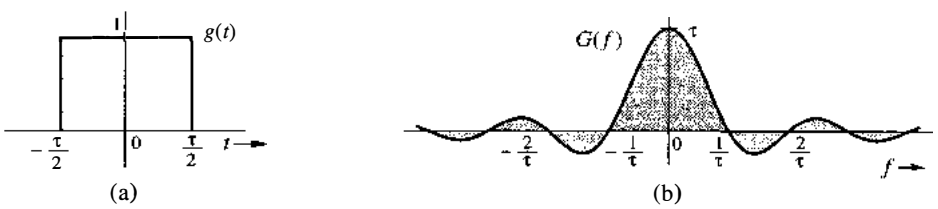


Figure 3.9 Rectangular pulse and its Fourier spectrum.

We have

$$G(f) = \int_{-\infty}^{\infty} \Pi\left(\frac{t}{\tau}\right) e^{-j2\pi ft} dt$$

Since $\Pi(t/\tau) = 1$ for $|t| < \tau/2$, and since it is zero for $|t| > \tau/2$,

$$\begin{aligned} G(f) &= \int_{-\tau/2}^{\tau/2} e^{-j2\pi ft}\, dt \\ &= -\frac{1}{j2\pi f}(e^{-j\pi f\tau} - e^{j\pi f\tau}) = \frac{2\sin(\pi f\tau)}{2\pi f} \\ &= \tau\frac{\sin(\pi f\tau)}{(\pi f\tau)} = \tau \text{ sinc } (\pi f\tau) \end{aligned}$$

Therefore,

$$\Pi\left(\frac{t}{\tau}\right) \Longleftrightarrow \tau \text{ sinc } \left(\frac{\omega\tau}{2}\right) = \tau \text{ sinc } (\pi f\tau) \tag{3.19}$$

Recall that $\text{sinc}(x) = 0$ when $x = \pm n\pi$. Hence, $\text{sinc}(\omega\tau/2) = 0$ when $\omega\tau/2 = \pm n\pi$; that is, when $f = \pm n/\tau$ $(n = 1, 2, 3, \ldots)$, as shown in Fig. 3.9b. Observe that in this case $G(f)$ happens to be real. Hence, we may convey the spectral information by a single plot of $G(f)$ shown in Fig. 3.9b.

Example 3.3 Find the Fourier transform of the unit impulse signal $\delta(t)$.

We use the sampling property of the impulse function [Eq. (2.11)] to obtain

$$\mathcal{F}[\delta(t)] = \int_{-\infty}^{\infty} \delta(t) e^{-j2\pi ft}\, dt = e^{-j2\pi f\cdot 0} = 1 \tag{3.20a}$$

or

$$\delta(t) \Longleftrightarrow 1 \tag{3.20b}$$

Figure 3.10 shows $\delta(t)$ and its spectrum.

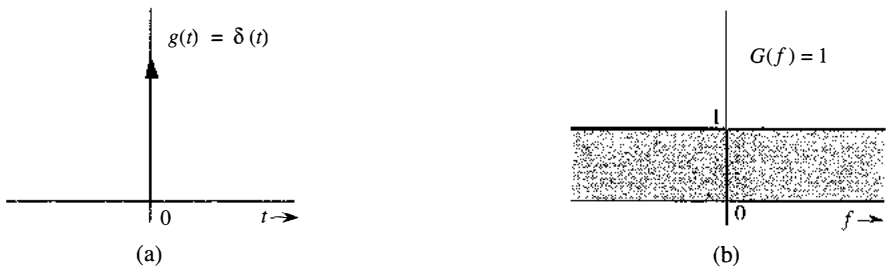


Figure 3.10 Unit impulse and its Fourier spectrum.

Example 3.4 Find the inverse Fourier transform of $\delta(2\pi f) = \frac{1}{2\pi}\delta(f)$.

From Eq. (3.9b) and the sampling property of the impulse function,

$$\mathcal{F}^{-1}[\delta(2\pi f)] = \int_{-\infty}^{\infty} \delta(2\pi f) e^{j2\pi ft}\, df = \frac{1}{2\pi}\int_{-\infty}^{\infty} \delta(2\pi f) e^{j2\pi ft}\, d(2\pi f)$$

$$= \frac{1}{2\pi}\cdot e^{-j2\pi f\cdot 0} = \frac{1}{2\pi}$$

Therefore,

$$\frac{1}{2\pi} \Longleftrightarrow \delta(2\pi f) \tag{3.21a}$$

or

$$1 \Longleftrightarrow \delta(f) \tag{3.21b}$$

This shows that the spectrum of a constant signal $g(t) = 1$ is an impulse $\delta(f) = 2\pi\delta(2\pi f)$, as shown in Fig. 3.11.

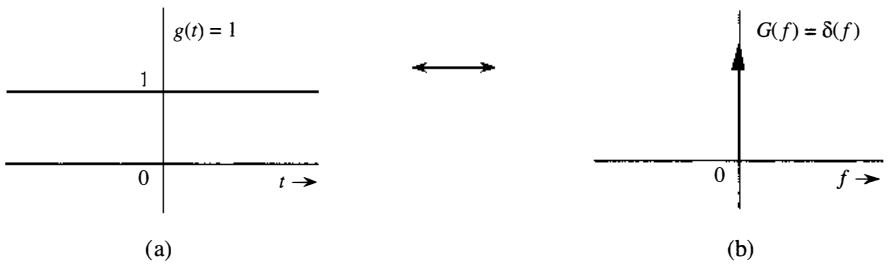


Figure 3.11 Constant (dc) signal and its Fourier spectrum.

The result [Eq. (3.21b)] also could have been anticipated on qualitative grounds. Recall that the Fourier transform of $g(t)$ is a spectral representation of $g(t)$ in terms of everlasting exponential components of the form $e^{j2\pi ft}$. Now to represent a constant signal $g(t) = 1$, we need a single everlasting exponential $e^{j2\pi ft}$ with $f = 0$. This results in a spectrum at a single frequency $f = 0$. We could also say that $g(t) = 1$ is a dc signal that has a single frequency component at $f = 0$ (dc).

If an impulse at $f = 0$ is a spectrum of a dc signal, what does an impulse at $f = f_0$ represent? We shall answer this question in the next example.

Example 3.5 Find the inverse Fourier transform of $\delta(f - f_0)$.

We the sampling property of the impulse function to obtain

$$\mathcal{F}^{-1}[\delta(f - f_0)] = \int_{-\infty}^{\infty} \delta(f - f_0) e^{j2\pi ft}\, df = e^{j2\pi f_0 t}$$

Therefore,

$$e^{j2\pi f_0 t} \Longleftrightarrow \delta(f - f_0) \tag{3.22a}$$

This result shows that the spectrum of an everlasting exponential $e^{j2\pi f_0 t}$ is a single impulse at $f = f_0$. We reach the same conclusion by qualitative reasoning. To represent the everlasting exponential $e^{j2\pi f_0 t}$, we need a single everlasting exponential $e^{j2\pi ft}$ with $\omega = 2\pi f_0$. Therefore, the spectrum consists of a single component at frequency $f = f_0$.
From Eq. (3.22a) it follows that

$$e^{-j2\pi f_0 t} \Longleftrightarrow \delta(f + f_0) \tag{3.22b}$$

Example 3.6 Find the Fourier transforms of the everlasting sinusoid $\cos 2\pi f_0$.

Recall the Euler formula

$$\cos 2\pi f_0 t = \frac{1}{2}(e^{j2\pi f_0 t} + e^{-j2\pi f_0 t})$$

Adding Eqs. (3.22a) and (3.22b), and using the preceding formula, we obtain

$$\cos 2\pi f_0 t \Longleftrightarrow \frac{1}{2}[\delta(f + f_0) + \delta(f - f_0)] \tag{3.23}$$

The spectrum of $\cos 2\pi f_0 t$ consists of two impulses at f_0 and $-f_0$ in the f-domain, or, two impulses at $\pm\omega_0 = \pm 2\pi f_0$ in the ω-domain as shown in Fig. 3.12. The result also follows from qualitative reasoning. An everlasting sinusoid $\cos \omega_0 t$ can be synthesized by two everlasting exponentials, $e^{j\omega_0 t}$ and $e^{-j\omega_0 t}$. Therefore, the Fourier spectrum consists of only two components of frequencies ω_0 and $-\omega_0$.

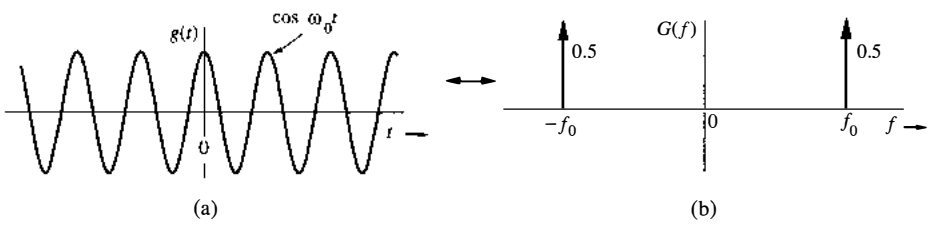


Figure 3.12 Cosine signal and its Fourier spectrum.

Example 3.7 Find the Fourier transform of the sign function sgn (t) (pronounced signum t), shown in Fig. 3.13. Its value is $+1$ or -1, depending on whether t is positive or negative:

$$\text{sgn}(t) = \begin{cases} 1 & t > 0 \\ 0 & t = 0 \\ -1 & t < 0 \end{cases} \tag{3.24}$$

We cannot use integration to find the transform of sgn (t) directly. This is because sgn (t) violates the Dirichlet condition [see E.g. (3.14) and the associated footnote]. Specifically, sgn (t) is not absolutely integrable. However, the transform can be obtained by considering sgn t as a sum of two exponentials, as shown in Fig. 3.13, in the limit as $a \to 0$:

$$\text{sgn}\, t = \lim_{a \to 0} \left[e^{-at} u(t) - e^{at} u(-t) \right]$$

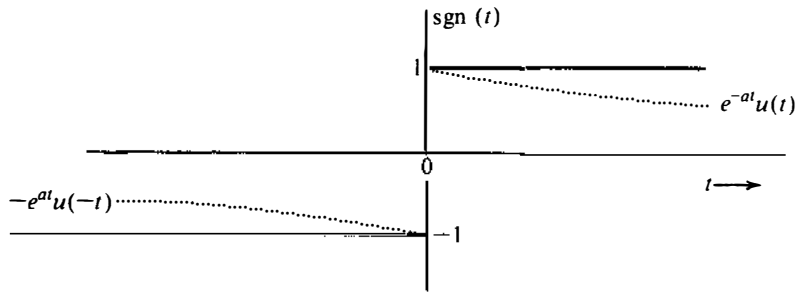


Figure 3.13 Sign function.

Therefore,

$$\begin{aligned} \mathcal{F}[\text{sgn}(t)] &= \lim_{a \to 0} \left\{ \mathcal{F}[e^{-at} u(t)] - \mathcal{F}[e^{at} u(-t)] \right\} \\ &= \lim_{a \to 0} \left(\frac{1}{a + j2\pi f} - \frac{1}{a - j2\pi f} \right) \qquad \text{(see pairs 1 and 2 in Table 3.1)} \\ &= \lim_{a \to 0} \left(\frac{-j4\pi f}{a^2 + 4\pi^2 f^2} \right) = \frac{1}{j\pi f} \end{aligned} \tag{3.25}$$

3.3 SOME PROPERTIES OF THE FOURIER TRANSFORM

We now study some of the important properties of the Fourier transform and their implications as well as their applications. Before embarking on this study, it is important to point out a pervasive aspect of the Fourier transform—the **time-frequency duality**.

TABLE 3.1
Short Table of Fourier Transforms

	$g(t)$	$G(f)$	
1	$e^{-at}u(t)$	$\frac{1}{a+j2\pi f}$	$a>0$
2	$e^{at}u(-t)$	$\frac{1}{a-j2\pi f}$	$a>0$
3	$e^{-a\|t\|}$	$\frac{2a}{a^2+(2\pi f)^2}$	$a>0$
4	$te^{-at}u(t)$	$\frac{1}{(a+j2\pi f)^2}$	$a>0$
5	$t^n e^{-at}u(t)$	$\frac{n!}{(a+j2\pi f)^{n+1}}$	$a>0$
6	$\delta(t)$	1	
7	1	$\delta(f)$	
8	$e^{j2\pi f_0 t}$	$\delta(f-f_0)$	
9	$\cos 2\pi f_0 t$	$0.5[\delta(f+f_0)+\delta(f-f_0)]$	
10	$\sin 2\pi f_0 t$	$j0.5[\delta(f+f_0)-\delta(f-f_0)]$	
11	$u(t)$	$\frac{1}{2}\delta(f)+\frac{1}{j2\pi f}$	
12	$\operatorname{sgn} t$	$\frac{2}{j2\pi f}$	
13	$\cos 2\pi f_0 t\, u(t)$	$\frac{1}{4}[\delta(f-f_0)+\delta(f+f_0)]+\frac{j2\pi f}{(2\pi f_0)^2-(2\pi f)^2}$	
14	$\sin 2\pi f_0 t\, u(t)$	$\frac{1}{4j}[\delta(f-f_0)-\delta(f+f_0)]+\frac{2\pi f_0}{(2\pi f_0)^2-(2\pi f)^2}$	
15	$e^{-at}\sin 2\pi f_0 t\, u(t)$	$\frac{2\pi f_0}{(a+j2\pi f)^2+4\pi^2 f_0^2}$	$a>0$
16	$e^{-at}\cos 2\pi f_0 t\, u(t)$	$\frac{a+j2\pi f}{(a+j2\pi f)^2+4\pi^2 f_0^2}$	$a>0$
17	$\Pi\left(\frac{t}{\tau}\right)$	$\tau\ \operatorname{sinc}(\pi f\tau)$	
18	$2B\operatorname{sinc}(2\pi Bt)$	$\Pi\left(\frac{f}{2B}\right)$	
19	$\Delta\left(\frac{t}{\tau}\right)$	$\frac{\tau}{2}\operatorname{sinc}^2\left(\frac{\pi f\tau}{2}\right)$	
20	$B\operatorname{sinc}^2(\pi Bt)$	$\Delta\left(\frac{f}{2B}\right)$	
21	$\sum_{n=-\infty}^{\infty}\delta(t-nT)$	$f_0\sum_{n=-\infty}^{\infty}\delta(f-nf_0)$	$f_0=\frac{1}{T}$
22	$e^{-t^2/2\sigma^2}$	$\sigma\sqrt{2\pi}e^{-2(\sigma\pi f)^2}$	

3.3.1 Time-Frequency Duality

Equations (3.9) show an interesting fact: the direct and the inverse transform operations are remarkably similar. These operations, required to go from $g(t)$ to $G(f)$ and then from $G(f)$ to $g(t)$, are shown graphically in Fig. 3.14. The only minor difference between these two operations lies in the opposite signs used in their exponential indices.

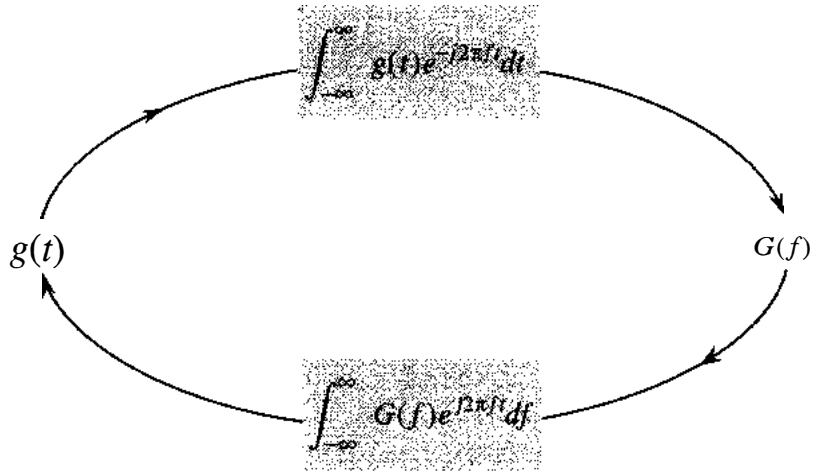


Figure 3.14 Near symmetry between direct and inverse Fourier transforms.

This similarity has far-reaching consequences in the study of Fourier transforms. It is the basis of the so-called duality of time and frequency. *The duality principle may be compared with a photograph and its negative. A photograph can be obtained from its negative, and by using an identical procedure, the negative can be obtained from the photograph.* For any result or relationship between $g(t)$ and $G(f)$, there exists a dual result or relationship, obtained by interchanging the roles of $g(t)$ and $G(f)$ in the original result (along with some minor modifications arising because of the factor 2π and a sign change). For example, the time-shifting property, to be proved later, states that if $g(t) \Longleftrightarrow G(f)$, then

$$g(t - t_0) \Longleftrightarrow G(f)e^{-j2\pi ft_0}$$

The dual of this property (the frequency-shifting property) states that

$$g(t)e^{j2\pi f_0 t} \Longleftrightarrow G(f - f_0)$$

Observe the role reversal of time and frequency in these two equations (with the minor difference of the sign change in the exponential index). The value of this principle lies in the fact that *whenever we derive any result, we can be sure that it has a dual.* This knowledge can give valuable insights about many unsuspected properties or results in signal processing.

The properties of the Fourier transform are useful not only in deriving the direct and the inverse transforms of many functions, but also in obtaining several valuable results in signal processing. The reader should not fail to observe the ever-present duality in this discussion. We begin with the duality property, which is one of the consequences of the duality principle.

3.3.2 Duality Property

The duality property states that if

$$g(t) \Longleftrightarrow G(f)$$

then

$$G(t) \Longleftrightarrow g(-f) \tag{3.26}$$

The duality property states that if the Fourier transform of $g(t)$ is $G(f)$ then the Fourier transform of $G(t)$, with f replaced by t, is the $g(-f)$ which is the original time domain signal with t replaced by $-f$.

Proof: From Eq. (3.9b),

$$g(t) = \int_{-\infty}^{\infty} G(x)e^{j2\pi xt}\,dx$$

Hence,

$$g(-t) = \int_{-\infty}^{\infty} G(x)e^{-j2\pi xt}\,dx$$

Changing t to f yields Eq. (3.26). ■

Example 3.8 In this example we shall apply the duality property [Eq. (3.26)] to the pair in Fig. 3.15a.

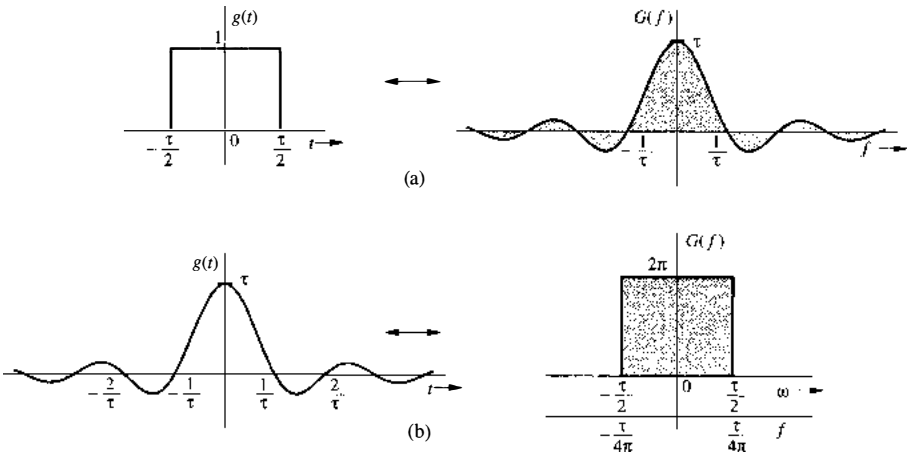


Figure 3.15 Duality property of the Fourier transform.

From Eq. (3.19) we have

$$\Pi\left(\frac{t}{\tau}\right) \Longleftrightarrow \tau\ \text{sinc}\,(\pi f\tau) \tag{3.27a}$$

$$\underbrace{\Pi\left(\frac{t}{\alpha}\right)}_{g(t)} \Longleftrightarrow \underbrace{\alpha\ \text{sinc}\,(\pi f\alpha)}_{G(f)} \tag{3.27b}$$

Also $G(t)$ is the same as $G(f)$ with f replaced by t, and $g(-f)$ is the same as $g(t)$ with t replaced by $-f$. Therefore, the duality property (3.26) yields

$$\underbrace{\alpha\ \text{sinc}\,(\pi\alpha t)}_{G(t)} \Longleftrightarrow \underbrace{\Pi\left(-\frac{f}{\alpha}\right)}_{g(-f)} = \Pi\left(\frac{f}{\alpha}\right) \tag{3.28a}$$

Substituting $\tau = 2\pi\alpha$, we obtain

$$\tau \text{ sinc}\left(\frac{\alpha t}{2}\right) \Longleftrightarrow 2\pi \Pi\left(\frac{2\pi f}{\tau}\right) \tag{3.28b}$$

In Eq. (3.8) we used the fact that $\Pi(-t) = \Pi(t)$ because $\Pi(t)$ is an even function. Figure 3.15b shows this pair graphically. Observe the interchange of the roles of t and $2\pi f$ (with the minor adjustment of the factor 2π). This result appears as pair 18 in Table 3.1 (with $\tau/2 = W$).

As an interesting exercise, generate a dual of every pair in Table 3.1 by applying the duality property.

3.3.3 Time-Scaling Property

If

$$g(t) \Longleftrightarrow G(f)$$

then, for any real constant a,

$$g(at) \Longleftrightarrow \frac{1}{|a|} G\left(\frac{f}{a}\right) \tag{3.29}$$

Proof: For a positive real constant a,

$$\mathcal{F}[g(at)] = \int_{-\infty}^{\infty} g(at) e^{-j2\pi ft}\, dt = \frac{1}{a}\int_{-\infty}^{\infty} g(x) e^{(-j2\pi f/a)x}\, dx = \frac{1}{a} G\left(\frac{f}{a}\right)$$

Similarly, it can be shown that if $a < 0$,

$$g(at) \Longleftrightarrow \frac{-1}{a} G\left(\frac{f}{a}\right)$$

Hence follows Eq. (3.29). ■

Significance of the Time-Scaling Property

The function $g(at)$ represents the function $g(t)$ compressed in time by a factor a ($|a| > 1$). Similarly, a function $G(f/a)$ represents the function $G(f)$ expanded in frequency by the same factor a. *The time-scaling property states that time compression of a signal results in its spectral expansion, and time expansion of the signal results in its spectral compression.* Intuitively, compression in time by a factor a means that the signal is varying **more rapidly** by the same factor. To synthesize such a signal, the frequencies of its sinusoidal components must be increased by the factor a, implying that its frequency spectrum is expanded by the factor a. Similarly, a signal expanded in time varies more slowly; hence, the frequencies of its components are lowered, implying that its frequency spectrum is compressed. For instance, the signal $\cos 4\pi f_0 t$ is the same as the signal $\cos 2\pi f_0 t$ time-compressed by a factor of 2. Clearly, the spectrum of the former (impulse at $\pm 2f_0$) is an expanded version of the spectrum of the latter (impulse at $\pm f_0$). The effect of this scaling is demonstrated in Fig. 3.16.

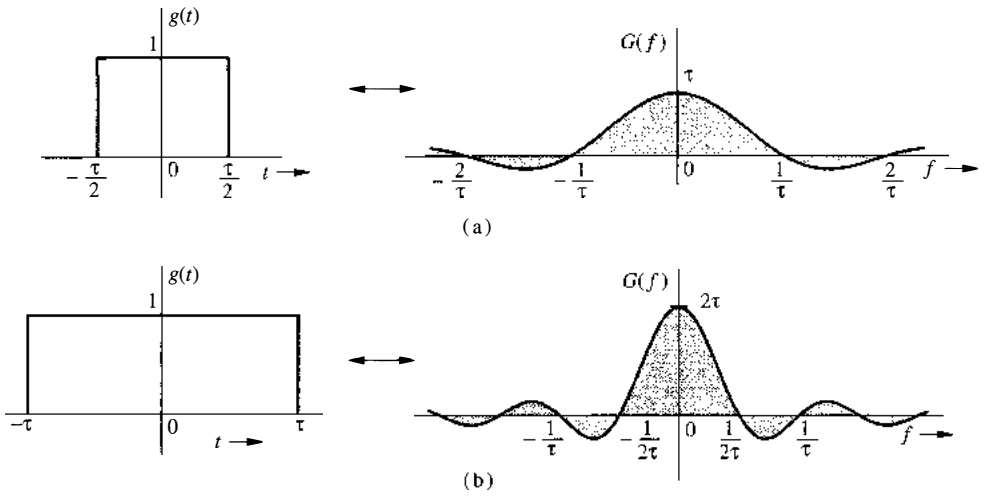


Figure 3.16 Scaling property of the Fourier transform.

Reciprocity of Signal Duration and Its Bandwidth

The time-scaling property implies that if $g(t)$ is wider, its spectrum is narrower, and vice versa. Doubling the signal duration halves its bandwidth, and vice versa. This suggests that the bandwidth of a signal is inversely proportional to the signal duration or width (in seconds). We have already verified this fact for the rectangular pulse, where we found that the bandwidth of a gate pulse of width τ seconds is $1/\tau$ Hz. More discussion of this interesting topic can be found in the literature.[2]

Example 3.9 Show that

$$g(-t) \Longleftrightarrow G(-f) \tag{3.30}$$

Use this result and the fact that $e^{-at}u(t) \Longleftrightarrow 1/(a+j2\pi f)$, to find the Fourier transforms of $e^{at}u(-t)$ and $e^{-a|t|}$.

Equation (3.30) follows from Eq. (3.29) by letting $a = -1$. Application of Eq. (3.30) to pair 1 of Table 3.1 yields

$$e^{at}u(-t) \Longleftrightarrow \frac{1}{a-j2\pi f}$$

Also

$$e^{-a|t|} = e^{-at}u(t) + e^{at}u(-t)$$

Therefore,

$$e^{-a|t|} \Longleftrightarrow \frac{1}{a+j2\pi f} + \frac{1}{a-j2\pi f} = \frac{2a}{a^2+(2\pi f)^2} \tag{3.31}$$

The signal $e^{-a|t|}$ and its spectrum are shown in Fig. 3.17.

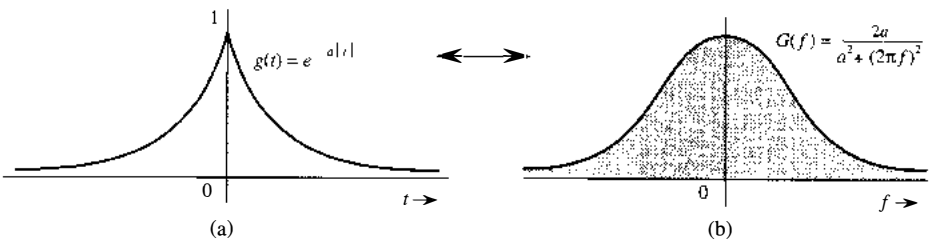


Figure 3.17 $e^{-a|t|}$ and its Fourier spectrum.

3.3.4 Time-Shifting Property

If

$$g(t) \Longleftrightarrow G(f)$$

then

$$g(t - t_0) \Longleftrightarrow G(f)e^{-j2\pi ft_0} \tag{3.32a}$$

Proof: By definition,

$$\mathcal{F}[g(t - t_0)] = \int_{-\infty}^{\infty} g(t - t_0)e^{-j2\pi ft}\, dt$$

Letting $t - t_0 = x$, we have

$$\begin{aligned}\mathcal{F}[g(t - t_0)] &= \int_{-\infty}^{\infty} g(x)e^{-j2\pi f(x+t_0)}\, dx \\ &= e^{-j2\pi ft_0} \int_{-\infty}^{\infty} g(x)e^{-j2\pi fx}\, dx = G(f)e^{-j2\pi ft_0}\end{aligned} \tag{3.32b}$$

This result shows that *delaying a signal by t_0 seconds does not change its amplitude spectrum. The phase spectrum, however, is changed by $-2\pi ft_0$.*

Physical Explanation of the Linear Phase

Time delay in a signal causes a linear phase shift in its spectrum. This result can also be derived by heuristic reasoning. Imagine $g(t)$ being synthesized by its Fourier components, which are sinusoids of certain amplitudes and phases. The delayed signal $g(t - t_0)$ can be synthesized by the same sinusoidal components, each delayed by t_0 seconds. The amplitudes of the components remain unchanged. Therefore, the amplitude spectrum of $g(t - t_0)$ is identical to that of $g(t)$. The time delay of t_0 in each sinusoid, however, does change the phase of each component. Now, a sinusoid $\cos 2\pi ft$ delayed by t_0 is given by

$$\cos 2\pi f(t - t_0) = \cos(2\pi ft - 2\pi ft_0)$$

Therefore, a time delay t_0 in a sinusoid of frequency f manifests as a phase delay of $2\pi ft_0$. This is a linear function of f, meaning that higher frequency components must undergo proportionately

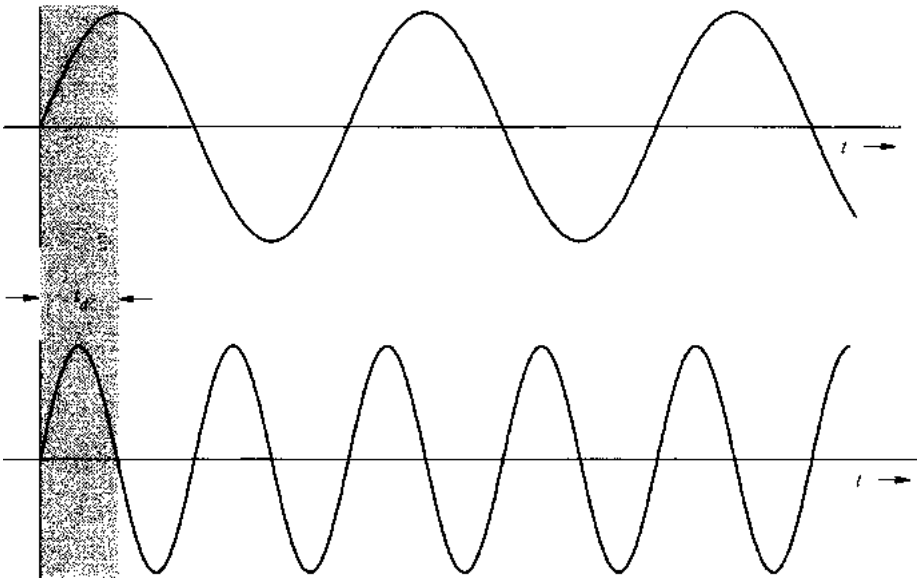


Figure 3.18 Physical explanation of the time-shifting property.

higher phase shifts to achieve the same time delay. This effect is shown in Fig. 3.18 with two sinusoids, the frequency of the lower sinusoid being twice that of the upper. The same time delay t_0 amounts to a phase shift of $\pi/2$ in the upper sinusoid and a phase shift of π in the lower sinusoid. This verifies that *to achieve the same time delay, higher frequency sinusoids must undergo proportionately higher phase shifts.*

Example 3.10 Find the Fourier transform of $e^{-a|t-t_0|}$.

This function, shown in Fig. 3.19a, is a time-shifted version of $e^{-a|t|}$ (shown in Fig. 3.17a). From Eqs. (3.31) and (3.32) we have

$$e^{-a|t-t_0|} \Longleftrightarrow \frac{2a}{a^2+(2\pi f)^2} e^{-j2\pi f t_0} \tag{3.33}$$

The spectrum of $e^{-a|t-t_0|}$ (Fig. 3.19b) is the same as that of $e^{-a|t|}$ (Fig. 3.17b), except for an added phase shift of $-2\pi f t_0$.

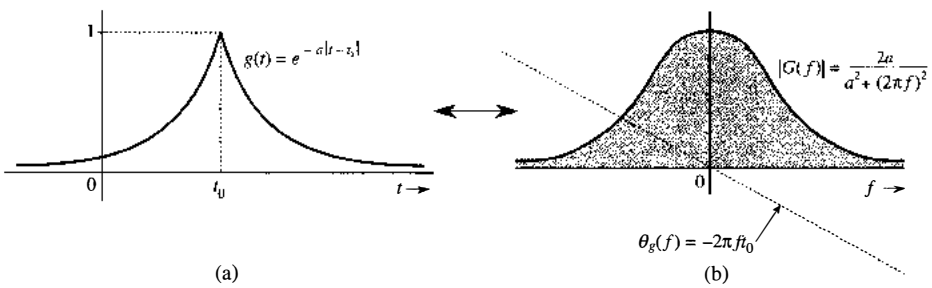


Figure 3.19 Effect of time shifting on the Fourier spectrum of a signal.

Observe that the time delay t_0 causes a **linear** phase spectrum $-2\pi f t_0$. This example clearly demonstrates the effect of time shift.

3.3.5 Frequency-Shifting Property

If

$$g(t) \Longleftrightarrow G(f)$$

then

$$g(t)e^{j2\pi f_0 t} \Longleftrightarrow G(f - f_0) \tag{3.34}$$

This property is also called the modulation property.

Proof: By definition,

$$\mathcal{F}[g(t)e^{j2\pi f_0 t}] = \int_{-\infty}^{\infty} g(t)e^{j2\pi f_0 t}e^{-j2\pi f t}\,dt = \int_{-\infty}^{\infty} g(t)e^{-j(2\pi f - 2\pi f_0)t}\,dt = G(f - f_0)$$

■

This property states that multiplication of a signal by a factor $e^{j2\pi f_0 t}$ shifts the spectrum of that signal by $f = f_0$. Note the duality between the time-shifting and the frequency-shifting properties.

Changing f_0 to $-f_0$ in Eq. (3.34) yields

$$g(t)e^{-j2\pi f_0 t} \Longleftrightarrow G(f + f_0) \tag{3.35}$$

Because $e^{j2\pi f_0 t}$ is not a real function that can be generated, frequency shifting in practice is achieved by multiplying $g(t)$ by a sinusoid. This can be seen from

$$g(t)\cos 2\pi f_0 t = \frac{1}{2}\left[g(t)e^{j2\pi f_0 t} + g(t)e^{-j2\pi f_0 t}\right]$$

From Eqs. (3.34) and (3.35), it follows that

$$g(t)\cos 2\pi f_0 t \Longleftrightarrow \frac{1}{2}[G(f - f_0) + G(f + f_0)] \tag{3.36}$$

This shows that the multiplication of a signal $g(t)$ by a sinusoid of frequency f_0 shifts the spectrum $G(f)$ by $\pm f_0$. Multiplication of a sinusoid $\cos 2\pi f_0 t$ by $g(t)$ amounts to modulating the sinusoid amplitude. This type of modulation is known as **amplitude modulation**. The sinusoid $\cos 2\pi f_0 t$ is called the **carrier**, the signal $g(t)$ is the **modulating signal**, and the signal $g(t)\cos 2\pi f_0 t$ is the **modulated signal**. Modulation and demodulation will be discussed in detail in Chapters 4 and 5.

To sketch a signal $g(t)\cos 2\pi f_0 t$, we observe that

$$g(t)\cos 2\pi f_0 t = \begin{cases} g(t) & \text{when } \cos 2\pi f_0 t = 1 \\ -g(t) & \text{when } \cos 2\pi f_0 t = -1 \end{cases}$$

Therefore, $g(t)\cos 2\pi f_0 t$ touches $g(t)$ when the sinusoid $\cos 2\pi f_0 t$ is at its positive peaks and touches $-g(t)$ when $\cos 2\pi f_0 t$ is at its negative peaks. This means that $g(t)$ and $-g(t)$ act as envelopes for the signal $g(t)\cos 2\pi f_0 t$ (see Fig. 3.20c). The signal $-g(t)$ is a mirror image of $g(t)$ about the horizontal axis. Figure 3.20 shows the signals $g(t)$, $g(t)\cos 2\pi f_0 t$, and their respective spectra.

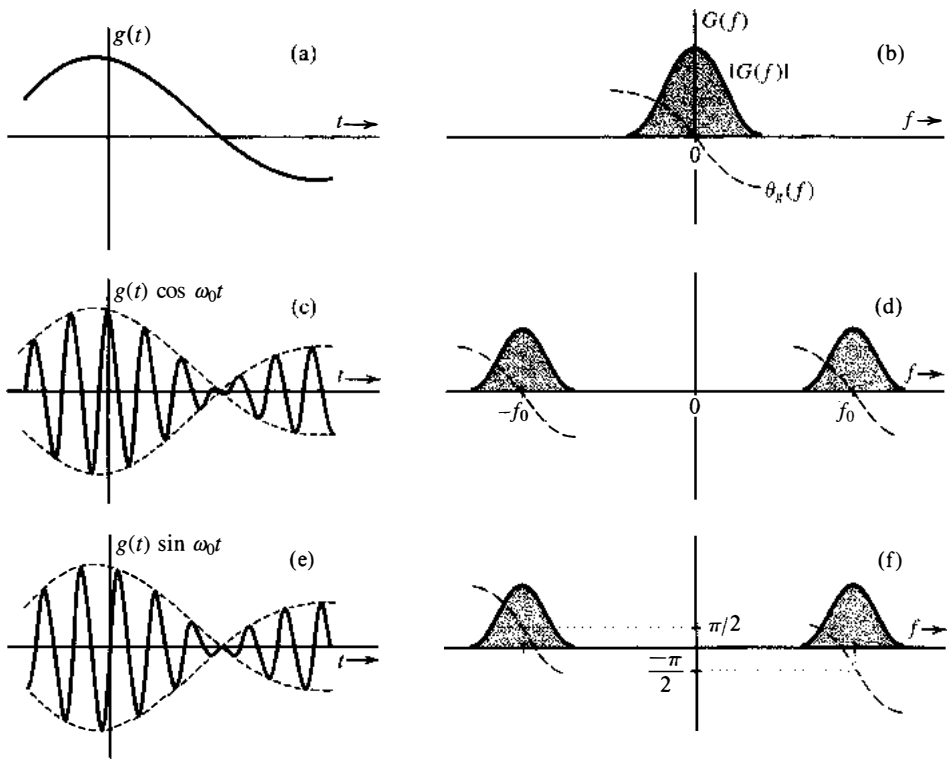


Figure 3.20 Amplitude modulation of a signal causes spectral shifting.

Shifting the Phase Spectrum of a Modulated Signal

We can shift the phase of each spectral component of a modulated signal by a constant amount θ_0 merely by using a carrier $\cos(2\pi f_0 t + \theta_0)$ instead of $\cos 2\pi f_0 t$. If a signal $g(t)$ is multiplied by $\cos(2\pi f_0 t + \theta_0)$, then we can use an argument similar to that used to derive Eq. (3.36), to show that

$$g(t)\cos(2\pi f_0 t + \theta_0) \Longleftrightarrow \frac{1}{2}\left[G(f - f_0)\, e^{j\theta_0} + G(f + f_0)\, e^{-j\theta_0}\right] \tag{3.37}$$

For a special case when $\theta_0 = -\pi/2$, Eq. (3.37) becomes

$$g(t)\sin 2\pi f_0 t \Longleftrightarrow \frac{1}{2}\left[G(f - f_0)\, e^{-j\pi/2} + G(f + f_0)\, e^{j\pi/2}\right] \tag{3.38}$$

Observe that $\sin 2\pi f_0 t$ is $\cos 2\pi f_0 t$ with a phase delay of $\pi/2$. Thus, shifting the carrier phase by $\pi/2$ shifts the phase of every spectral component by $\pi/2$. Figures 3.20e and f show the signal $g(t)\sin 2\pi f_0 t$ and its spectrum.

Modulation is a common application that shifts signal spectra. In particular, If several message signals, each occupying the same frequency band, are transmitted simultaneously over a common transmission medium, they will all interfere; it will be impossible to separate or retrieve them at a receiver. For example, if all radio stations decide to broadcast audio signals simultaneously, receivers will not be able to separate them. This problem is solved by using modulation, whereby each radio station is assigned a distinct carrier frequency. Each station transmits a modulated signal, thus shifting the signal spectrum to its allocated band, which is not occupied by any other station. A radio receiver can pick up any station by tuning to the

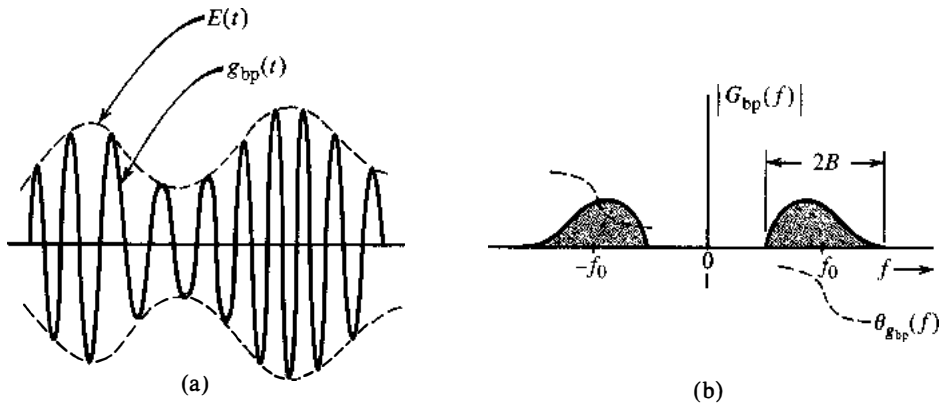


Figure 3.21
Bandpass signal and its spectrum.

band of the desired station. The receiver must now demodulate the received signal (undo the effect of modulation). Demodulation therefore consists of another spectral shift required to restore the signal to its original band.

Bandpass Signals

Figure 3.20(d)(f) shows that if $g_c(t)$ and $g_s(t)$ are low-pass signals, each with a bandwidth B Hz or $2\pi B$ rad/s, then the signals $g_c(t)\cos 2\pi f_0 t$ and $g_s(t)\sin 2\pi f_0 t$ are both bandpass signals occupying the same band, and each having a bandwidth of $2B$ Hz. Hence, a linear combination of both these signals will also be a bandpass signal occupying the same band as that of the either signal, and with the same bandwidth ($2B$ Hz). Hence, a general bandpass signal $g_{bp}(t)$ can be expressed as*

$$g_{bp}(t) = g_c(t)\cos 2\pi f_0 t + g_s(t)\sin 2\pi f_0 t \tag{3.39}$$

The spectrum of $g_{bp}(t)$ is centered at $\pm f_0$ and has a bandwidth $2B$, as shown in Fig. 3.21. Although the magnitude spectra of both $g_c(t)\cos 2\pi f_0 t$ and $g_s(t)\sin 2\pi f_0 t$ are symmetrical about $\pm f_0$, the magnitude spectrum of their sum, $g_{bp}(t)$, is not necessarily symmetrical about $\pm f_0$. This is because the different phases of the two signals do not allow their amplitudes to add directly for the reason that

$$a_1 e^{j\varphi_1} + a_2 e^{j\varphi_2} \neq (a_1 + a_2)e^{j(\varphi_1+\varphi_2)}$$

A typical bandpass signal $g_{bp}(t)$ and its spectra are shown in Fig. 3.21. We can use a well-known trigonometric identity to express Eq. (3.39) as

$$g_{bp}(t) = E(t)\cos[2\pi f_0 t + \psi(t)] \tag{3.40}$$

where

$$E(t) = +\sqrt{g_c^2(t) + g_s^2(t)} \tag{3.41a}$$

$$\psi(t) = -\tan^{-1}\left[\frac{g_s(t)}{g_c(t)}\right] \tag{3.41b}$$

* See Sec. 9.9 for a rigorous proof of this statement.

Because $g_c(t)$ and $g_s(t)$ are low-pass signals, $E(t)$ and $\psi(t)$ are also low-pass signals. Because $E(t)$ is nonnegative [Eq. (3.41a)], it follows from Eq. (3.40) that $E(t)$ is a slowly varying envelope and $\psi(t)$ is a slowly varying phase of the bandpass signal $g_{bp}(t)$, as shown in Fig. 3.21. Thus, the bandpass signal $g_{bp}(t)$ will appear as a sinusoid of slowly varying amplitude. Because of the time-varying phase $\psi(t)$, the frequency of the sinusoid also varies slowly* with time about the center frequency f_0.

Example 3.11 Find the Fourier transform of a general periodic signal $g(t)$ of period T_0, and hence, determine the Fourier transform of the periodic impulse train $\delta_{T_0}(t)$ shown in Fig. 3.22a.

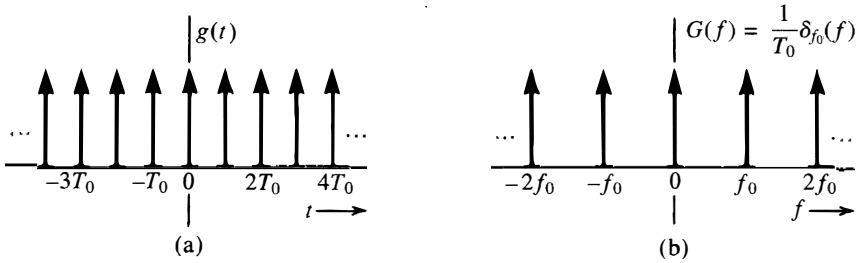


Figure 3.22 Impulse train and its spectrum.

A periodic signal $g(t)$ can be expressed as an exponential Fourier series as

$$g(t) = \sum_{n=-\infty}^{\infty} D_n e^{jn2\pi f_0 t} \qquad f_0 = \frac{1}{T_0}$$

Therefore,

$$g(t) \Longleftrightarrow \sum_{n=-\infty}^{\infty} \mathcal{F}[D_n e^{jn2\pi f_0 t}]$$

Now from Eq. (3.22a), it follows that

$$g(t) \Longleftrightarrow \sum_{n=-\infty}^{\infty} D_n \delta(f - nf_0) \tag{3.42}$$

Equation (2.67) shows that the impulse train $\delta_{T_0}(t)$ can be expressed as an exponential Fourier series as

$$\delta_{T_0}(t) = \frac{1}{T_0} \sum_{n=-\infty}^{\infty} e^{jn2\pi f_0 t} \qquad f_0 = \frac{1}{T_0}$$

---

* It is necessary that $B \ll f_0$ for a well-defined envelope. Otherwise the variations of $E(t)$ are of the same order as the carrier, and it will be difficult to separate the envelope from the carrier.

Here $D_n = 1/T_0$. Therefore, from Eq. (3.42),

$$\delta_{T_0}(t) \Longleftrightarrow \frac{1}{T_0} \sum_{n=-\infty}^{\infty} \delta(f - nf_0)$$

$$= \frac{1}{T_0} \delta_{f_0}(f) \qquad f_0 = \frac{1}{T_0} \tag{3.43}$$

Thus, the spectrum of the impulse train also happens to be an impulse train (in the frequency domain), as shown in Fig. 3.23b.

3.3.6 Convolution Theorem

The convolution of two functions $g(t)$ and $w(t)$, denoted by $g(t) * w(t)$, is defined by the integral

$$g(t) * w(t) = \int_{-\infty}^{\infty} g(\tau) w(t - \tau)\, d\tau$$

The time convolution property and its dual, the frequency convolution property, state that if

$$g_1(t) \Longleftrightarrow G_1(f) \qquad \text{and} \qquad g_2(t) \Longleftrightarrow G_2(f)$$

then (**time convolution**)

$$g_1(t) * g_2(t) \Longleftrightarrow G_1(f) G_2(f) \tag{3.44}$$

and (**frequency convolution**)

$$g_1(t) g_2(t) \Longleftrightarrow G_1(f) * G_2(f) \tag{3.45}$$

These two relationships of the convolution theorem state that convolution of two signals in the time domain becomes multiplication in the frequency domain, while multiplication of two signals in the time domain becomes convolution in the frequency domain.

Proof: By definition,

$$\mathcal{F}|g_1(t) * g_2(t)| = \int_{-\infty}^{\infty} e^{-j2\pi ft} \left[\int_{-\infty}^{\infty} g_1(\tau) g_2(t - \tau) d\tau \right] dt$$

$$= \int_{-\infty}^{\infty} g_1(\tau) \left[\int_{-\infty}^{\infty} e^{-j2\pi ft} g_2(t - \tau) dt \right] d\tau$$

The inner integral is the Fourier transform of $g_2(t - \tau)$, given by [time-shifting property in Eq. (3.32a)] $G_2(f) e^{-j2\pi f\tau}$. Hence,

$$\mathcal{F}[g_1(t) * g_2(t)] = \int_{-\infty}^{\infty} g_1(\tau) e^{-j2\pi f\tau} G_2(f)\, d\tau$$

$$= G_2(f) \int_{-\infty}^{\infty} g_1(\tau) e^{-j2\pi f\tau}\, d\tau = G_1(f) G_2(f) \qquad \blacksquare$$

The frequency convolution property (3.45) can be proved in exactly the same way by reversing the roles of $g(t)$ and $G(f)$.

Bandwidth of the Product of Two Signals

If $g_1(t)$ and $g_2(t)$ have bandwidths B_1 and B_2 Hz, respectively, the bandwidth of $g_1(t)g_2(t)$ is $B_1 + B_2$ Hz. This result follows from the application of the width property of convolution[3] to Eq. (3.45). This property states that the width of $x * y$ is the sum of the widths of x and y. Consequently, if the bandwidth of $g(t)$ is B Hz, then the bandwidth of $g^2(t)$ is $2B$ Hz, and the bandwidth of $g^n(t)$ is nB Hz.*

Example 3.12 Using the time convolution property, show that if

$$g(t) \Longleftrightarrow G(f)$$

then

$$\int_{-\infty}^{t} g(\tau)d\tau \Longleftrightarrow \frac{G(f)}{j2\pi f} + \frac{1}{2}G(0)\delta(f) \tag{3.46}$$

Because

$$u(t-\tau) = \begin{cases} 1 & \tau \le t \\ 0 & \tau > t \end{cases}$$

it follows that

$$g(t) * u(t) = \int_{-\infty}^{\infty} g(\tau)u(t-\tau)\,d\tau = \int_{-\infty}^{t} g(\tau)\,d\tau$$

Now from the time convolution property [Eq. (3.44)], it follows that

$$\begin{aligned} g(t) * u(t) &\Longleftrightarrow G(f)U(f) \\ &= G(f)\left[\frac{1}{j2\pi f} + \frac{1}{2}\delta(f)\right] \\ &= \frac{G(f)}{j2\pi f} + \frac{1}{2}G(0)\delta(f) \end{aligned}$$

In deriving the last result we used pair 11 of Table 3.1 and Eq. (2.10a).

3.3.7 Time Differentiation and Time Integration

If

$$g(t) \Longleftrightarrow G(f),$$

* The width property of convolution does not hold in some pathological cases. It fails when the convolution of two functions is zero over a range even when both functions are nonzero [e.g., $\sin 2\pi f_0 t\, u(t) * u(t)$]. Technically the property holds even in this case if in calculating the width of the convolved function, we take into account the range in which the convolution is zero.

then **(time differentiation)***

$$\frac{dg(t)}{dt} \Longleftrightarrow j2\pi fG(f) \tag{3.47}$$

and **(time integration)**

$$\int_{-\infty}^{t} g(\tau)d\tau \Longleftrightarrow \frac{G(f)}{j2\pi f} + \frac{1}{2}G(0)\delta(f) \tag{3.48}$$

Proof: Differentiation of both sides of Eq. (3.9b) yields

$$\frac{dg(t)}{dt} = \int_{-\infty}^{\infty} j2\pi fG(f)e^{j2\pi ft}\,df$$

This shows that

$$\frac{dg(t)}{dt} \Longleftrightarrow j2\pi fG(f)$$

Repeated application of this property yields

$$\frac{d^n g(t)}{dt^n} \Longleftrightarrow (j2\pi f)^n G(f) \tag{3.49}$$

The time integration property [Eq. (3.48)] already has been proved in Example 3.12. ■

Example 3.13 Use the time differentiation property to find the Fourier transform of the triangular pulse $\Delta(t/\tau)$ shown in Fig. 3.23a.

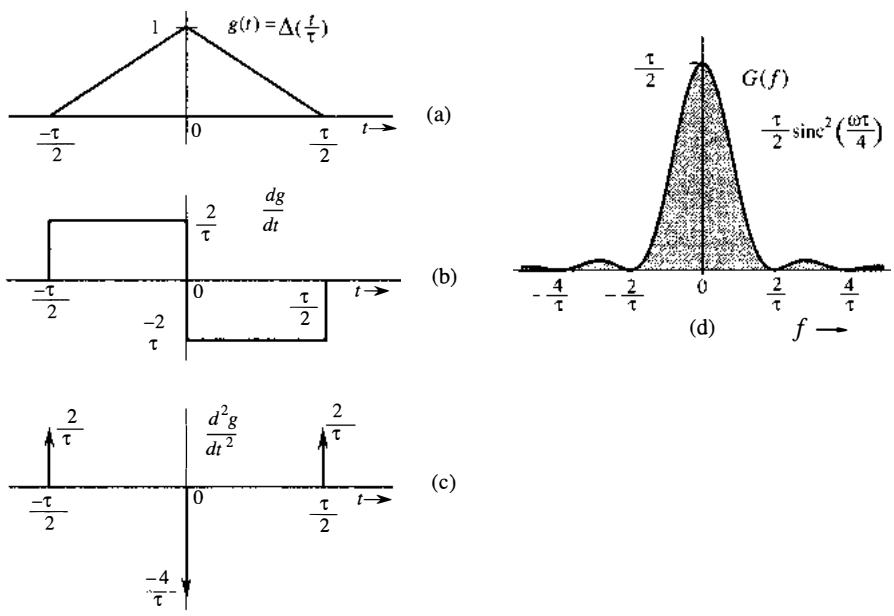


Figure 3.23 Using the time differentiation property to finding the Fourier transform of a piecewise-linear signal.

* Valid only if the transform of $dg(t)/dt$ exists.

To find the Fourier transform of this pulse, we differentiate it successively, as shown in Fig. 3.23b and c. The second derivative consists of a sequence of impulses (Fig. 3.23c). Recall that the derivative of a signal at a jump discontinuity is an impulse of strength equal to the amount of jump. The function $dg(t)/dt$ has a positive jump of $2/\tau$ at $t = \pm\tau/2$, and a negative jump of $4/\tau$ at $t = 0$. Therefore,

$$\frac{d^2g(t)}{dt^2} = \frac{2}{\tau}\left[\delta\left(t+\frac{\tau}{2}\right) - 2\delta(t) + \delta\left(t-\frac{\tau}{2}\right)\right] \tag{3.50}$$

From the time differentiation property [Eq. (3.49)],

$$\frac{d^2g}{dt^2} \Longleftrightarrow (j2\pi f)^2 G(f) = -(2\pi f)^2 G(f) \tag{3.51a}$$

Also, from the time-shifting property [Eqs. (3.32)],

$$\delta(t - t_0) \Longleftrightarrow e^{-j2\pi f t_0} \tag{3.51b}$$

Taking the Fourier transform of Eq. (3.50) and using the results in Eq. (3.51), we obtain

$$(j2\pi f)^2 G(f) = \frac{2}{\tau}\left(e^{j\pi f\tau} - 2 + e^{-j\pi f\tau}\right) = \frac{4}{\tau}(\cos \pi f\tau - 1) = -\frac{8}{\tau}\sin^2\left(\frac{\pi f\tau}{2}\right)$$

and

$$G(f) = \frac{8}{(2\pi f)^2\tau}\sin^2\left(\frac{\pi f\tau}{2}\right) = \frac{\tau}{2}\left[\frac{\sin(\pi f\tau/2)}{\pi f\tau/2}\right]^2 = \frac{\tau}{2}\operatorname{sinc}^2\left(\frac{\pi f\tau}{2}\right) \tag{3.52}$$

The spectrum $G(f)$ is shown in Fig. 3.23d. This procedure of finding the Fourier transform can be applied to any function $g(t)$ made up of straight-line segments with $g(t) \to 0$ as $|t| \to \infty$. The second derivative of such a signal yields a sequence of impulses whose Fourier transform can be found by inspection. This example suggests a numerical method of finding the Fourier transform of an arbitrary signal $g(t)$ by approximating the signal by straight-line segments.

To provide easy reference, several important properties of Fourier transform are summarized in Table 3.2.

3.4 SIGNAL TRANSMISSION THROUGH A LINEAR SYSTEM

A linear time-invariant (LTI) continuous time system can be characterized equally well in either the time domain or the frequency domain. The LTI system model, illustrated in Fig. 3.24, can often be used to characterize communication channels. In communication systems and in signal processing, we are interested only in bounded-input–bounded-output (BIBO) stable linear systems. Detailed discussions on system stability can be found in the textbook by Lathi.[3]

TABLE 3.2
Properties of Fourier Transform Operations

Operation	$g(t)$	$G(f)$
Superposition	$g_1(t) + g_2(t)$	$G_1(f) + G_2(f)$
Scalar multiplication	$kg(t)$	$kG(f)$
Duality	$G(t)$	$g(-f)$
Time scaling	$g(at)$	$\frac{1}{\lvert a\rvert} G\left(\frac{f}{a}\right)$
Time shifting	$g(t - t_0)$	$G(f)e^{-j2\pi f t_0}$
Frequency shifting	$g(t)e^{j2\pi f_0 t}$	$G(f - f_0)$
Time convolution	$g_1(t) * g_2(t)$	$G_1(f)G_2(f)$
Frequency convolution	$g_1(t)g_2(t)$	$G_1(f) * G_2(f)$
Time differentiation	$\frac{d^n g(t)}{dt^n}$	$(j2\pi f)^n G(f)$
Time integration	$\int_{-\infty}^{t} g(x)\,dx$	$\frac{G(f)}{j2\pi f} + \frac{1}{2}G(0)\delta(f)$

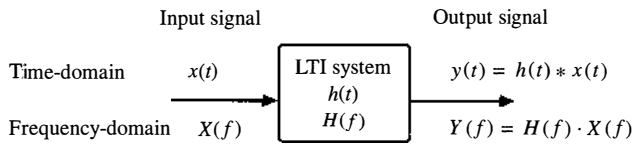


Figure 3.24 Signal transmission through a linear time-invariant system.

A stable LTI system can be characterized in the time domain by its impulse response $h(t)$, which is the system response to a unit impulse input, that is,

$$y(t) = h(t) \qquad \text{when} \qquad x(t) = \delta(t)$$

The system response to a bounded input signal $x(t)$ follows the convolutional relationship

$$y(t) = h(t) * x(t) \tag{3.53}$$

The frequency domain relationship between the input and the output is obtained by taking Fourier transform of both sides of Eq. (3.53). We let

$$x(t) \Longleftrightarrow X(f)$$
$$y(t) \Longleftrightarrow Y(f)$$
$$h(t) \Longleftrightarrow H(f)$$

Then according to the convolution theorem, Eq. (3.53) becomes

$$Y(f) = H(f) \cdot X(f) \tag{3.54}$$

Generally $H(f)$, the Fourier transform of the impulse response $h(t)$, is referred to as the **transfer function** or the **frequency response** of the LTI system. Again, in general, $H(f)$ is

complex and can be written as

$$H(f) = |H(f)|e^{j\theta_h(f)}$$

where $|H(f)|$ is the amplitude response and $\theta_h(f)$ is the phase response of the LTI system.

3.4.1 Signal Distortion during Transmission

The transmission of an input signal $x(t)$ through a system changes it into the output signal $y(t)$. Equation (3.54) shows the nature of this change or modification. Here $X(f)$ and $Y(f)$ are the spectra of the input and the output, respectively. Therefore, $H(f)$ is the spectral response of the system. The output spectrum is given by the input spectrum multiplied by the spectral response of the system. Equation (3.54) clearly brings out the spectral shaping (or modification) of the signal by the system. Equation (3.54) can be expressed in polar form as

$$|Y(f)|e^{j\theta_y(f)} = |X(f)||H(f)|e^{j[\theta_x(f)+\theta_h(f)]}$$

Therefore, we have the amplitude and phase relationships

$$|Y(f)| = |X(f)|\,|H(f)| \tag{3.55a}$$

$$\theta_y(f) = \theta_x(f) + \theta_h(f) \tag{3.55b}$$

During the transmission, the input signal amplitude spectrum $|X(f)|$ is changed to $|X(f)| \cdot |H(f)|$. Similarly, the input signal phase spectrum $\theta_x(f)$ is changed to $\theta_x(f) + \theta_h(f)$.

An input signal spectral component of frequency f is modified in amplitude by a factor $|H(f)|$ and is shifted in phase by an angle $\theta_h(f)$. Clearly, $|H(f)|$ is the amplitude response, and $\theta_h(f)$ is the phase response of the system. The plots of $|H(f)|$ and $\theta_h(f)$ as functions of f show at a glance how the system modifies the amplitudes and phases of various sinusoidal inputs. This is why $H(f)$ is called the **frequency response** of the system. During transmission through the system, some frequency components may be boosted in amplitude, while others may be attenuated. The relative phases of the various components also change. In general, the output waveform will be different from the input waveform.

3.4.2 Distortionless Transmission

In several applications, such as signal amplification or message signal transmission over a communication channel, we require the output waveform to be a replica of the input waveform. In such cases, we need to minimize the distortion caused by the amplifier or the communication channel. It is therefore of practical interest to determine the characteristics of a system that allows a signal to pass without distortion (**distortionless transmission**).

Transmission is said to be distortionless if the input and the output have identical wave shapes within a multiplicative constant. A delayed output that retains the input waveform is also considered distortionless. Thus, in distortionless transmission, the input $x(t)$ and the output $y(t)$ satisfy the condition

$$y(t) = k \cdot x(t - t_d) \tag{3.56}$$

The Fourier transform of this equation yields

$$Y(f) = kX(f)e^{-j2\pi f t_d}$$

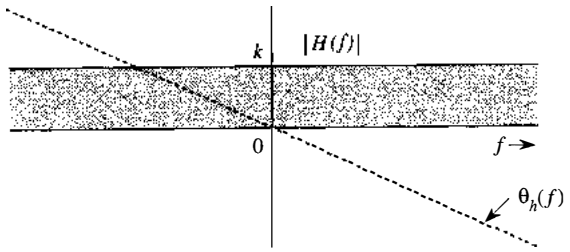


Figure 3.25 Linear time-invariant system frequency response for distortionless transmission.

But because

$$Y(f) = X(f)H(f)$$

we therefore have

$$H(f) = k\, e^{-j2\pi f t_d}$$

This is the transfer function required for distortionless transmission. From this equation it follows that

$$|H(f)| = k \tag{3.57a}$$

$$\theta_h(f) = -2\pi f t_d \tag{3.57b}$$

This shows that for distortionless transmission, the amplitude response $|H(f)|$ must be a constant, and the phase response $\theta_h(f)$ must be a linear function of f going through the origin $f = 0$, as shown in Fig. 3.25. The slope of $\theta_h(f)$ with respect to the angular frequency $\omega = 2\pi f$ is $-t_d$, where t_d is the delay of the output with respect to the input.*

All-Pass vs. Distortionless System

In circuit analysis and filter designs, we sometimes are mainly concerned with the gain of a system response. An all-pass system has a constant gain for all frequencies [i.e., $|H(f)| = k$], without the linear phase requirement. Note from Eq. (3.57) that a distortionless system is always an all-pass system, whereas the converse is not true. Because it is very common for beginners to be confused by the difference between all-pass and distortionless systems, now is the best time to clarify.

To see how an all-pass system may lead to distortion, let us consider an illustrative example. Imagine that we would like to transmit a recorded music signal from a violin-cello duet. The violin contributes to the high frequency part of this music signal, while the cello contributes to the bases part. When this music signal is transmitted through a particular *all-pass* system, both parts have the same gain. However, suppose that this all-pass system would cause a 1-second *extra* delay on the high-frequency content of the music (from the violin). As a result, the audience on the receiving end will hear a "music" signal that is totally out of sync even though *all signal components have the same gain and all are present.* The difference in transmission delay for components of different frequencies is contributed by the nonlinear phase of $H(f)$ in the all-pass filter.

---

* In addition, we require that $\theta_h(0)$ either be 0 (as shown in Fig. 3.25) or have a constant value $n\pi$ (n an integer), that is, $\theta_h(f) = n\pi - 2\pi f t_d$. The addition of the excess phase of $n\pi$ may at most change the sign of the signal.

To be more precise, the transfer function gain $|H(f)|$ determines the gain of each input frequency component, whereas $\angle H(f)$ determines the delay of each component. Imagine a system input $x(t)$ consisting of multiple sinusoids (its spectral components). For the output signal $y(t)$ to be distortionless, it should be the input signal multiplied by a gain k and delayed by t_d. To synthesize such a signal, $y(t)$ needs exactly the same components as those of $x(t)$, with each component multiplied by k and delayed by t_d. This means that the system transfer function $H(f)$ should be such that each sinusoidal component encounters the same gain (or loss) k and each component undergoes the same time delay of t_d seconds. The first condition requires that

$$|H(f)| = k$$

We have seen earlier (Sec. 3.3) that to achieve the same time delay t_d for every frequency component requires a linear phase delay $2\pi f t_d$ (Fig. 3.18) through the origin

$$\theta_h(f) = -2\pi f t_d$$

In practice, many systems have a phase characteristic that may be only approximately linear. A convenient method of checking phase linearity is to plot the slope of $\angle H(f)$ as a function of frequency. This slope can be a function of f in the general case and is given by

$$t_d(f) = -\frac{1}{2\pi} \cdot \frac{d\theta_h(f)}{df} \tag{3.58}$$

If the slope of θ_h is constant (that is, if θ_h is linear with respect to f), all the components are delayed by the same time interval t_d. But if the slope is not constant, then the time delay t_d varies with frequency. This means that different frequency components undergo different amounts of time delay, and consequently the output waveform will not be a replica of the input waveform (as in the example of the violin-cello duet). For a signal transmission to be distortionless, $t_d(f)$ should be a constant t_d over the frequency band of interest.*

Thus, there is a clear distinction between all-pass and distortionless systems. It is a common mistake to think that flatness of amplitude response $|H(f)|$ alone can guarantee signal quality. A system that has a flat amplitude response may yet distort a signal beyond recognition if the phase response is not linear (t_d not constant).

The Nature of Distortion in Audio and Video Signals

Generally speaking, a human ear can readily perceive amplitude distortion, although it is relatively insensitive to phase distortion. For the phase distortion to become noticeable, the

* Figure 3.25 shows that for distortionless transmission, the phase response not only is linear but also must pass through the origin. This latter requirement can be somewhat relaxed for bandpass signals. The phase at the origin may be any constant [$\theta_h(f) = \theta_0 - 2\pi f t_d$ or $\theta_h(0) = \theta_0$]. The reason for this can be found in Eq. (3.37), which shows that the addition of a constant phase θ_0 to a spectrum of a bandpass signal amounts to a phase shift of the carrier by θ_0. The modulating signal (the envelope) is not affected. The output envelope is the same as the input envelope delayed by

$$t_g = -\frac{1}{2\pi}\frac{d\theta_h(f)}{df}$$

called the **group delay** or **envelope delay**, and the output carrier is the same as the input carrier delayed by

$$t_p = -\frac{\theta_h(f)}{2\pi f}$$

called the **phase delay**, where f_0 is the center frequency of the passband.

variation in delay (variation in the slope of θ_h) should be comparable to the signal duration (or the physically perceptible duration, in case the signal itself is long). In the case of audio signals, each spoken syllable can be considered to be an individual signal. The average duration of a spoken syllable is of a magnitude on the order of 0.01 to 0.1 second. The audio systems may have nonlinear phases, yet no noticeable signal distortion results because in practical audio systems, maximum variation in the slope of θ_h is only a small fraction of a millisecond. This is the real reason behind the statement that "the human ear is relatively insensitive to phase distortion.[4] As a result, the manufacturers of audio equipment make available only $|H(f)|$, the amplitude response characteristic of their systems.

For video signals, on the other hand, the situation is exactly the opposite. The human eye is sensitive to phase distortion but is relatively insensitive to amplitude distortion. The amplitude distortion in television signals manifests itself as a partial destruction of the relative half-tone values of the resulting picture, which is not readily apparent to the human eye. The phase distortion (nonlinear phase), on the other hand, causes different time delays in different picture elements. This results in a smeared picture, which is readily apparent to the human eye. Phase distortion is also very important in digital communication systems because the nonlinear phase characteristic of a channel causes pulse dispersion (spreading out), which in turn causes pulses to interfere with neighboring pulses. This interference can cause an error in the pulse amplitude at the receiver: a binary **1** may read as **0**, and vice versa.

3.5 IDEAL VERSUS PRACTICAL FILTERS

Ideal filters allow distortionless transmission of a certain band of frequencies and suppress all the remaining frequencies. The ideal low-pass filter (Fig. 3.26), for example, allows all components below $f = B$ Hz to pass without distortion and suppresses all components above $f = B$. Figure 3.27 shows ideal high-pass and bandpass filter characteristics.

The ideal low-pass filter in Fig. 3.26a has a linear phase of slope $-t_d$, which results in a time delay of t_d seconds for all its input components of frequencies below B Hz. Therefore, if the input is a signal $g(t)$ band-limited to B Hz, the output $y(t)$ is $g(t)$ delayed by t_d, that is,

$$y(t) = g(t - t_d)$$

The signal $g(t)$ is transmitted by this system without distortion, but with time delay t_d. For this filter $|H(f)| = \Pi(f/2B)$, and $\theta_h(f) = -2\pi f t_d$, so that

$$H(f) = \Pi\left(\frac{f}{2B}\right) e^{-j2\pi f t_d} \tag{3.59a}$$

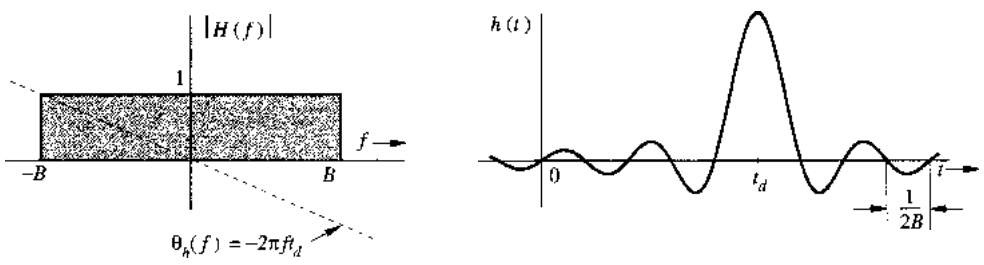


Figure 3.26 Ideal low-pass filter frequency response and its impulse response.

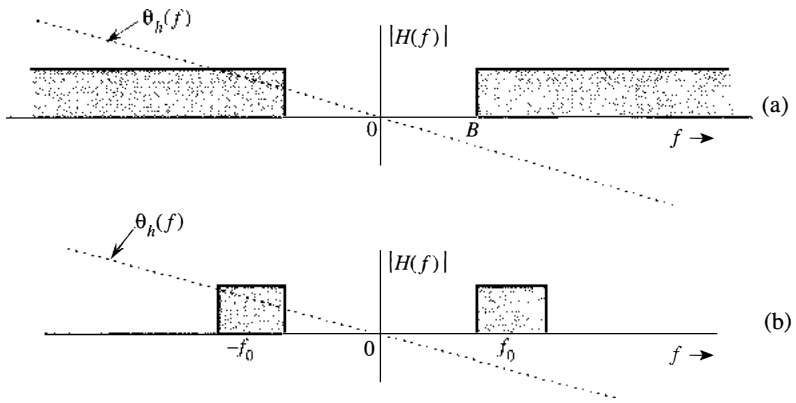


Figure 3.27 Ideal high-pass and bandpass filter frequency responses.

The unit impulse response $h(t)$ of this filter is found from pair 18 in Table 3.1 and the time-shifting property:

$$h(t) = \mathcal{F}^{-1}\left[\Pi\left(\frac{f}{2B}\right)e^{-j2\pi f t_d}\right]$$

$$= 2B \text{ sinc } [2\pi B(t - t_d)] \tag{3.59b}$$

Recall that $h(t)$ is the system response to impulse input $\delta(t)$, which is applied at $t = 0$. Figure 3.26b shows a curious fact: the response $h(t)$ begins even before the input is applied (at $t = 0$). Clearly, the filter is noncausal and therefore unrealizable; that is, such a system is physically impossible, since no sensible system can respond to an input **before** it is applied to the system. Similarly, one can show that other ideal filters (such as the ideal high-pass or the ideal bandpass filters shown in Fig. 3.27) are also physically unrealizable.

For a physically realizable system, $h(t)$ must be causal; that is,

$$h(t) = 0 \qquad \text{for } t < 0$$

In the frequency domain, this condition is equivalent to the **Paley-Wiener criterion**, which states that the necessary and sufficient condition for $|H(f)|$ to be the amplitude response of a realizable (or causal) system is*

$$\int_{-\infty}^{\infty} \frac{|\ln |H(f)|\,|}{1 + (2\pi f)^2}\,df < \infty \tag{3.60}$$

If $H(f)$ does not satisfy this condition, it is unrealizable. Note that if $|H(f)| = 0$ over any finite band, $|\ln|H(f)|| = \infty$ over that band, and the condition (3.60) is violated. If, however, $H(f) = 0$ at a single frequency (or a set of discrete frequencies), the integral in Eq. (3.60) may still be finite even though the integrand is infinite. Therefore, for a physically realizable system, $H(f)$ may be zero at some discrete frequencies, but it cannot be zero over any finite band. According to this criterion, ideal filter characteristics (Figs. 3.26 and 3.27) are clearly unrealizable.

---

* $|H(f)|$ is assumed to be square integrable. That is,

$$\int_{-\infty}^{\infty} |H(f)|^2\,df$$

is assumed to be finite.

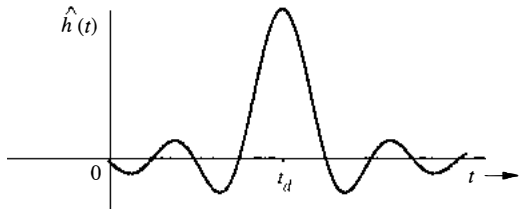


Figure 3.28 Approximate realization of an ideal low-pass filter by truncating its impulse response.

The impulse response $h(t)$ in Fig. 3.26 is not realizable. One practical approach to filter design is to cut off the tail of $h(t)$ for $t < 0$. The resulting causal impulse response $\hat{h}(t)$, where

$$\hat{h}(t) = h(t)u(t)$$

is physically realizable because it is causal (Fig. 3.28). If t_d is sufficiently large, $\hat{h}(t)$ will be a close approximation of $h(t)$, and the resulting filter $\hat{H}(f)$ will be a good approximation of an ideal filter. This close realization of the ideal filter is achieved because of the increased value of time delay t_d. This means that the price of close physical approximation is higher delay in the output; this is often true of noncausal systems. Of course, theoretically a delay $t_d = \infty$ is needed to realize the ideal characteristics. But a glance at Fig. 3.27b shows that a delay t_d of three or four times π/W will make $\hat{h}(t)$ a reasonably close version of $h(t - t_d)$. For instance, audio filters are required to handle frequencies of up to 20 kHz (the highest frequency the human ear can hear). In this case a t_d of about 10^{-4} (0.1 ms) would be a reasonable choice. The truncation operation [cutting the tail of $h(t)$ to make it causal], however, creates some unsuspected problems of spectral spread and leakage, and which can be partly corrected by using a tapered window function to truncate $h(t)$ gradually (rather than abruptly).[5]

In practice, we can realize a variety of filter characteristics to approach ideal characteristics. Practical (realizable) filter characteristics are gradual, without jump discontinuities in the amplitude response $|H(f)|$. For example, Butterworth and Chebychev filters are used extensively in various applications including practical communication circuits.

Analog signals can also be processed by digital means (A/D conversion). This involves sampling, quantizing, and coding. The resulting digital signal can be processed by a small, special-purpose digital computer designed to convert the input sequence into a desired output sequence. The output sequence is converted back into the desired analog signal. A special algorithm of the processing digital computer can be used to achieve a given signal operation (e.g., low-pass, bandpass, or high-pass filtering). The subject of digital filtering is somewhat beyond our scope in this book. Several excellent books are available on the subject.[3]

3.6 SIGNAL DISTORTION OVER A COMMUNICATION CHANNEL

A signal transmitted over a channel is distorted because of various channel imperfections. The nature of signal distortion will now be studied.

3.6.1 Linear Distortion

We shall first consider linear time-invariant channels. Signal distortion can be caused over such a channel by nonideal characteristics of magnitude distortion, phase distortion, or both.

We can identify the effects these **nonidealities** will have on a pulse $g(t)$ transmitted through such a channel. Let the pulse exist over the interval (a, b) and be zero outside this interval. The components of the Fourier spectrum of the pulse have such a perfect and delicate balance of magnitudes and phases that they add up precisely to the pulse $g(t)$ over the interval (a, b) and to zero outside this interval. The transmission of $g(t)$ through an ideal channel that satisfies the conditions of distortionless transmission also leaves this balance undisturbed, because a distortionless channel multiplies each component by the same factor and delays each component by the same amount of time. Now, if the amplitude response of the channel is not ideal [i.e., if $|H(f)|$ is not equal to a constant], this delicate balance will be disturbed, and the sum of all the components cannot be zero outside the interval (a, b). In short, the pulse will spread out (see Example 3.14). The same thing happens if the channel phase characteristic is not ideal, that is, if $\theta_h(f) \neq -2\pi f t_d$. Thus, spreading, or **dispersion**, of the pulse will occur if either the amplitude response or the phase response, or both, are nonideal.

Linear channel distortion (dispersion in time) is particularly damaging to digital communication systems. It introduces what is known as intersymbol interferences (ISI). In other words, a digital symbol, when transmitted over a dispersive channel, tends to spread more widely than its allotted time. Therefore, adjacent symbols will interfere with one another, thereby increasing the probability of detection error at the receiver.

Example 3.14 A low-pass filter (Fig. 3.29a) transfer function $H(f)$ is given by

$$H(f) = \begin{cases} (1 + k\cos 2\pi f T)e^{-j2\pi f t_d} & |f| < B \\ 0 & |f| > B \end{cases} \tag{3.61}$$

A pulse $g(t)$ band-limited to B Hz (Fig. 3.29b) is applied at the input of this filter. Find the output $y(t)$.

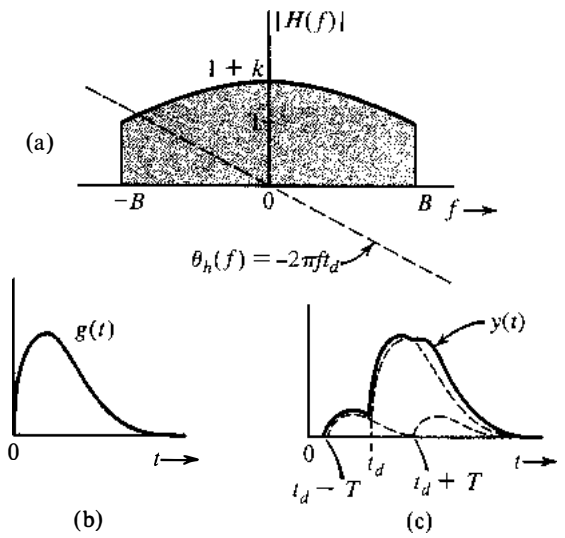


Figure 3.29 Pulse is dispersed when it passes through a system that is not distortionless.

This filter has ideal phase and nonideal magnitude characteristics. Because $g(t) \Leftrightarrow G(f)$, $y(t) \Leftrightarrow Y(f)$ and

$$\begin{aligned} Y(f) &= G(f)H(f) \\ &= G(f) \cdot \Pi\left(\frac{f}{2B}\right)(1 + k\cos 2\pi fT)e^{-j2\pi ft_d} \\ &= G(f)e^{-j2\pi ft_d} + k\,[G(f)\cos 2\pi fT]\; e^{-j2\pi ft_d} \end{aligned} \tag{3.62}$$

Note that in the derivation of Eq. (3.62) because $g(t)$ is band-limited to B Hz, we have $G(f) \cdot \Pi\left(\frac{f}{2B}\right) = G(f)$. Then, by using the time-shifting property and Eq. (3.32a), we have

$$y(t) = g(t - t_d) + \frac{k}{2}[g(t - t_d - T) + g(t - t_d + T)] \tag{3.63}$$

The output is actually $g(t) + (k/2)[g(t-T) + g(t+T)]$ delayed by t_d. It consists of $g(t)$ and its echoes shifted by $\pm t_d$. The dispersion of the pulse caused by its echoes is evident from Fig. 3.29c. Ideal amplitude but nonideal phase response of $H(f)$ has a similar effect (see Prob. 3.6-1).

3.6.2 Distortion Caused by Channel Nonlinearities

Until now we have considered the channel to be linear. This approximation is valid only for small signals. For large signal amplitudes, nonlinearities cannot be ignored. A general discussion of nonlinear systems is beyond our scope. Here we shall consider a simple case of a memoryless nonlinear channel where the input g and the output y are related by some (memoryless) nonlinear equation,

$$y = f(g)$$

The right-hand side of this equation can be expanded in a Maclaurin series as

$$y(t) = a_0 + a_1 g(t) + a_2 g^2(t) + a_3 g^3(t) + \cdots + a_k g^k(t) + \cdots$$

Recall the result in Sec. 3.3.6 (convolution) that if the bandwidth of $g(t)$ is B Hz, then the bandwidth of $g^k(t)$ is kB Hz. Hence, the bandwidth of $y(t)$ is **greater than** kB Hz. Consequently, the output spectrum spreads well beyond the input spectrum, and the output signal contains new frequency components not contained in the input signal. In broadcast communication, we need to amplify signals at very high power levels, where high-efficiency (class C) amplifiers are desirable. Unfortunately, these amplifiers are nonlinear, and they cause distortion when used to amplify signals. This is one of the serious problems in AM signals. However, FM signals are not affected by nonlinear distortion, as shown in Chapter 5. If a signal is transmitted over a nonlinear channel, the nonlinearity not only distorts the signal but also causes interference with other signals on the channel because of its spectral dispersion (spreading).

For digital communication systems, the nonlinear distortion effect is in contrast to the time dispersion effect due to linear distortion. Linear distortion causes interference among signals within the same channel, whereas spectral dispersion due to nonlinear distortion causes interference among signals using different frequency channels.

Example 3.15 The input $x(t)$ and the output $y(t)$ of a certain nonlinear channel are related as

$$y(t) = x(t) + 0.000158x^2(t)$$

Find the output signal $y(t)$ and its spectrum $Y(f)$ if the input signal is $x(t) = 2000\,\text{sinc}\,(2000\pi t)$. Verify that the bandwidth of the output signal is twice that of the input signal. This is the result of signal squaring. Can the signal $x(t)$ be recovered (without distortion) from the output $y(t)$?

Since

$$x(t) = 2000\,\text{sinc}\,(2000\pi t) \qquad \Longleftrightarrow \qquad X(f) = \Pi\left(\frac{f}{2000}\right)$$

We have

$$y(t) = x(t) + 0.000158x^2(t) = 2000\,\text{sinc}\,(2000\pi t) + 0.316 \cdot 2000\,\text{sinc}^2\,(2000\pi t)$$

$$\Longleftrightarrow$$

$$Y(f) = \Pi\left(\frac{f}{2000}\right) + 0.316\,\Delta\left(\frac{f}{4000}\right)$$

Observe that $0.316 \cdot 2000\text{sinc}^2\,(2000\pi t)$ is the unwanted (distortion) term in the received signal. Figure 3.30a shows the input (desired) signal spectrum $X(f)$; Fig. 3.30b shows the spectrum of the undesired (distortion) term; and Fig. 3.30c shows the received signal spectrum $Y(f)$. We make the following observations.

1. The bandwidth of the received signal $y(t)$ is twice that of the input signal $x(t)$ (because of signal squaring).
2. The received signal contains the input signal $x(t)$ plus an unwanted signal $632\,\text{sinc}^2(2000\pi t)$. The spectra of these two signals are shown in Fig. 3.30a and b. Figure 3.30c shows $Y(f)$, the spectrum of the received signal. Note that spectra of the desired signal and the distortion signal overlap, and it is impossible to recover the signal $x(t)$ from the received signal $y(t)$ without some distortion.
3. We can reduce the distortion by passing the received signal through a low-pass filter of bandwidth 1000 Hz. The spectrum of the output of this filter is shown in Fig. 3.30d. Observe that the output of this filter is the desired input signal $x(t)$ with some residual distortion.
4. We have an additional problem of interference with other signals if the input signal $x(t)$ is frequency-division-multiplexed along with several other signals on this channel. This means that several signals occupying nonoverlapping frequency bands are transmitted simultaneously on the same channel. Spreading the spectrum $X(f)$ outside its original band of 1000 Hz will interfere with the signal in the band of 1000 to 2000 Hz. Thus, in addition to the distortion of $x(t)$, we have an interference with the neighboring band.

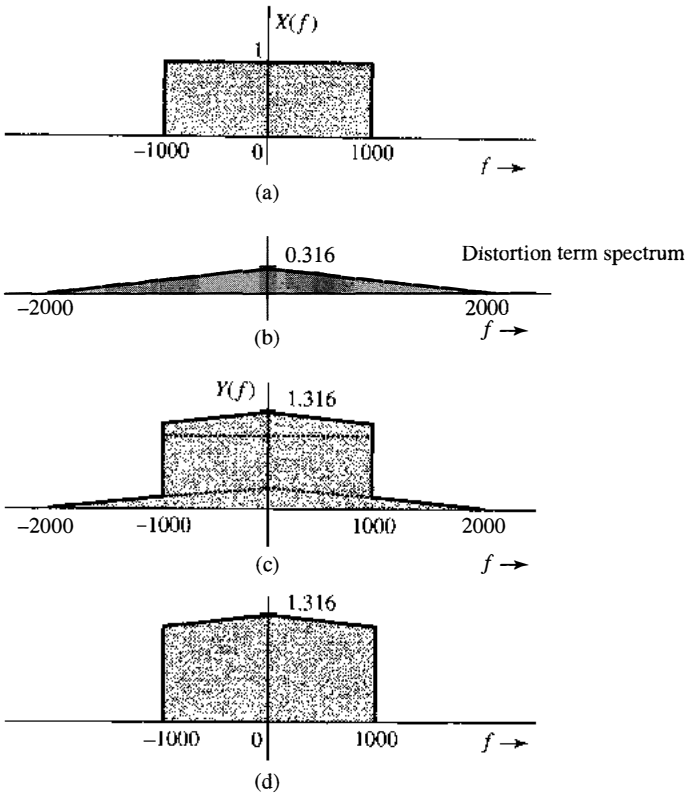


Figure 3.30 Signal distortion caused by nonlinear operation. (a) Desired (input) signal spectrum. (b) Spectrum of the unwanted signal (distortion) in the received signal. (c) Spectrum of the received signal. (d) Spectrum of the received signal after low-pass filtering.

5. If $x(t)$ were a digital signal consisting of a pulse train, each pulse would be distorted, but there would be no interference with the neighboring pulses. Moreover even with distorted pulses, data can be received without loss because digital communication can withstand considerable pulse distortion without loss of information. Thus, if this channel were used to transmit a time-division multiplexed signal consisting of two interleaved pulse trains, the data in the two trains would be recovered at the receiver.

3.6.3 Distortion Caused by Multipath Effects

A multipath transmission occurs when a transmitted signal arrives at the receiver by two or more paths of different delays. For example, if a signal is transmitted over a cable that has impedance irregularities (mismatching) along the path, the signal will arrive at the receiver in the form of a direct wave plus various reflections with various delays. In radio links, the signal can be received by direct path between the transmitting and the receiving antennas and also by reflections from other objects, such as hills and buildings. In long-distance radio links using the ionosphere, similar effects occur because of one-hop and multihop paths. In each of these cases, the transmission channel can be represented as several channels in parallel, each with a different relative attenuation and a different time delay. Let us consider the case of only two paths: one with a unity gain and a delay t_d, and the other with a gain α and a delay $t_d + \Delta t$, as shown in Fig. 3.31a. The transfer functions of the two paths are given by $e^{-j2\pi f t_d}$ and $\alpha e^{-j2\pi f(t_d+\Delta t)}$, respectively. The overall transfer function of such a channel is

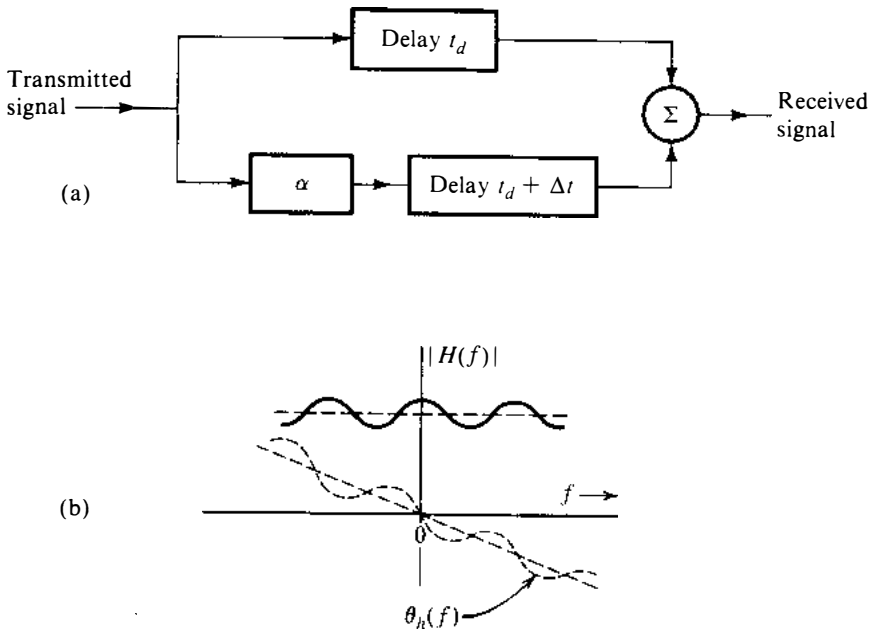


Figure 3.31 Multipath transmission.

$H(f)$, given by

$$\begin{aligned} H(f) &= e^{-j2\pi f t_d} + \alpha e^{-j2\pi f(t_d+\Delta t)} \\ &= e^{-j2\pi f t_d}(1 + \alpha e^{-j2\pi f \Delta t}) \qquad (3.64a) \\ &= e^{-j2\pi f t_d}(1 + \alpha \cos 2\pi f \Delta t - j\alpha \sin 2\pi f \Delta t) \end{aligned}$$

$$= \underbrace{\sqrt{1 + \alpha^2 + 2\alpha \cos 2\pi f \Delta t}}_{|H(f)|} \exp\left[-j \underbrace{\left(2\pi f t_d + \tan^{-1} \frac{\alpha \sin 2\pi f \Delta t}{1 + \alpha \cos 2\pi f \Delta t} \right)}_{\theta_h(f)} \right] \qquad (3.64b)$$

Both the magnitude and the phase characteristics of $H(f)$ are periodic in f with a period of $1/\Delta t$ (Fig. 3.31b). The multipath channel, therefore, can exhibit nonidealities in the magnitude and the phase characteristics of the channel and can cause linear distortion (pulse dispersion), as discussed earlier.

If, for instance, the gains of the two paths are very close, that is, $\alpha \approx 1$, then the signals received from the two paths may have opposite phase (π radians apart) at certain frequencies. This means that at those frequencies where the two paths happen to result in opposite phases, the signals from the two paths will almost cancel each other. Equation (3.64b) shows that at frequencies where $f = n/(2\Delta t)$ (n odd), $\cos 2\pi f \Delta t = -1$, and $|H(f)| \approx 0$ when $\alpha \approx 1$. These frequencies are the multipath null frequencies. At frequencies $f = n/(2\Delta t)$ (n even), the two signals interfere constructively to enhance the gain. Such channels cause **frequency-selective fading** of transmitted signals. Such distortion can be partly corrected by using the tapped delay-line equalizer, as shown in Prob. 3.6-2. These equalizers are useful in several applications in communications. Their design issues are addressed later in Chapters 7 and 12.

3.6.4 Fading Channels

Thus far, the channel characteristics have been assumed to be constant with time. In practice, we encounter channels whose transmission characteristics vary with time. These include troposcatter channels and channels using the ionosphere for radio reflection to achieve long-distance communication. The time variations of the channel properties arise because of semi periodic and random changes in the propagation characteristics of the medium. The reflection properties of the ionosphere, for example, are related to meteorological conditions that change seasonally, daily, and even from hour to hour, much like the weather. Periods of sudden storms also occur. Hence, the effective channel transfer function varies semi periodically and randomly, causing random attenuation of the signal. This phenomenon is known as **fading**. One way to reduce the effects of slow fading is to use **automatic gain control (AGC)**.*

Fading may be strongly frequency dependent where different frequency components are affected unequally. Such fading, known as frequency-selective fading, can cause serious problems in communication. Multipath propagation can cause frequency-selective fading.

3.7 SIGNAL ENERGY AND ENERGY SPECTRAL DENSITY

The energy E_g of a signal $g(t)$ is defined as the area under $|g(t)|^2$. We can also determine the signal energy from its Fourier transform $G(f)$ through Parseval's theorem.

3.7.1 Parseval's Theorem

Signal energy can be related to the signal spectrum $G(f)$ by substituting Eq. (3.9b) in Eq. (2.2):

$$E_g = \int_{-\infty}^{\infty} g(t)g^*(t)\,dt = \int_{-\infty}^{\infty} g(t)\left[\int_{-\infty}^{\infty} G^*(f)e^{-j2\pi ft}\,df\right]dt$$

Here, we used the fact that $g^*(t)$, being the conjugate of $g(t)$, can be expressed as the conjugate of the right-hand side of Eq. (3.9b). Now, interchanging the order of integration yields

$$\begin{aligned} E_g &= \int_{-\infty}^{\infty} G^*(f)\left[\int_{-\infty}^{\infty} g(t)e^{-j2\pi ft}\,dt\right]df \\ &= \int_{-\infty}^{\infty} G(f)G^*(f)\,df \\ &= \int_{-\infty}^{\infty} |G(f)|^2\,df \end{aligned} \tag{3.65}$$

This is the well-known statement of Parseval theorem. A similar result was obtained for a periodic signal and its Fourier series in Eq. (2.68). This result allows us to determine the signal energy from either the time domain specification $g(t)$ or the frequency domain specification $G(f)$ of the same signal.

* AGC will also suppress slow variations of the original signal.

Example 3.16 Verify Parseval's theorem for the signal $g(t) = e^{-at}u(t)$ $(a > 0)$.

We have

$$E_g = \int_{-\infty}^{\infty} g^2(t)\,dt = \int_0^{\infty} e^{-2at}\,dt = \frac{1}{2a} \tag{3.66}$$

We now determine E_g from the signal spectrum $G(f)$ given by

$$G(f) = \frac{1}{j2\pi f + a}$$

and from Eq. (3.65),

$$E_g = \int_{-\infty}^{\infty} |G(f)|^2\,df = \int_{-\infty}^{\infty} \frac{1}{(2\pi f)^2 + a^2}\,df = \frac{1}{2\pi a}\tan^{-1}\frac{2\pi f}{a}\bigg|_{-\infty}^{\infty} = \frac{1}{2a}$$

which verifies Parseval's theorem.

3.7.2 Energy Spectral Density (ESD)

Equation (3.65) can be interpreted to mean that the energy of a signal $g(t)$ is the result of energies contributed by all the spectral components of the signal $g(t)$. The contribution of a spectral component of frequency f is proportional to $|G(f)|^2$. To elaborate this further, consider a signal $g(t)$ applied at the input of an ideal bandpass filter, whose transfer function $H(f)$ is shown in Fig. 3.32a. This filter suppresses all frequencies except a narrow band Δf $(\Delta f \to 0)$ centered at angular frequency ω_0 (Fig. 3.32b). If the filter output is $y(t)$, then its Fourier transform $Y(f) = G(f)H(f)$, and E_y, the energy of the output $y(t)$, is

$$E_y = \int_{-\infty}^{\infty} |G(f)H(f)|^2\,df \tag{3.67}$$

Because $H(f) = 1$ over the passband Δf, and zero everywhere else, the integral on the right-hand side is the sum of the two shaded areas in Fig. 3.32b, and we have (for $\Delta f \to 0$)

$$E_y = 2\,|G(f_0)|^2\,df$$

Thus, $2|G(f)|^2\,df$ is the energy contributed by the spectral components within the two narrow bands, each of width Δf Hz, centered at $\pm f_0$. Therefore, we can interpret $|G(f)|^2$ as the energy per unit bandwidth (in hertz) of the spectral components of $g(t)$ centered at frequency f. In other words, $|G(f)|^2$ is the energy spectral density (per unit bandwidth in hertz) of $g(t)$. Actually, since both the positive- and the negative-frequency components combine to form the components in the band Δf, the energy contributed per unit bandwidth is $2|G(f)|^2$. However, for the sake of convenience we consider the positive- and negative-frequency components to be independent. The **energy spectral density (ESD)** $\Psi_g(t)$ is thus defined as

$$\Psi_g(f) = |G(f)|^2 \tag{3.68}$$

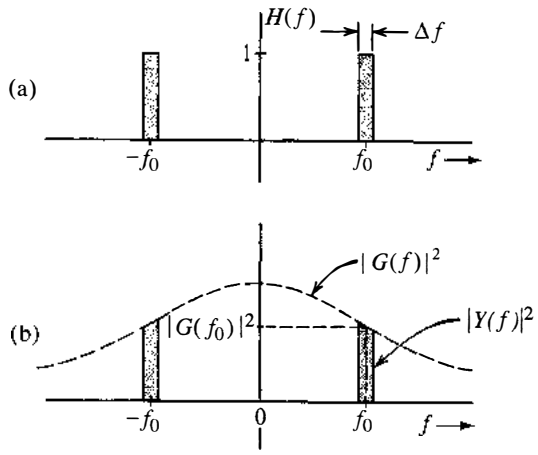


Figure 3.32 Interpretation of the energy spectral density of a signal.

and Eq. (3.65) can be expressed as

$$E_g = \int_{-\infty}^{\infty} \Psi_g(f)\, df \tag{3.69a}$$

From the results in Example 3.16, the ESD of the signal $g(t) = e^{-at}u(t)$ is

$$\Psi_g(f) = |G(f)|^2 = \frac{1}{(2\pi f)^2 + a^2} \tag{3.69b}$$

3.7.3 Essential Bandwidth of a Signal

The spectra of most signals extend to infinity. However, because the energy of a practical signal is finite, the signal spectrum must approach 0 as $f \to \infty$. Most of the signal energy is contained within a certain band of B Hz, and the energy content of the components of frequencies greater than B Hz is negligible. We can therefore suppress the signal spectrum beyond B Hz with little effect on the signal shape and energy. The bandwidth B is called the **essential bandwidth** of the signal. The criterion for selecting B depends on the error tolerance in a particular application. We may, for instance, select B to be that bandwidth that contains 95% of the signal energy.* The energy level may be higher or lower than 95%, depending on the precision needed. We can use such a criterion to determine the essential bandwidth of a signal. Suppression of all the spectral components of $g(t)$ beyond the essential bandwidth results in a signal $\hat{g}(t)$, which is a close approximation of $g(t)$.† If we use the 95% criterion for the essential bandwidth, the energy of the error (the difference) $g(t) - \hat{g}(t)$ is 5% of E_g. The following example demonstrates the bandwidth estimation procedure.

* Essential bandwidth for a low-pass signal may also be defined as a frequency at which the value of the amplitude spectrum is a small fraction (about 5–10%) of its peak value. In Example 3.16, the peak of $|G(f)|$ is $1/a$, and it occurs at $f = 0$.

† In practice the truncation is performed gradually, by using tapered windows, to avoid excessive spectral leakage due to the abrupt truncation.[5]

Example 3.17 Estimate the essential bandwidth W (in rad/s) of the signal $e^{-at}u(t)$ if the essential band is required to contain 95% of the signal energy.

In this case,

$$G(f) = \frac{1}{j2\pi f + a}$$

and the ESD is

$$|G(f)^2| = \frac{1}{(2\pi f)^2 + a^2}$$

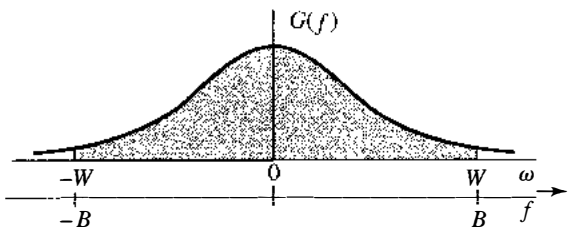


Figure 3.33 Estimating the essential bandwidth of a signal.

This ESD is shown in Fig. 3.33. Moreover, the signal energy E_g is the area under this ESD, which has already been found to be $1/2a$. Let W rad/s be the essential bandwidth, which contains 95% of the total signal energy E_g. This means $1/2\pi$ times the shaded area in Fig. 3.33 is $0.95/2a$, that is,

$$\frac{0.95}{2a} = \int_{-W/2\pi}^{W/2\pi} \frac{df}{(2\pi f)^2 + a^2}$$

$$= \frac{1}{2\pi a} \tan^{-1} \frac{2\pi f}{a}\Bigg|_{-W/2\pi}^{W/2\pi} = \frac{1}{\pi a} \tan^{-1} \frac{W}{a}$$

or

$$\frac{0.95\pi}{2} = \tan^{-1} \frac{W}{a} \Longrightarrow W = 12.7\,a \text{ rad/s}$$

In terms of hertz, the essential bandwidth is

$$B = \frac{W}{2\pi} = 2.02\,a \quad \text{Hz}$$

This means that in the band from 0 (dc) to 12.7 $\times$ a rad/s (2.02 $\times$ a Hz), the spectral components of $g(t)$ contribute 95% of the total signal energy; all the remaining spectral components (in the band from 2.02 $\times$ a Hz to ∞) contribute only 5% of the signal energy.*

* Note that although the ESD exists over the band $-\infty$ to ∞, the trigonometric spectrum exists only over the band 0 to ∞. The spectrum range $-\infty$ to ∞ applies to the exponential spectrum. In practice, whenever we talk about a bandwidth, we mean it in the trigonometric sense. Hence, the essential band is from 0 to B Hz (or W rad/s), not from $-B$ to B.

Example 3.18 Estimate the essential bandwidth of a rectangular pulse $g(t) = \Pi(t/T)$ (Fig. 3.34a), where the essential bandwidth is to contain at least 90% of the pulse energy.

For this pulse, the energy E_g is

$$E_g = \int_{-\infty}^{\infty} g^2(t)\,dt = \int_{-T/2}^{T/2} dt = T$$

Also because

$$\Pi\left(\frac{t}{T}\right) \Longleftrightarrow T\ \text{sinc}\,(\pi fT)$$

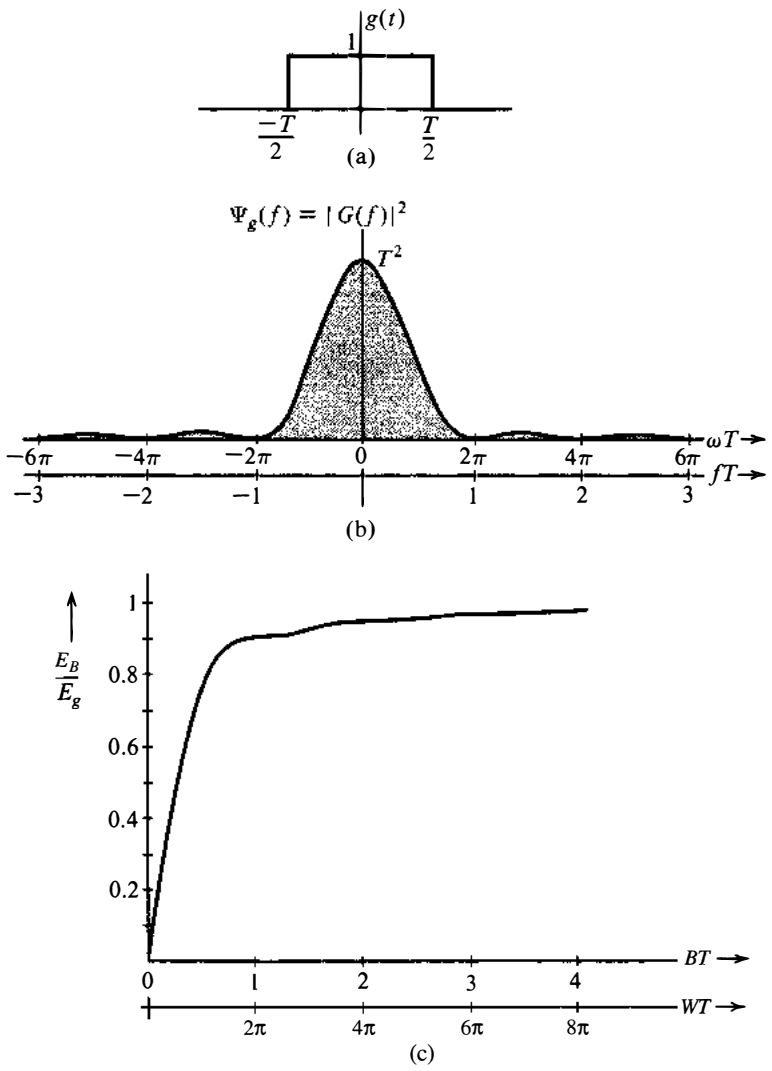


Figure 3.34 (a) EX-FGNI/FGC rectangular function, (b) its energy spectral density, and (c) fraction of energy inside $B(H_2)$.

the ESD for this pulse is

$$\Psi_g(f) = |G(f)|^2 = T^2 \operatorname{sinc}^2(\pi f T)$$

This ESD is shown in Fig. 3.34b as a function of ωT as well as fT, where f is the frequency in hertz. The energy E_B within the band from 0 to B Hz is given by

$$E_B = \int_{-B}^{B} T^2 \operatorname{sinc}^2(\pi f T)\, df$$

Setting $2\pi fT = x$ in this integral so that $df = dx/(2\pi T)$, we obtain

$$E_B = \frac{T}{\pi} \int_0^{2\pi BT} \operatorname{sinc}^2\left(\frac{x}{2}\right) dx$$

Also because $E_g = T$, we have

$$\frac{E_B}{E_g} = \frac{1}{\pi} \int_0^{2\pi BT} \operatorname{sinc}^2\left(\frac{x}{2}\right) dx$$

The integral on the right-hand side is numerically computed, and the plot of E_B/E_g vs. BT is shown in Fig. 3.34c. Note that 90.28% of the total energy of the pulse $g(t)$ is contained within the band $B = 1/T$ Hz. Therefore, by the 90% criterion, the bandwidth of a rectangular pulse of width T seconds is $1/T$ Hz.

3.7.4 Energy of Modulated Signals

We have seen that modulation shifts the signal spectrum $G(f)$ to the left and right by f_0. We now show that a similar thing happens to the ESD of the modulated signal.

Let $g(t)$ be a baseband signal band-limited to B Hz. The amplitude-modulated signal $\varphi(t)$ is

$$\varphi(t) = g(t) \cos 2\pi f_0 t$$

and the spectrum (Fourier transform) of $\varphi(t)$ is

$$\Phi(f) = \frac{1}{2}[G(f + f_0) + G(f - f_0)]$$

The ESD of the modulated signal $\varphi(t)$ is $|\Phi(f)|^2$, that is,

$$\Psi_\varphi(f) = \frac{1}{4}|G(f + f_0) + G(f - f_0)|^2$$

If $f_0 \geq B$, then $G(f + f_0)$ and $G(f - f_0)$ are nonoverlapping (see Fig. 3.35), and

$$\begin{aligned}\Psi_\varphi(f) &= \frac{1}{4}\left[|G(f + f_0)|^2 + |G(f - f_0)|^2\right] \\ &= \frac{1}{4}\Psi_g(f + f_0) + \frac{1}{4}\Psi_g(f - f_0)\end{aligned} \tag{3.70}$$

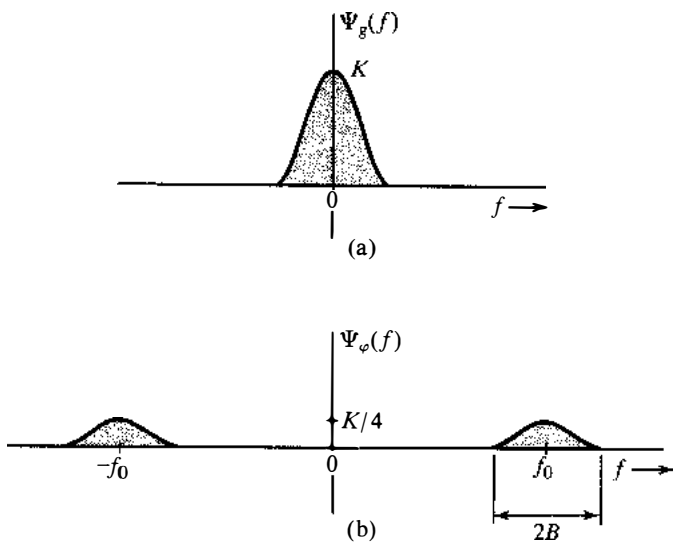


Figure 3.35 Energy spectral densities of modulating and modulated signals.

The ESDs of both $g(t)$ and the modulated signal $\varphi(t)$ are shown in Fig. 3.35. It is clear that modulation shifts the ESD of $g(t)$ by $\pm f_0$. Observe that the area under $\Psi_\varphi(f)$ is half the area under $\Psi_g(f)$. Because the energy of a signal is proportional to the area under its ESD, it follows that the energy of $\varphi(t)$ is half the energy of $g(t)$, that is,

$$E_\varphi = \frac{1}{2}E_g \qquad f_0 \geq B \tag{3.71}$$

It may seem surprising that a signal $\varphi(t)$, which appears so energetic in comparison to $g(t)$, should have only half the energy of $g(t)$. Appearances are deceiving, as usual. The energy of a signal is proportional to the square of its amplitude, and higher amplitudes contribute more energy. Signal $g(t)$ remains at higher amplitude levels most of the time. On the other hand, $\varphi(t)$, because of the factor $\cos 2\pi f_0 t$, dips to zero amplitude levels many times, which reduces its energy.

3.7.5 Time Autocorrelation Function and the Energy Spectral Density

In Chapter 2, we showed that a good measure of comparing two signals $g(t)$ and $z(t)$ is the cross-correlation function $\psi_{gz}(\tau)$ defined in Eq. (2.46). We also defined the correlation of a signal $g(t)$ with itself [the autocorrelation function $\psi_g(\tau)$] in Eq. (2.47). For a real signal $g(t)$, the autocorrelation function $\psi_g(\tau)$ is given by*

$$\psi_g(\tau) = \int_{-\infty}^{\infty} g(t)g(t+\tau)\,dt \tag{3.72a}$$

* For a complex signal $g(t)$, we define

$$\psi_g(\tau) = \int_{-\infty}^{\infty} g(t)g^*(t-\tau)\,dt = \int_{-\infty}^{\infty} g^*(t)g(t+\tau)\,dt$$

Setting $x = t + \tau$ in Eq. (3.72a) yields

$$\psi_g(\tau) = \int_{-\infty}^{\infty} g(x)g(x - \tau)\, dx$$

In this equation, x is a dummy variable and could be replaced by t. Thus,

$$\psi_g(\tau) = \int_{-\infty}^{\infty} g(t)g(t \pm \tau)\, dt \tag{3.72b}$$

This shows that for a real $g(t)$, the autocorrelation function is an even function of τ, that is,

$$\psi_g(\tau) = \psi_g(-\tau) \tag{3.72c}$$

There is, in fact, a very important relationship between the autocorrelation of a signal and its ESD. Specifically, the autocorrelation function of a signal $g(t)$ and its ESD $\Psi_g(f)$ form a Fourier transform pair, that is,

$$\psi_g(\tau) \Longleftrightarrow \Psi_g(f) \tag{3.73a}$$

Thus,

$$\Psi_g(f) = \mathcal{F}\left\{\psi_g(\tau)\right\} = \int_{-\infty}^{\infty} \psi_g(\tau)e^{-j2\pi f\tau}\, d\tau \tag{3.73b}$$

$$\psi_g(\tau) = \mathcal{F}^{-1}\left\{\Psi_g(f)\right\} = \int_{-\infty}^{\infty} \Psi_g(f)e^{-j2\pi f\tau}\, df \tag{3.73c}$$

Note that the Fourier transform of Eq. (3.73a) is performed with respect to τ in place of t.

We now prove that the ESD $\Psi_g(f) = |G(f)|^2$ is the Fourier transform of the autocorrelation function $\psi_g(\tau)$. Although the result is proved here for real signals, it is valid for complex signals also. Note that the autocorrelation function is a function of τ, not t. Hence, its Fourier transform is $\int \psi_g(\tau)e^{-j2\pi f\tau}\, d\tau$. Thus,

$$\begin{aligned}\mathcal{F}[\psi_g(\tau)] &= \int_{-\infty}^{\infty} e^{-j2\pi f\tau}\left[\int_{-\infty}^{\infty} g(t)g(t+\tau)\, dt\right] d\tau \\ &= \int_{-\infty}^{\infty} g(t)\left[\int_{-\infty}^{\infty} g(\tau + t)e^{-j2\pi f\tau} d\tau\right] dt\end{aligned}$$

The inner integral is the Fourier transform of $g(\tau + t)$, which is $g(\tau)$ left-shifted by t. Hence, it is given by $G(f)e^{j2\pi ft}$, in accordance with the time-shifting property in Eq. (3.32a). Therefore,

$$\mathcal{F}[\psi_g(\tau)] = G(f)\int_{-\infty}^{\infty} g(t)e^{j2\pi ft} dt = G(f)G(-f) = |G(f)|^2$$

This completes the proof that

$$\psi_g(\tau) \Longleftrightarrow \Psi_g(f) = |G(f)|^2 \tag{3.74}$$

A careful observation of the operation of correlation shows a close connection to convolution. Indeed, the autocorrelation function $\psi_g(\tau)$ is the convolution of $g(\tau)$ with $g(-\tau)$ because

$$g(\tau) * g(-\tau) = \int_{-\infty}^{\infty} g(x)g[-(\tau - x)]\, dx = \int_{-\infty}^{\infty} g(x)g(x - \tau)\, dx = \psi_g(\tau)$$

Application of the time convolution property [Eq. (3.44)] to this equation yields Eq. (3.74).

ESD of the Input and the Output

If $x(t)$ and $y(t)$ are the input and the corresponding output of a linear time-invariant (LTI) system, then

$$Y(f) = H(f)X(f)$$

Therefore,

$$|Y(f)|^2 = |H(f)|^2|X(f)|^2$$

This shows that

$$\Psi_y(f) = |H(f)|^2\Psi_x(f) \tag{3.75}$$

Thus, the output signal ESD is $|H(f)|^2$ times the input signal ESD.

3.8 SIGNAL POWER AND POWER SPECTRAL DENSITY

For a power signal, a meaningful measure of its size is its power [defined in Eq. (2.4)] as the time average of the signal energy averaged over the infinite time interval. The power P_g of a real-valued signal $g(t)$ is given by

$$P_g = \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{T/2} g^2(t)\,dt \tag{3.76}$$

The signal power and the related concepts can be readily understood by defining a truncated signal $g_T(t)$ as

$$g_T(t) = \begin{cases} g(t) & |t| \le T/2 \\ 0 & |t| > T/2 \end{cases}$$

The truncated signal is shown in Fig. 3.36. The integral on the right-hand side of Eq. (3.76) yields E_{g_T}, which is the energy of the truncated signal $g_T(t)$. Thus,

$$P_g = \lim_{T\to\infty} \frac{E_{g_T}}{T} \tag{3.77}$$

This equation describes the relationship between power and energy of nonperiodic signals. Understanding this relationship will be very helpful in understanding and relating all the power concepts to the energy concepts. Because the signal power is just the time average of energy, all the concepts and results of signal energy also apply to signal power if we modify the concepts properly by taking their time averages.

3.8.1 Power Spectral Density (PSD)

If the signal $g(t)$ is a power signal, then its power is finite, and the truncated signal $g_T(t)$ is an energy signal as long as T is finite. If $g_T(t) \Longleftrightarrow G_T(f)$, then from Parseval's theorem,

$$E_{g_T} = \int_{-\infty}^{\infty} g_T^2(t)\,dt = \int_{-\infty}^{\infty} |G_T(f)|^2\,df$$

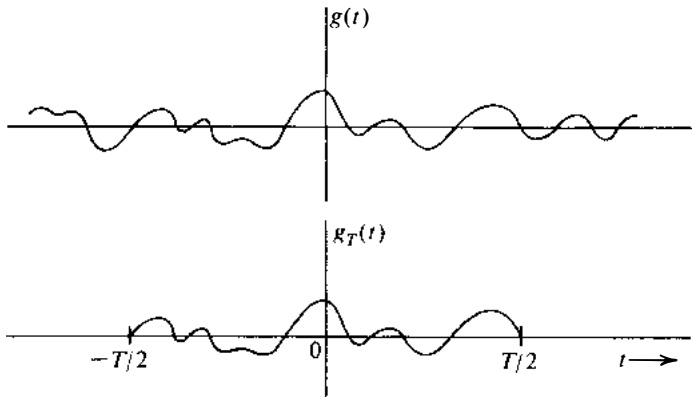


Figure 3.36 Limiting process in derivation of PSD.

Hence, P_g, the power of $g(t)$, is given by

$$P_g = \lim_{T\to\infty} \frac{E_{gT}}{T} = \lim_{T\to\infty} \frac{1}{T}\left[\int_{-\infty}^{\infty} |G_T(f)|^2\, df\right] \tag{3.78}$$

As T increases, the duration of $g_T(t)$ increases, and its energy E_{gT} also increases proportionately. This means that $|G_T(f)|^2$ also increases with T, and as $T \to \infty$, $|G_T(f)|^2$ also approaches ∞. However, $|G_T(f)|^2$ must approach ∞ at the same rate as T because for a power signal, the right-hand side of Eq. (3.78) must converge. This convergence permits us to interchange the order of the limiting process and integration in Eq. (3.78), and we have

$$P_g = \int_{-\infty}^{\infty} \lim_{T\to\infty} \frac{|G_T(f)|^2}{T}\, df \tag{3.79}$$

We define the **power spectral density (PSD)** $S_g(\omega)$ as

$$S_g(f) = \lim_{T\to\infty} \frac{|G_T(f)|^2}{T} \tag{3.80}$$

Consequently,*

$$P_g = \int_{-\infty}^{\infty} S_g(f)\, df \tag{3.81a}$$

$$= 2\int_{0}^{\infty} S_g(f)\, df \tag{3.81b}$$

This result is parallel to the result [Eq. (3.69a)] for energy signals. The power is the area under the PSD. Observe that the PSD is the time average of the ESD of $g_T(t)$ [Eq. (3.80)].

As is the case with ESD, the PSD is also a positive, real, and even function of f. If $g(t)$ is a voltage signal, the units of PSD are volts squared per hertz.

* One should be cautious in using a unilateral expression such as $P_g = 2\int_0^{\infty} S_g(f)\, df$ when $S_g(f)$ contains an impulse at the origin (a dc component). The impulse part should not be multiplied by the factor 2.

3.8.2 Time Autocorrelation Function of Power Signals

The (time) autocorrelation function $\mathcal{R}_g(\tau)$ of a real power signal $g(t)$ is defined as*

$$\mathcal{R}_g(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} g(t)g(t-\tau)\,dt \tag{3.82a}$$

We can use the same argument as that used for energy signals [Eqs. (3.72b) and (3.72c)] to show that $\mathcal{R}_g(\tau)$ is an even function of τ. This means that for a real $g(t)$,

$$\mathcal{R}_g(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} g(t)g(t+\tau)\,dt \tag{3.82b}$$

and

$$\mathcal{R}_g(\tau) = \mathcal{R}_g(-\tau) \tag{3.83}$$

For energy signals, the ESD $\Psi_g(f)$ is the Fourier transform of the autocorrelation function $\psi_g(\tau)$. A similar result applies to power signals. We now show that for a power signal, the PSD $S_g(f)$ is the Fourier transform of the autocorrelation function $\mathcal{R}_g(\tau)$. From Eq. (3.82b) and Fig. 3.36,

$$\mathcal{R}_g(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-\infty}^{\infty} g_T(t)g_T(t+\tau)\,dt = \lim_{T\to\infty} \frac{\psi_{g_T}(\tau)}{T} \tag{3.84}$$

Recall from the Wiener-Khintchine theorem that $\psi_{g_T}(\tau) \Longleftrightarrow |G_T(f)|^2$. Hence, the Fourier transform of the preceding equation yields

$$\mathcal{R}_g(\tau) \Longleftrightarrow \lim_{T\to\infty} \frac{|G_T(f)|^2}{T} = S_g(f) \tag{3.85}$$

Although we have proved these results for a real $g(t)$, Eqs. (3.80), (3.81a), (3.81b), and (3.85) are equally valid for a complex $g(t)$.

The concept and relationships for signal power are parallel to those for signal energy. This is brought out in Table 3.3.

Signal Power Is Its Mean Square Value

A glance at Eq. (3.76) shows that the signal power is the time average or mean of its squared value. In other words P_g is the mean square value of $g(t)$. We must remember, however, that this is a time mean, not a statistical mean (to be discussed in later chapters). Statistical means are denoted by overbars. Thus, the (statistical) mean square of a variable x is denoted by $\overline{x^2}$. To distinguish from this kind of mean, we shall use a wavy overbar to denote a time average. Thus, the time mean square value of $g(t)$ will be denoted by $\overbrace{g^2(t)}$. The time averages are conventionally denoted by angle brackets, written as $\langle g^2(t)\rangle$. We shall, however, use the wavy

* For a complex $g(t)$, we define

$$\mathcal{R}_g(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} g(t)g^*(t-\tau)\,dt = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} g^*(t)g(t+\tau)\,dt$$

TABLE 3.3

$E_g = \int_{-\infty}^{\infty} g^2(t)\,dt$	$P_g = \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{T/2} g^2(t)\,dt = \lim_{T\to\infty} \frac{E_{gT}}{T}$
$\psi_g(\tau) = \int_{-\infty}^{\infty} g(t)g(t+\tau)\,dt$	$\mathcal{R}_g(\tau) = \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{T/2} g(t)g(t+\tau)\,dt = \lim_{T\to\infty} \frac{\psi_{gT}(\tau)}{T}$
$\Psi_g(f) = \lvert G(f)\rvert^2$	$S_g(f) = \lim_{T\to\infty} \frac{\lvert G_T(f)\rvert^2}{T} = \lim_{T\to\infty} \frac{\Psi_{gT}(f)}{T}$
$\psi_g(\tau) \Longleftrightarrow \Psi_g(f)$	$\mathcal{R}_g(\tau) \Longleftrightarrow S_g(f)$
$E_g = \int_{-\infty}^{\infty} \Psi_g(f)\,df$	$P_g = \int_{-\infty}^{\infty} S_g(f)\,df$

overbar notation because it is much easier to associate means with a bar on top than with brackets. Using this notation, we see that

$$P_g = \overbrace{g^2(t)} = \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{T/2} g^2(t)\,dt \tag{3.86a}$$

Note that the rms value of a signal is the square root of its mean square value. Therefore,

$$[g(t)]_{\text{rms}} = \sqrt{P_g} \tag{3.86b}$$

From Eqs. (3.82), it is clear that for a real signal $g(t)$, the time autocorrelation function $\mathcal{R}_g(\tau)$ is the time mean of $g(t)g(t \pm \tau)$. Thus,

$$\mathcal{R}_g(\tau) = \overbrace{g(t)g(t \pm \tau)} \tag{3.87}$$

This discussion also explains why we have been using "time autocorrelation" rather than just "autocorrelation". This is to distinguish clearly the present autocorrelation function (a time average) from the statistical autocorrelation function (a statistical average) to be introduced in Chapter 9 in the context of probability theory and random processes.

Interpretation of Power Spectral Density

Because the PSD is the time average of the ESD of $g(t)$, we can argue along the lines used in the interpretation of ESD. We can readily show that the PSD $S_g(f)$ represents the power per unit bandwidth (in hertz) of the spectral components at the frequency f. The amount of power contributed by the spectral components within the band f_1 to f_2 is given by

$$\Delta P_g = 2\int_{f_1}^{f_2} S_g(f)\,df \tag{3.88}$$

Autocorrelation Method: A Powerful Tool

For a signal $g(t)$, the ESD, which is equal to $|G(f)|^2$, can also be found by taking the Fourier transform of its autocorrelation function. If the Fourier transform of a signal is enough to determine its ESD, then why do we needlessly complicate our lives by talking about autocorrelation functions? The reason for following this alternate route is to lay a foundation for dealing with power signals and random signals. The Fourier transform of a power signal generally does not

exist. Moreover, the luxury of finding the Fourier transform is available only for deterministic signals, which can be described as functions of time. The random message signals that occur in communication problems (e.g., random binary pulse train) cannot be described as functions of time, and it is impossible to find their Fourier transforms. However, the autocorrelation function for such signals can be determined from their statistical information. This allows us to determine the PSD (the spectral information) of such a signal. Indeed, we may consider the autocorrelation approach to be the generalization of Fourier techniques to power signals and random signals. The following example of a random binary pulse train dramatically illustrates the power of this technique.

Example 3.19 Figure 3.37a shows a random binary pulse train $g(t)$. The pulse width is $T_b/2$, and one binary digit is transmitted every T_b seconds. A binary **1** is transmitted by the positive pulse, and a binary **0** is transmitted by the negative pulse. The two symbols are equally likely and occur randomly. We shall determine the autocorrelation function, the PSD, and the essential bandwidth of this signal.

We cannot describe this signal as a function of time because the precise waveform, being random, is not known. We do, however, know its behavior in terms of the averages (the statistical information). The autocorrelation function, being an average parameter (time average) of the signal, is determinable from the given statistical (average) information. We have [Eq. (3.82a)]

$$\mathcal{R}_g(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} g(t)g(t-\tau)\,dt$$

Figure 3.37b shows $g(t)$ by solid lines and $g(t-\tau)$, which is $g(t)$ delayed by τ, by dashed lines. To determine the integrand on the right-hand side of the preceding equation, we multiply $g(t)$ with $g(t-\tau)$, find the area under the product $g(t)g(t-\tau)$, and divide it by the averaging interval T. Let there be N bits (pulses) during this interval T so that $T = NT_b$, and as $T \to \infty$, $N \to \infty$. Thus,

$$\mathcal{R}_g(\tau) = \lim_{N\to\infty} \frac{1}{NT_b} \int_{-NT_b/2}^{NT_b/2} g(t)g(t-\tau)\,dt$$

Let us first consider the case of $\tau < T_b/2$. In this case there is an overlap (shaded region) between each pulse of $g(t)$ and of $g(t-\tau)$. The area under the product $g(t)g(t-\tau)$ is $T_b/2 - \tau$ for each pulse. Since there are N pulses during the averaging interval, the total area under $g(t)g(t-\tau)$ is $N(T_b/2-\tau)$, and

$$\begin{aligned}\mathcal{R}_g(\tau) &= \lim_{N\to\infty} \frac{1}{NT_b}\left[N\left(\frac{T_b}{2}-\tau\right)\right] \\ &= \frac{1}{2}\left(1-\frac{2\tau}{T_b}\right) \qquad \tau < \frac{T_b}{2}\end{aligned}$$

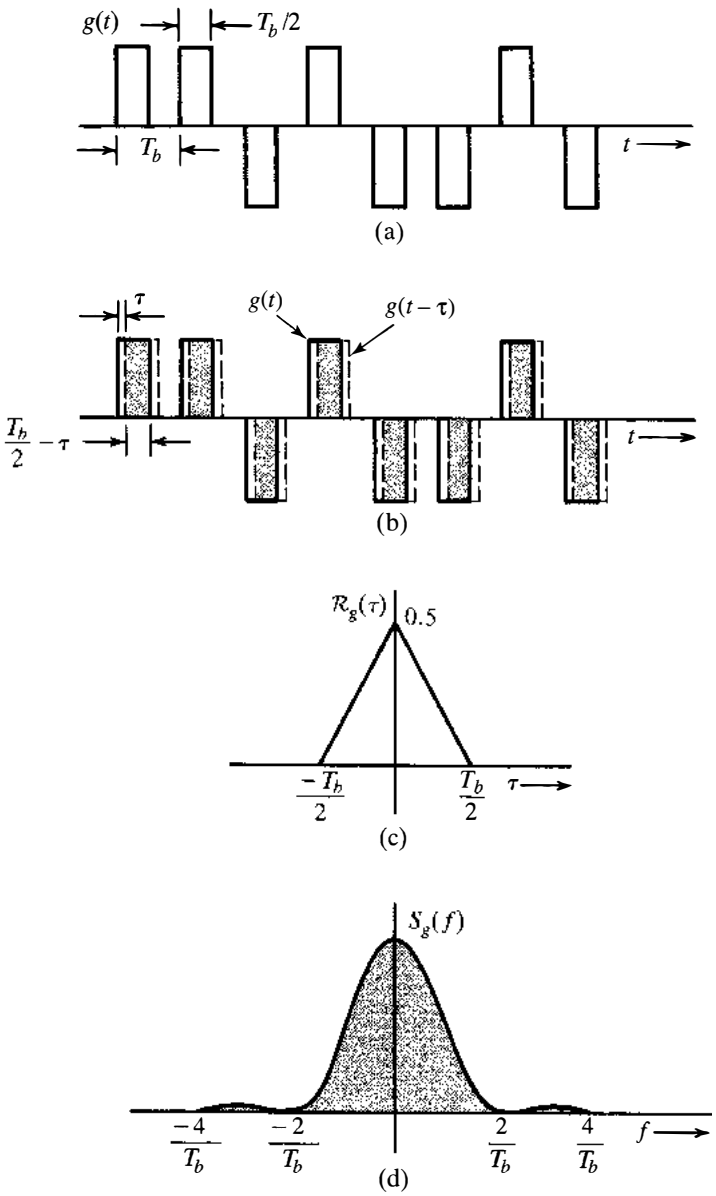


Figure 3.37 Autocorrelation function and power spectral density function of a random binary pulse train.

Because $\mathcal{R}_g(\tau)$ is an even function of τ,

$$\mathcal{R}_g(\tau) = \frac{1}{2}\left(1 - \frac{2|\tau|}{T_b}\right) \qquad |\tau| < \frac{T_b}{2} \tag{3.89a}$$

as shown in Fig. 3.37c.

As we increase τ beyond $T_b/2$, there will be overlap between each pulse and its immediate neighbor. The two overlapping pulses are equally likely to be of the same polarity or of opposite polarity. Their product is equally likely to be 1 or -1 over the overlapping interval. On the average, half the pulse products will be 1 (positive-positive or negative-negative

pulse combinations), and the remaining half pulse products will be -1 (positive-negative or negative-positive combinations). Consequently, the area under $g(t)g(t-\tau)$ will be zero when averaged over an infinitely large time ($T \to \infty$), and

$$\mathcal{R}_g(\tau) = 0 \qquad |\tau| > \frac{T_b}{2} \tag{3.89b}$$

The two parts of Eq. (3.89) show that the autocorrelation function in this case is the triangular function $\frac{1}{2}\Delta(t/T_b)$ shown in Fig. 3.37c. The PSD is the Fourier transform of $\frac{1}{2}\Delta(t/T_b)$, which is found in Example 3.13 (or Table 3.1, pair 19) as

$$S_g(f) = \frac{T_b}{4}\,\text{sinc}^2\left(\frac{\pi f T_b}{2}\right) \tag{3.90}$$

The PSD is the square of the sinc function, as shown in Fig. 3.37d. From the result in Example 3.18, we conclude that 90.28% of the area of this spectrum is contained within the band from 0 to $4\pi/T_b$ rad/s, or from 0 to $2/T_b$ Hz. Thus, the essential bandwidth may be taken as $2/T_b$ Hz (assuming a 90% power criterion). This example illustrates dramatically how the autocorrelation function can be used to obtain the spectral information of a (random) signal when conventional means of obtaining the Fourier spectrum are not usable.

3.8.3 Input and Output Power Spectral Densities

Because the PSD is a time average of ESDs, the relationship between the input and output signal PSDs of a linear time-invariant (LTI) system is similar to that of ESDs. Following the argument used for ESD [Eq. (3.75)], we can readily show that if $g(t)$ and $y(t)$ are the input and output signals of an LTI system with transfer function $H(f)$, then

$$S_y(f) = |H(f)|^2 S_g(f) \tag{3.91}$$

Example 3.20 A noise signal $n_i(t)$ with PSD $S_{n_i}(f) = K$ is applied at the input of an ideal differentiator (Fig. 3.38a). Determine the PSD and the power of the output noise signal $n_o(t)$.

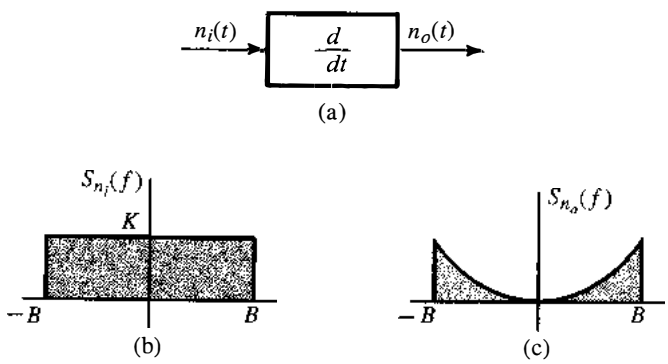


Figure 3.38 Power spectral densities at the input and the output of an ideal differentiator.

The transfer function of an ideal differentiator is $H(f) = j2\pi f$. If the noise at the demodulator output is $n_o(t)$, then from Eq. (3.91),

$$S_{n_o}(f) = |H(f)|^2 S_{n_i}(f) = |j2\pi f|^2 K$$

The output PSD $S_{n_o}(f)$ is parabolic, as shown in Fig. 3.38c. The output noise power N_o is the area under the output PSD. Therefore,

$$N_o = \int_{-B}^{B} K(2\pi f)^2 \, df = 2K \int_0^B (2\pi f)^2 \, df = \frac{8\pi^2 B^3 K}{3}$$

3.8.4 PSD of Modulated Signals

Following the argument in deriving Eqs. (3.70) and (3.71) for energy signals, we can derive similar results for power signals by taking the time averages. We can show that for a power signal $g(t)$, if

$$\varphi(t) = g(t) \cos 2\pi f_0 t$$

then the PSD $S_\varphi(f)$ of the modulated signal $\varphi(t)$ is given by

$$S_\varphi(f) = \frac{1}{4}\left[S_g(f + f_0) + S_g(f - f_0)\right] \tag{3.92}$$

The detailed derivation is provided in Sec. 7.8. Thus, modulation shifts the PSD of $g(t)$ by $\pm f_0$. The power of $\varphi(t)$ is half the power of $g(t)$, that is,

$$P_\varphi = \frac{1}{2} P_g \qquad f_0 \geq B \tag{3.93}$$

3.9 NUMERICAL COMPUTATION OF FOURIER TRANSFORM: THE DFT

To compute $G(f)$, the Fourier transform of $g(t)$, numerically, we have to use the samples of $g(t)$. Moreover, we can determine $G(f)$ only at some finite number of frequencies. Thus, we can compute only samples of $G(f)$. For this reason, we shall now find the relationships between samples of $g(t)$ and samples of $G(f)$.

In numerical computations, the data must be finite. This means that the number of samples of $g(t)$ and $G(f)$ must be finite. In other words, we must deal with time-limited signals. If the signal is not time-limited, then we need to truncate it to make its duration finite. The same is true of $G(f)$. To begin, let us consider a signal $g(t)$ of duration τ seconds, starting at $t = 0$, as shown in Fig. 3.39a. However, for reasons that will become clear as we go along, we shall consider the duration of $g(t)$ to be T_0, where $T_0 \geq \tau$, which makes $g(t) = 0$ in the interval $\tau < t \leq T_0$, as shown in Fig. 3.39a. Clearly, this makes no difference in the computation of $G(f)$. Let us take samples of $g(t)$ at uniform intervals of T_s seconds. There are a total of N_0 samples, where

$$N_0 = \frac{T_0}{T_s} \tag{3.94}$$

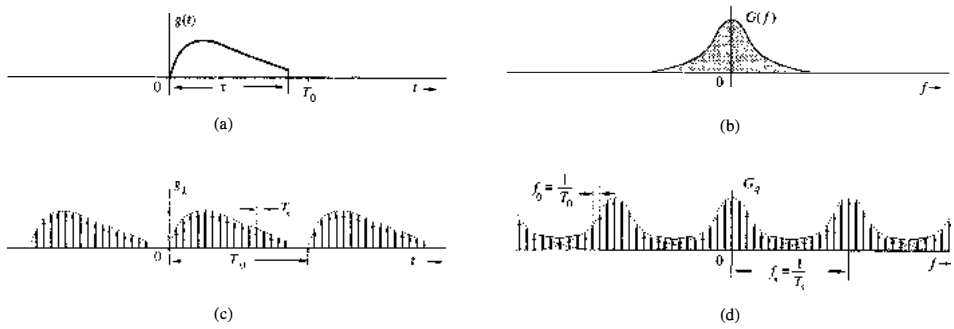


Figure 3.39 Relationship between samples of $g(t)$ and $G(f)$.

Now,*

$$G(f) = \int_0^{T_0} g(t)e^{-j2\pi ft}\,dt$$

$$= \lim_{T_s \to 0} \sum_{k=0}^{N_0-1} g(kT_s)e^{-j2\pi fkT_s}\,T_s \tag{3.95}$$

Let us consider the samples of $G(f)$ at uniform intervals of f_0. If G_q is the q-th sample, that is, $G_q = G(qf_0)$, then from Eq. (3.95), we obtain

$$G_q = \sum_{k=0}^{N_0-1} T_s g(kT_s)e^{-jq2\pi f_0 T_s k}$$

$$= \sum_{k=0}^{N_0-1} g_k e^{-jq\Omega_0 k} \tag{3.96}$$

where

$$g_k = T_s g(kT_s), \qquad G_q = G(qf_0), \qquad \Omega_0 = 2\pi f_0 T_s \tag{3.97}$$

Thus, Eq. (3.96) relates the samples of $g(t)$ to the samples of $G(f)$. In this derivation, we have assumed that $T_s \to 0$. In practice, it is not possible to make $T_s \to 0$ because this would increase the data enormously. We strive to make T_s as small as is practicable. This will result in some computational error.

We make an interesting observation from Eq. (3.96). The samples G_q are periodic with a period of $2\pi/\Omega_0$ samples. This follows from Eq. (3.96), which shows that $G_{(q+2\pi/\Omega_0)} = G_q$. Thus, only $2\pi/\Omega_0$ samples G_q can be independent. Equation (3.96) shows that G_q is determined by N_0 independent values g_k. Hence, for unique inverses of these equations, there can be only N_0 independent sample values G_q. This means that

$$N_0 = \frac{2\pi}{\Omega_0} = \frac{2\pi}{2\pi f_0 T_s} = \frac{2\pi N_0}{2\pi f_0 T_0} \tag{3.98}$$

In other words, we have

$$2\pi f_0 = \frac{2\pi}{T_0} \qquad \text{and} \qquad f_0 = \frac{1}{T_0} \tag{3.99}$$

* The upper limit on the summation in Eq. (3.95) is $N_0 - 1$ (not N_0) because the last term in the sum starts at $(N_0 - 1)T_s$ and covers the area under the summand up to $N_0 T_s = T_0$.

Thus, the spectral sampling interval f_0 Hz can be adjusted by a proper choice of T_0: the larger the T_0, the smaller the f_0. The wisdom of selecting $T_0 \geq \tau$ is now clear. When T_0 is greater than τ, we shall have several zero-valued samples g_k in the interval from τ to T_0. Thus, by increasing the number of zero-valued samples of g_k, we reduce f_0 [more closely spaced samples of $G(f)$], yielding more details of $G(f)$. This process of reducing f_0 by the inclusion of zero-valued samples g_k is known as **zero padding**. Also, for a given sampling interval T_s, larger T_0 implies larger N_0. Thus, by selecting a suitably large value of N_0, we can obtain samples of $G(f)$ as close as possible.

To find the inverse relationship, we multiply both sides of Eq. (3.96) by $e^{jm\Omega_0 q}$ and sum over q as

$$\sum_{q=0}^{N_0-1} G_q e^{jm\Omega_0 q} = \sum_{q=0}^{N_0-1} \left[\sum_{k=0}^{N_0-1} g_k e^{-jq\Omega_0 k} \right] e^{jm\Omega_0 q}$$

Upon interchanging the order of summation on the right-hand side,

$$\sum_{q=0}^{N_0-1} G_q e^{jm\Omega_0 q} = \sum_{k=0}^{N_0-1} g_k \left[\sum_{q=0}^{N_0-1} e^{j(m-k)\Omega_0 q} \right] \tag{3.100}$$

To find the inner sum on the right-hand side, we shall now show that

$$\sum_{k=0}^{N_0-1} e^{jn\Omega_0 k} = \begin{cases} N_0 & n = 0, \pm N_0, \pm 2N_0, \ldots \\ 0 & \text{otherwise} \end{cases} \tag{3.101}$$

To show this, recall that $\Omega_0 N_0 = 2\pi$ and $e^{jn\Omega_0 k} = 1$ for $n = 0, \pm N_0, \pm 2N_0, \ldots$, so that

$$\sum_{k=0}^{N_0-1} e^{jn\Omega_0 k} = \sum_{k=0}^{N_0-1} 1 = N_0 \qquad n = 0, \pm N_0, \pm 2N_0, \ldots$$

To compute the sum for other values of n, we note that the sum on the left-hand side of Eq. (3.101) is a geometric series with common ratio $\alpha = e^{jn\Omega_0}$. Therefore, its partial sum of the first N_0 terms is

$$\sum_{k=0}^{N_0-1} e^{jn\Omega_0 k} = \frac{e^{jn\Omega_0 N_0} - 1}{e^{jn\Omega_0} - 1} = 0,$$

where $\qquad e^{jn\Omega_0 N_0} = e^{j2\pi n} = 1$

This proves Eq. (3.101).

It now follows that the inner sum on the right-hand side of Eq. (3.100) is zero for $k \neq m$, and the sum is N_0 when $k = m$. Therefore, the outer sum will have only one nonzero term when $k = m$, and it is $N_0 g_k = N_0 g_m$. Therefore,

$$g_m = \frac{1}{N_0} \sum_{q=0}^{N_0-1} G_q e^{jm\Omega_0 q} \qquad \Omega_0 = \frac{2\pi}{N_0} \tag{3.102}$$

Equation (3.102) reveals the interesting fact that $g_{(m+N_0)} = g_m$. This means that the sequence g_k is also periodic with a period of N_0 samples (representing the time duration $N_0T_s = T_0$ seconds). Moreover, G_q is also periodic with a period of N_0 samples, representing a frequency interval $N_0f_0 = (T_0/T_s)(T_0) = 1/T_s = f_s$ hertz. But $1/T_s$ is the number of samples of $g(t)$ per second. Thus, $1/T_s = f_s$ is the sampling frequency (in hertz) of $g(t)$. This means that G_q is N_0-periodic, repeating every f_s Hz. Let us summarize the results derived so far. We have proved the discrete Fourier transform (DFT) pair

$$G_q = \sum_{k=0}^{N_0-1} g_k e^{-jq\Omega_0 k} \tag{3.103a}$$

$$g_k = \frac{1}{N_0} \sum_{q=0}^{N_0-1} G_q e^{jk\Omega_0 q} \tag{3.103b}$$

where

$$\begin{aligned} g_k &= T_s g(kT_s) & G_q &= G(qf_0) \\ 2\pi f_0 &= \frac{2\pi}{T_0} & 2\pi f_s &= \frac{2\pi}{T_s} \\ N_0 &= \frac{T_0}{T_s} = \frac{f_s}{f_0} & \Omega_0 &= 2\pi f_0 T_s = \frac{2\pi}{N_0} \end{aligned} \tag{3.104}$$

Both the sequences g_k and G_q are periodic with a period of N_0 samples. This results in g_k repeating with period T_0 seconds and G_q repeating with period $f_s = 1/T_s$ rad/s, or $f_s = 1/T_s$ Hz (the sampling frequency). The sampling interval of g_k is T_s seconds and the sampling interval of G_q is $f_0 = 1/T_0$ Hz. This is shown in Fig. 3.39c and d. For convenience, we have used the frequency variable f (in hertz) rather than ω (in radians per second).

We have assumed $g(t)$ to be time-limited to τ seconds. This makes $G(f)$ non-band-limited.* Hence, the periodic repetition of the spectra G_q, as shown in Fig. 3.39d, will cause overlapping of spectral components, resulting in error. The nature of this error, known as **aliasing error**, is explained in more detail in Chapter 6. The spectrum G_q repeats every f_s Hz. The aliasing error is reduced by increasing f_s, the repetition frequency (see Fig. 3.39d). To summarize, the computation of G_q using DFT has aliasing error when $g(t)$ is time-limited. This error can be made as small as desired by increasing the sampling frequency $f_s = 1/T_s$ (or reducing the sampling interval T_s). The aliasing error is the direct result of the nonfulfillment of the requirement $T_s \to 0$ in Eq. (3.95).

When $g(t)$ is not time-limited, we need to truncate it to make it time-limited. This will cause further error in G_q. This error can be reduced as much as desired by appropriately increasing the truncating interval T_0.†

In computing the inverse Fourier transform [by using the inverse DFT in Eq. (3.103b)], we have similar problems. If $G(f)$ is band-limited, $g(t)$ is not time-limited, and the periodic repetition of samples g_k will overlap (aliasing in the time domain). We can reduce the aliasing error by increasing T_0, the period of g_k (in seconds). This is equivalent to reducing the frequency

---

* We can show that a signal cannot be simultaneously time-limited and band-limited. If it is one, it cannot be the other, and vice versa.[3]

† The DFT relationships represent a transform in their own right, and they are exact. If, however, we identify g_k and G_q as the samples of a signal $g(t)$ and its Fourier transform $G(f)$, respectively, then the DFT relationships are approximations because of the aliasing and truncating effects.

sampling interval $f_0 = 1/T_0$ of $G(f)$. Moreover, if $G(f)$ is not band-limited, we need to truncate it. This will cause an additional error in the computation of g_k. By increasing the truncation bandwidth, we can reduce this error. In practice, (tapered) window functions are often used for truncation[5] in order to reduce the severity of some problems caused by straight truncation (also known as rectangular windowing).

Because G_q is N_0-periodic, we need to determine the values of G_q over any one period. It is customary to determine G_q over the range $(0, N_0 - 1)$ rather than over the range $(-N_0/2, N_0/2 - 1)$. The identical remark applies to g_k.

Choice of T_s, T_0, and N_0

In DFT computation, we first need to select suitable values for N_0, T_s, and T_0. For this purpose we should first decide on B, the essential bandwidth of $g(t)$. From Fig. 3.39d, it is clear that the spectral overlapping (aliasing) occurs at the frequency $f_s/2$ Hz. This spectral overlapping may also be viewed as the spectrum beyond $f_s/2$ folding back at $f_s/2$. Hence, this frequency is also called the **folding frequency**. If the folding frequency is chosen such that the spectrum $G(f)$ is negligible beyond the folding frequency, aliasing (the spectral overlapping) is not significant. Hence, the folding frequency should at least be equal to the highest significant frequency, that is, the frequency beyond which $G(f)$ is negligible. We shall call this frequency the **essential bandwidth** B (in hertz). If $g(t)$ is band-limited, then clearly, its bandwidth is identical to the essential bandwidth. Thus,

$$\frac{f_s}{2} \geq B \quad \text{Hz} \tag{3.105a}$$

Moreover, the sampling interval $T_s = 1/f_s$ [Eq. (3.104)]. Hence,

$$T_s \leq \frac{1}{2B} \tag{3.105b}$$

Once we pick B, we can choose T_s according to Eq. (3.105b). Also,

$$f_0 = \frac{1}{T_0} \tag{3.106}$$

where f_0 is the **frequency resolution** [separation between samples of $G(f)$]. Hence, if f_0 is given, we can pick T_0 according to Eq. (3.106). Knowing T_0 and T_s, we determine N_0 from

$$N_0 = \frac{T_0}{T_s} \tag{3.107}$$

In general, if the signal is time-limited, $G(f)$ is not band-limited, and there is aliasing in the computation of G_q. To reduce the aliasing effect, we need to increase the folding frequency; that is, we must reduce T_s (the sampling interval) as much as is practicable. If the signal is band-limited, $g(t)$ is not time-limited, and there is aliasing (overlapping) in the computation of g_k. To reduce this aliasing, we need to increase T_0, the period of g_k. This results in reducing the frequency sampling interval f_0 (in hertz). In either case (reducing T_s in the time-limited case or increasing T_0 in the band-limited case), for higher accuracy, we need to increase the number of samples N_0 because $N_0 = T_0/T_s$. There are also signals that are neither time-limited nor band-limited. In such cases, we need to reduce T_s and increase T_0.

Points of Discontinuity

If $g(t)$ has a jump discontinuity at a sampling point, the sample value should be taken as the average of the values on the two sides of the discontinuity because the Fourier representation at a point of discontinuity converges to the average value.

Using the FFT Algorithm in DFT Computations

The number of computations required in performing the DFT was dramatically reduced by an algorithm developed by Tukey and Cooley in 1965.[6] This algorithm, known as the **fast Fourier transform (FFT)**, reduces the number of computations from something on the order of N_0^2 to $N_0 \log N_0$. To compute one sample G_r from Eq. (3.103a), we require N_0 complex multiplications and $N_0 - 1$ complex additions. To compute N_0 values of G_r ($r = 0, 1, \ldots, N_0 - 1$), we require a total of N_0^2 complex multiplications and $N_0(N_0 - 1)$ complex additions. For large N_0, this can be prohibitively time-consuming, even for a very high-speed computer. The FFT is, thus, a lifesaver in signal processing applications. The FFT algorithm is simplified if we choose N_0 to be a power of 2, although this is not necessary, in general. Details of the FFT can be found in any book on signal processing (e.g., Ref. 3).

3.10 MATLAB EXERCISES

Computing Fourier Transforms

In this section of computer exercises, let us consider two examples illustrating the use of DFT in finding the Fourier transform. We shall use MATLAB to find DFT by the FFT algorithm. In the first example, the signal $g(t) = e^{-2t}u(t)$ starts at $t = 0$. In the second example, we use $g(t) = \Pi(t)$, which starts at $t = -\frac{1}{2}$.

COMPUTER EXAMPLE C3.1

Use DFT (implemented by the FFT algorithm) to compute the Fourier transform of $e^{-2t}u(t)$. Plot the resulting Fourier spectra.

We first determine T_s and T_0. The Fourier transform of $e^{-2t}u(t)$ is $1/(j2\pi f + 2)$. This low-pass signal is not band-limited. Let us take its essential bandwidth to be that frequency where $|G(f)|$ becomes 1% of its peak value, which occurs at $f = 0$. Observe that

$$|G(f)| = \frac{1}{\sqrt{(2\pi f)^2 + 4}} \approx \frac{1}{2\pi f} \qquad 2\pi f \gg 2$$

Also, the peak of $|G(f)|$ is at $f = 0$, where $|G(0)| = 0.5$. Hence, the essential bandwidth B is at $f = B$, where

$$|G(f)| \approx \frac{1}{2\pi B} = 0.5 \times 0.01 \quad \Rightarrow B = \frac{100}{\pi} \text{ Hz}$$

and from Eq. (3.105b),

$$T_s \leq \frac{1}{2B} = 0.005\pi = 0.0157$$

Let us round this value down to $T_s = 0.015625$ second so that we have 64 samples per second. The second issue is to determine T_0. The signal is not time-limited. We need to truncate it at T_0 such that $g(T_0) \ll 1$. We shall pick $T_0 = 4$ (eight time constants of the signal), which yields $N_0 = T_0/T_s = 256$. This is a power of 2. Note that there is a great deal of flexibility in determining T_s and T_0, depending on the accuracy desired and the computational capacity available. We could just as well have picked $T_0 = 8$ and $T_s = 1/32$, yielding $N_0 = 256$, although this would have given a slightly higher aliasing error.

Because the signal has a jump discontinuity at $t = 0$, the first sample (at $t = 0$) is 0.5, the averages of the values on the two sides of the discontinuity. The MATLAB program, which implements the DFT by using the FFT algorithm is as follows:

```
Ts=1/64; T0=4; N0=T0/Ts;
t=0:Ts:Ts*(N0-1);t=t';
g=Ts*exp(-2*t);
g(1)=Ts*0.5;
G=fft(g);
$[Gp,Gm]$=cart2pol($real(G),imag(G)$);
k=0:N0-1; k=k';
w=2*pi*k/T0;
subplot(211),stem(w(1:32),Gm(1:32));
subplot(212),stem(w(1:32),Gp(1:32))
```

Because G_q is N_0-periodic, $G_q = G_{(q+256)}$ so that $G_{256} = G_0$. Hence, we need to plot G_q over the range $q = 0$ to 255 (not 256). Moreover, because of this periodicity, $G_{-q} = G_{(-q+256)}$, and the G_q over the range of $q = -127$ to -1 are identical to the G_q over the range of $q = 129$ to 255. Thus, $G_{-127} = G_{129}$, $G_{-126} = G_{130}$, ..., $G_{-1} = G_{255}$. In addition, because of the property of conjugate symmetry of the Fourier transform, $G_{-q} = G_q^*$, it follows that $G_{129} = G_{127}^*$, $G_{130} = G_{126}^*$, ..., $G_{255} = G_1^*$. Thus, the plots beyond $q = N_0/2$ (128 in this case) are not necessary for real signals (because they are conjugates of G_q for $q = 0$ to 128).

The plot of the Fourier spectra in Fig. 3.40 shows the samples of magnitude and phase of $G(f)$ at the intervals of $1/T_0 = 1/4$ Hz or $\omega_0 = 1.5708$ rad/s. In Fig. 3.40, we have shown only the first 28 points (rather than all 128 points) to avoid too much crowding of the data.

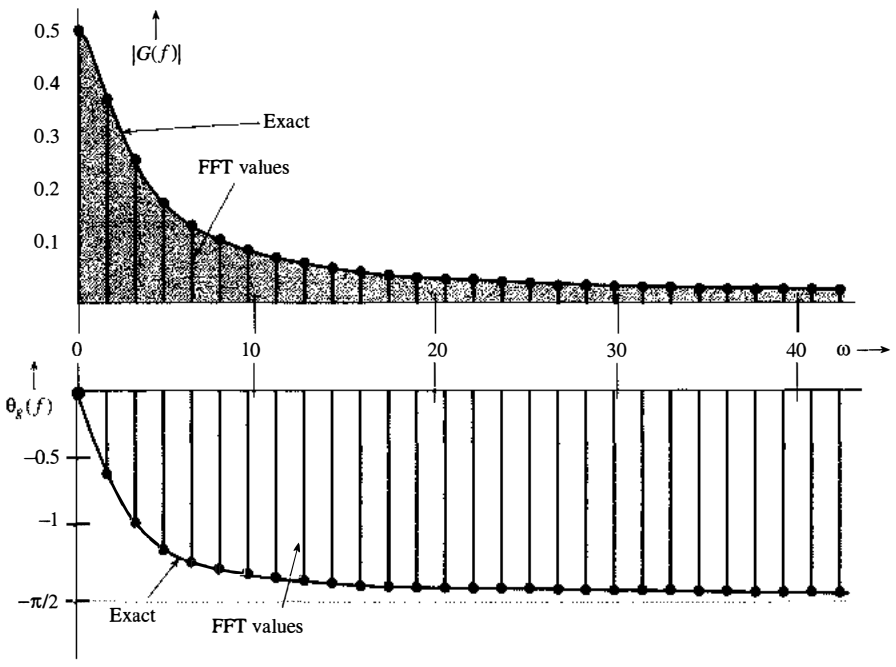


Figure 3.40 Discrete Fourier transform of an exponential signal $e^{-2t}u(t)$. Notice that the horizontal axis in this case is ω (in radians per second).

In this example, we knew $G(f)$ beforehand and hence could make INTELLIGENT choices for B (or the sampling frequency f_s). In practice, we generally do not know $G(f)$ beforehand. In fact, that is the very thing we are trying to determine. In such a case, we must make an intelligent guess for B or f_s from circumstantial evidence. We then continue reducing the value of T_s and recomputing the transform until the result stabilizes within the desired number of significant digits.

Next, we compute the Fourier transform of $g(t) = 8\,\Pi(t)$.

COMPUTER EXAMPLE C3.2

Use DFT (implemented by the FFT algorithm) to compute the Fourier transform of $8\,\Pi(t)$. Plot the resulting Fourier spectra.

This rectangular function and its Fourier transform are shown in Fig. 3.41a and b. To determine the value of the sampling interval T_s, we must first decide on the essential bandwidth B. From Fig. 3.41b, we see that $G(f)$ decays rather slowly with f. Hence, the essential bandwidth B is rather large. For instance, at $B = 15.5$ Hz (97.39 rad/s), $G(f) = -0.1643$, which is about 2% of the peak at $G(0)$. Hence, the essential bandwidth may be taken as 16 Hz. However, we shall deliberately take $B = 4$ for two reasons: (1) to show the effect of aliasing and (2) because the use of $B > 4$ will give an enormous number of samples, which cannot be conveniently displayed on a book-sized page without losing sight of the essentials. Thus, we shall intentionally accept approximation for the sake of clarifying the concepts of DFT graphically.

The choice of $B = 4$ results in the sampling interval $T_s = 1/2B = 1/8$. Looking again at the spectrum in Fig. 3.41b, we see that the choice of the frequency resolution $f_0 = 1/4$ Hz is reasonable. This will give four samples in each lobe of $G(f)$. In this case $T_0 = 1/f_0 = 4$ seconds and $N_0 = T_0/T = 32$. The duration of $g(t)$ is only 1 second. We must repeat it every 4 seconds ($T_0 = 4$), as shown in Fig. 3.41c, and take samples every 0.125 second. This gives us 32 samples ($N_0 = 32$). Also,

$$\begin{aligned} g_k &= T_s g(kT) \\ &= \frac{1}{8} g(kT) \end{aligned}$$

Since $g(t) = 8\,\Pi(t)$, the values of g_k are 1, 0, or 0.5 (at the points of discontinuity), as shown in Fig. 3.41c, where for convenience, g_k is shown as a function of t as well as k.

In the derivation of the DFT, we assumed that $g(t)$ begins at $t = 0$ (Fig. 3.39a), and then took N_0 samples over the interval $(0, T_0)$. In the present case, however, $g(t)$ begins at $-\frac{1}{2}$. This difficulty is easily resolved when we realize that the DFT found by this procedure is actually the DFT of g_k repeating periodically every T_0 seconds. From Fig. 3.41c, it is clear that repeating the segment of g_k over the interval from -2 to 2 seconds periodically is identical to repeating the segment of g_k over the interval from 0 to 4 seconds. Hence, the DFT of the samples taken from -2 to 2 seconds is the same as that of the samples taken from 0 to 4 seconds. Therefore, regardless of where $g(t)$ starts, we can always take the samples of $g(t)$ and its periodic extension over the interval from 0 to T_0. In the present

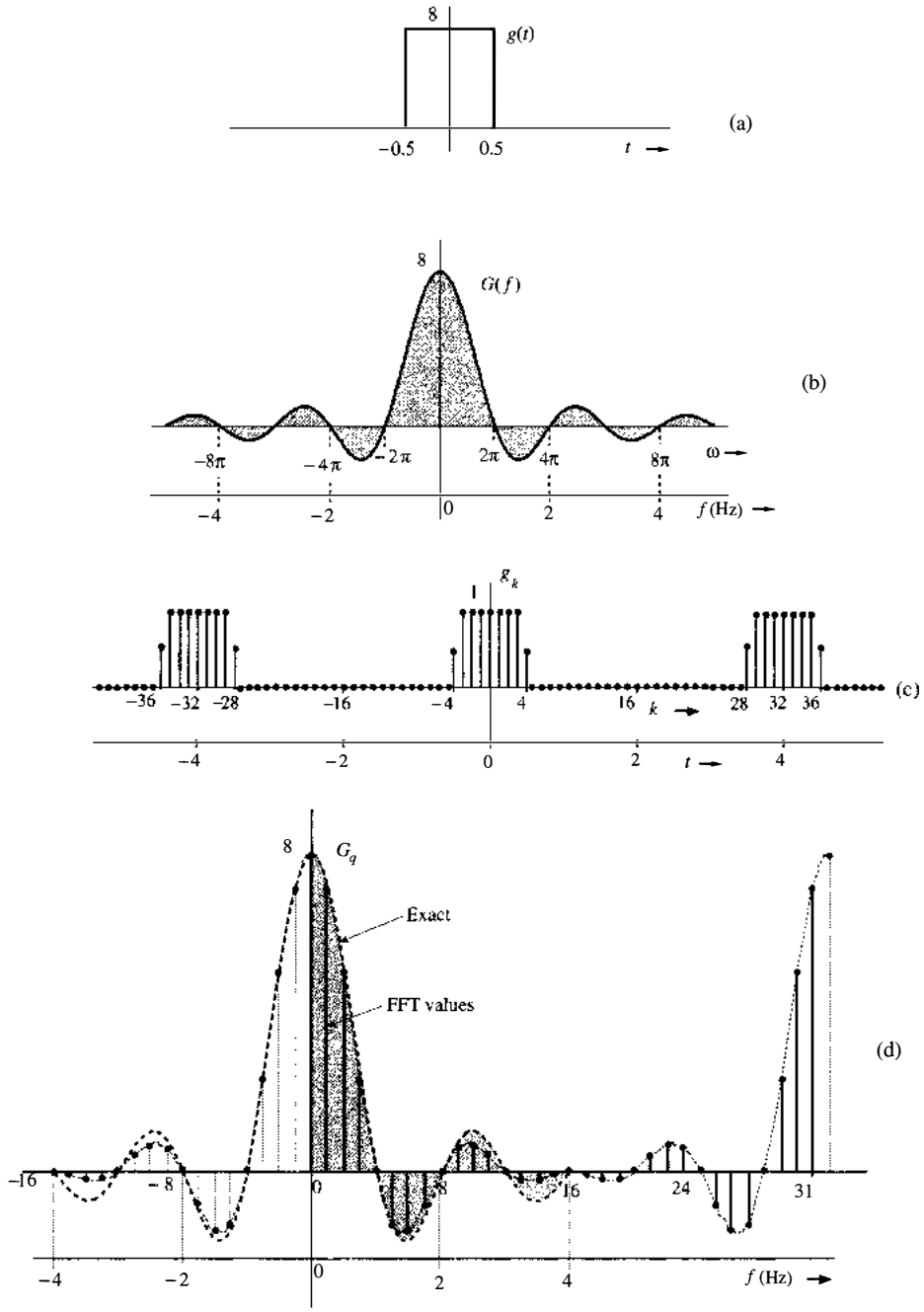


Figure 3.41 Discrete Fourier transform of a rectangular pulse.

example, the 32 sample values are

$$g_k = \begin{cases} 1 & 0 \le k \le 3 \quad \text{and} \quad 29 \le k \le 31 \\ 0 & 5 \le k \le 27 \\ 0.5 & k = 4, 28 \end{cases}$$

Observe that the last sample is at $t = 31/8$, not at 4, because the signal repetition starts at $t = 4$, and the sample at $t = 4$ is the same as the sample at $t = 0$. Now, $N_0 = 32$ and $\Omega_0 = 2\pi/32 = \pi/16$. Therefore [see Eq. (3.103a)],

$$G_q = \sum_{k=0}^{31} g_k e^{-jq\frac{\pi}{16}k}$$

The MATLAB program, which uses the FFT algorithm to implement this DFT equation, is given next. First we write a MATLAB program to generate 32 samples of g_k, and then we compute the DFT.

```
% (c32.m)
B=4;     f0=1/4;
Ts=1/(2*B); T0=1/f0;
N0=T0/Ts;
k=0:N0; k=k';
for m=1:length(k)
$  $ if k(m)$>$=0 & k(m)$<$=3, gk(m)=1; end
$  $ if k(m)==4 & k(m)==28 gk(m)=0.5; end
$  $ if k(m)$>$=5 & k(m)$<$=27, gk(m)=0; end
$  $ if k(m)$>$=29 & k(m)$<$=31, gk(m)=1; end
end
gk=gk';
Gr=fft(gk);
subplot(211),stem(k,gk)
subplot(212),stem(k,Gr)
```

Figure 3.41d shows the plot of G_q.

The samples G_q are separated by $f_0 = 1/T_0$ Hz. In this case $T_0 = 4$, so the frequency resolution f_0 is $\frac{1}{4}$ Hz, as desired. The folding frequency $f_s/2 = B = 4$ Hz corresponds to $q = N_0/2 = 16$. Because G_q is N_0-periodic ($N_0 = 32$), the values of G_q for $q = -16$ to -1 are the same as those for $q = 16$ to 31. The DFT gives us the samples of the spectrum $G(f)$.

For the sake of comparison, Fig. 3.41d also shows the shaded curve $8\,\mathrm{sinc}(\pi f)$, which is the Fourier transform of $8\,\Pi(t)$. The values of G_q computed from the DFT equation show aliasing error, which is clearly seen by comparing the two superimposed plots. The error in G_2 is just about 1.3%. However, the aliasing error increases rapidly with r. For instance, the error in G_6 is about 12%, and the error in G_{10} is 33%. The error in G_{14} is a whopping 72%. The percent error increases rapidly near the folding frequency ($r = 16$) because $g(t)$ has a jump discontinuity, which makes $G(f)$ decay slowly as $1/f$. Hence, near the folding frequency, the inverted tail (due to aliasing) is very nearly equal to $G(f)$ itself. Moreover, the final values are the difference between the exact and the folded values (which are very close to the exact values). Hence, the percent error near the folding frequency ($r = 16$ in this case) is very high, although the absolute error is very small. Clearly, for signals with jump discontinuities, the aliasing error near the folding frequency will always be high (in percentage terms), regardless of the choice of N_0. To ensure a negligible aliasing error at any value q, we must make sure that $N_0 \gg q$. This observation is valid for all signals with jump discontinuities.

Filtering

We generally think of filtering in terms of a hardware-oriented solution (namely, building a circuit with *RLC* components and operational amplifiers). However, filtering also has a software-oriented solution [a computer algorithm that yields the filtered output $y(t)$ for a given input $g(t)$]. This can be conveniently accomplished by using the DFT. If $g(t)$ is the signal to be filtered, then G_q, the DFT of g_k, is found. The spectrum G_q is then shaped (filtered) as desired by multiplying G_q by H_q, where H_q are the samples of the filter transfer function $H(f)$ [$H_q = H(qf_0)$]. Finally, we take the inverse DFT or (IDFT) of G_qH_q to obtain the filtered output y_k [$y_k = T_s y(kT)$]. This procedure is demonstrated in the following example.

COMPUTER EXAMPLE C3.3

The signal $g(t)$ in Fig. 3.42a is passed through an ideal low-pass filter of transfer function $H(f)$, shown in Fig. 3.42b. Use DFT to find the filter output.

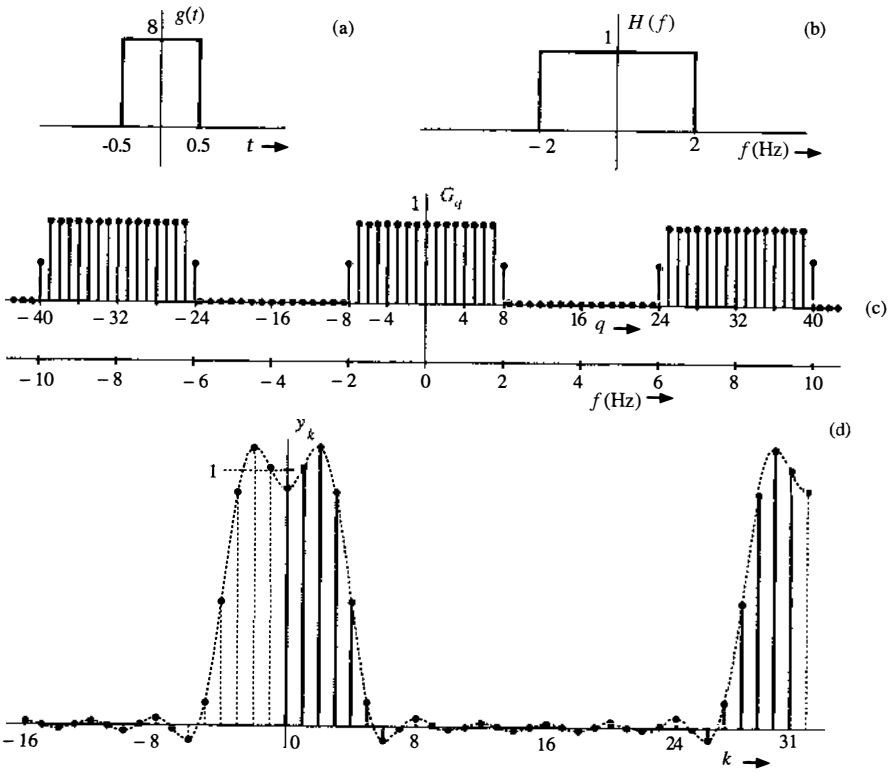


Figure 3.42 Filtering $g(t)$ through $H(f)$.

We have already found the 32-point DFT of $g(t)$ (see Fig. 3.41d). Next we multiply G_q by H_q. To compute H_q, we remember that in computing the 32-point DFT of $g(t)$, we have used $f_0 = 0.25$. Because G_q is 32-periodic, H_q must also be 32-periodic with samples separated by 0.25 Hz. This means that H_q must be repeated every 8 Hz or 16π rad/s (see Fig. 3.42c). This gives the 32 samples of H_q over $0 \le f \le 8$ as follows:

$$H_q = \begin{cases} 1 & 0 \le q \le 7 \quad \text{and} \quad 25 \le q \le 31 \\ 0 & 9 \le q \le 23 \\ 0.5 & q = 8, 24 \end{cases}$$

We multiply G_q by H_q and take the inverse DFT. The resulting output signal is shown in Fig. 3.42d. Table 3.4 gives a printout of g_k, G_q, H_q, Y_q, and y_k.

We have already found the 32-point DFT (G_q) of $g(t)$ in Example C3.2. The MATLAB program of Example C3.2 should be saved as an m-file (e.g., "c32.m"). We can import G_q in the MATLAB environment by the command "c32". Next, we generate 32-point samples of H_q, multiply G_q by H_q, and take the inverse DFT to compute y_k. We can also find y_k by convolving g_k with h_k.

```
c32;
q=0:32; q=q';
for m=1:length(q)
     if q(m)$>$=0 & q(m)$<$=7, Hq(m)=1; end
     if q(m)$>$=25 & q(m)$<$=31, Hq(m)=1; end
     if q(m)$>$=9 & q(m)$<$=23, Hq(m)=0; end
     if q(m)==8 & q(m)==24, Hq(m)=0.5; end
```

TABLE 3.4

No.	gk	G_q	H_q	G_qH_q	y_k
0	1	8,000	1	8.000	0.9285
1	1	7,179	1	7.179	1.009
2	1	5,027	1	5.027	1.090
3	1	2,331	1	2.331	0.9123
4	1	0.000	1	0.000	0.4847
5	0.5	−1.323	1	−1.323	0.08884
6	0	−1.497	1	−1.497	−0.05698
7	0	−0.8616	1	−0.8616	−0.01383
8	0	0.000	0.5	0.000	0.02933
9	0	0.5803	0	0.000	0.004837
10	0	0.6682	0	0.000	−0.01966
11	0	0.3778	0	0.000	−0.002156
12	0	0.000	0	0.000	0.01534
13	0	−0.2145	0	0.000	0.0009828
14	0	−0.1989	0	0.000	−0.01338
15	0	−0.06964	0	0.000	−0.0002876
16	0	0.000	0	0.000	0.01280
17	0	−0.06964	0	0.000	−0.0002876
18	0	−0.1989	0	0.000	−0.01338
19	0	−0.2145	0	0.000	0.0009828
20	0	0.000	0	0.000	0.01534
21	0	0.3778	0	0.000	−0.002156
22	0	0.6682	0	0.000	−0.01966
23	0	0.5803	0	0.000	0.004837
24	0	0.000	0.5	0.000	0.03933
25	0	−0.8616	1	−0.8616	−0.01383
26	0	−1.497	1	−1.497	−0.05698
27	0	−1.323	1	−1.323	0.08884
28	0.5	0.000	1	0.000	0.4847
29	1	2.331	1	2.331	0.9123
30	1	5.027	1	5.027	1.090
31	1	7.179	1	7.179	1.009

```
end
Hq=Hq';
Yq=Gq.*Hq;
yk=ifft(Yq);
clf,stem(k,yk)
```

REFERENCES

1. R. V. Churchill and J. W. Brown, *Fourier Series and Boundary Value Problems*, 3rd ed., McGraw-Hill, New York, 1978.
2. R. N. Bracewell, *Fourier Transform and Its Applications*, rev. 2nd ed., McGraw-Hill, New York, 1986.
3. B. P. Lathi, *Signal Processing and Linear Systems*, Oxford University Press, 2000.
4. E. A. Guillemin, *Theory of Linear Physical Systems*, Wiley, New York, 1963.
5. F. J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," *Proc. IEEE*, vol. 66, pp. 51–83, Jan. 1978.
6. J. W. Tukey and J. Cooley, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, Vol. 19, pp. 297–301, April 1965.

PROBLEMS

3.1-1 Show that the Fourier transform of $g(t)$ may be expressed as

$$G(f) = \int_{-\infty}^{\infty} g(t)\cos 2\pi ft\, dt - j\int_{-\infty}^{\infty} g(t)\sin 2\pi ft\, dt$$

Hence, show that if $g(t)$ is an even function of t, then

$$G(f) = 2\int_{0}^{\infty} g(t)\cos 2\pi ft\, dt$$

and if $g(t)$ is an odd function of t, then

$$G(f) = -2j\int_{0}^{\infty} g(t)\sin 2\pi ft\, dt$$

Hence, prove that the following.

If $g(t)$ is:	*Then $G(f)$ is*:
a real and even function of t	a real and even function of f
a real and odd function of t	an imaginary and odd function of f
an imaginary and even function of t	an imaginary and even function of f
a complex and even function of t	a complex and even function of f
a complex and odd function of t	a complex and odd function of f

3.1-2 **(a)** Show that for a real $g(t)$, the inverse transform, Eq. (3.9b), can be expressed as

$$g(t) = 2\int_{0}^{\infty} |G(f)|\cos[2\pi ft + \theta_g(2\pi f)]\, df$$

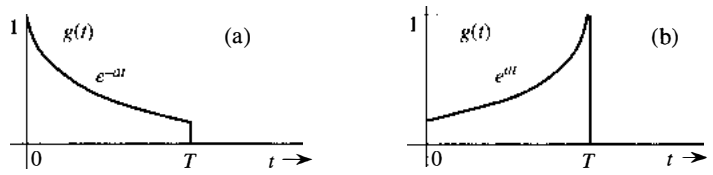
This is the trigonometric form of the (inverse) Fourier transform.

(b) Express the Fourier integral (inverse Fourier transform) for $g(t) = e^{-at}u(t)$ in the trigonometric form given in part **(a)**.

3.1-3 If $g(t) \Longleftrightarrow G(f)$, then show that $g^*(t) \Longleftrightarrow G^*(-f)$.

3.1-4 From definition (3.9a), find the Fourier transforms of the signals shown in Fig. P3.1-4.

Figure P.3.1-4



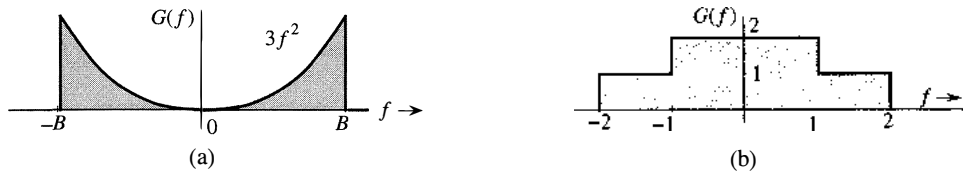
3.1-5 From definition (3.9a), find the Fourier transforms of the signals shown in Fig. P3.1-5.

Figure P.3.1-5



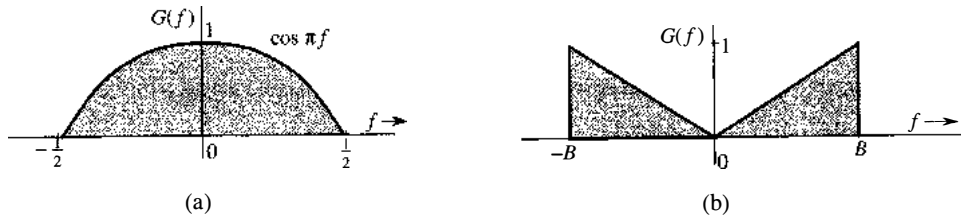
3.1-6 From definition (3.9b), find the inverse Fourier transforms of the spectra shown in Fig. P3.1-6.

Figure P.3.1-6



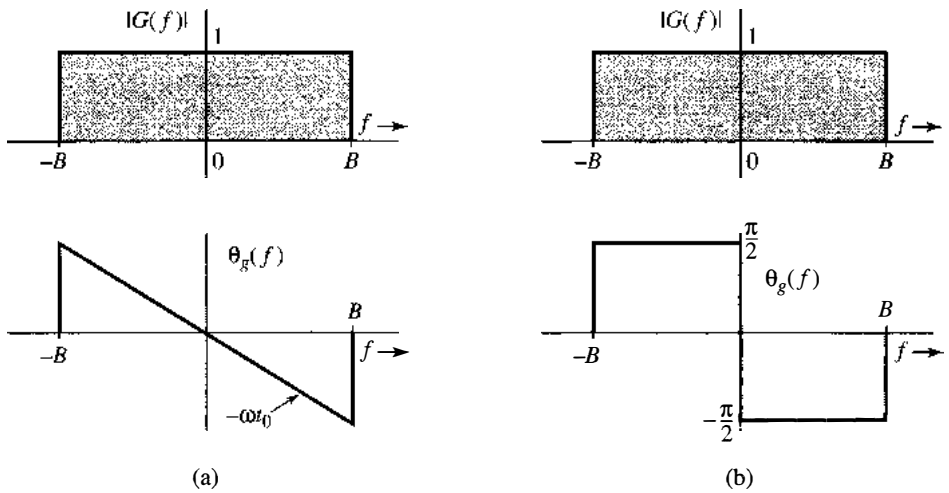
3.1-7 From definition (3.9b), find the inverse Fourier transforms of the spectra shown in Fig. P3.1-7.

Figure P.3.1-7



3.1-8 Show that the two signals in parts **(a)** and **(b)** of Fig. P3.1-8 are totally different in the time domain, despite their similarity.

Figure P.3.1-8



Hint: $G(f) = |G(f)|e^{j\theta_g(f)}$. For part **(a)**, $G(f) = 1 \cdot e^{-j2\pi ft_0}$, $|f| \le B$, whereas for part **(b)**,

$$G(f) = \begin{cases} 1e^{-j\pi/2} = -j & 0 < f \le B \\ 1e^{j\pi/2} = j & 0 > f \ge -B \end{cases}$$

3.2-1 Sketch the following functions: **(a)** $\Pi(t/2)$; **(b)** $\Delta(3\omega/100)$; **(c)** $\Pi(t-10/8)$; **(d)** sinc $(\pi\omega//5)$; **(e)** sinc $[(\omega - 10\pi)/5]$; **(f)** sinc $(t/5)\,\Pi(t/10\pi)$.

Hint: $g(\frac{x-a}{b})$ is $g(\frac{x}{b})$ right-shifted by a.

3.2-2 From definition (3.9a), show that the Fourier transform of rect $(t - 5)$ is sinc $(\pi f)e^{-j10\pi f}$.

3.2-3 From definition (3.9b), show that the inverse Fourier transform of rect $[(2\pi f - 10)/2\pi]$ is sinc $(\pi t)\, e^{j10t}$.

3.2-4 Using pairs 7 and 12 (Table 3.1) show that $u(t) \Longleftrightarrow 0.5\,\delta(f) + 1/j2\pi f$.

Hint: Add 1 to sgn (t), and see what signal comes out.

3.2-5 Show that $\cos(2\pi f_0 t + \theta) \Longleftrightarrow \frac{1}{2}[\delta(f + f_0)e^{-j\theta} + \delta(f - f_0)e^{j\theta}]$.
Hint: Express $\cos(2\pi f_0 t + \theta)$ in terms of exponentials using Euler's formula.

3.3-1 Apply the duality property to the appropriate pair in Table 3.1 to show that:

(a) $0.5[\delta(t) + (j/\pi t)] \Longleftrightarrow u(f)$

(b) $\delta(t + T) + \delta(t - T) \Longleftrightarrow 2\cos 2\pi fT$

(c) $\delta(t + T) - \delta(t - T) \Longleftrightarrow 2j \sin 2\pi fT$

Hint: $g(-t) \Longleftrightarrow G(-f)$ and $\delta(t) = \delta(-t)$.

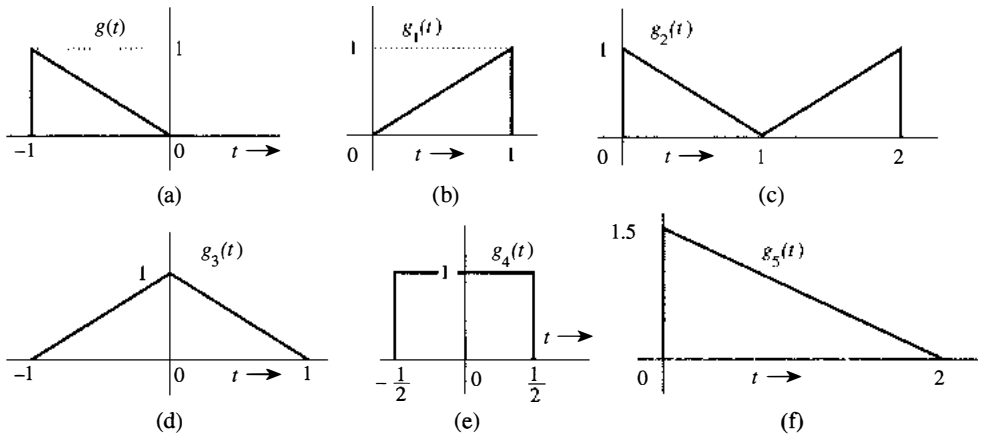
3.3-2 The Fourier transform of the triangular pulse $g(t)$ in Fig. P3.3-2a is given as

$$G(f) = \frac{1}{(2\pi f)^2}(e^{j2\pi f} - j2\pi f e^{j2\pi f} - 1)$$

Use this information, and the time-shifting and time-scaling properties, to find the Fourier transforms of the signals shown in Fig. P3.3-2b–f.

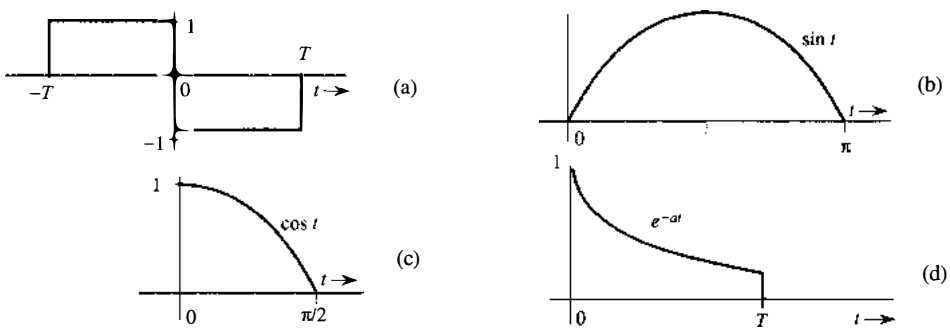
Hint: Time inversion in $g(t)$ results in the pulse $g_1(t)$ in Fig. P3.3-2b; consequently $g_1(t) = g(-t)$. The pulse in Fig. P3.3-2c can be expressed as $g(t-T)+g_1(t-T)$ [the sum of $g(t)$ and $g_1(t)$ both delayed by T]. Both pulses in Fig. P3.3-2d and e can be expressed as $g(t-T)+g_1(t+T)$ [the sum of $g(t)$ delayed by T and $g_1(t)$ advanced by T] for some suitable choice of T. The pulse in Fig. P3.3-2f can be obtained by time-expanding $g(t)$ by a factor of 2 and then delaying the resulting pulse by 2 seconds [or by first delaying $g(t)$ by 1 second and then time-expanding by a factor of 2].

Figure P.3.3-2



3.3-3 Using only the time-shifting property and Table 3.1, find the Fourier transforms of the signals shown in Fig. P3.3-3.

Figure P.3.3-3



Hint: The signal in Fig. P3.3-3a is a sum of two shifted rectangular pulses. The signal in Fig. P3.3-3b is $\sin t\,[u(t) - u(t-\pi)] = \sin t\,u(t) - \sin t\,u(t-\pi) = \sin t\,u(t) + \sin(t-\pi)$

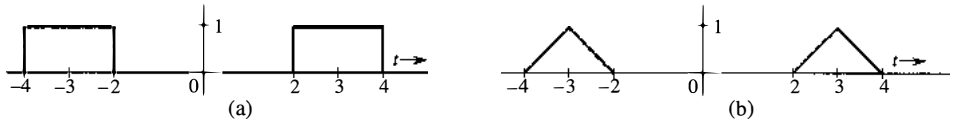
$u(t-\pi)$. The reader should verify that the addition of these two sinusoids indeed results in the pulse in Fig. P3.3-3b. In the same way, we can express the signal in Fig. P3.3-3c as $\cos t\, u(t) + \sin(t-\pi/2)u(t-\pi/2)$ (verify this by sketching these signals). The signal in Fig. P3.3-3d is $e^{-at}[u(t)-u(t-T)] = e^{-at}u(t) - e^{-aT}e^{-a(t-T)}u(t-T)$.

3.3-4 Use the time-shifting property to show that if $g(t) \Longleftrightarrow G(f)$, then

$$g(t+T) + g(t-T) \Longleftrightarrow 2G(f)\cos 2\pi fT$$

This is the dual of Eq. (3.36). Use this result and pairs 17 and 19 in Table 3.1 to find the Fourier transforms of the signals shown in Fig. P3.3-4.

Figure P.3.3-4



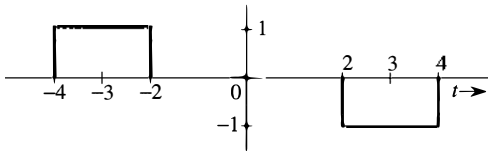
3.3-5 Prove the following results:

$$g(t)\sin 2\pi f_0 t \Longleftrightarrow \frac{1}{2j}[G(f-f_0) - G(f+f_0)]$$

$$\frac{1}{2j}[g(t+T) - g(t-T)] \Longleftrightarrow G(f)\sin 2\pi fT$$

Use the latter result and Table 3.1 to find the Fourier transform of the signal in Fig. P3.3-5.

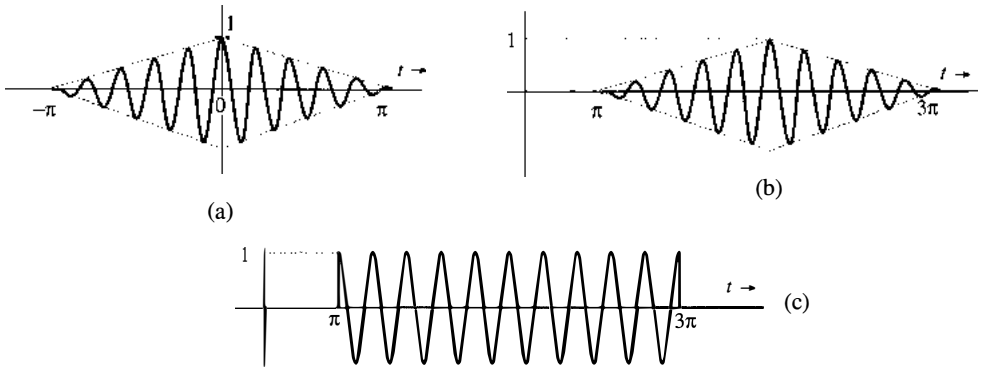
Figure P.3.3-5



3.3-6 The signals in Fig. P3.3-6 are modulated signals with carrier $\cos 10t$. Find the Fourier transforms of these signals by using the appropriate properties of the Fourier transform and Table 3.1. Sketch the amplitude and phase spectra for Fig. P3.3–6a and b.

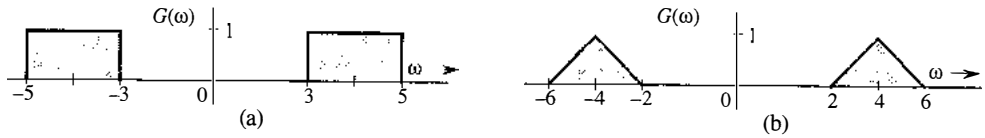
Hint: These functions can be expressed in the form $g(t)\cos 2\pi f_0 t$.

Figure P.3.3-6



3.3-7 Use the frequency shift property and Table 3.1 to find the inverse Fourier transform of the spectra shown in Fig. P3.3-7. Notice that this time, the Fourier transform is in the ω domain.

Figure P.3.3-7



3.3-8 A signal $g(t)$ is band-limited to B Hz. Show that the signal $g^n(t)$ is band-limited to nB Hz.

Hint: $g^2(t) \Longleftrightarrow [G(f) * G(f)]$, and so on. Use the width property of convolution.

3.3-9 Find the Fourier transform of the signal in Fig. P3.3-3a by three different methods:

(a) By direct integration using the definition (3.9a).

(b) Using only pair 17 Table 3.1 and the time-shifting property.

(c) Using the time differentiation and time-shifting properties, along with the fact that $\delta(t) \Longleftrightarrow 1$.

Hint: $1 - \cos 2x = 2\sin^2 x$.

3.3-10 The process of recovering a signal $g(t)$ from the modulated signal $g(t)\cos 2\pi f_0 t$ is called **demodulation**. Show that the signal $g(t)\cos 2\pi f_0 t$ can be demodulated by multiplying it by $2\cos 2\pi f_0 t$ and passing the product through a low-pass filter of bandwidth B Hz [the bandwidth of $g(t)$]. Assume $B < f_0$. *Hint*: $2\cos^2 2\pi f_0 t = 1 + \cos 4\pi f_0 t$. Recognize that the spectrum of $g(t)\cos 4\pi f_0 t$ is centered at $2f_0$ and will be suppressed by a low-pass filter of bandwidth B Hz.

3.4-1 Signals $g_1(t) = 10^4\Pi(10^4 t)$ and $g_2(t) = \delta(t)$ are applied at the inputs of the ideal low-pass filters $H_1(f) = \Pi(f/20,000)$ and $H_2(f) = \Pi(f/10,000)$ (Fig. P3.4-1). The outputs $y_1(t)$ and $y_2(t)$ of these filters are multiplied to obtain the signal $y(t) = y_1(t)y_2(t)$.

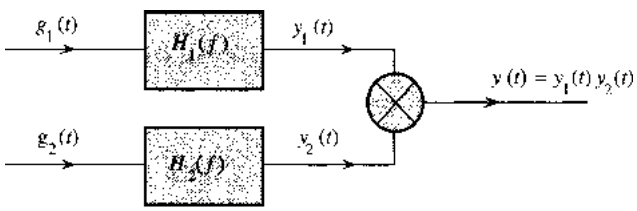
(a) Sketch $G_1(f)$ and $G_2(f)$.

(b) Sketch $H_1(f)$ and $H_2(f)$.

(c) Sketch $Y_1(f)$ and $Y_2(f)$.

(d) Find the bandwidths of $y_1(t), y_2(t)$, and $y(t)$.

Figure P.3.4-1



3.5-1 For systems with the following impulse responses, which system is causal?

(a) $h(t) = e^{-at}u(t), \quad a > 0$

(b) $h(t) = e^{-a|t|}, \quad a > 0$

(c) $h(t) = e^{-a(t-t_0)}u(t - t_0), \quad a > 0$

(d) $h(t) = \text{sinc}(at), \quad a > 0$

(e) $h(t) = \text{sinc}[a(t - t_0)], \quad a > 0.$

3.5-2 Consider a filter with the transfer function

$$H(f) = e^{-k(2\pi kf)^2 - j2\pi ft_0}$$

Show that this filter is physically unrealizable by using the time domain criterion [noncausal $h(t)$] and the frequency domain (Paley-Wiener) criterion. Can this filter be made approximately realizable by choosing a sufficiently large t_0? Use your own (reasonable) criterion of approximate realizability to determine t_0.

Hint: Use pair 22 in Table 3.1.

3.5-3 Show that a filter with transfer function

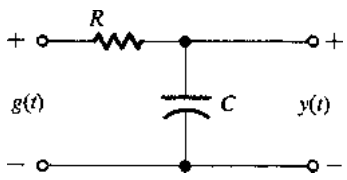
$$H(f) = \frac{2(10^5)}{(2\pi f)^2 + 10^{10}} e^{-j2\pi ft_0}$$

is unrealizable. Can this filter be made approximately realizable by choosing a sufficiently large t_0? Use your own (reasonable) criterion of approximate realizability to determine t_0.

Hint: Show that the impulse response is noncausal.

3.5-4 Determine the maximum bandwidth of a signal that can be transmitted through the low-pass RC filter in Fig. P3.5-4 with $R = 1000$ and $C = 10^{-9}$ if, over this bandwidth, the amplitude response (gain) variation is to be within 5% and the time delay variation is to be within 2%.

Figure P.3.5-4



3.5-5 A bandpass signal $g(t)$ of bandwidth $B = 2000$ Hz centered at $f = 10^5$ Hz is passed through the RC filter in Fig. P3.5-4 with $RC = 10^{-3}$. If over the passband, a variation of less than 2% in amplitude response and less than 1% in time delay is considered distortionless transmission, would $g(t)$ be transmitted without distortion? Find the approximate expression for the output signal.

3.6-1 A certain channel has ideal amplitude, but nonideal phase response (Fig. P3.6-1), given by

$$|H(f)| = 1$$
$$\theta_h(f) = -2\pi ft_0 - k \sin 2\pi fT \qquad k \ll 1$$

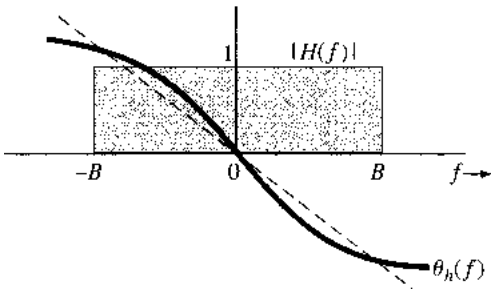
(a) Show that $y(t)$, the channel response to an input pulse $g(t)$ band-limited to B Hz, is

$$y(t) = g(t - t_0) + \frac{k}{2}[g(t - t_0 - T) - g(t - t_0 + T)]$$

Hint: Use $e^{-jk \sin 2\pi fT} \approx 1 - jk \sin 2\pi fT$.

(b) Discuss how this channel will affect TDM and FDM systems from the viewpoint of interference among the multiplexed signals.

Figure P.3.6-1

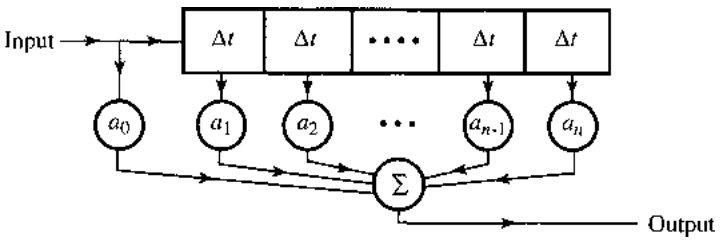


3.6-2 The distortion caused by multipath transmission can be partly corrected by a tapped delay-line equalizer. Show that if $\alpha \ll 1$, the distortion in the multipath system in Fig. 3.31a can be approximately corrected if the received signal in Fig. 3.31a is passed through the tapped delay-line equalizer shown in Fig. P3.6-2.

Hint: From Eq. (3.64a), it is clear that the equalizer filter transfer function should be $H_{eq}(f) = 1/(1+\alpha e^{-j2\pi f \Delta t})$. Use the fact that $1/(1-x) = 1+x+x^2+x^3+\cdots$ if $x \ll 1$ to show what should be the tap parameters a_i to make the resulting transfer function

$$H(f)H_{eq}(f) \approx e^{-j2\pi f t_d}$$

Figure P.3.6-2



3.7-1 Show that the energy of the Gaussian pulse

$$g(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2}}$$

from direct integration is $1/2\sigma\sqrt{\pi}$. Verify this result by using Parseval's theorem to derive the energy E_g from $G(f)$. *Hint:* See pair 22 in Table 3.1. Use the fact that

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-x^2-y^2}\,dx\,dy = \pi \Rightarrow \int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi}.$$

3.7-2 Show that

$$\int_{-\infty}^{\infty} \text{sinc}^2\,(kt)dt = \frac{\pi}{4}$$

Hint: Recognize that the integral is the energy of $g(t) = \text{sinc}\,(kt)$. Use Parseval's theorem to find this energy.

3.7-3 Generalize Parseval's theorem to show that for real Fourier transformable signals $g_1(t)$ and $g_2(t)$,

$$\int_{-\infty}^{\infty} g_1(t)g_2(t)\,dt = \int_{-\infty}^{\infty} G_1(-f)G_2(f)\,df = \int_{-\infty}^{\infty} G_1(f)G_2(-f)\,df$$

3.7-4 Show that

$$\int_{-\infty}^{\infty} \text{sinc}\,(2\pi Bt - m\pi)\,\text{sinc}\,(2\pi Bt - n\pi)\,dt = \begin{cases} 0 & m \neq n \\ \frac{1}{2B} & m = n \end{cases}$$

Hint: Recognize that

$$\text{sinc}\,(2\pi Bt - k\pi) = \text{sinc}\left[2\pi B\left(t - \frac{k}{2B}\right)\right] \Longleftrightarrow \frac{1}{2B}\,\Pi\left(\frac{f}{2B}\right)e^{-j\pi fk/B}$$

Use this fact and the result in Prob. 3.7-2 to show that

$$\int_{-\infty}^{\infty} \text{sinc}\,(2\pi Bt - m\pi)\,\text{sinc}\,(2\pi Bt - n\pi)\,dt = \frac{1}{4B^2}\int_{-B}^{B} e^{j[(n-m)/2B]2\pi f}\,df$$

The desired result follows from this integral.

3.7-5 For the signal

$$g(t) = \frac{2a}{t^2 + a^2}$$

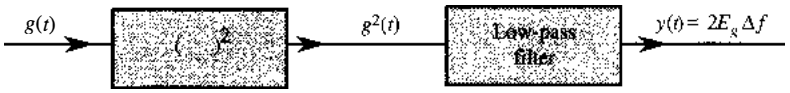
determine the essential bandwidth B Hz of $g(t)$ such that the energy contained in the spectral components of $g(t)$ of frequencies below B Hz is 99% of the signal energy E_g.

Hint: Determine $G(f)$ by applying the duality property [Eq. (3.26)] to pair 3 of Table 3.1.

3.7-6 A low-pass signal $g(t)$ is applied to a squaring device. The squarer output $g^2(t)$ is applied to a unity gain ideal low-pass filter of bandwidth Δf Hz (Fig. P3.7-6). Show that if Δf is very small ($\Delta f \to 0$), the filter output is a dc signal of amplitude $2E_g\,\Delta f$, where E_g is the energy of $g(t)$.

Hint: The output $y(t)$ is a dc signal because its spectrum $Y(f)$ is concentrated at $f = 0$ from $-\Delta f$ to Δf with $\Delta f \to 0$ (impulse at the origin). If $g^2(t) \Longleftrightarrow A(f)$, and $y(t) \Longleftrightarrow Y(f)$, then $Y(f) \approx [2A(0)\Delta f]\delta(f)$. Now, show that $E_g = A(0)$.

Figure P.3.7-6



3.8-1 Show that the autocorrelation function of $g(t) = C\cos(2\pi f_0 t + \theta_0)$ is given by $\mathcal{R}_g(\tau) = (C^2/2)\cos 2\pi f_0\tau$, and the corresponding PSD is $S_g(f) = (C^2/4)[\delta(f - f_0) + \delta(f + f_0)]$. Hence, show that for a signal $y(t)$ given by

$$y(t) = C_0 + \sum_{n=1}^{\infty} C_n \cos(n2\pi f_0 t + \theta_n)$$

the autocorrelation function and the PSD are given by

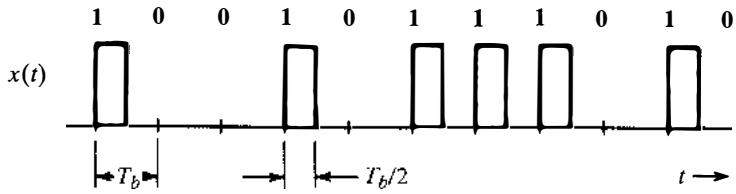
$$\mathcal{R}_y(\tau) = C_0{}^2 + \frac{1}{2}\sum_{n=1}^{\infty} C_n{}^2 \cos n2\pi f_0\tau$$

$$S_y(f) = C_0{}^2\delta(f) + \frac{1}{4}\sum_{n=1}^{\infty} C_n{}^2[\delta(f - nf_0) + \delta(f + nf_0)]$$

Hint: Show that if $g(t) = g_1(t)+g_2(t)$, then $\mathcal{R}_g(\tau) = \mathcal{R}_{g_1}(\tau)+\mathcal{R}_{g_2}(\tau)+\mathcal{R}_{g_1g_2}(\tau)+\mathcal{R}_{g_2g_1}(\tau)$, where $\mathcal{R}_{g_1g_2}(\tau) = \lim_{T\to\infty}(1/T)\int_{-T/2}^{T/2} g_1(t)g_2(t+\tau)\,dt$. If $g_1(t)$ and $g_2(t)$ represent any two of the infinite terms in $y(t)$, then show that $\mathcal{R}_{g_1g_2}(\tau) = \mathcal{R}_{g_2g_1}(\tau) = 0$. To show this, use the fact that the area under any sinusoid over a very large time interval is at most equal to the area of the half-cycle of the sinusoid.

3.8-2 The random binary signal $x(t)$ shown in Fig. P3.8-2 transmits one digit every T_b seconds. A binary **1** is transmitted by a pulse $p(t)$ of width $T_b/2$ and amplitude A; a binary **0** is transmitted by no pulse. The digits **1** and **0** are equally likely and occur randomly. Determine the autocorrelation function $\mathcal{R}_x(\tau)$ and the PSD $S_x(f)$.

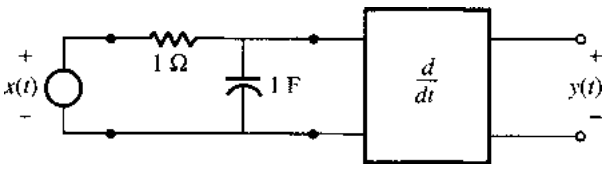
Figure P.3.8-2



3.8-3 Find the mean square value (or power) of the output voltage $y(t)$ of the RC network shown in Fig. P3.5-4 with $RC = 2\pi$ if the input voltage PSD $S_x(f)$ is given by **(a)** K; **(b)** $\Pi(\pi f)$; **(c)** $[\delta(f + 1) + \delta(f - 1)]$. In each case calculate the power (mean square value) of the input signal $x(t)$.

3.8-4 Find the mean square value (or power) of the output voltage $y(t)$ of the system shown in Fig. P3.8-4 if the input voltage PSD $S_x(f) = \Pi(\pi f)$. Calculate the power (mean square value) of the input signal $x(t)$.

Figure P.3.8-4



4 AMPLITUDE MODULATIONS AND DEMODULATIONS

Modulation often refers to a process that moves the message signal into a specific frequency band that is dictated by the physical channel (e.g. voiceband telephone modems). Modulation provides a number of advantages mentioned in Chapter 1 including ease of RF transmission and frequency division multiplexing. Modulations can be analog or digital. Though traditional communication systems such as AM/FM radios and NTSC television signals are based on analog modulations, more recent systems such as 2G and 3G cellphones, HDTV, and DSL are all digital.

In this chapter and the next, we will focus on the classic analog modulations: amplitude modulation and angle modulation. Before we begin our discussion of different analog modulations, it is important to distinguish between communication systems that do not use modulation (**baseband communications**) and systems that use modulation (**carrier communications**).

4.1 BASEBAND VERSUS CARRIER COMMUNICATIONS

The term **baseband** is used to designate the frequency band of the original message signal from the source or the input transducer (see Fig. 1.2). In telephony, the baseband is the audio band (band of voice signals) of 0 to 3.5 kHz. In NTSC television, the video baseband is the video band occupying 0 to 4.3 MHz. For digital data or pulse code modulation (PCM) that uses bipolar signaling at a rate of R_b pulses per second, the baseband is approximately 0 to R_b Hz.

Baseband Communications

In baseband communication, message signals are directly transmitted without any modification. Because most baseband signals such as audio and video contain significant low-frequency content, they cannot be effectively transmitted over radio (wireless) links. Instead, dedicated user channels such as twisted pairs of copper wires and coaxial cables are assigned to each user for long-distance communications. Because baseband signals have overlapping bands, they would interfere severely if sharing a common channel. Thus, baseband communications leave much of the channel spectrum unused. By modulating several baseband signals and shifting their spectra to nonoverlapping bands, many users can share one channel by utilizing

most of the available bandwidth through frequency division multiplexing (FDM). Long-haul communication over a radio link also requires modulation to shift the signal spectrum to higher frequencies in order to enable efficient power radiation using antennas of reasonable dimensions. Yet another use of modulation is to exchange transmission bandwidth for better performance against interferences.

Carrier Modulations

Communication that uses modulation to shift the frequency spectrum of a signal is known as **carrier communication**. In terms of analog modulation, one of the basic parameters (amplitude, frequency, or phase) of a **sinusoidal carrier** of high frequency f_c Hz (or $\omega_c = 2\pi f_c$ rad/s) is varied linearly with the baseband signal $m(t)$. This results in amplitude modulation (AM), frequency modulation (FM), or phase modulation (PM), respectively. Amplitude modulation is linear, while the latter two types of carrier modulation are similar and nonlinear, often known collectively as **angle modulation**.

A comment about pulse-modulated signals [pulse amplitude modulation (PAM), pulse width modulation (PWM), pulse position modulation (PPM), pulse code modulation (PCM), and delta modulation (DM)] is in order here. Despite the term *modulation*, these signals are baseband digital signals. "Modulation" is used here not in the sense of frequency or band shifting. Rather, in these cases it is in fact describing digital pulse coding schemes used to represent the original analog signals. In other words, the analog message signal is modulating parameters of a digital pulse train. These signals can still modulate a carrier in order to shift their spectra.

Amplitude Modulations and Angle Modulations

We denote as $m(t)$ the source message signal that is to be transmitted by the sender to its receivers; its Fourier transform is denoted as $M(f)$. To move the frequency response of $m(t)$ to a new frequency band centered at f_c Hz, we begin by noting that the Fourier transform has already revealed a very strong property known as the *frequency shifting* property to achieve this goal. In other words, all we need to do is to multiply $m(t)$ by a sinusoid of frequency f_c such that

$$s_1(t) = m(t) \cos\ 2\pi f_c t$$

This immediately achieves the basic aim of modulation by moving the signal frequency content to be centered at $\pm f_c$ via

$$S_1(f) = \frac{1}{2}M(f - f_c) + \frac{1}{2}M(f + f_c)$$

This simple multiplication is in fact allowing changes in the amplitude of the sinusoid $s_1(t)$ to be proportional to the message signal. This method is indeed a very valuable modulation known as amplitude modulation.

More broadly, consider a sinusoidal signal

$$s(t) = A(t) \cos\ [\omega_c t + \phi(t)]$$

There are three variables in a sinusoid: amplitude, (instantaneous) frequency, and phase. Indeed, the message signal can be used to modulate any one of these three parameters to allow $s(t)$ to carry the information from the transmitter to the receiver:

Amplitude $A(t)$ linearly varies with $m(t)$	$\Longleftrightarrow$	amplitude modulation
Frequency linearly varies with $m(t)$	$\Longleftrightarrow$	frequency modulation
Phase $\phi(t)$ linearly varies with $m(t)$	$\Longleftrightarrow$	phase modulation

These are known, respectively, as amplitude modulation, frequency modulation, and phase modulation. In this chapter, we describe various forms of amplitude modulation in practical communication systems. Amplitude modulations are linear, and their analysis in the time and frequency domains is simpler. In Chapter 5, we will separately discuss nonlinear angle modulations.

The Interchangeable Use of f and ω

In Chapter 3, we noted the equivalence of frequency response denoted by frequency f with angular frequency ω. Each of these two notations has its own advantages and disadvantages. After the examples and problems of Chapter 3, readers should be familiar and comfortable with the use of either notation. Thus, from this point on, we will use the two different notations interchangeably, selecting one or the other on the basis of notational or graphical simplicity.

4.2 DOUBLE-SIDEBAND AMPLITUDE MODULATION

Amplitude modulation is characterized by an information-bearing **carrier** amplitude $A(t)$ that is a linear function of the baseband (message) signal $m(t)$. At the same time, the carrier frequency ω_c and the phase θ_c remain constant. We can assume $\theta_c = 0$ without loss of generality. If the carrier amplitude A is made directly proportional to the modulating signal $m(t)$, then *modulated signal* is $m(t) \cos \omega_c t$ (Fig. 4.1). As we saw earlier [Eq. (3.36)], this type of modulation simply shifts the spectrum of $m(t)$ to the carrier frequency (Fig. 4.1a). Thus, if

$$m(t) \Longleftrightarrow M(f)$$

then

$$m(t) \cos 2\pi f_c t \Longleftrightarrow \tfrac{1}{2}[M(f + f_c) + M(f - f_c)] \tag{4.1}$$

Recall that $M(f - f_c)$ is $M(f)$ shifted to the right by f_c, and $M(f + f_c)$ is $M(f)$ shifted to the left by f_c. Thus, the process of modulation shifts the spectrum of the modulating signal to the left and to the right by f_c. Note also that if the bandwidth of $m(t)$ is B Hz, then, as seen from Fig. 4.1c, the modulated signal now has bandwidth of $2B$ Hz. We also observe that the modulated signal spectrum centered at $\pm f_c$ (or $\pm\omega_c$ in rad/s) consists of two parts: a portion that lies outside $\pm f_c$, known as the *upper sideband (USB)*, and a portion that lies inside $\pm f_c$, known as the *lower sideband (LSB)*. We can also see from Fig. 4.1c that, unless the message signal $M(f)$ has an impulse at zero frequency, the modulated signal in this scheme does not contain a discrete component of the carrier frequency f_c. In other words, the modulation process does not introduce a sinusoid at f_c. For this reason it is called **double-sideband suppressed carrier (DSB-SC) modulation.***

The relationship of B to f_c is of interest. Figure 4.1c shows that $f_c \geq B$, thus avoiding overlap of the modulated spectra centered at f_c and $-f_c$. If $f_c < B$, then the two copies of message spectra overlap and the information of $m(t)$ is lost during modulation, which makes it impossible to get back $m(t)$ from the modulated signal $m(t) \cos \omega_c t$.

Note that practical factors may impose additional restrictions on f_c. For instance, in broadcast applications, a transmit antenna can radiate only a narrow band without distortion. This means that to avoid distortion caused by the transmit antenna, we must have $f_c/B \gg 1$. The

* The term **suppressed carrier** does not necessarily mean absence of the spectrum at the carrier frequency f_c. It means that there is no discrete component of the carrier frequency. This implies that the spectrum of the DSB-SC does not have impulses at $\pm f_c$, which also implies that the modulated signal $m(t) \cos 2\pi f_c t$ does not contain a term of the form $k \cos 2\pi f_c t$ [assuming that $m(t)$ has a zero mean value].

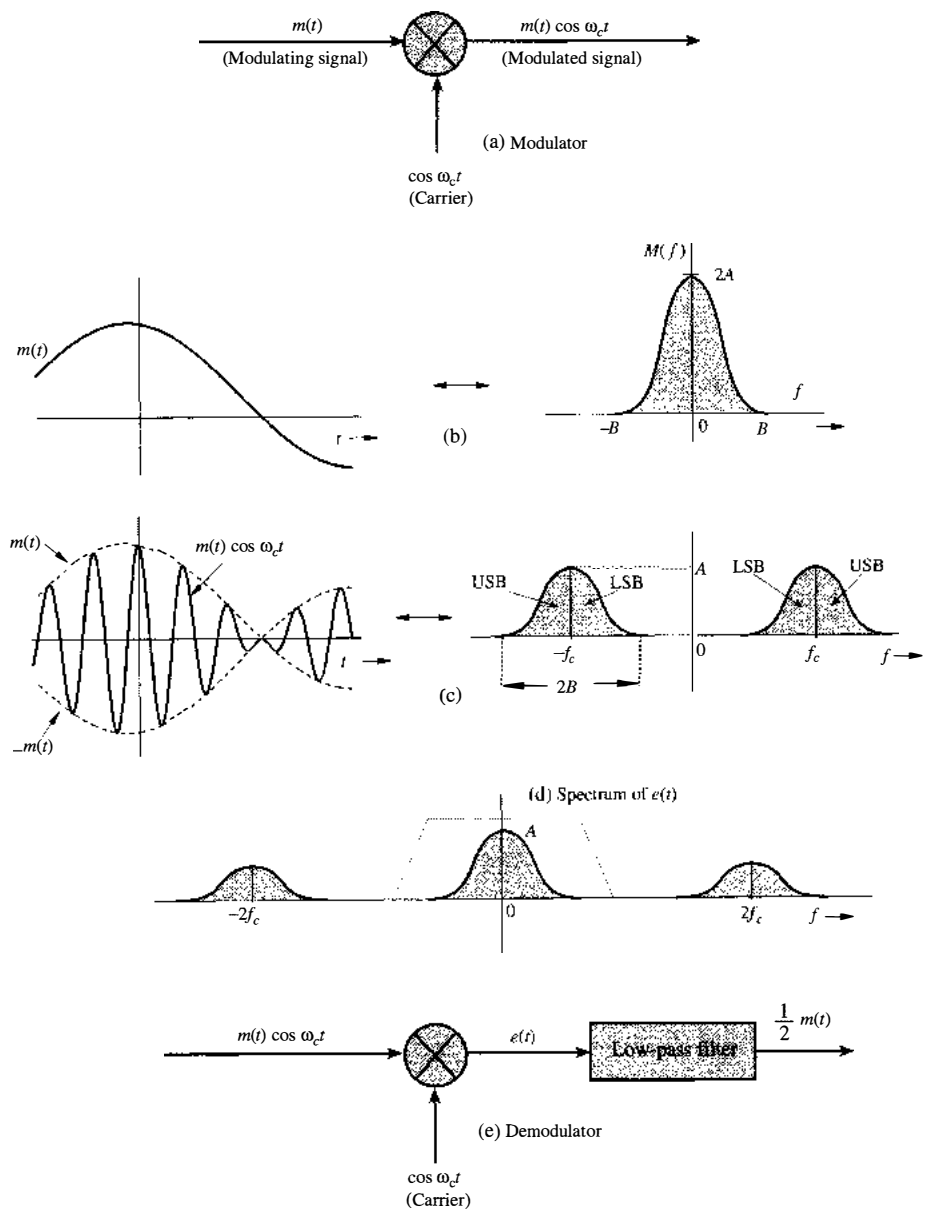


Figure 4.1 DSB-SC modulation and demodulation.

broadcast band AM radio, for instance, with $B = 5$ kHz and the band of 550 to 1600 kHz for the carrier frequencies, gives a ratio of f_c/B roughly in the range of 100 to 300.

Demodulation

The DSB-SC modulation translates or shifts the frequency spectrum to the left and the right by f_c (i.e., at $+f_c$ and $-f_c$), as seen from Eq. (4.1). To recover the original signal $m(t)$ from the modulated signal, it is necessary to retranslate the spectrum to its original position. The process of recovering the signal from the modulated signal (retranslating the spectrum to its original position) is referred to as **demodulation**. Observe that if the modulated signal spectrum in Fig. 4.1c is shifted to the left and to the right by f_c (and multiplied by one-half), we obtain the

spectrum shown in Fig. 4.1d, which contains the desired baseband spectrum plus an unwanted spectrum at $\pm 2f_c$. The latter can be suppressed by a low-pass filter. Thus, demodulation, which is almost identical to modulation, consists of multiplication of the incoming modulated signal $m(t)\cos\ \omega_c t$ by a carrier $\cos\ \omega_c t$ followed by a low-pass filter, as shown in Fig. 4.1e. We can verify this conclusion directly in the time domain by observing that the signal $e(t)$ in Fig. 4.1e is

$$\begin{aligned} e(t) &= m(t)\cos^2 \omega_c t \\ &= \frac{1}{2}[m(t) + m(t)\cos\ 2\omega_c t] \end{aligned} \tag{4.2a}$$

Therefore, the Fourier transform of the signal $e(t)$ is

$$E(f) = \frac{1}{2}M(f) + \frac{1}{4}[M(f + 2f_c) + M(f - 2f_c)] \tag{4.2b}$$

This analysis shows that the signal $e(t)$ consists of two components $(1/2)m(t)$ and $(1/2)m(t)\cos\ 2\omega_c t$, with their nonoverlapping spectra as shown in Fig. 4.1d. The spectrum of the second component, being a modulated signal with carrier frequency $2f_c$, is centered at $\pm 2f_c$. Hence, this component is suppressed by the low-pass filter in Fig. 4.1e. The desired component $(1/2)M(f)$, being a low-pass spectrum (centered at $f = 0$), passes through the filter unharmed, resulting in the output $(1/2)m(t)$. A possible form of low pass filter characteristics is shown (under the dotted line) in Fig. 4.1d. The filter leads to a distortionless demodulation of the message signal $m(t)$ from the DSB-SC signal. We can get rid of the inconvenient fraction $1/2$ in the output by using a carrier $2\cos\ \omega_c t$ instead of $\cos\ \omega_c t$. In fact, later on, we shall often use this strategy, which does not affect general conclusions.

This method of recovering the baseband signal is called *synchronous detection*, or *coherent detection*, where we use a carrier of exactly the same frequency (and phase) as the carrier used for modulation. Thus, for demodulation, we need to generate a local carrier at the receiver in frequency and phase coherence (synchronism) with the carrier used at the modulator.

Example 4.1 For a baseband signal

$$m(t) = \cos\ \omega_m t = \cos\ 2\pi f_m t,$$

find the DSB-SC signal, and sketch its spectrum. Identify the USB and LSB. Verify that the DSB-SC modulated signal can be demodulated by the demodulator in Fig. 4.1e.

The case in this example is referred to as *tone modulation* because the modulating signal is a pure sinusoid, or tone, $\cos\ \omega_m t$. To clarify the basic concepts of DSB-SC modulation, we shall work this problem in the frequency domain as well as the time domain. In the frequency domain approach, we work with the signal spectra. The spectrum of the baseband signal $m(t) = \cos\ \omega_m t$ is given by

$$\begin{aligned} M(f) &= \frac{1}{2}[\delta(f - f_m) + \delta(f + f_m)] \\ &= \pi[\delta(\omega - \omega_m) + \delta(\omega + \omega_m)] \end{aligned}$$

The message spectrum consists of two impulses located at $\pm f_m$, as shown in Fig. 4.2a. The DSB-SC (modulated) spectrum, as seen from Eq. (4.1), is the baseband spectrum in Fig. 4.2a shifted to the right and the left by f_c (times one-half), as shown in Fig. 4.2b. This spectrum consists of impulses at angular frequencies $\pm(f_c - f_m)$ and $\pm(f_c + f_m)$. The spectrum beyond f_c is the USB, and the one below f_c is the LSB. Observe that the DSB-SC spectrum does not have the component of the carrier frequency f_c. This is why it is called *suppressed carrier.*

In the time domain approach, we work directly with signals in the time domain. For the baseband signal $m(t) = \cos\ \omega_m t$, the DSB-SC signal $\varphi_{\text{DSB-SC}}(t)$ is

$$\begin{aligned}\varphi_{\text{DSB-SC}}(t) &= m(t)\cos\ \omega_c t \\ &= \cos\ \omega_m t \cos\ \omega_c t \\ &= \frac{1}{2}[\cos\ (\omega_c + \omega_m)t + \cos\ (\omega_c - \omega_m)t]\end{aligned}$$

This shows that when the baseband (message) signal is a single sinusoid of frequency f_m, the modulated signal consists of two sinusoids: the component of frequency $f_c + f_m$ (the USB) and the component of frequency $f_c - f_m$ (the LSB). Figure 4.2b shows precisely the spectrum of $\varphi_{\text{DSB-SC}}(t)$. Thus, each component of frequency f_m in the modulating signal turns into two components of frequencies $f_c + f_m$ and $f_c - f_m$ in the modulated signal. Note the curious fact that there is no component of the carrier frequency f_c on the right-hand side of the preceding equation. As mentioned, this is why it is called double-sideband suppressed carrier (DSB-SC) modulation.

We now verify that the modulated signal $\varphi_{\text{DSB-SC}}(t) = \cos\ \omega_m t \cos\ \omega_c t$, when applied to the input of the demodulator in Fig. 4.1e, yields the output proportional to the desired baseband signal $\cos\ \omega_m t$. The signal $e(t)$ in Fig. 4.1e is given by

$$\begin{aligned}e(t) &= \cos\ \omega_m t \cos^2 \omega_c t \\ &= \frac{1}{2}\cos\ \omega_m t\ (1 + \cos\ 2\omega_c t)\end{aligned}$$

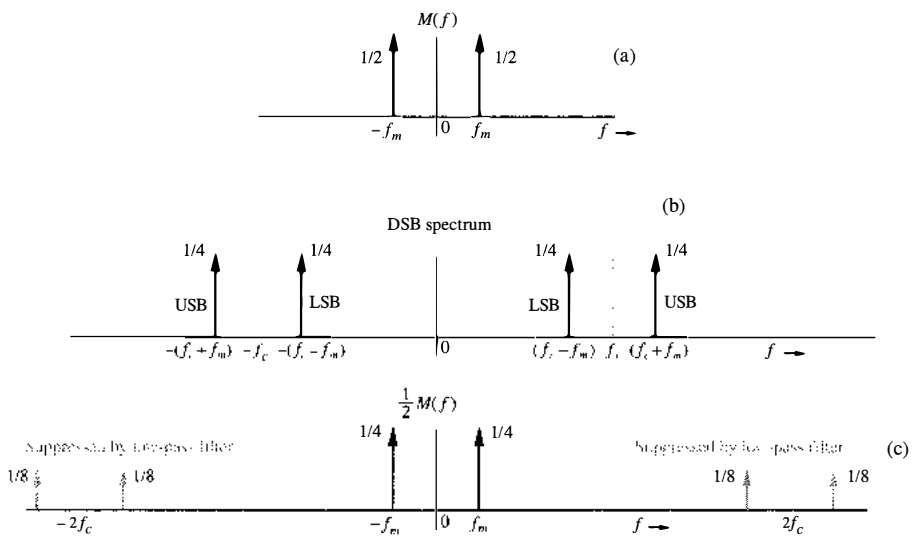


Figure 4.2 Example of DSB-SC modulation.

The spectrum of the term $\cos \omega_m t \cos 2\omega_c t$ is centered at $2\omega_c$ and will be suppressed by the low-pass filter, yielding $\frac{1}{2} \cos \omega_m t$ as the output. We can also derive this result in the frequency domain. Demodulation causes the spectrum in Fig. 4.2b to shift left and right by ω_c (and to be multiplied by one-half). This results in the spectrum shown in Fig. 4.2c. The low-pass filter suppresses the spectrum centered at $\pm 2\omega_c$, yielding the spectrum $\frac{1}{2}M(f)$.

Modulators

Modulation can be achieved in several ways. We shall discuss some important categories of modulators.

Multiplier Modulators: Here modulation is achieved directly by multiplying $m(t)$ with $\cos \omega_c t$, using an analog multiplier whose output is proportional to the product of two input signals. Typically, such a multiplier may be obtained from a variable-gain amplifier in which the gain parameter (such as the β of a transistor) is controlled by one of the signals, say, $m(t)$. When the signal $\cos \omega_c t$ is applied at the input of this amplifier, the output is proportional to $m(t) \cos \omega_c t$.

In the early days, multiplication of two signals over a sizable dynamic range was a challenge to circuit designers. However, as semiconductor technologies continued to advance, signal multiplication ceased to be a major concern. Still, we will present several classical modulators that avoid the use of multipliers. Studying these modulators can provide unique insight and an excellent opportunity to pick up some new signal analysis skills.

Nonlinear Modulators: Modulation can also be achieved by using nonlinear devices, such as a semiconductor diode or a transistor. Figure 4.3 shows one possible scheme, which uses two identical nonlinear elements (boxes marked NL).

Let the input-output characteristics of either of the nonlinear elements be approximated by a power series

$$y(t) = ax(t) + bx^2(t) \tag{4.3}$$

where $x(t)$ and $y(t)$ are the input and the output, respectively, of the nonlinear element. The summer output $z(t)$ in Fig. 4.3 is given by

$$z(t) = y_1(t) - y_2(t) = [ax_1(t) + bx_1{}^2(t)] - [ax_2(t) + bx_2{}^2(t)]$$

Substituting the two inputs $x_1(t) = \cos \omega_c t + m(t)$ and $x_2(t) = \cos \omega_c t - m(t)$ in this equation yields

$$z(t) = 2a \cdot m(t) + 4b \cdot m(t) \cos \omega_c t$$

The spectrum of $m(t)$ is centered at the origin, whereas the spectrum of $m(t) \cos \omega_c t$ is centered at $\pm \omega_c$. Consequently, when $z(t)$ is passed through a bandpass filter tuned to ω_c, the signal $am(t)$ is suppressed and the desired modulated signal $4bm(t) \cos \omega_c t$ can pass through the system without distortion.

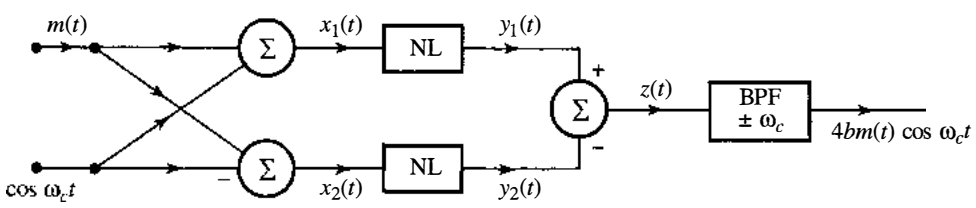


Figure 4.3 Nonlinear DSB-SC modulator.

In this circuit there are two inputs: $m(t)$ and $\cos\ \omega_c t$. The output of the last summer, $z(t)$, no longer contains one of the inputs, the carrier signal $\cos\ \omega_c t$. Consequently, the carrier signal does not appear at the input of the final bandpass filter. The circuit acts as a balanced bridge for one of the inputs (the carrier). Circuits that have this characteristic are called *balanced circuits*. The nonlinear modulator in Fig. 4.3 is an example of a class of modulators known as *balanced modulators*. This circuit is balanced with respect to only one input (the carrier); the other input $m(t)$ still appears at the final bandpass filter, which must reject it. For this reason, it is called a *single balanced modulator*. A circuit balanced with respect to both inputs is called a *double balanced modulator*, of which the ring modulator (see later: Fig. 4.6) is an example.

Switching Modulators: The multiplication operation required for modulation can be replaced by a simpler switching operation if we realize that a modulated signal can be obtained by multiplying $m(t)$ not only by a pure sinusoid but by any periodic signal $\phi(t)$ of the fundamental radian frequency ω_c. Such a periodic signal can be expressed by a trigonometric Fourier series as

$$\phi(t) = \sum_{n=0}^{\infty} C_n \cos\ (n\omega_c t + \theta_n) \tag{4.4a}$$

Hence,

$$m(t)\phi(t) = \sum_{n=0}^{\infty} C_n\, m(t) \cos\ (n\omega_c t + \theta_n) \tag{4.4b}$$

This shows that the spectrum of the product $m(t)\phi(t)$ is the spectrum $M(\omega)$ shifted to $\pm\omega_c,\ \pm 2\omega_c, \ldots, \pm n\omega_c, \ldots$. If this signal is passed through a bandpass filter of bandwidth $2B$ Hz and tuned to ω_c, then we get the desired modulated signal $c_1 m(t) \cos\ (\omega_c t + \theta_1)$.*

The square pulse train $w(t)$ in Fig. 4.4b is a periodic signal whose Fourier series was found earlier (by rewriting the results of Example 2.4) as

$$w(t) = \frac{1}{2} + \frac{2}{\pi}\left(\cos\ \omega_c t - \frac{1}{3}\cos\ 3\omega_c t + \frac{1}{5}\cos\ 5\omega_c t - \cdots\right) \tag{4.5}$$

The signal $m(t)w(t)$ is given by

$$m(t)w(t) = \frac{1}{2}m(t) + \frac{2}{\pi}\left[m(t)\cos\ \omega_c t - \frac{1}{3}m(t)\cos\ 3\omega_c t + \frac{1}{5}m(t)\cos\ 5\omega_c t - \cdots\right] \tag{4.6}$$

The signal $m(t)w(t)$ consists not only of the component $m(t)$ but also of an infinite number of modulated signals with carrier frequencies $\omega_c,\ 3\omega_c,\ 5\omega_c, \ldots$. Therefore, the spectrum of $m(t)w(t)$ consists of multiple copies of the message spectrum $M(f)$, shifted to $0, \pm f_c, \pm 3f_c, \pm 5f_c, \ldots$ (with decreasing relative weights), as shown in Fig. 4.4c.

For modulation, we are interested in extracting the modulated component $m(t)\cos\ \omega_c t$ only. To separate this component from the rest of the crowd, we pass the signal $m(t)w(t)$ through a bandpass filter of bandwidth $2B$ Hz (or $4\pi B$ rad/s), centered at the frequency $\pm f_c$. Provided the carrier frequency $f_c \geq 2B$ (or $\omega_c \geq 4\pi B$), this will suppress all the spectral components not centered at $\pm f_c$ to yield the desired modulated signal $(2/\pi)m(t)\cos\ \omega_c t$ (Fig. 4.4d).

We now see the real payoff of this method. Multiplication of a signal by a square pulse train is *in reality* a switching operation in which the signal $m(t)$ is switched on and off periodically; it

* The phase θ_1 is not important.

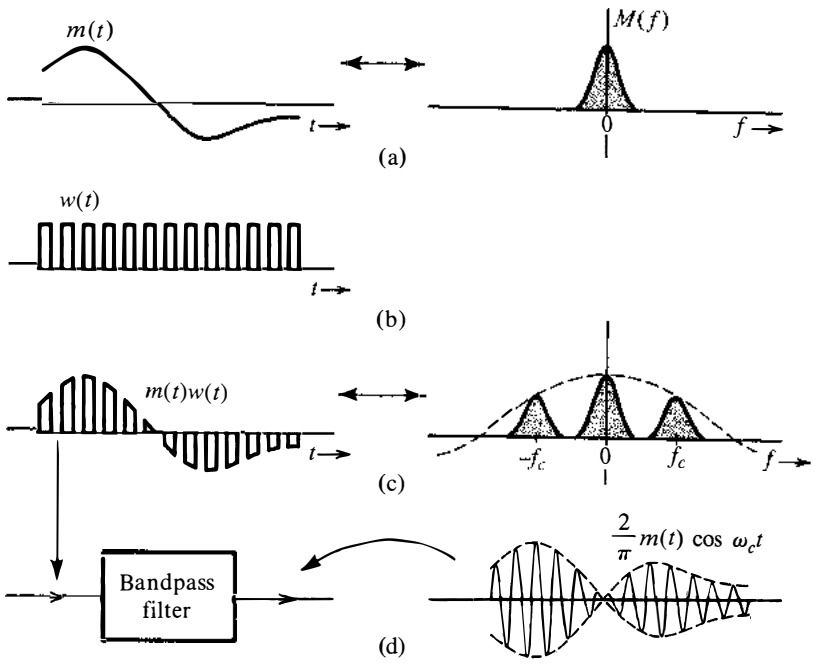


Figure 4.4 Switching modulator for DSB-SC.

can be accomplished by simple switching elements controlled by $w(t)$. Figure 4.5a shows one such electronic switch, the **diode-bridge modulator**, driven by a sinusoid $A \cos \omega_c t$ to produce the switching action. Diodes D_1, D_2 and D_3, D_4 are matched pairs. When the signal $\cos \omega_c t$ is of a polarity that will make terminal c positive with respect to d, all the diodes conduct. Because diodes D_1 and D_2 are matched, terminals a and b have the same potential and are effectively shorted. During the next half-cycle, terminal d is positive with respect to c, and all four diodes open, thus opening terminals a and b. The diode bridge in Fig. 4.5a, therefore, serves as a desired electronic switch, where terminals a and b open and close periodically with carrier frequency f_c when a sinusoid $A \cos \omega_c t$ is applied across terminals c and d. To obtain the signal $m(t)w(t)$, we may place this electronic switch (terminals a and b) in series (Fig. 4.5b) or across (in parallel) $m(t)$, as shown in Fig. 4.5c. These modulators are known as the **series-bridge diode modulator** and the **shunt-bridge diode modulator**, respectively. This switching on and off of $m(t)$ repeats for each cycle of the carrier, resulting in the switched signal $m(t)w(t)$, which when bandpass-filtered, yields the desired modulated signal $(2/\pi)m(t) \cos \omega_c t$.

Another switching modulator, known as the **ring modulator**, is shown in Fig. 4.6a. During the positive half-cycles of the carrier, diodes D_1 and D_3 conduct, and D_2 and D_4 are open. Hence, terminal a is connected to c, and terminal b is connected to d. During the negative half-cycles of the carrier, diodes D_1 and D_3 are open, and D_2 and D_4 are conducting, thus connecting terminal a to d and terminal b to c. Hence, the output is proportional to $m(t)$ during the positive half-cycle and to $-m(t)$ during the negative half-cycle. In effect, $m(t)$ is multiplied by a square pulse train $w_0(t)$, as shown in Fig. 4.6b. The Fourier series for $w_0(t)$ can be found by using the signal $w(t)$ of Eq. (4.5) to yield $w_0(t) = 2w(t) - 1$. Therefore, we can use the Fourier series of $w(t)$ [Eq. (4.5)] to determine the Fourier series of $w_0(t)$ as

$$w_0(t) = \frac{4}{\pi}\left(\cos \omega_c t - \frac{1}{3}\cos 3\omega_c t + \frac{1}{5}\cos 5\omega_c t - \cdots\right) \tag{4.7a}$$

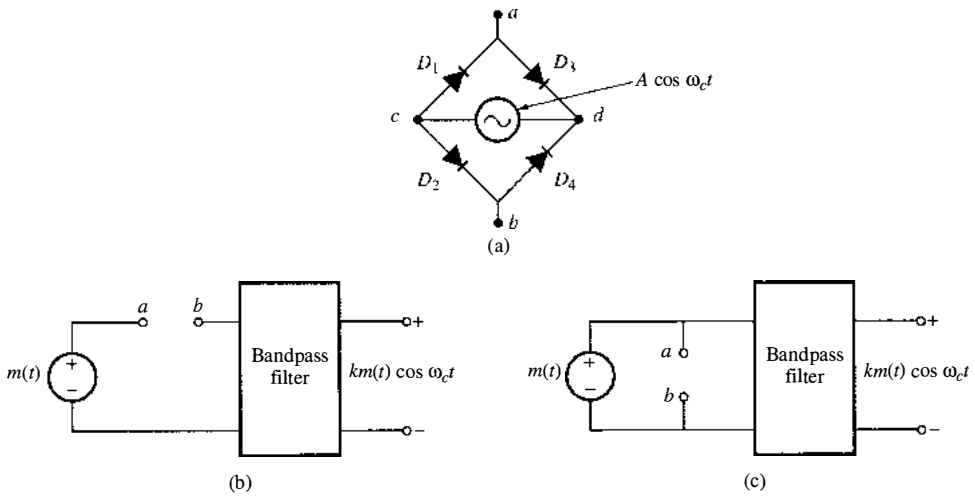


Figure 4.5
(a) Diode-bridge electronic switch. (b) Series-bridge diode modulator. (c) Shunt-bridge diode modulator.

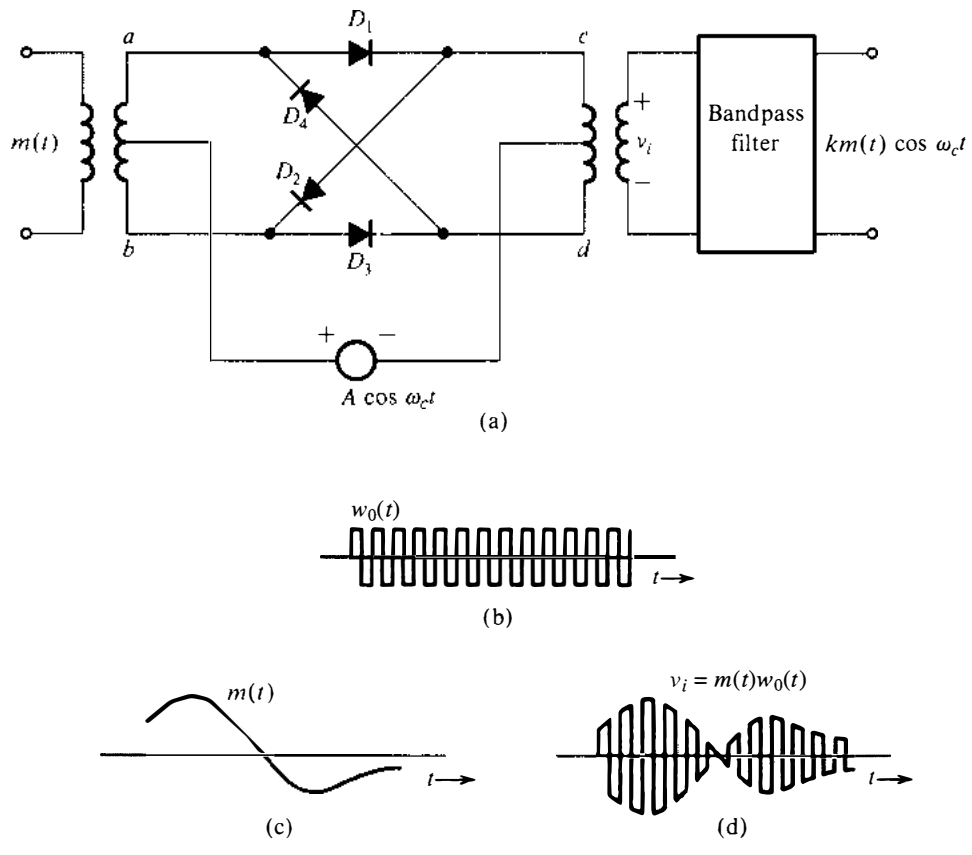


Figure 4.6
Ring modulator.

Hence, we have

$$v_i(t) = m(t)w_0(t) = \frac{4}{\pi}\left[m(t)\cos\ \omega_c t - \frac{1}{3}m(t)\cos\ 3\omega_c t + \frac{1}{5}m(t)\cos\ 5\omega_c t - \cdots\right] \quad (4.7b)$$

The signal $m(t)w_0(t)$ is shown in Fig. 4.6d. When this waveform is passed through a bandpass filter tuned to ω_c (Fig. 4.6a), the filter output will be the desired signal $(4/\pi)m(t)\cos\ \omega_c t$.

In this circuit there are two inputs: $m(t)$ and $\cos\ \omega_c t$. The input to the final bandpass filter does not contain either of these inputs. Consequently, this circuit is an example of a **double balanced modulator**.

Example 4.2 Frequency Mixer or Converter

We shall analyze a frequency mixer, or frequency converter, used to change the carrier frequency of a modulated signal $m(t)\cos\ \omega_c t$ from ω_c to another frequency ω_I.

This can be done by multiplying $m(t)\cos\ \omega_c t$ by $2\cos\ \omega_{\text{mix}} t$, where $\omega_{\text{mix}} = \omega_c + \omega_I$ or $\omega_c - \omega_I$, and then bandpass-filtering the product, as shown in Fig. 4.7a.

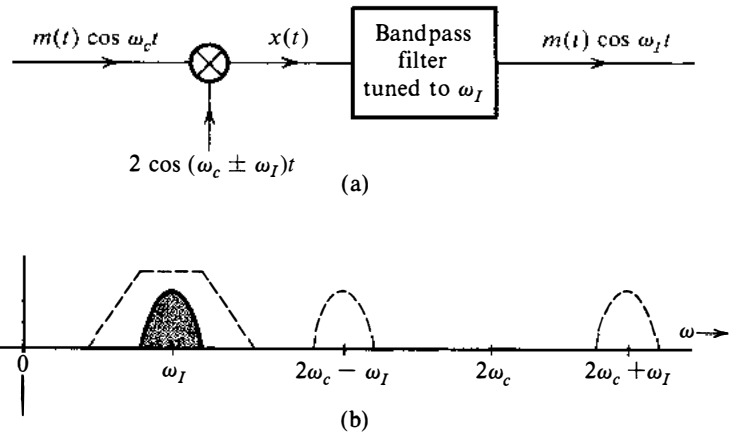


Figure 4.7 Frequency mixer or converter.

The product $x(t)$ is

$$\begin{aligned} x(t) &= 2m(t)\cos\ \omega_c t\cos\ \omega_{\text{mix}} t \\ &= m(t)[\cos\ (\omega_c - \omega_{\text{mix}})t + \cos\ (\omega_c + \omega_{\text{mix}})t] \end{aligned}$$

If we select $\omega_{\text{mix}} = \omega_c - \omega_I$, then

$$x(t) = m(t)[\cos\ \omega_I t + \cos\ (2\omega_c - \omega_I)t]$$

If we select $\omega_{\text{mix}} = \omega_c + \omega_I$, then

$$x(t) = m(t)[\cos\ \omega_I t + \cos\ (2\omega_c + \omega_I)t]$$

In either case, as long as $\omega_c - \omega_I \geq 2\pi B$ and $\omega_I \geq 2\pi B$, the various spectra in Fig. 4.7b will not overlap. Consequently, a bandpass filter at the output, tuned to ω_I, will pass the term $m(t)\cos\ \omega_I t$ and suppress the other term, yielding the output $m(t)\cos\ \omega_I t$. Thus, the carrier frequency has been translated to ω_I from ω_c.

The operation of frequency mixing/conversion (also known as heterodyning) is basically a shifting of spectra by an additional ω_{mix}. This is equivalent to the operation of modulation with a modulating carrier frequency (the mixer oscillator frequency ω_{mix}) that differs from the incoming carrier frequency by ω_I. Any one of the modulators discussed earlier can be used for frequency mixing. When we select the local carrier frequency $\omega_{\text{mix}} = \omega_c + \omega_I$, the operation is called **upconversion**, and when we select $\omega_{\text{mix}} = \omega_c - \omega_I$, the operation is **downconversion**.

Demodulation of DSB-SC Signals

As discussed earlier, demodulation of a DSB-SC signal essentially involves multiplication by the carrier signal and is identical to modulation (see Fig. 4.1). At the receiver, we multiply the incoming signal by a local carrier of frequency and phase in synchronism with the incoming carrier. The product is then passed through a low-pass filter. The **only difference** between the modulator and the demodulator lies in the input signal and the output filter. In the modulator, message $m(t)$ is the input while the multiplier output is passed through a bandpass filter tuned to ω_c, whereas in the demodulator, the DSB-SC signal is the input while the multiplier output is passed through a low-pass filter. Therefore, all the modulators discussed earlier without multipliers can also be used as demodulators, provided the bandpass filters at the output are replaced by low-pass filters of bandwidth B.

For demodulation, the receiver must generate a carrier in phase and frequency synchronism with the incoming carrier. These demodulators are synonymously called **synchronous** or **coherent** (also **homodyne**) demodulators.

Example 4.3 Analyze the switching demodulator that uses the electronic switch (diode bridge) in Fig. 4.5a as a switch (either in series or in parallel).

The input signal is $m(t)\cos\,\omega_c t$. The carrier causes the periodic switching on and off of the input signal. Therefore, the output is $m(t)\cos\,\omega_c t \times w(t)$. Using the identity $\cos\,x\,\cos\,y = 0.5[\cos\,(x+y) + \cos\,(x-y)]$, we obtain

$$\begin{aligned} m(t)\cos\,\omega_c t \times w(t) &= m(t)\cos\,\omega_c t\left[\frac{1}{2} + \frac{2}{\pi}\left(\cos\,\omega_c t - \frac{1}{3}\cos\,3\omega_c t + \cdots\right)\right] \\ &= \frac{2}{\pi}m(t)\cos^2\omega_c t + \text{terms of the form } m(t)\cos\,n\omega_c t \\ &= \frac{1}{\pi}m(t) + \frac{1}{\pi}m(t)\cos\,2\omega_c t + \text{terms of the form } m(t)\cos\,n\omega_c t \end{aligned}$$

Spectra of the terms of the form $m(t)\cos\,n\omega_c t$ are centered at $\pm n\omega_c$ and are filtered out by the low-pass filter, yielding the output $(1/\pi)m(t)$. It is left as an exercise for the reader to show that the output of the ring circuit in Fig. 4.6a operating as a demodulator (with the low-pass filter at the output) is $(2/\pi)m(t)$ (twice that of the switching demodulator in this example).

4.3 AMPLITUDE MODULATION (AM)

In the last section, we began our discussion of amplitude modulation by introducing the DSB-SC amplitude modulation because it is easy to understand and to analyze in both the time

and frequency domains. However, analytical simplicity does not always equate to simplicity in practical implementation. The (coherent) demodulation of a DSB-SC signal requires the receiver to possess a carrier signal that is synchronized with the incoming carrier. This requirement is not easy to achieve in practice. Because the modulated signal may have traveled hundreds of miles and could even suffer from some unknown frequency shift, the bandpass received signal in fact has the form of

$$r(t) = A_c m(t - t_0) \cos [(\omega_c + \Delta\omega)(t - t_0)] = A_c m(t - t_0) \cos [(\omega_c + \Delta\omega)t - \theta_d)]$$

in which $\Delta\omega$ represents the Doppler effect while

$$\theta_d = (\omega_c + \Delta\omega)t_d$$

comes from the unknown delay t_0. To utilize the coherent demodulator, the receiver must be sophisticated enough to generate a local oscillator $\cos [(\omega_c + \Delta\omega)t - \theta_d)]$ purely from the received signal $r(t)$. Such a receiver would be harder to implement and could be quite costly. This cost is particularly to be avoided in broadcasting systems, which have many receivers for every transmitter.

The alternative to a coherent demodulator is for the transmitter to send a carrier $A \cos \omega_c t$ [along with the modulated signal $m(t) \cos \omega_c t$] so that there is no need to generate a carrier at the receiver. In this case the transmitter needs to transmit at a much higher power level, which increases its cost as a trade-off. In point-to-point communications, where there is one transmitter for every receiver, substantial complexity in the receiver system can be justified, provided its cost is offset by a less expensive transmitter. On the other hand, for a broadcast system with a huge number of receivers for each transmitter, it is more economical to have one expensive high-power transmitter and simpler, less expensive receivers because any cost saving at the receiver is multiplied by the number of receiver units. For this reason, broadcasting systems tend to favor the trade-off by migrating cost from the (many) receivers to the (fewer) transmitters.

The second option of transmitting a carrier along with the modulated signal is the obvious choice in broadcasting because of its desirable trade-offs. This leads to the so-called AM (amplitude modulation), in which the transmitted signal $\varphi_{\text{AM}}(t)$ is given by

$$\varphi_{\text{AM}}(t) = A \cos \omega_c t + m(t) \cos \omega_c t \tag{4.8a}$$

$$= [A + m(t)] \cos \omega_c t \tag{4.8b}$$

The spectrum of $\varphi_{\text{AM}}(t)$ is basically the same as that of $\varphi_{\text{DSB-SC}}(t) = m(t) \cos \omega_c t$ except for the two additional impulses at $\pm f_c$,

$$\varphi_{\text{AM}}(t) \iff \frac{1}{2}[M(f + f_c) + M(f - f_c)] + \frac{A}{2}[\delta(f + f_c) + \delta(f - f_c)] \tag{4.8c}$$

Upon comparing $\varphi_{\text{AM}}(t)$ with $\varphi_{\text{DSB-SC}}(t) = m(t) \cos \omega_c t$, it is clear that the AM signal is identical to the DSB-SC signal with $A + m(t)$ as the modulating signal [instead of $m(t)$]. The value of A is always chosen to be positive. Therefore, to sketch $\varphi_{\text{AM}}(t)$, we sketch the envelope $|A + m(t)|$ and its mirror image $-|A + m(t)|$ and fill in between with the sinusoid of the carrier frequency f_c. The size of A affects the time domain envelope of the modulated signal.

Two cases are considered in Fig. 4.8. In the first case, A is large enough that $A + m(t) \geq 0$ is always nonnegative. In the second case, A is not large enough to satisfy this condition. In the first case, the envelope has the same shape as $m(t)$ (although riding on a dc of magnitude A). In the second case, the envelope shape differs from the shape of $m(t)$ because the negative

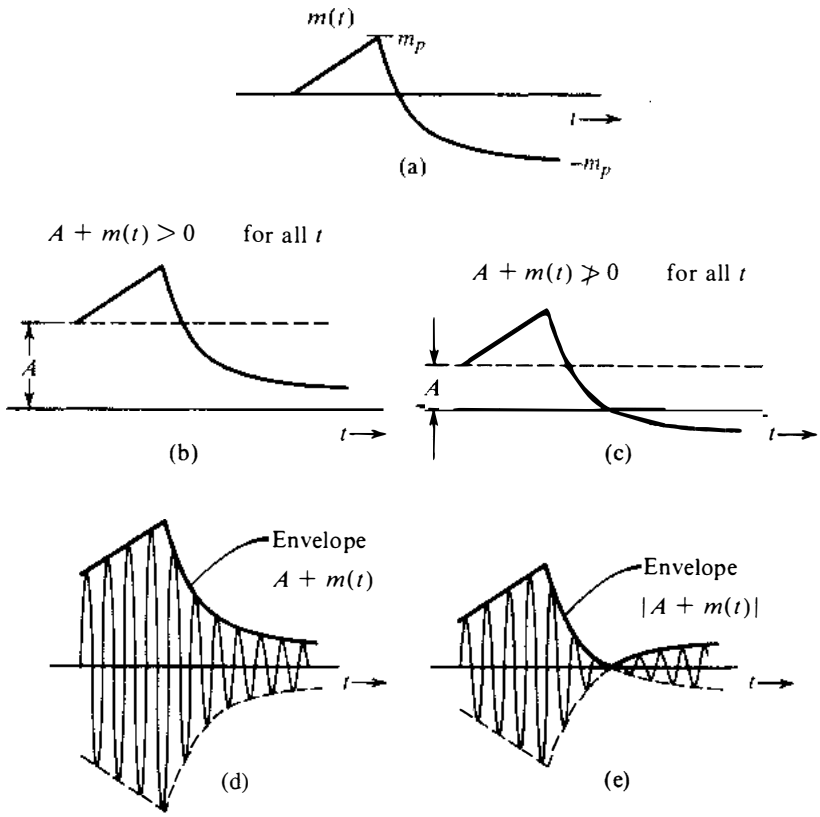


Figure 4.8
AM signal and its envelope.

part of $A + m(t)$ is rectified. This means we can detect the desired signal $m(t)$ by detecting the envelope in the first case when $A + m(t) > 0$. Such detection is not possible in the second case. We shall see that envelope detection is an extremely simple and inexpensive operation, which does not require generation of a local carrier for the demodulation. But as seen earlier, the AM envelope has the information about $m(t)$ only if the AM signal $[A + m(t)] \cos \omega_c t$ satisfies the condition $A + m(t) > 0$ for all t.

Let us now be more precise about the definition of "envelope." Consider a signal $E(t) \cos \omega_c t$. If $E(t)$ varies slowly in comparison with the sinusoidal carrier $\cos \omega_c t$, then the **envelope** of $E(t) \cos \omega_c t$ is $|E(t)|$. This means [see Eq. (4.8b)] that if and only if $A + m(t) \geq 0$ for all t, the envelope of $\varphi_{\text{AM}}(t)$ is

$$|A + m(t)| = A + m(t)$$

In other words, for envelope detection to properly detect $m(t)$, two conditions must be met:

(a) $f_c \gg$ bandwidth of $m(t)$

(b) $A + m(t) \geq 0$

This conclusion is readily verified from Fig. 4.8d and e. In Fig. 4.8d, where $A + m(t) \geq 0$, $A + m(t)$ is indeed the envelope, and $m(t)$ can be recovered from this envelope. In Fig. 4.8e, where $A + m(t)$ is not always positive, the envelope $|A + m(t)|$ is rectified from $A + m(t)$, and $m(t)$ cannot be recovered from the envelope. Consequently, demodulation of $\varphi_{\text{AM}}(t)$ in Fig. 4.8d amounts to simple envelope detection. Thus, the **condition for envelope detection**

of an AM signal is

$$A + m(t) \geq 0 \qquad \text{for all } t \tag{4.9a}$$

If $m(t) \geq 0$ for all t, then $A = 0$ already satisfies condition (4.9a). In this case there is no need to add any carrier because the envelope of the DSB-SC signal $m(t) \cos \omega_c t$ is $m(t)$ and such a DSB-SC signal can be detected by envelope detection. In the following discussion we assume that $m(t) \not\geq 0$ for all t; that is, $m(t)$ can be negative over some range of t.

Message Signals $m(t)$ with Zero Offset: Let $\pm m_p$ be the maximum and the minimum values of $m(t)$, respectively (see Fig. 4.8). This means that $m(t) \geq -m_p$. Hence, the condition of envelope detection (4.9a) is equivalent to

$$A \geq -m_{\min} \tag{4.9b}$$

Thus, the minimum carrier amplitude required for the viability of envelope detection is m_p. This is quite clear from Fig. 4.8. We define the modulation index μ as

$$\mu = \frac{m_p}{A} \tag{4.10a}$$

For envelope detection to be distortionless, the condition is $A \geq m_p$. Hence, it follows that

$$0 \leq \mu \leq 1 \tag{4.10b}$$

is the required condition for the distortionless demodulation of AM by an envelope detector.

When $A < m_p$, Eq. (4.10a) shows that $\mu > 1$ (overmodulation). In this case, the option of envelope detection is no longer viable. We then need to use synchronous demodulation. Note that synchronous demodulation can be used for any value of μ, since the demodulator will recover signal $A + m(t)$. Only an additional dc block is needed to remove the DC voltage A. The envelope detector, which is considerably simpler and less expensive than the synchronous detector, can be used only for $\mu \leq 1$.

Message Signals $m(t)$ with Nonzero Offset: On rare occasions, the message signal $m(t)$ will have a nonzero offset such that its maximum $m_{\max}$ and its minimum $m_{\min}$ are not symmetric, that is,

$$m_{\min} \neq -m_{\max}$$

In this case, it can be recognized that any offset to the envelope does not change the shape of the envelope detector output. In fact, one should note that constant offset does not carry any fresh information.

In this case, envelope detection would still remain distortionless if

$$0 \leq \mu \leq 1 \tag{4.11a}$$

with a modified modulation index definition of

$$\mu = \frac{m_{\max} - m_{\min}}{2A + m_{\max} + m_{\min}} \tag{4.11b}$$

Example 4.4 Sketch $\varphi_{\text{AM}}(t)$ for modulation indices of $\mu = 0.5$ and $\mu = 1$, when $m(t) = b \cos \omega_m t$. This case is referred to as **tone modulation** because the modulating signal is a pure sinusoid (or tone).

In this case, $m_{\max} = b$ and $m_{\min} = -b$. Hence the modulation index according to Eq. (4.10a) is

$$\mu = \frac{b - (-b)}{2A + b + (-b)} = \frac{b}{A}$$

Hence, $b = \mu A$ and

$$m(t) = b \cos \omega_m t = \mu A \cos \omega_m t$$

Therefore,

$$\varphi_{\text{AM}}(t) = [A + m(t)] \cos \omega_c t = A[1 + \mu \cos \omega_m t] \cos \omega_c t$$

Figure 4.9 shows the modulated signals corresponding to $\mu = 0.5$ and $\mu = 1$, respectively.

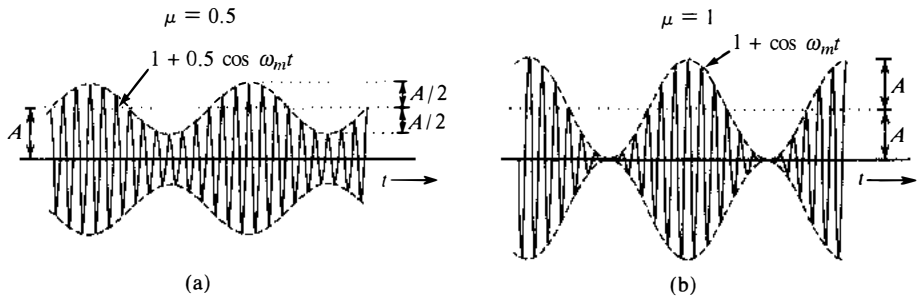


Figure 4.9 Tone-modulated AM. (a) $\mu = 0.5$. (b) $\mu = 1$.

Sideband and Carrier Power

The advantage of envelope detection in AM comes at a price. In AM, the carrier term does not carry any information, and hence, the carrier power is wasteful from this point of view:

$$\varphi_{\text{AM}}(t) = \underbrace{A \cos \omega_c t}_{\text{carrier}} + \underbrace{m(t) \cos \omega_c t}_{\text{sidebands}}$$

The carrier power P_c is the mean square value of $A \cos \omega_c t$, which is $A^2/2$. The sideband power P_s is the power of $m(t) \cos \omega_c t$, which is $0.5\ \overline{m^2(t)}$ [see Eq. (3.93)]. Hence,

$$P_c = \frac{A^2}{2} \quad \text{and} \quad P_s = \frac{1}{2}\overline{m^2(t)}$$

The useful message information resides in the sideband power, whereas the carrier power is the used for convenience in modulation and demodulation. The total power is the sum of the carrier (wasted) power and the sideband (useful) power. Hence, η, the power efficiency, is

$$\eta = \frac{\text{useful power}}{\text{total power}} = \frac{P_s}{P_c + P_s} = \frac{\overline{m^2(t)}}{A^2 + \overline{m^2(t)}} 100\%$$

For the special case of tone modulation,

$$m(t) = \mu A \cos \omega_m t \quad \text{and} \quad \overline{m^2(t)} = \frac{(\mu A)^2}{2}$$

Hence

$$\eta = \frac{\mu^2}{2+\mu^2}100\%$$

with the condition that $0 \leq \mu \leq 1$. It can be seen that η increases monotonically with μ, and $\eta_{\max}$ occurs at $\mu = 1$, for which

$$\eta_{\max} = 33\%$$

Thus, for tone modulation, under the best conditions ($\mu = 1$), only one-third of the transmitted power is used for carrying messages. For practical signals, the efficiency is even worse—on the order of 25% or lower—compared with the DSB-SC case. The best condition implies $\mu = 1$. Smaller values of μ degrade efficiency further. For this reason, volume compression and peak limiting are commonly used in AM to ensure that full modulation ($\mu = 1$) is maintained most of the time.

Example 4.5 Determine η and the percentage of the total power carried by the sidebands of the AM wave for tone modulation when $\mu = 0.5$ and when $\mu = 0.3$.

For $\mu = 0.5$,

$$\eta = \frac{\mu^2}{2+\mu^2}100\% = \frac{(0.5)^2}{2+(0.5)^2}100\% = 11.11\%$$

Hence, only about 11% of the total power is in the sidebands. For $\mu = 0.3$,

$$\eta = \frac{(0.3)^2}{2+(0.3)^2}100\% = 4.3\%$$

Hence, only 4.3% of the total power is in the sidebands that contain the message signal.

Generation of AM Signals

In principle, the generation of AM signals is identical to that of the DSB-SC modulations discussed in Sec. 4.2 except that an additional carrier component $A \cos \omega_c t$ needs to be added to the DSB-SC signal.

Demodulation of AM Signals

Like DSB-SC signals, the AM signal can be demodulated coherently by a locally generated carrier. Coherent, or synchronous, demodulation of AM, however, defeats the purpose of AM because it does not take advantage of the additional carrier component $A \cos \omega_c t$. As we have seen earlier, in the case of $\mu \leq 1$, the envelope of the AM signal follows the message signal $m(t)$. Hence, we shall consider here two noncoherent methods of AM demodulation under the condition of $0 < \mu \leq 1$: rectifier detection and envelope detection.

Rectifier Detector: If an AM signal is applied to a diode and a resistor circuit (Fig. 4.10), the negative part of the AM wave will be removed. The output across the resistor is a half-wave-rectified version of the AM signal. Visually, the diode acts like a pair of scissors by cutting off any negative half-cycle of the modulated sinusoid. In essence, at the rectifier output, the AM

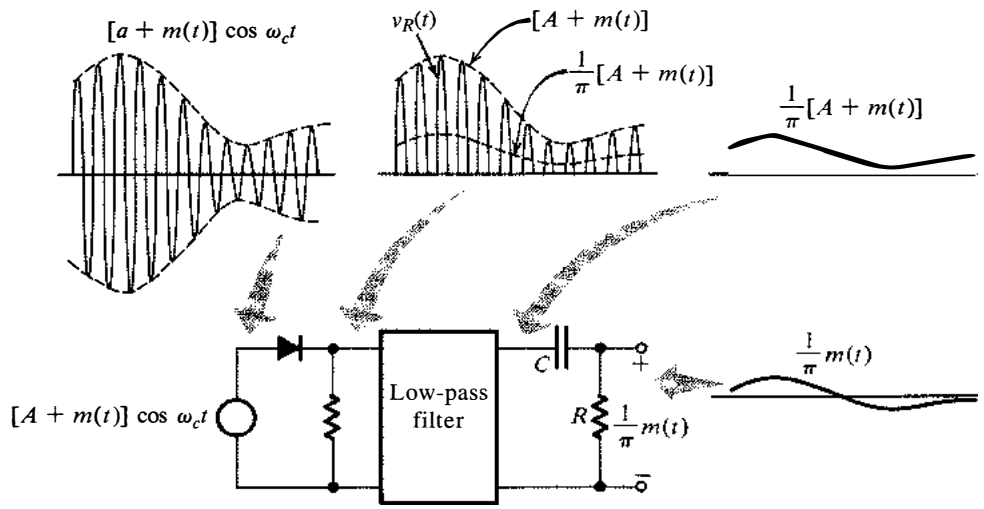


Figure 4.10
Rectifier detector for AM.

signal is multiplied by $w(t)$. Hence, the half-wave-rectified output $v_R(t)$ is

$$v_R(t) = \{[A + m(t)]\cos\ \omega_c t\}\, w(t) \tag{4.12}$$

$$= [A + m(t)]\cos\ \omega_c t\left[\frac{1}{2} + \frac{2}{\pi}\left(\cos\ \omega_c t - \frac{1}{3}\cos\ 3\omega_c t + \frac{1}{5}\cos\ 5\omega_c t - \cdots\right)\right] \tag{4.13}$$

$$= \frac{1}{\pi}[A + m(t)] + \text{other terms of higher frequencies} \tag{4.14}$$

When $v_R(t)$ is applied to a low-pass filter of cutoff B Hz, the output is $[A + m(t)]/\pi$, and all the other terms in v_R of frequencies higher than B Hz are suppressed. The dc term A/π may be blocked by a capacitor (Fig. 4.10) to give the desired output $m(t)/\pi$. The output can be doubled by using a full-wave rectifier.

It is interesting to note that because of the multiplication with $w(t)$, rectifier detection is in effect synchronous detection performed without using a local carrier. The high carrier content in AM ensures that its zero crossings are periodic and the information about the frequency and phase of the carrier at the transmitter is built in to the AM signal itself.

Envelope Detector: The output of an envelope detector follows the envelope of the modulated signal. The simple circuit shown in Fig. 4.11a functions as an envelope detector. On the positive cycle of the input signal, the input grows and may exceed the charged voltage on the capacity $v_C(t)$, turning on the diode and allowing the capacitor C to charge up to the peak voltage of the input signal cycle. As the input signal falls below this peak value, it falls quickly below the capacitor voltage (which is very nearly the peak voltage), thus causing the diode to open. The capacitor now discharges through the resistor R at a slow rate (with a time constant RC). During the next positive cycle, the same drama repeats. As the input signal rises above the capacitor voltage, the diode conducts again. The capacitor again charges to the peak value of this (new) cycle. The capacitor discharges slowly during the cutoff period.

During each positive cycle, the capacitor charges up to the peak voltage of the input signal and then decays slowly until the next positive cycle as shown in Fig. 4.11b. The output voltage $v_C(t)$, thus, closely follows the (rising) envelope of the input AM signal. Equally important, the slow capacity discharge via the resistor R allows the capacity voltage to follow a declining

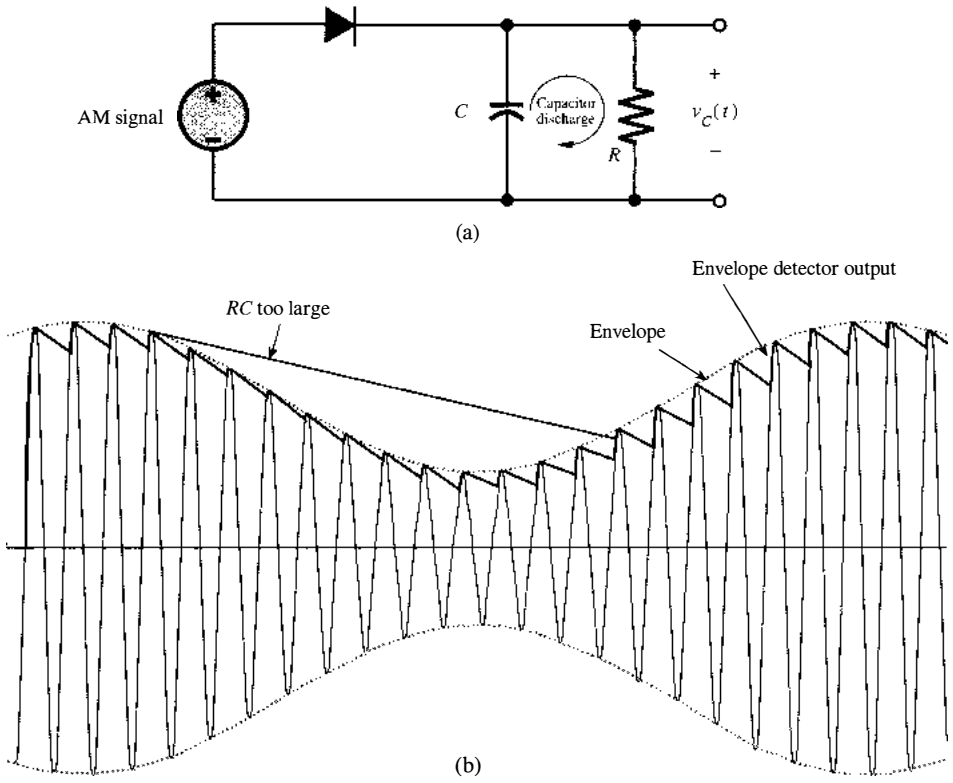


Figure 4.11
Envelope detector for AM.

envelope. Capacitor discharge between positive peaks causes a ripple signal of frequency ω_c in the output. This ripple can be reduced by choosing a larger time constant RC so that the capacitor discharges very little between the positive peaks ($RC \gg 1/\omega_c$). Picking RC too large, however, would make it impossible for the capacitor voltage to follow a fast-declining envelope (see Fig. 4.11b). Because the maximum rate of AM envelope decline is dominated by the bandwidth B of the message signal $m(t)$, the design criterion of RC should be

$$1/\omega_c \ll RC < 1/(2\pi B) \qquad \text{or} \qquad 2\pi B < \frac{1}{RC} \ll \omega_c$$

The envelope detector output is $v_C(t) = A + m(t)$ with a ripple of frequency ω_c. The dc term A can be blocked out by a capacitor or a simple RC high-pass filter. The ripple may be reduced further by another (low-pass) RC filter.

4.4 BANDWIDTH-EFFICIENT AMPLITUDE MODULATIONS

As seen from Fig. 4.12, the DSB spectrum (including suppressed carrier and AM) has two sidebands: the upper sideband (USB) and the lower sideband (LSB), each containing the complete information of the baseband signal $m(t)$. As a result, for a baseband signal $m(t)$ with bandwidth B Hz, DSB modulations require twice the radio-frequency bandwidth to transmit. To improve the spectral efficiency of amplitude modulation, there exist two basic schemes to

either utilize or remove the 100% spectral redundancy:

- Single-sideband (SSB) modulation, which removes either the LSB or the USB that uses only bandwidth of B Hz for one message signal $m(t)$;
- Quadrature amplitude modulation (QAM), which utilizes the spectral redundancy by sending two messages over the same bandwidth of $2B$ Hz.

Amplitude Modulation: Single Sideband (SSB)

As shown in Fig. 4.13, either the LSB or the USB can be suppressed from the DSB signal via bandpass filtering. Such a scheme in which only one sideband is transmitted is known as **single-sideband (SSB) transmission**, and requires only one-half the bandwidth of the DSB signal.

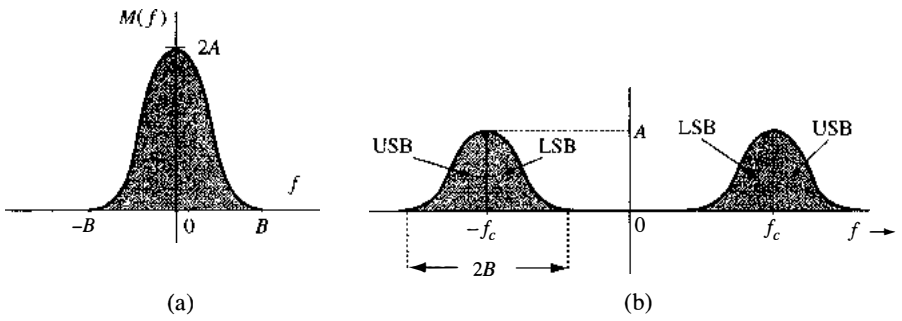


Figure 4.12 (a) Original message spectrum. (b) The redundant bandwidth consumption in DSB modulations.

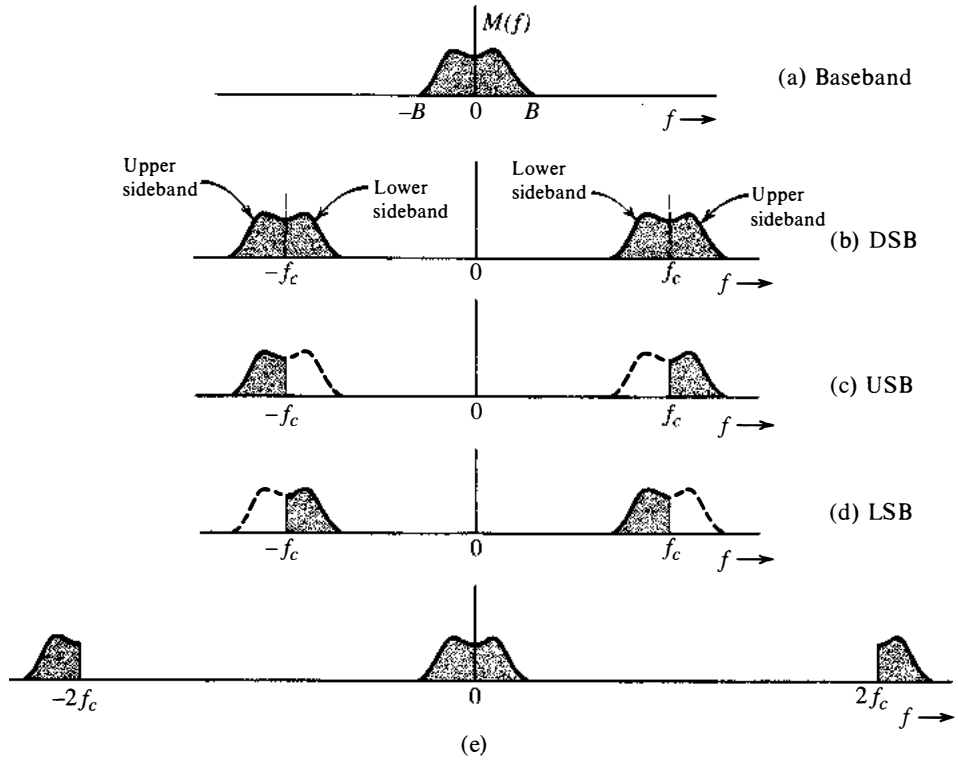


Figure 4.13 SSB spectra from suppressing one DSB sideband.

An SSB signal can be coherently (synchronously) demodulated just like DSB-SC signals. For example, multiplication of a USB signal (Fig. 4.13c) by cos $\omega_c t$ shifts its spectrum to the left and right by ω_c, yielding the spectrum in Fig. 4.13e. Low-pass filtering of this signal yields the desired baseband signal. The case is similar with LSB signals. Since the demodulation of SSB signals is identical to that of DSB-SC signals, the transmitters can now utilize only half the DSB-SC signal bandwidth without any additional cost to the receivers. Since no additional carrier accompanies the modulated SSB signal, the resulting modulator outputs are known as suppressed carrier signals (SSB-SC).

Hilbert Transform

We now introduce for later use a new tool known as the **Hilbert transform**. We use $x_h(t)$ and $\mathcal{H}\{x(t)\}$ to denote the Hilbert transform of signal $x(t)$

$$x_h(t) = \mathcal{H}\{x(t)\} = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{x(\alpha)}{t-\alpha}\, d\alpha \tag{4.15}$$

Observe that the right-hand side of Eq. (4.15) has the form of a convolution

$$x(t) * \frac{1}{\pi t}$$

Now, application of the duality property to pair 12 of Table 3.1 yields $1/\pi t \Longleftrightarrow -j\,\mathrm{sgn}\,(f)$. Hence, application of the time convolution property to the convolution (of Eq. (4.15) yields

$$X_h(f) = -jX(f)\ \mathrm{sgn}\ (f) \tag{4.16}$$

From Eq. (4.16), it follows that if $m(t)$ passes through a transfer function $H(f) = -j\ \mathrm{sgn}\ (f)$, then the output is $m_h(t)$, the Hilbert transform of $m(t)$. Because

$$H(f) = -j\ \mathrm{sgn}\ (f) \tag{4.17}$$

$$= \begin{cases} -j = 1 \cdot e^{-j\pi/2} & f > 0 \\ j = 1 \cdot e^{j\pi/2} & f < 0 \end{cases} \tag{4.18}$$

it follows that $|H(f)| = 1$ and that $\theta_h(f) = -\pi/2$ for $f > 0$ and $\pi/2$ for $f < 0$, as shown in Fig. 4.14. Thus, if we change the phase of every component of $m(t)$ by $\pi/2$ (without changing its amplitude), the resulting signal is $m_h(t)$, the Hilbert transform of $m(t)$. Therefore, a Hilbert transformer is an ideal phase shifter that shifts the phase of every spectral component by $-\pi/2$.

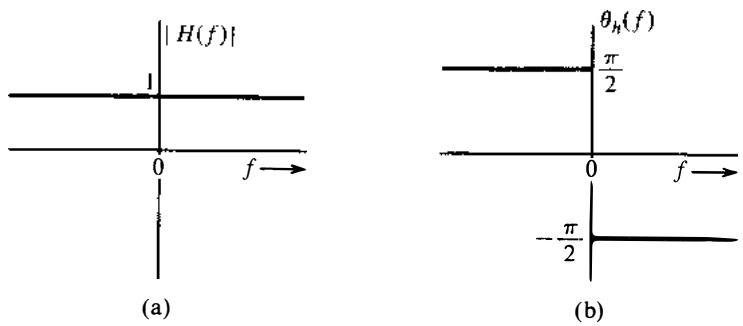


Figure 4.14 Transfer function of an ideal $\pi/2$ phase shifter (Hilbert transformer).

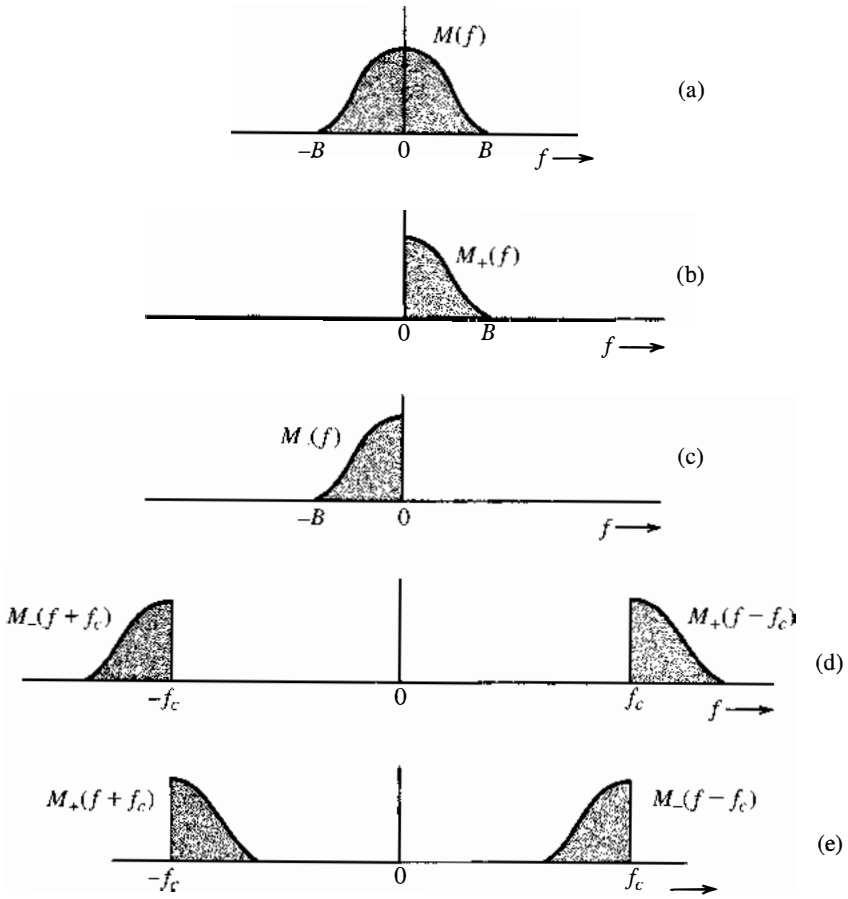


Figure 4.15 Expressing SSB spectra in terms of $M_+(f)$ and $M_-(f)$.

Time Domain Representation of SSB Signals

Because the building blocks of an SSB signal are the sidebands, we shall first obtain a time domain expression for each sideband.

Figure 4.15a shows the message spectrum $M(f)$. Figure 4.15b shows its right half $M_+(f)$, and Fig. 4.15c shows its left half $M_-(f)$. From Fig. 4.15b and c, we observe that

$$M_+(f) = M(f) \cdot u(f) = M(f)\frac{1}{2}[1 + \text{sgn}(f)] = \frac{1}{2}[M(f) + jM_h(f)] \tag{4.19a}$$

$$M_-(f) = M(f)u(-f) = M(f)\frac{1}{2}[1 - \text{sgn}(f)] = \frac{1}{2}[M(f) - jM_h(f)] \tag{4.19b}$$

We can now express the SSB signal in terms of $m(t)$ and $m_h(t)$. From Fig. 4.15d it is clear that the USB spectrum $\Phi_{\text{USB}}(f)$ can be expressed as

$$\begin{aligned}\Phi_{\text{USB}}(f) &= M_+(f - f_c) + M_-(f + f_c) \\ &= \frac{1}{2}[M(f - f_c) + M(f + f_c)] - \frac{1}{2j}[M(f - f_c) - M(f + f_c)]\end{aligned}$$

From the frequency-shifting property, the inverse transform of this equation yields

$$\varphi_{\text{USB}}(t) = m(t)\cos\ \omega_c t - m_h(t)\sin\ \omega_c t \tag{4.20a}$$

Similarly, we can show that

$$\varphi_{\text{LSB}}(t) = m(t)\cos\ \omega_c t + m_h(t)\sin\ \omega_c t \tag{4.20b}$$

Hence, a general SSB signal $\varphi_{\text{SSB}}(t)$ can be expressed as

$$\varphi_{\text{SSB}}(t) = m(t)\cos\ \omega_c t \mp m_h(t)\sin\omega_c t \tag{4.20c}$$

where the minus sign applies to USB and the plus sign applies to LSB.

Given the time domain expression of SSB-SC signals, we can now confirm analytically (instead of graphically) that SSB-SC signals can be coherently demodulated:

$$\begin{aligned}\varphi_{\text{SSB}}(t)\cos\ \omega_c t &= [m(t)\cos\ \omega_c t \mp m_h(t)\sin\omega_c t]\,2\cos\ \omega_c t\\ &= m(t)[1+\cos\ 2\omega_c t] \mp m_h(t)\sin\ 2\omega_c t\\ &= m(t) + \underbrace{[m(t)\cos\ 2\omega_c t \mp m_h(t)\sin\ 2\omega_c t]}_{\text{SSB-SC signal with carrier } 2\omega_c}\end{aligned}$$

Thus, the product $\varphi_{\text{SSB}}(t)\cdot 2\cos\ \omega_c t$ yields the baseband signal and another SSB signal with a carrier $2\omega_c$. The spectrum in Fig. 4.13e shows precisely this result. A low-pass filter will suppress the unwanted SSB terms, giving the desired baseband signal $m(t)$. Hence, the demodulator is identical to the synchronous demodulator used for DSB-SC. Thus, any one of the synchronous DSB-SC demodulators discussed earlier in Sec. 4.2 can be used to demodulate an SSB-SC signal.

Example 4.6 Tone Modulation: SSB

Find $\varphi_{\text{SSB}}(t)$ for a simple case of a tone modulation, that is, when the modulating signal is a sinusoid $m(t) = \cos\ \omega_m t$. Also demonstrate the coherent demodulation of this SSB signal.

Recall that the Hilbert transform delays the phase of each spectral component by $\pi/2$. In the present case, there is only one spectral component of frequency ω_m. Delaying the phase of $m(t)$ by $\pi/2$ yields

$$m_h(t) = \cos\left(\omega_m t - \frac{\pi}{2}\right) = \sin\ \omega_m t$$

Hence, from Eq. (4.20c),

$$\begin{aligned}\varphi_{\text{SSB}}(t) &= \cos\ \omega_m t\cos\ \omega_c t \mp \sin\ \omega_m t\sin\ \omega_c t\\ &= \cos\ (\omega_c \pm \omega_m)t\end{aligned}$$

Thus,

$$\varphi_{\text{USB}}(t) = \cos\ (\omega_c + \omega_m)t \qquad \text{and} \qquad \varphi_{\text{LSB}}(t) = \cos\ (\omega_c - \omega_m)t$$

To verify these results, consider the spectrum of $m(t)$ (Fig. 4.16a) and its DSB-SC (Fig. 4.16b), USB (Fig. 4.16c), and LSB (Fig. 4.16d) spectra. It is evident that the spectra in Fig. 4.16c and d do indeed correspond to the $\varphi_{\text{USB}}(t)$ and $\varphi_{\text{LSB}}(t)$ derived earlier.

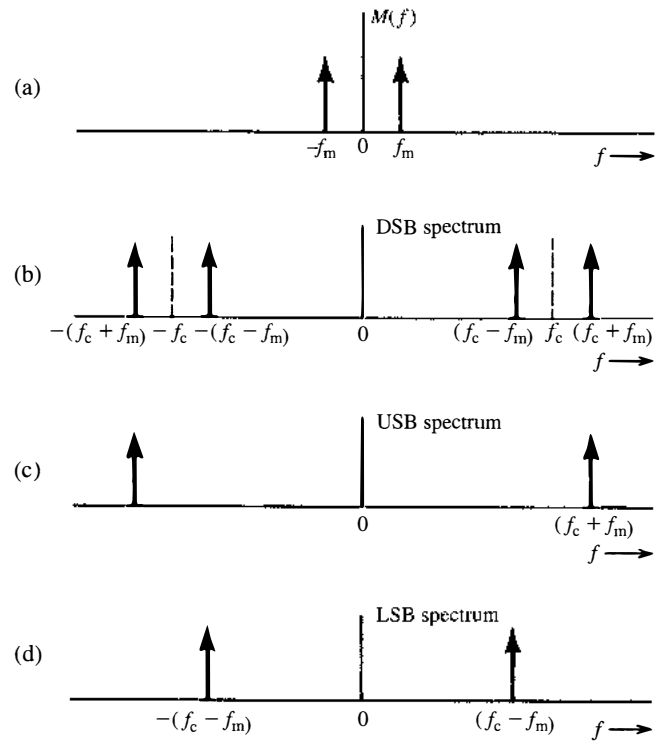


Figure 4.16 SSB spectra for tone modulation.

Finally, the coherent demodulation of the SSB tone modulation is can be achieved by

$$\begin{aligned}\varphi_{\text{SSB}}(t)2\cos\ \omega_c t &= 2\cos\ (\omega_c \pm \omega_m)t\cos\ \omega_c t\\ &= \cos\ \omega_m t + \cos\ (\omega_c + \omega_m)t\end{aligned}$$

which can be sent to a lowpass filter to retrieve the message tone $\cos\ \omega_m t$.

SSB Modulation Systems

Three methods are commonly used to generate SSB signals: phase shifting, selective filtering, and the Weaver method.[1] None of these modulation methods are precise, and all generally require that the baseband signal spectrum have little power near the origin.

The **phase shift method** directly uses Eq. (4.20) as its basis. Figure 4.17 shows its implementation. The box marked "$-\pi/2$" is a phase shifter, which delays the phase of every positive spectral component by $\pi/2$. Hence, it is a Hilbert transformer. Note that an ideal Hilbert phase shifter is unrealizable. This is because the Hilbert phase shifter requires an abrupt phase change of π at zero frequency. When the message $m(t)$ has a dc null and very little low-frequency content, the practical approximation of this ideal phase shifter has almost no real effect and does not affect the accuracy of SSB modulation.

In the **selective-filtering method**, the most commonly used method of generating SSB signals, a DSB-SC signal is passed through a sharp cutoff filter to eliminate the undesired sideband. To obtain the USB, the filter should pass all components above frequency f_c unattenuated and completely suppress all components below f_c. Such an operation requires an ideal filter, which is unrealizable. It can, however, be approximated closely if there is some separation

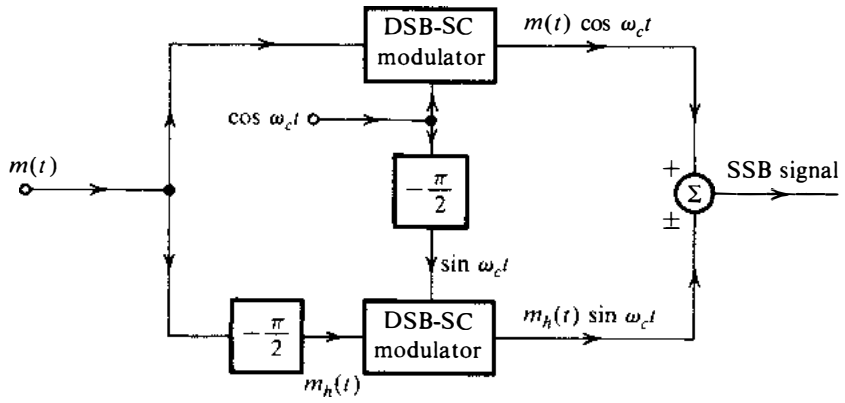


Figure 4.17 Generating SSB using the phase shift method.

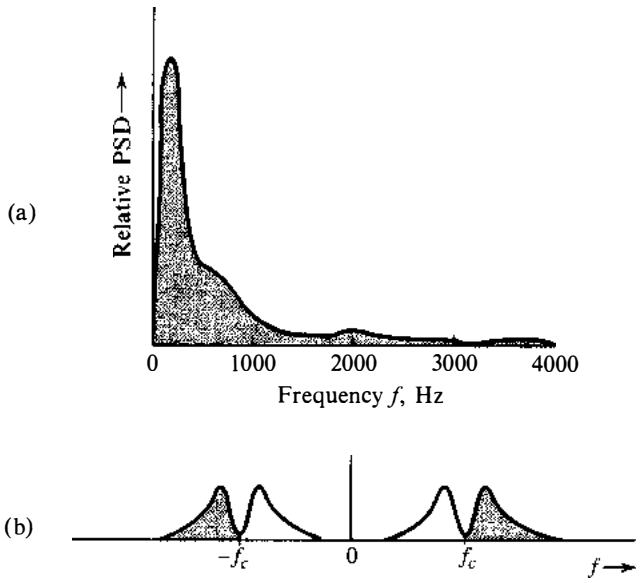


Figure 4.18 (a) Relative power spectrum of speech signal and (b) corresponding USB spectrum.

between the passband and the stopband. Fortunately, the voice signal provides this condition, because its spectrum shows little power content at the origin (Fig. 4.18a). In addition, articulation tests have shown that for speech signals, frequency components below 300 Hz are not important. In other words, we may suppress all speech components below 300 Hz (and above 3500 Hz) without affecting intelligibility appreciably. Thus, filtering of the unwanted sideband becomes relatively easy for speech signals because we have a 600 Hz transition region around the cutoff frequency f_c. To minimize adjacent channel interference, the undesired sideband should be attenuated at least 40 dB.

For very high carrier frequency f_c, the ratio of the gap band (600 Hz) to the carrier frequency may be too small, and, thus, a transition of 40 dB in amplitude over 600 Hz may be difficult. In such a case, a third method, known as **Weaver's method**,[1] utilizes two stages of SSB amplitude modulation. First, the modulation is carried out by using a smaller carrier frequency (f_{c_1}). The resulting SSB signal effectively widens the gap to $2f_{c_1}$ (see shaded spectra in Fig. 4.18b). Now by treating this signal as the new baseband signal, it is possible to achieve SSB-modulation at a higher carrier frequency.

Detection of SSB Signals with a Carrier (SSB+C)

We now consider SSB signals with an additional carrier (SSB+C). Such a signal can be expressed as

$$\varphi_{\text{SSB+C}} = A\cos\ \omega_c t + [m(t)\cos\ \omega_c t + m_h(t)\sin\ \omega_c t]$$

and $m(t)$ can be recovered by synchronous detection [multiplying $\varphi_{\text{SSB+C}}$ by $\cos\ \omega_c t$] if the carrier component $A\cos\ \omega_c t$ can be extracted (by narrowband filtering of) $\varphi_{\text{SSB+C}}$. Alternatively, if the carrier amplitude A is large enough, $m(t)$ can also be (approximately) recovered from $\varphi_{\text{SSB+C}}$ by envelope or rectifier detection. This can be shown by rewriting $\varphi_{\text{SSB+C}}$ as

$$\begin{aligned}\varphi_{\text{SSB+C}} &= [A + m(t)]\cos\ \omega_c t + m_h(t)\sin\ \omega_c t \\ &= E(t)\cos\ \ (\omega_c t + \theta)\end{aligned} \tag{4.21}$$

where $E(t)$, the envelope of $\varphi_{\text{SSB+C}}$, is given by [see Eq. (3.41a)]

$$\begin{aligned}E(t) &= \{[A + m(t)]^2 + m_h^2(t)\}^{1/2} \\ &= A\left[1 + \frac{2m(t)}{A} + \frac{m^2(t)}{A^2} + \frac{m_h^2(t)}{A^2}\right]^{1/2}\end{aligned}$$

If $A \gg |m(t)|$, then in general* $A \gg |m_h(t)|$, and the terms $m^2(t)/A^2$ and $m_h^2(t)/A^2$ can be ignored. Thus,

$$E(t) \simeq A\left[1 + \frac{2m(t)}{A}\right]^{1/2}$$

Using Taylor series expansion and discarding higher order terms [because $m(t)/A \ll 1$], we get

$$\begin{aligned}E(t) &\simeq A\left[1 + \frac{m(t)}{A}\right] \\ &= A + m(t)\end{aligned}$$

It is evident that for a large carrier, the SSB + C can be demodulated by an envelope detector.

In AM, envelope detection requires the condition $A \geq |m(t)|$, whereas for SSB+C, the condition is $A \gg |m(t)|$. Hence, in SSB case, the required carrier amplitude is much larger than that in AM, and, consequently, the efficiency of SSB+C is pathetically low.

Quadrature Amplitude Modulation (QAM)

Because SSB-SC signals are difficult to generate accurately, quadrature amplitude modulation (QAM) offers an attractive alternative to SSB-SC. QAM can be exactly generated without requiring sharp-cutoff bandpass filters. QAM operates by transmitting two DSB signals using carriers of the same frequency but in phase quadrature, as shown in Fig. 4.19. This scheme is known as **quadrature amplitude modulation (QAM)** or **quadrature multiplexing**.

As shown Figure 4.19, the boxes labeled $-\pi/2$ are phase shifters that delay the phase of an input sinusoid by $-\pi/2$ rad. If the two baseband message signals for transmission are $m_1(t)$ and $m_2(t)$, the corresponding QAM signal $\varphi_{\text{QAM}}(t)$, the sum of the two DSB-modulated signals, is

$$\varphi_{\text{QAM}}(t) = m_1(t)\cos\ \omega_c t + m_2(t)\sin\ \omega_c t$$

* This may not be true for all t, but it is true for most t.

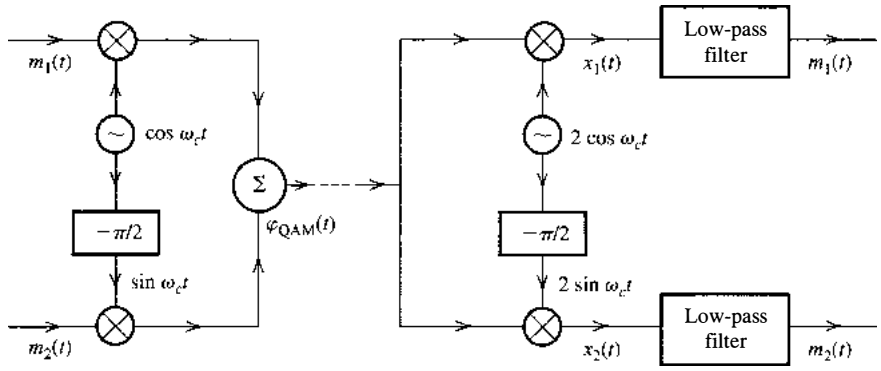


Figure 4.19 Quadrature amplitude multiplexing.

Both modulated signals occupy the same band. Yet two baseband signals can be separated at the receiver by synchronous detection if two local carriers are used in phase quadrature, as shown in Fig. 4.19. This can be shown by considering the multiplier output $x_1(t)$ of the upper arm of the receiver (Fig. 4.19):

$$\begin{aligned} x_1(t) &= 2\varphi_{\text{QAM}}(t)\cos\ \omega_c t = 2[m_1(t)\cos\ \omega_c t + m_2(t)\sin\ \omega_c t]\cos\ \omega_c t \\ &= m_1(t) + m_1(t)\cos\ 2\omega_c t + m_2(t)\sin\ 2\omega_c t \end{aligned} \tag{4.22a}$$

The last two terms are bandpass signals centered around $2\omega_c$. In fact, they actually form a QAM signal with $2\omega_c$ as the carrier frequency. They are suppressed by the low-pass filter, yielding the desired demodulation output $m_1(t)$. Similarly, the output of the lower receiver branch can be shown to be $m_2(t)$.

$$\begin{aligned} x_2(t) &= 2\varphi_{\text{QAM}}(t)\sin\omega_c t = 2[m_1(t)\cos\ \omega_c t + m_2(t)\sin\ \omega_c t]\sin\ \omega_c t \\ &= m_2(t) - m_2(t)\cos\ 2\omega_c t + m_1(t)\sin\ 2\omega_c t \end{aligned} \tag{4.22b}$$

Thus, two baseband signals, each of bandwidth B Hz, can be transmitted simultaneously over a bandwidth $2B$ by using DSB transmission and quadrature multiplexing. The upper channel is also known as the **in-phase** (I) channel and the lower channel is the **quadrature** (Q) channel. Both signals $m_1(t)$ and $m_2(t)$ can be separately demodulated.

Note, however, that QAM demodulation must be totally synchronous. An error in the phase or the frequency of the carrier at the demodulator in QAM will result in loss and interference between the two channels. To show this, let the carrier at the demodulator be $2\cos(\omega_c t + \theta)$. In this case,

$$\begin{aligned} x_1(t) &= 2[m_1(t)\cos\ \omega_c t + m_2(t)\sin\ \omega_c t]\cos\ (\omega_c t + \theta) \\ &= m_1(t)\cos\theta - m_2(t)\sin\ \theta + m_1(t)\cos\ (2\omega_c t + \theta) + m_2(t)\sin\ (2\omega_c t + \theta) \end{aligned}$$

The low-pass filter suppresses the two signals modulated by carrier of angular frequency $2\omega_c$, resulting in the first demodulator output

$$m_1(t)\cos\theta - m_2(t)\sin\theta$$

Thus, in addition to the desired signal $m_1(t)$, we also receive signal $m_2(t)$ in the upper receiver branch. A similar phenomenon can be shown for the lower branch. This so-called **cochannel interference** is undesirable. Similar difficulties arise when the local frequency is in error (see

Prob. 4.4-1). In addition, unequal attenuation of the USB and the LSB during transmission leads to cross talk or cochannel interference.

Quadrature multiplexing is used in analog color television to multiplex the so-called chrominance signals, which carry the information about colors. There, the synchronization is achieved by periodic insertion of a short burst of carrier signal (called **color burst** in the transmitted signal). Digital satellite television transmission also applies QAM.

In terms of bandwidth requirement, SSB is similar to QAM but less exacting in terms of the carrier frequency and phase or the requirement of a distortionless transmission medium. However, SSB is difficult to generate if the baseband signal $m(t)$ has significant spectral content near the dc.

4.5 AMPLITUDE MODULATIONS: VESTIGIAL SIDEBAND (VSB)

As discussed earlier, it is rather difficult to generate exact SSB signals. They generally require that the message signal $m(t)$ have a null around dc. A phase shifter, required in the phase shift method, is unrealizable, or only approximately realizable. The generation of DSB signals is much simpler, but it requires twice the signal bandwidth. **Vestigial sideband (VSB)** modulation, also called the asymmetric sideband system, is a compromise between DSB and SSB. It inherits the advantages of DSB and SSB but avoids their disadvantages at a small cost. VSB signals are relatively easy to generate, and, at the same time, their bandwidth is only a little (typically 25%) greater than that of SSB signals.

In VSB, instead of rejecting one sideband completely (as in SSB), a gradual cutoff of one sideband as shown in Fig. 4.20d, is accepted. The baseband signal can be recovered exactly by a synchronous detector in conjunction with an appropriate equalizer filter $H_o(f)$ at the receiver output (Fig. 4.21). If a large carrier is transmitted along with the VSB signal, the baseband signal can be recovered by an envelope (or a rectifier) detector.

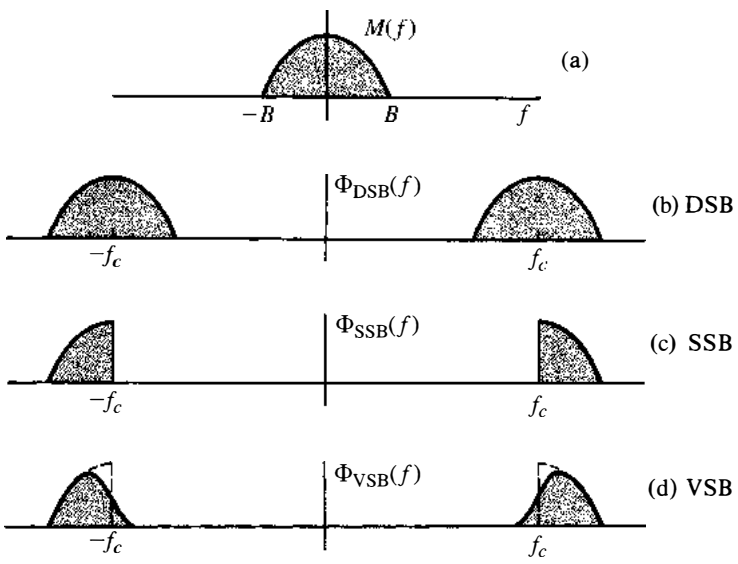


Figure 4.20 Spectra of the modulating signal and corresponding DSB, SSB, and VSB signals.

If the vestigial shaping filter that produces VSB from DSB is $H_i(f)$ (Fig. 4.21), then the resulting VSB signal spectrum is

$$\Phi_{VSB}(f) = [M(f+f_c) + M(f-f_c)]H_i(f) \tag{4.23}$$

This VSB shaping filter $H_i(f)$ allows the transmission of one sideband but suppresses the other sideband, not completely, but gradually. This makes it easy to realize such a filter, but the transmission bandwidth is now somewhat higher than that of the SSB (where the other sideband is suppressed completely). The bandwidth of the VSB signal is typically 25 to 33% higher than that of the SSB signals.

We require that $m(t)$ be recoverable from $\varphi_{VSB}(t)$ by using synchronous demodulation at the receiver. This is done by multiplying the incoming VSB signal $\varphi_{VSB}(t)$ by $2\cos \omega_c t$. The product $e(t)$ is given by

$$e(t) = 2\varphi_{VSB}(t)\cos \omega_c t \Longleftrightarrow [\Phi_{VSB}(f+f_c) + \Phi_{VSB}(f-f_c)]$$

The signal $e(t)$ is further passed through the low-pass equalizer filter of the transfer function $H_o(f)$. The output of the equalizer filter is required to be $m(t)$. Hence, the output signal spectrum is given by

$$M(f) = [\Phi_{VSB}(f+f_c) + \Phi_{VSB}(f-f_c)]H_o(f)$$

Substituting Eq. (4.23) into this equation and eliminating the spectra at $\pm 4f_c$ [suppressed by a low-pass filter $H_o(f)$], we obtain

$$M(f) = M(f)[H_i(f+f_c) + H_i(f-f_c)]H_o(f) \tag{4.24}$$

Hence

$$H_o(f) = \frac{1}{H_i(f+f_c) + H_i(f-f_c)} \qquad |f| \le B \tag{4.25}$$

Note that because $H_i(f)$ is a bandpass filter, the terms $H_i(f \pm f_c)$ contain low-pass components.

Complementary VSB Filter and Envelope Detection of VSB + C Signals

As a special case of a filter at the VSB modulator, we can choose $H_i(f)$ such that

$$H_i(f+f_c) + H_i(f-f_c) = 1 \qquad |f| \le B \tag{4.26}$$

The output filter is just a simple low-pass filter with transfer function:

$$H_o(f) = 1 \qquad |f| \le B$$

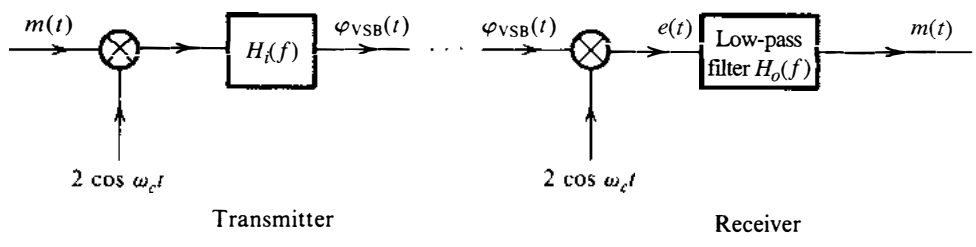


Figure 4.21 VSB modulator and demodulator.

The resulting VSB signal plus carrier (VSB + C) can be envelope-detected. This demodulation method may be proved by using exactly the same argument used in proving the case for SSB + C signals. In particular, because of Eq. (4.26), we can define a new low-pass filter

$$F(f) = j[1 - 2H_i(f - f_c)] = -j[1 - 2H_i(f + f_c)] \qquad |f| \leq B$$

Defining a new (complex) low-pass signal as

$$m_v(t) \iff M_v(f) = F(f)M(f)$$

we can rewrite the VSB signal as

$$\Phi_{\text{VSB}}(f) = \frac{M(f - f_c) + M(f + f_c)}{2} + \frac{M_v(f - f_c) - M_v(f + f_c)}{2j} \tag{4.27a}$$

$$\iff$$

$$\varphi_{\text{VSB}}(t) = m(t)\cos 2\pi f_c t + m_v(t)\sin 2\pi f_c t \tag{4.27b}$$

Clearly, both the SSB and the VSB modulated signals have the same form, with $m_h(t)$ in SSB replaced by a low-pass signal $m_v(t)$ in VSB. Applying the same analysis from the SSB+C envelope detection, a large carrier addition to $\varphi_{\text{VSB}}(t)$ would allow the envelope detection of VSB + C.

We have shown that SSB+C requires a much larger carrier than DSB+C (AM) for envelope detection. Because VSB+C is an in-between case, the added carrier required in VSB is larger than that in AM, but smaller than that in SSB + C.

Example 4.7 The carrier frequency of a certain VSB signal is $f_c = 20$ kHz, and the baseband signal bandwidth is 6 kHz. The VSB shaping filter $H_i(f)$ at the input, which cuts off the lower sideband gradually over 2 kHz, is shown in Fig. 4.22a. Find the output filter $H_o(f)$ required for distortionless reception.

Figure 4.22b shows the low-pass segments of $H_i(f + f_c) + H_i(f - f_c)$. We are interested in this spectrum only over the baseband (the remaining undesired portion is suppressed by the output filter). This spectrum, which is 0.5 over the band of 0 to 2 kHz, is 1 from 2 to 6 kHz, as shown in Fig. 4.22b. Figure 4.22c shows the desired output filter $H_o(f)$, which is the reciprocal of the spectrum in Fig. 4.22b [see Eq. (4.25)].

Use of VSB in Broadcast Television

VSB is a clever compromise between SSB and DSB, which makes it very attractive for television broadcast systems. The baseband video signal of television occupies an enormous bandwidth of 4.5 MHz, and a DSB signal needs a bandwidth of 9 MHz. It would seem desirable to use SSB to conserve bandwidth. Unfortunately, doing this creates several problems. First, the baseband video signal has sizable power in the low-frequency region, and consequently it is difficult to suppress one sideband completely. Second, for a broadcast receiver, an envelope detector is preferred over a synchronous one to reduce the receiver cost. We saw earlier that SSB+C has a very low power efficiency. Moreover, using SSB will increase the receiver cost.

The spectral shaping of television VSBs signals can be illustrated by Fig. 4.23. The vestigial spectrum is controlled by two filters: the transmitter RF filter $H_T(f)$ and the receiver RF filter

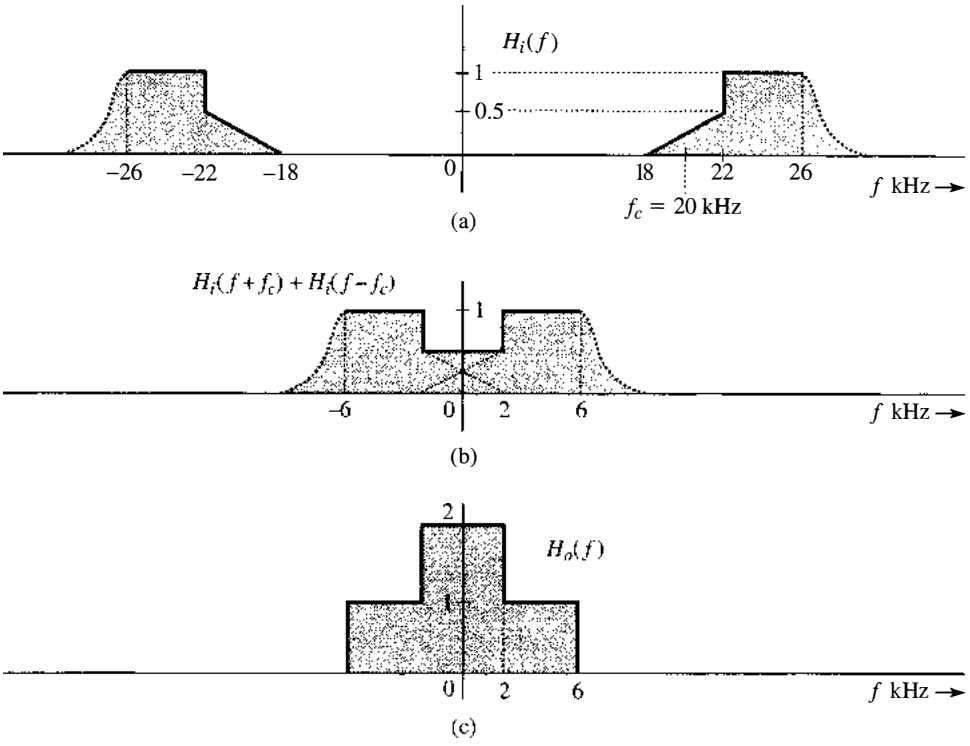


Figure 4.22 VSB modulator and receiver filters.

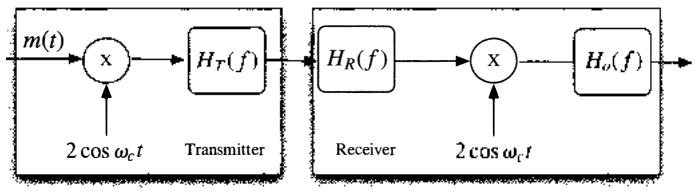


Figure 4.23 Transmitter filter $H_T(f)$, receiver front-end filter $H_R(f)$, and the receiver output low-pass filter $H_o(f)$ in VSB television systems.

$H_R(f)$. Jointly we have

$$H_i(f) = H_T(f)H_R(f)$$

Hence, the design of the receiver output filter $H_o(f)$ follows Eq. (4.25).

The DSB spectrum of a television signal is shown in Fig. 4.24a. The vestigial shaping filter $H_i(f)$ cuts off the lower sideband spectrum gradually, starting at 0.75 MHz to 1.25 MHz below the carrier frequency f_c, as shown in Fig. 4.24b. The receiver output filter $H_o(f)$ is designed according to Eq. (4.25). The resulting VSB spectrum bandwidth is 6 MHz. Compare this with the DSB bandwidth of 9 MHz and the SSB bandwidth of 4.5 MHz.

4.6 LOCAL CARRIER SYNCHRONIZATION

In a suppressed carrier, amplitude-modulated system (DSB-SC, SSB-SC, and VSB-SC), the coherent receiver must generate a local carrier that is synchronous with the incoming carrier (frequency and phase). As discussed earlier, any discrepancy in the frequency or phase of the local carrier gives rise to distortion in the detector output.

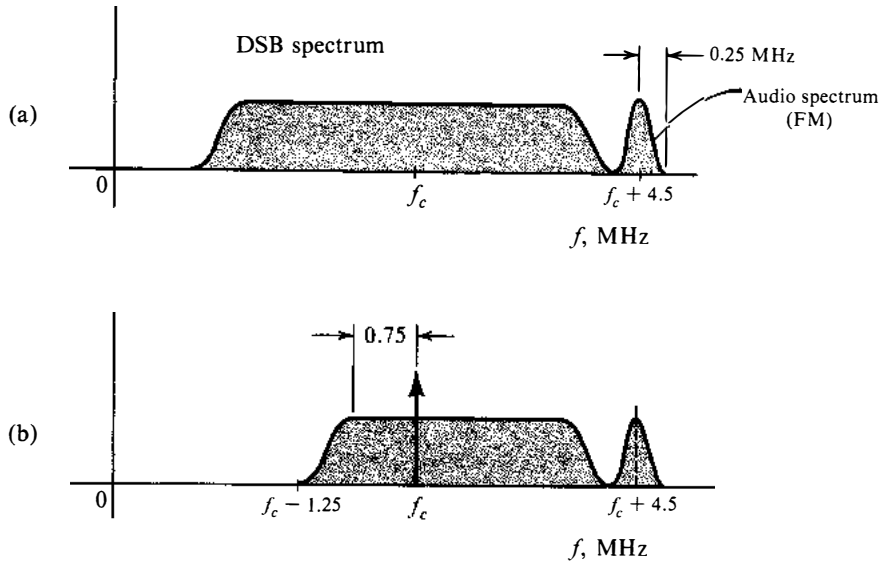


Figure 4.24
Television signal spectra: (a) DSB signal. (b) signal transmitted.

Consider an SSB-SC case where a received signal is

$$m(t)\cos\,[(\omega_c + \Delta\omega)t + \delta] - m_h(t)\sin\,[(\omega_c + \Delta\omega)t + \delta]$$

because of propagation delay and Doppler frequency shift. The local carrier remains as $2\cos\,\omega_c t$. The product of the received signal and the local carrier is $e(t)$, given by

$$\begin{aligned} e(t) &= 2\cos\,\omega_c t\,[m(t)\cos\,(\omega_c t + \Delta\omega t + \delta) - m_h(t)\sin\,(\omega_c t + \Delta\omega t + \delta)] \\ &= m(t)\cos(\Delta\omega t + \delta) - m_h(t)\sin\,(\Delta\omega t + \delta) \\ &+ \underbrace{m(t)\cos[(2\omega_c + \Delta\omega)t + \delta] - m_h(t)\sin\,[(2\omega_c + \Delta\omega)t + \delta]}_{\text{bandpass SSB-SC signal around } 2\omega_c + \Delta\omega} \end{aligned} \tag{4.28}$$

The bandpass component is filtered out by the receiver low-pass filter, leaving the output $e_o(t)$ as

$$e_o(t) = m(t)\cos\,(\Delta\omega t + \delta) - m_h(t)\sin\,(\Delta\omega t + \delta) \tag{4.29}$$

If $\Delta\omega$ and δ are both zero (no frequency or phase error), then

$$e_o(t) = m(t)$$

as expected.

In practice, if the radio wave travels a distance of d meters at the speed of light c, then the phase delay is

$$\delta = -(\omega_c + \Delta\omega)d/c$$

which can be any value within the interval $[-\pi, +\pi]$. Two oscillators initially of identical frequency can also drift apart. Moreover, if the receiver or the transmitter is traveling at a velocity of v_e, then the maximum Doppler frequency shift would be

$$\Delta f_{\max} = \frac{v_e}{c} f_c$$

The velocity v_e depends on the actual vehicles (e.g. spacecrafts, airplanes, cars). For example, if the mobile velocity v_e is 108 km/ph, then for a carrier frequency at 100 MHz, the maximum Doppler frequency shift would be 10 Hz. Such a shift of every frequency component by a fixed amount $\Delta\omega$ destroys the harmonic relationship between frequency components. For $\Delta f = 10$ Hz, the components of frequencies 1000 and 2000 Hz will be shifted to frequencies 1010 and 2010 Hz, respectively. This upsets their harmonic relationship and the quality of nonaudio signals.

It is interesting to note that audio signals are highly redundant, and unless Δf is very large, such a change does not destroy intelligibility of the output. For audio signals $\Delta f < 30$ Hz does not significantly affect the signal quality. $\Delta f > 30$ Hz results in a sound quality similar to that of Donald Duck. But the intelligibility is not completely lost.

Generally, there are two ways to recover the incoming carrier at the receiver. One way is for the transmitter to transmit a pilot (sinusoid) signal that can be either the exact carrier or directly related to the carrier (e.g., a pilot at half the carrier frequency). The pilot is separated at the receiver by a very narrowband filter tuned to the pilot frequency. It is amplified and used to synchronize the local oscillator. Another method, in which no pilot is transmitted, is for the receiver to use a nonlinear device to process the received signal, to generate a separate carrier component that can be extracted using narrow bandpass filters. Clearly, effective and narrow bandpass filters are very important to both methods. Moreover, the bandpass filter should also have the ability to adaptively adjust its center frequency to combat significant frequency drift or Doppler shift. Aside from some typical bandpass filter designs, the phase-locked loop (PLL), which plays an important role in carrier acquisition of various modulations, can be viewed as such a narrow and adaptive bandpass filter. The principles of PLL will be discussed later in this chapter.

4.7 FREQUENCY DIVISION MULTIPLEXING (FDM)

Signal multiplexing allows the transmission of several signals on the same channel. In Chapter 6, we shall discuss time division multiplexing (TDM), where several signals time-share the same channel. In FDM, several signals share the band of a channel. Each signal is modulated by a different carrier frequency. These carriers, referred to as **subcarriers**, are adequately separated to avoid overlap (or interference) between the spectra of various modulated signals. Each signal may use a different kind of modulation (e.g., DSB-SC, AM, SSB-SC, VSB-SC, or even frequency modulation or phase modulation). The modulated signal spectra may be separated by a small guard band to avoid interference and facilitate signal separation at the receiver.

When all the modulated spectra are added, we have a composite signal that may be considered to be a baseband signal to further modulate a radio-frequency (RF) carrier for the purpose of transmission.

At the receiver, the incoming signal is first demodulated by the RF carrier to retrieve the composite baseband, which is then bandpass-filtered to separate all the modulated signals. Then each modulated signal is demodulated individually by an appropriate subcarrier to obtain all the basic baseband signals.

One simple example of FDM is the analog telephone long-haul system. There are two types of long-haul telephone carrier system: the legacy analog L-carrier hierarchy systems and the digital T-carrier hierarchy systems in North America (or the E-carrier in Europe).[3] Both were standardized by the predecessor of the International Telecommunications Union known (before 1992) as the CCITT (Comité Consultatif International Téléphonique et Télégraphique).

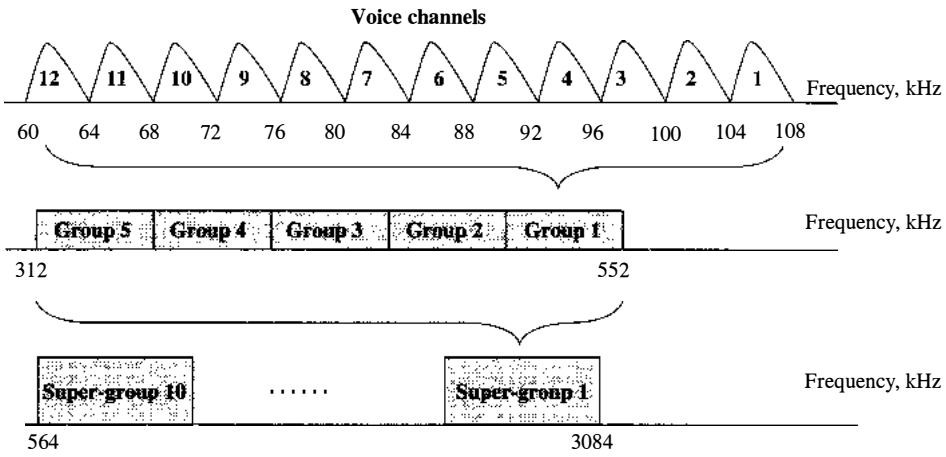


Figure 4.25 L-carrier hierarchical long-haul analog telephone frequency division multiplexing system.

We will first describe the analog telephone hierarchy that utilizes FDM and SSB modulation here and defer the digital hierarchy discussion until later (Chapter 6).

In the analog L-carrier hierarchy,[4] each voice channel is modulated using SSB+C. Twelve voice channels form a basic channel **group** occupying the bandwidth of 60 to 108 kHz. As shown in Fig. 4.25, each user channel uses LSB, and frequency division multiplexing (FDM) is achieved by maintaining the channel carrier separation of 4 kHz.

Further up the hierarchy,[5] five groups form a **supergroup**, via FDM. Multiplexing 10 supergroups generates a **mastergroup**, and multiplexing six supergroups forms a **jumbo group**, which consists of 3600 voice channels over a frequency band of 16.984 MHz in the L4 system. At each level of the hierarchy from the supergroup, additional frequency gaps are provided for interference reduction and for inserting pilot frequencies. The multiplexed signal can be fed into the baseband input of a microwave radio channel or directly into a coaxial transmission system.

4.8 PHASE-LOCKED LOOP AND SOME APPLICATIONS

Phase-Locked Loop (PLL)

The **phase-locked loop (PLL)** is a very important device typically used to track the phase and the frequency of the carrier component of an incoming signal. It is, therefore, a useful device for synchronous demodulation of AM signals with a suppressed carrier or with a little carrier (the pilot). It can also be used for the demodulation of angle-modulated signals, especially under conditions of low signal-to-noise ratio (SNR). It also has important applications in a number of clock recovery systems including timing recovery in digital receivers. For these reasons, the PLL plays a key role in nearly every modern digital and analog communication system.

A PLL has three basic components:

1. A voltage-controlled oscillator (VCO).
2. A multiplier, serving as a phase detector (PD) or a phase comparator.
3. A loop filter $H(s)$.

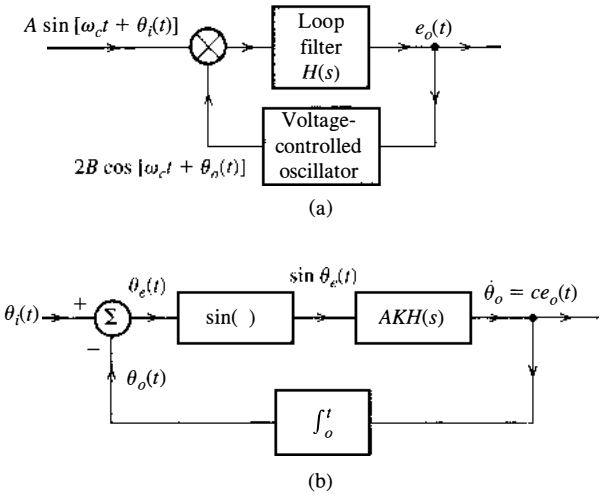


Figure 4.26 Phase-locked loop and its equivalent circuit.

Basic PLL Operation

The operation of the PLL is similar to that of a feedback system (Fig. 4.26a). In a typical feedback system, the feedback signal tends to follow the input signal. If the feedback signal is not equal to the input signal, the difference (known as the error) will change the feedback signal until it is close to the input signal. A PLL operates on a similar principle, except that the quantity fed back and compared is not the amplitude, but the phase. The VCO adjusts its own frequency such that its frequency and phase can track those of the input signal. At this point, the two signals are in synchronism (except for a possible difference of a constant phase).

The **voltage-controlled oscillator (VCO)** is an oscillator whose frequency can be linearly controlled by an input voltage. If a VCO input voltage is $e_o(t)$, its output is a sinusoid with instantaneous frequency given by

$$\omega(t) = \omega_c + ce_o(t) \tag{4.30}$$

where c is a constant of the VCO and ω_c is the **free-running frequency** of the VCO [when $e_o(t) = 0$]. The multiplier output is further low-pass-filtered by the loop filter and then applied to the input of the VCO. This voltage changes the frequency of the oscillator and keeps the loop **locked** by forcing the VCO output to track the phase (and hence the frequency) of the input sinusoid.

If the VCO output is $B \cos [\omega_c t + \theta_o(t)]$, then its instantaneous frequency is $\omega_c + \dot{\theta}_o(t)$. Therefore,

$$\dot{\theta}_o(t) = ce_o(t) \tag{4.31}$$

Note that c and B are constant parameters of the PLL.

Let the incoming signal (input to the PLL) be $A \sin [\omega_c t + \theta_i(t)]$. If the incoming signal happens to be $A \sin [\omega_o t + \psi(t)]$, it can still be expressed as $A \sin [\omega_c t + \theta_i(t)]$, where $\theta_i(t) = (\omega_o - \omega_c)t + \psi(t)$. Hence, the analysis that follows is general and not restricted to equal frequencies of the incoming signal and the free-running VCO signal.

The multiplier output is

$$AB \sin (\omega_c t + \theta_i) \cos (\omega_c t + \theta_o) = \frac{AB}{2}[\sin(\theta_i - \theta_o) + \sin (2\omega_c t + \theta_i + \theta_o)]$$

The sum frequency term is suppressed by the loop filter. Hence, the effective input to the loop filter is $\frac{1}{2}AB \sin [\theta_i(t) - \theta_o(t)]$. If $h(t)$ is the unit impulse response of the loop filter,

$$\begin{aligned} e_o(t) &= h(t) * \frac{1}{2}AB \sin [\theta_i(t) - \theta_o(t)] \\ &= \frac{1}{2}AB \int_0^t h(t-x) \sin [\theta_i(x) - \theta_o(x)]\, dx. \end{aligned} \tag{4.32}$$

Substituting Eq. (4.32) into Eq. (4.31) and letting $K = \frac{1}{2}cB$ lead to

$$\dot{\theta}_o(t) = AK \int_0^t h(t-x) \sin \theta_e(x)\, dx \tag{4.33}$$

where $\theta_e(t)$ is the phase error, defined as

$$\theta_e(t) = \theta_i(t) - \theta_o(t)$$

These equations [along with Eq. (4.31)] immediately suggest a model for the PLL, as shown in Fig. 4.26b.

The PLL design requires careful selection of the loop filter $H(s)$ and the loop gain AK. Different loop filters can enable the PLL to capture and track input signals with different types of frequency variation. On the other hand, the loop gain can affect the range of the trackable frequency variation.

Small-Error PLL Analysis

In small-error PLL analysis, $\sin \theta_e \simeq \theta_e$, and the block diagram in Fig. 4.26b reduces to the linear (time-invariant) system shown in Fig. 4.27a. Straightforward feedback analysis gives

$$\frac{\Theta_o(s)}{\Theta_i(s)} = \frac{AKH(s)/s}{1 + [AKH(s)/s]} = \frac{AKH(s)}{s + AKH(s)} \tag{4.34}$$

Therefore, the PLL acts as a filter with transfer function $AKH(s)/[s + AKH(s)]$, as shown in Fig. 4.27b. The error $\Theta_e(s)$ is given by

$$\begin{aligned} \Theta_e(s) = \Theta_i(s) - \Theta_o(s) &= \left[1 - \frac{\Theta_o(s)}{\Theta_i(s)}\right] \Theta_i(s) \\ &= \frac{s}{s + AKH(s)} \Theta_i(s) \end{aligned} \tag{4.35}$$

One of the important applications of the PLL is in the acquisition of the frequency and the phase for the purpose of synchronization. Let the incoming signal be $A \sin (\omega_0 t + \varphi_0)$. We wish

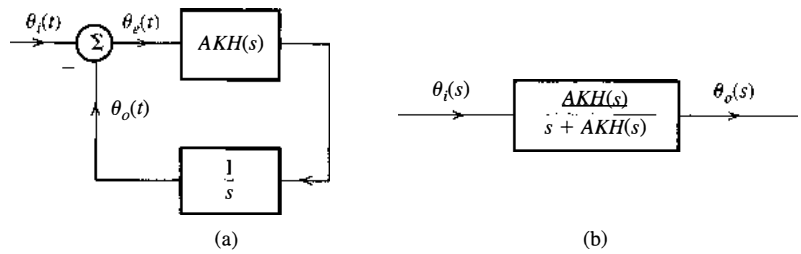


Figure 4.27 Equivalent circuits of a linearized PLL.

to generate a local signal of frequency ω_0 and phase* φ_0. Assuming the quiescent frequency of the VCO to be ω_c, the incoming signal can be expressed as $A \sin [\omega_c t + \theta_i(t)]$, where

$$\theta_i(t) = (\omega_0 - \omega_c)t + \varphi_0$$

and

$$\Theta_i(s) = \frac{\omega_0 - \omega_c}{s^2} + \frac{\varphi_0}{s}$$

Consider the special case of $H(s) = 1$. Substituting this equation into Eq. (4.35),

$$\begin{aligned}\Theta_e(s) &= \frac{s}{s + AK}\left[\frac{\omega_0 - \omega_c}{s^2} + \frac{\varphi_0}{s}\right] \\ &= \frac{(\omega_o - \omega_c)/AK}{s} - \frac{(\omega_0 - \omega_c)/AK}{s + AK} + \frac{\varphi_0}{s + AK}\end{aligned}$$

Hence,

$$\theta_e(t) = \frac{(\omega_0 - \omega_c)}{AK}\left(1 - e^{-AKt}\right) + \varphi_0 e^{-AKt} \tag{4.36a}$$

Observe that

$$\lim_{t\to\infty} \theta_e(t) = \frac{\omega_0 - \omega_c}{AK} \tag{4.36b}$$

Hence, after the transient dies (in about $4/AK$ seconds), the phase error maintains a constant value of $(\omega_0 - \omega_c)/AK$. This means the PLL frequency eventually equals the incoming frequency ω_0. There is, however, a constant phase error. The PLL output is

$$B \cos\left[\omega_0 t + \varphi_0 - \frac{\omega_0 - \omega_c}{AK}\right]$$

For a second-order PLL using

$$H(s) = \frac{s + a}{s} \tag{4.37a}$$

$$\begin{aligned}\Theta_e(s) &= \frac{s}{s + AKH(s)}\Theta_i(s) \qquad (4.37b)\\ &= \frac{s^2}{s^2 + AK(s + a)}\left[\frac{\omega_0 - \omega_c}{s^2} + \frac{\varphi_0}{s}\right]\end{aligned}$$

the final value theorem directly yields,[6]

$$\lim_{t\to\infty} \theta_e(t) = \lim_{s\to 0} s\Theta_e(s) = 0 \tag{4.38}$$

In this case, the PLL eventually acquires both the frequency and the phase of the incoming signal.

* With a difference $\pi/2$.

We can use small-error analysis, to show that a first-order loop cannot track an incoming signal whose instantaneous frequency varies linearly with time. Moreover, such a signal can be tracked within a constant phase (constant phase error) by using a second-order loop [Eq. (4.37)], and it can be tracked with zero phase error by using a third-order loop.[7]

It must be remembered that the preceding analysis assumes a linear model, which is valid only when $\theta_e(t) \ll \pi/2$. This means the frequencies ω_0 and ω_c must be very close for this analysis to be valid. For a general case, one must use the nonlinear model in Fig. 4.26b. For such an analysis, the reader is referred to Viterbi,[7] Gardner,[8] or Lindsey.[9]

First-Order Loop Analysis

Here we shall use the nonlinear model in Fig. 4.26b, but for the simple case of $H(s) = 1$. For this case $h(t) = \delta(t)$,* and Eq. (4.33) gives

$$\dot{\theta}_o(t) = AK \sin \theta_e(t)$$

Because $\theta_e = \theta_i - \theta_o$,

$$\dot{\theta}_e = \dot{\theta}_i - AK \sin \theta_e(t). \tag{4.39}$$

Let us here consider the problem of frequency and phase acquisition. Let the incoming signal be $A \sin (\omega_0 t + \varphi_0)$ and let the VCO have a quiescent frequency ω_c. Hence,

$$\theta_i(t) = (\omega_0 - \omega_c)t + \varphi_0$$

and

$$\dot{\theta}_e = (\omega_0 - \omega_c) - AK \sin \theta_e(t) \tag{4.40}$$

For a better understanding of PLL behavior, we use Eq. (4.40) to sketch $\dot{\theta}_e$ vs. θ_e. Equation (4.40) shows that $\dot{\theta}_e$ is a vertically shifted sinusoid, as shown in Fig. 4.28. To satisfy Eq. (4.40), the loop operation must stay along the sinusoidal trajectory shown in Fig. 4.28. When $\dot{\theta}_e = 0$, the system is in equilibrium, because at these points, θ_e stops varying with time. Thus $\theta_e = \theta_1$, θ_2, θ_3, and θ_4 are all equilibrium points.

If the initial phase error $\theta_e(0) = \theta_{e0}$ (Fig. 4.28), then $\dot{\theta}_e$ corresponding to this value of θ_e is negative. Hence, the phase error will start decreasing along the sinusoidal trajectory until it

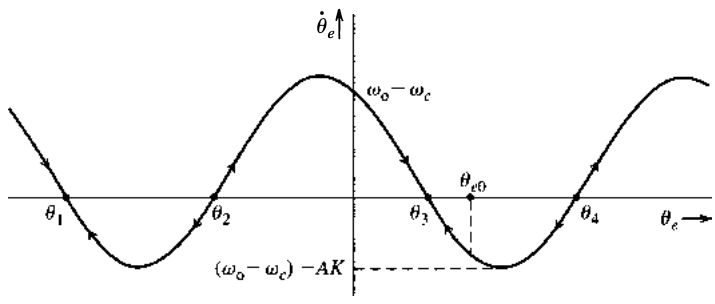


Figure 4.28 Trajectory of a first-order PLL.

* Actually $h(t) = 2B \operatorname{sinc} (2\pi Bt)$, where B is the bandwidth of the loop filter. This is a low-pass, narrow band filter, which suppresses the high-frequency signal centered at $2\omega_c$. This makes $H(s) = 1$ over a low-pass narrow band of B Hz.

reaches the value θ_3, where equilibrium is attained. Hence, in steady state, the phase error is a constant θ_3. This means the loop is in frequency lock; that is, the VCO frequency is now ω_0, but there is a phase error of θ_3. Note, however, that if $|\omega_0 - \omega_c| > AK$, there are no equilibrium points in Fig. 4.28, the loop never achieves lock, and θ_e continues to move along the trajectory forever. Hence, this simple loop can achieve phase lock provided the incoming frequency ω_0 does not differ from the quiescent VCO frequency ω_c by more than AK.

In Fig. 4.28, several equilibrium points exist. Half of these points, however, are unstable equilibrium points, meaning that a slight perturbation in the system state will move the operating point farther away from these equilibrium points. Points θ_1 and θ_3 are stable points because any small perturbation in the system state will tend to bring it back to these points. Consider, for example, the point θ_3. If the state is perturbed along the trajectory toward the right, $\dot{\theta}_e$ is negative, which tends to reduce θ_e and bring it back to θ_3. If the operating point is perturbed from θ_3 toward the left, $\dot{\theta}_e$ is positive, θ_e will tend to increase, and the operating point will return to θ_3. On the other hand, at point θ_2 if the point is perturbed toward the right, $\dot{\theta}_e$ is positive, and θ_e will increase until it reaches θ_3. Similarly, if at θ_2 the operating point is perturbed toward the left, $\dot{\theta}_e$ is negative, and θ_e will decrease until it reaches θ_1. Hence, θ_2 is an unstable equilibrium point. The slightest disturbance, such as noise, will dislocate it either to θ_1 or to θ_3. In a similar way, we can show that θ_4 is an unstable point and that θ_1 is a stable equilibrium point.

The equilibrium point θ_3 occurs where $\dot{\theta}_e = 0$. Hence, from Eq. (4.40),

$$\theta_3 = \sin^{-1} \frac{\omega_0 - \omega_c}{AK}$$

If $\theta_3 \ll \pi/2$, then

$$\theta_3 \simeq \frac{\omega_0 - \omega_c}{AK}$$

which agrees with our previous result of the small-error analysis [Eq. (4.36b)].

The first-order loop suffers from the fact that it has a constant phase error. Moreover, it can acquire frequency lock only if the incoming frequency and the VCO quiescent frequency differ by not more than AK rad/s. Higher order loops overcome these disadvantages, but they create a new problem of stability. More detailed analysis can be found in Gardener.[8]

Generalization of PLL Behaviors

To generalize, suppose that the loop is *locked*, meaning that the frequencies of both the input and the output sinusoids are identical. The two signals are said to be mutually **phase coherent** or **in phase lock**. The VCO thus tracks the frequency and the phase of the incoming signal. A PLL can track the incoming frequency only over a finite range of frequency shift. This range is called the **hold-in** or **lock** range. Moreover, if initially the input and output frequencies are not close enough, the loop may not acquire lock. The frequency range over which the input will cause the loop to lock is called the **pull-in** or **capture** range. Also if the input frequency changes too rapidly, the loop may not lock.

If the input sinusoid is noisy, the PLL not only tracks the sinusoid, but also cleans it up. The PLL can also be used as a frequency modulation (FM) demodulator and frequency synthesizer, as shown later, in the next chapter. Frequency multipliers and dividers can also be built using PLL. The PLL, being a relatively inexpensive integrated circuit, has become one of the most frequently used communication circuits.

In space vehicles, because of the Doppler shift and oscillator drift, the frequency of the received signal has a lot of uncertainty. The Doppler shift of the carrier itself could be as high as ± 75 kHz, whereas the desired modulated signal band may be just 10 Hz. To receive such a

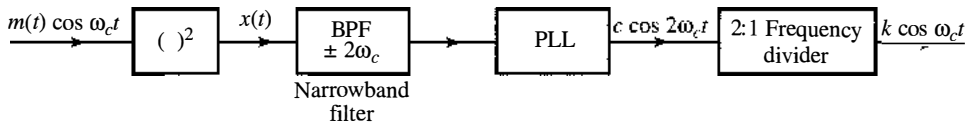


Figure 4.29 Using signal squaring to generate a coherent demodulation carrier.

signal by conventional receivers would require a filter of bandwidth 150 kHz, when the desired signal has a bandwidth of only 10 Hz. This would cause an undesirable increase in the received noise (by a factor of 15,000), since the noise power is proportional to the bandwidth. The PLL proves convenient here because it tracks the received frequency continuously, and the filter bandwidth required is only 10 Hz.

Carrier Acquisition in DSB-SC

We shall now discuss two methods of carrier regeneration using PLL at the receiver in DSB-SC: signal squaring and the Costas loop.

Signal-Squaring Method:

An outline of this scheme is given in Fig. 4.29. The incoming signal is squared and then passed through a narrow (high Q) bandpass filter tuned to $2\omega_c$. The output of this filter is the sinusoid $k\cos\ 2\omega_c t$, with some residual unwanted signal. This signal is applied to a PLL to obtain a cleaner sinusoid of twice the carrier frequency, which is passed through a 2:1 frequency divider to obtain a local carrier in phase and frequency synchronism with the incoming carrier. The analysis is straightforward. The squarer output $x(t)$ is

$$x(t) = [m(t)\cos\ \omega_c t]^2 = \tfrac{1}{2}m^2(t) + \tfrac{1}{2}m^2(t)\cos\ 2\omega_c t$$

Now $m^2(t)$ is a nonnegative signal, and therefore has a nonzero average value [in contrast to $m(t)$, which generally has a zero average value]. Let the average value, which is the dc component of $m^2(t)/2$, be k. We can now express $m^2(t)/2$ as

$$\frac{1}{2}m^2(t) = k + \phi(t)$$

where $\phi(t)$ is a zero mean baseband signal [$m^2(t)/2$ minus its dc component]. Thus,

$$\begin{aligned} x(t) &= \frac{1}{2}m^2(t) + \frac{1}{2}m^2(t)\cos\ 2\omega_c t \\ &= \frac{1}{2}m^2(t) + k\ \cos\ 2\omega_c t + \phi(t)\cos\ 2\omega_c t \end{aligned}$$

The bandpass filter is a narrowband (high-Q) filter tuned to frequency $2\omega_c$. It completely suppresses the signal $m^2(t)$, whose spectrum is centered at $\omega = 0$. It also suppresses most of the signal $\phi(t)\cos\ 2\omega_c t$. This is because although this signal spectrum is centered at $2\omega_c$, it has zero (infinitesimal) power at $2\omega_c$ since $\phi(t)$ has a zero dc value. Moreover this component is distributed over the band of $4B$ Hz centered at $2\omega_c$. Hence, very little of this signal passes through the narrowband filter.* In contrast, the spectrum of $k\ \cos\ 2\omega_c t$ consists of impulses

* This will also explain why we cannot extract the carrier directly from $m(t)\cos\ \omega_c t$ by passing it through a narrowband filter centered at ω_c. The reason is that the power of $m(t)\cos\ \omega_c t$ at ω_c is zero because $m(t)$ has no dc component [the average value of $m(t)$ is zero].

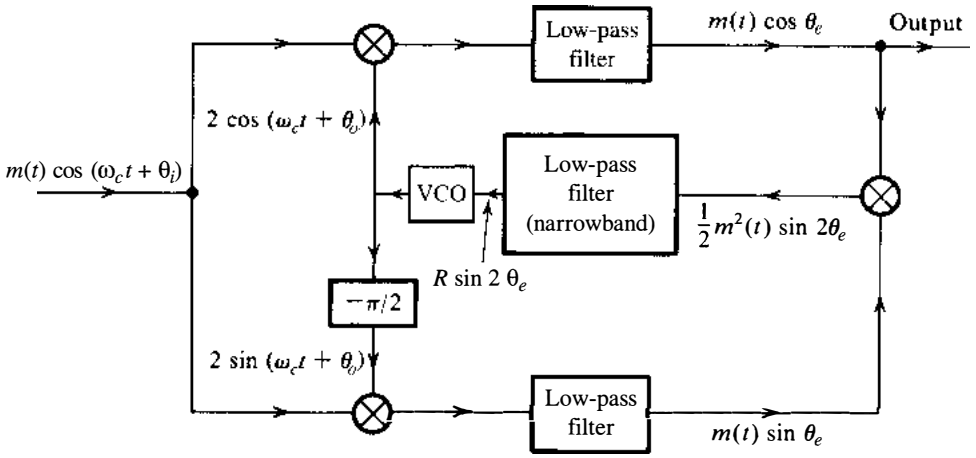


Figure 4.30 Costas phase-locked loop for the generation of a coherent demodulation carrier.

located at $\pm 2\omega_c$. Hence, all its power is concentrated at $2\omega_c$ and will pass through. Thus, the filter output is $k\cos 2\omega_c t$ plus a small undesired residue from $\phi(t)\cos 2\omega_c t$. This residue can be suppressed by using a PLL, which tracks $k\cos 2\omega_c t$. The PLL output, after passing through a 2:1 frequency divider, yields the desired carrier. One qualification is in order. Because the incoming signal sign is lost in the squarer, we have a sign ambiguity (or phase ambiguity of π) in the carrier generated. This is immaterial for analog signals. For a digital baseband signal, however, the carrier sign is essential, and this method, therefore, cannot be used directly.

Costas Loop: Yet another scheme for generating a local carrier, proposed by Costas,[10] is shown in Fig. 4.30. The incoming signal is $m(t)\cos(\omega_c t+\theta_i)$. At the receiver, a VCO generates the carrier $\cos(\omega_c t+\theta_o)$. The phase error is $\theta_e=\theta_i-\theta_o$. Various signals are indicated in Fig. 4.30. The two low-pass filters suppress high-frequency terms to yield $m(t)\cos\theta_e$ and $m(t)\sin\theta_e$, respectively. These outputs are further multiplied to give $m^2(t)\sin 2\theta_e$. When this is passed through a narrowband low-pass filter, the output is $R\sin 2\theta_e$, where R is the dc component of $m^2(t)/2$. The signal $R\sin 2\theta_e$ is applied to the input of a VCO with quiescent frequency ω_c. The input $R\sin 2\theta_e$ increases the output frequency, which, in turn, reduces θ_e. This mechanism was fully discussed earlier in connection with Fig. 4.26.

Carrier Acquisition in SSB-SC

For the purpose of synchronization at the SSB receiver, one may use highly stable crystal oscillators, with crystals cut for the same frequency at the transmitter and the receiver. At very high frequencies, where even quartz crystals may not have adequate performance, a pilot carrier may be transmitted. These are the same methods used for DSB-SC. However, neither the received-signal squaring technique nor the Costas loop used in DSB-SC can be used for SSB-SC. This can be seen by expressing the SSB signal as

$$\begin{aligned}\varphi_{\text{SSB}}(t) &= m(t)\cos\omega_c t \mp m_h(t)\sin\omega_c t\\ &= E(t)\cos[\omega_c t+\theta(t)]\end{aligned}$$

where

$$E(t) = \sqrt{m^2(t) + m_h^2(t)}$$

$$\theta(t) = \tan^{-1}\left[\frac{\pm m_h(t)}{m(t)}\right]$$

Squaring this signal yields

$$\varphi_{\text{SSB}}^2(t) = E^2(t)\cos^2[\omega_c t + \theta(t)]$$

$$= \frac{E^2(t)}{2}\{1 + \cos[2\omega_c t + 2\theta(t)]\}$$

The signal $E^2(t)$ is eliminated by a bandpass filter. Unfortunately, the remaining signal is not a pure sinusoid of frequency $2\omega_c$ (as was the case for DSB). There is nothing we can do to remove the time-varying phase $2\theta(t)$ from this sinusoid. Hence, for SSB, the squaring technique does not work. The same argument can be used to show that the Costas loop will not work either. These conclusions also apply to VSB signals.

4.9 MATLAB EXERCISES

In this section, we provide MATLAB exercises to reinforce some of the basic concepts on analog modulations covered in earlier sections. We will cover examples that illustrate the modulation and demodulation of DSB-SC, AM, SSB-SC, and QAM.

DSB-SC Modulation and Demodulation

The first MATLAB program, `triplesinc.m`, is to generate a signal that is (almost) strictly band-limited and consists of three different delayed version of the sinc signal:

$$m_2(t) = 2\,\text{sinc}\,(2t/T_a) + \text{sinc}\,(2t/T_a + 1) + \text{sinc}\,(2t/T_a - 1)$$

```
% (triplesinc.m)
%   Baseband signal for AM
%   Usage m=triplesinc(t,Ta)
    function m=triplesinc(t,Ta)
%   t is the length of the signal
%   Ta is the parameter, equaling twice the delay
%
    sig_1=sinc(2*t/Ta);
    sig_2=sinc(2*t/Ta-1);
    sig_3=sinc(2*t/Ta+1);
    m=2*sig_1+sig_2+sig_3;
end
```

The DSB-SC signal can be generated with the MATLAB file `ExampleDSB.m` that generates a DSB-SC signal for $t = \in (-0.04, 0.04)$. The carrier frequency is 300 Hz. The original

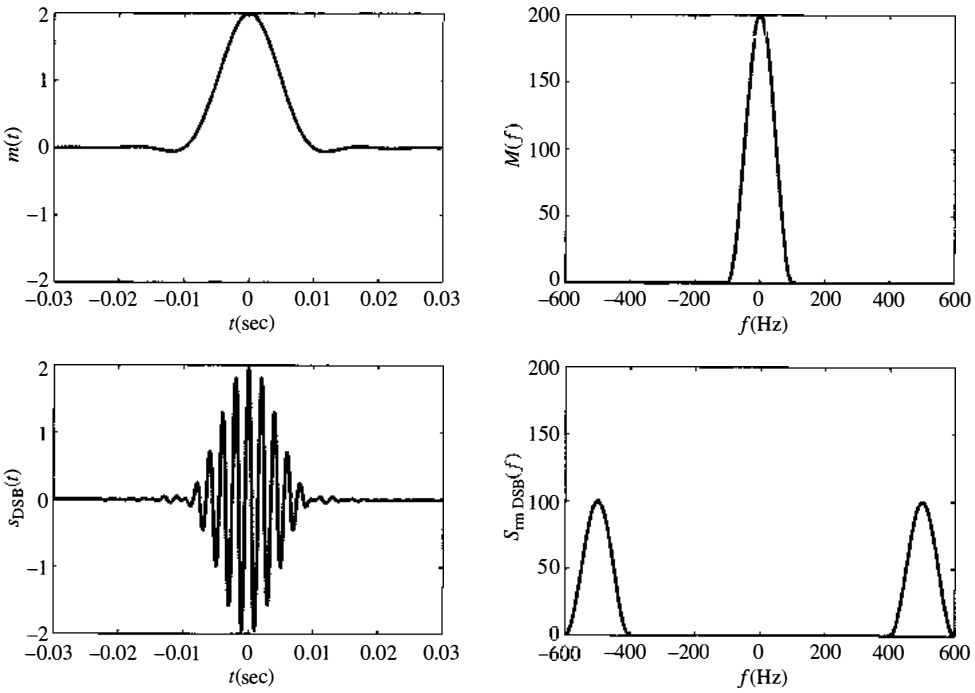


Figure 4.31 Example signals in time and frequency domains during DSB-SC modulation.

message signal and the DSB-SC signal for both time and frequency domains are illustrated in Fig. 4.31.

```
% (ExampleDSB.m)
% This program uses triplesinc.m to illustrate DSB modulation
% and demodulation

ts=1.e-4

t=-0.04:ts:0.04;
Ta=0.01;
m_sig=triplesinc(t,Ta);
Lfft=length(t);  Lfft=2^ceil(log2(Lfft));
M_fre=fftshift(fft(m_sig,Lfft));
freqm=(-Lfft/2:Lfft/2-1)/(Lfft*ts);

s_dsb=m_sig.*cos(2*pi*500*t);
Lfft=length(t);  Lfft=2^ceil(log2(Lfft)+1);
S_dsb=fftshift(fft(s_dsb,Lfft));
freqs=(-Lfft/2:Lfft/2-1)/(Lfft*ts);

Trange=[-0.03 0.03 -2 2]
figure(1)
subplot(221);td1=plot(t,m_sig);
axis(Trange); set(td1,'Linewidth',2);
```

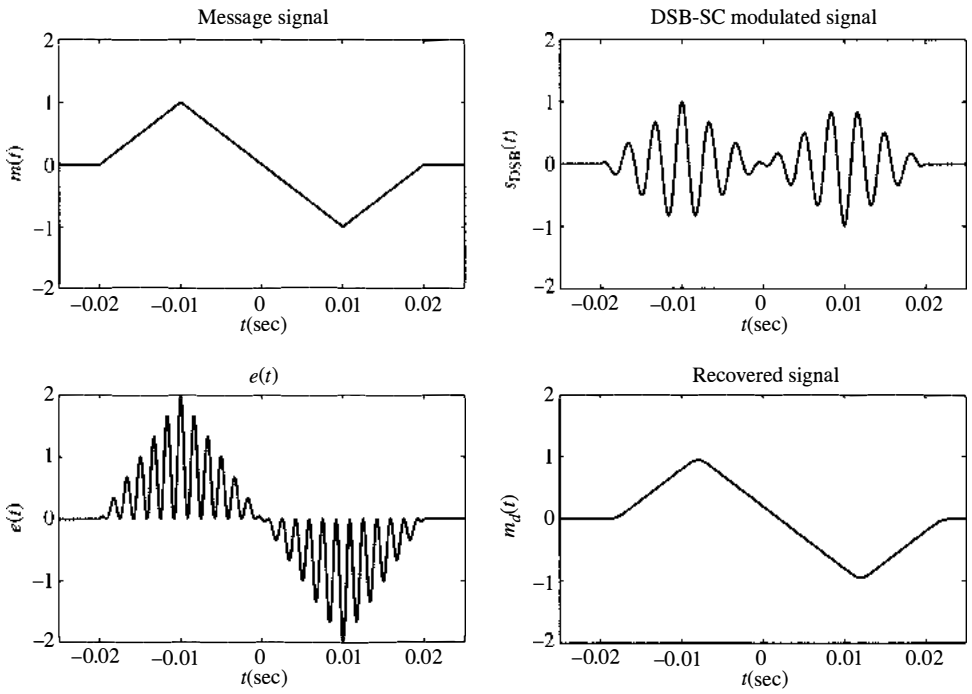


Figure 4.32 Time domain signals during DSB-SC modulation and demodulation.

```
xlabel('{\it t} (sec)'); ylabel('{\it m}({\it t})')
subplot(223);td2=plot(t,s_dsb);
axis(Trange); set(td2,'Linewidth',2);
xlabel('{\it t} (sec)'); ylabel('{\it s}_{\rm DSB}({\it t})')

Frange=[-600 600 0 200]
subplot(222);fd1=plot(freqm,abs(M_fre));
axis(Frange); set(fd1,'Linewidth',2);
xlabel('{\it f} (Hz)'); ylabel('{\it M}({\it f})')
subplot(224);fd2=plot(freqs,abs(S_dsb));
axis(Frange); set(fd2,'Linewidth',2);
xlabel('{\it f} (Hz)'); ylabel('{\it S}_{rm DSB}({\it f})')
```

The first modulation example, `ExampleDSBdemfilt.m` is based on a strictly low-pass message signal $m_0(t)$. Next, we will generate a different message signal that is not strictly band-limited. In effect, the new message signal consists of two triangles:

$$m_1(t) = \Delta\left(\frac{t+0.01}{0.01}\right) - \Delta\left(\frac{t-0.01}{0.01}\right)$$

Coherent demodulation is also implemented with a finite impulse response (FIR) low-pass filter of order 40. The original message signal $m(t)$, the DSB-SC signal $m(t)\cos\ \omega_c t$, the demodulator signal $e(t) = m(t)\cos^2 \omega_c t$, and the recovered message signal $m_d(t)$ after low-pass filtering are all given in Fig. 4.32 for the time domain and in Fig. 4.33 for the frequency domain. The low-pass filter at the demodulator has bandwidth of 150 Hz. The demodulation result shows almost no distortion.

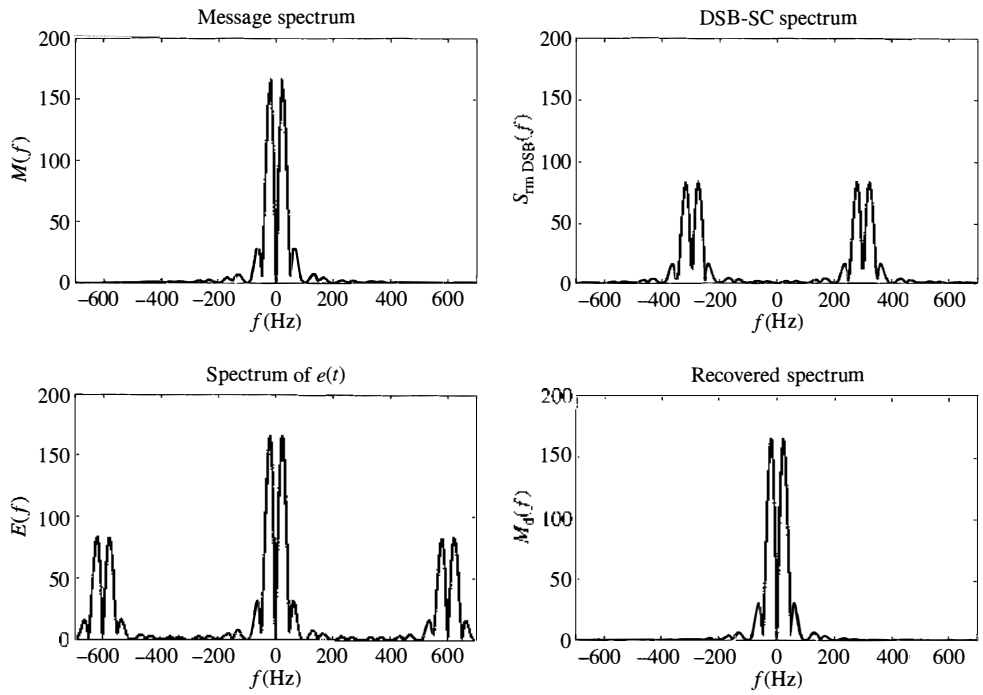


Figure 4.33 Frequency domain signals during DSB-SC modulation and demodulation.

```
% (ExampleDSBdemfilt.m)
% This program uses triangl.m to illustrate DSB modulation
% and demodulation

ts=1.e-4;

t=-0.04:ts:0.04;
Ta=0.01;
m_sig=triangl((t+0.01)/0.01)-triangl((t-0.01)/0.01);
Lm_sig=length(m_sig);
Lfft=length(t);
Lfft=2^ceil(log2(Lfft));
M_fre=fftshift(fft(m_sig,Lfft));
freqm=(-Lfft/2:Lfft/2-1)/(Lfft*ts);
B_m=150;    %Bandwidth of the signal is B_m Hz.
h=fir1(40,[B_m*ts]);

t=-0.04:ts:0.04;
Ta=0.01;fc=300;
s_dsb=m_sig.*cos(2*pi*fc*t);
Lfft=length(t);  Lfft=2^ceil(log2(Lfft)+1);
S_dsb=fftshift(fft(s_dsb,Lfft));
freqs=(-Lfft/2:Lfft/2-1)/(Lfft*ts);

%  Demodulation begins by multiplying with the carrier
```

```
s_dem=s_dsb.*cos(2*pi*fc*t)*2;
S_dem=fftshift(fft(s_dem,Lfft));

%  Using an ideal LPF with bandwidth 150 Hz
s_rec=filter(h,1,s_dem);
S_rec=fftshift(fft(s_rec,Lfft));

Trange=[-0.025 0.025 -2 2];
figure(1)
subplot(221);td1=plot(t,m_sig);
axis(Trange); set(td1,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it m}({\it t})');
title('message signal');
subplot(222);td2=plot(t,s_dsb);
axis(Trange); set(td2,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it s}_{\rm DSB}({\it t})')
title('DSB-SC modulated signal');
subplot(223);td3=plot(t,s_dem);
axis(Trange); set(td3,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it e}({\it t})')
title('{\it e}({\it t})');
subplot(224);td4=plot(t,s_rec);
axis(Trange); set(td4,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it m}_d({\it t})')
title('Recovered signal');

Frange=[-700 700 0 200];
figure(2)
subplot(221);fd1=plot(freqm,abs(M_fre));
axis(Frange); set(fd1,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it M}({\it f})');
title('message spectrum');
subplot(222);fd2=plot(freqs,abs(S_dsb));
axis(Frange); set(fd2,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it S}_{rm DSB}({\it f})');
title('DSB-SC spectrum');
subplot(223);fd3=plot(freqs,abs(S_dem));
axis(Frange); set(fd3,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it E}({\it f})');
title('spectrum of {\it e}({\it t})');
subplot(224);fd4=plot(freqs,abs(S_rec));
axis(Frange); set(fd4,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it M}_d({\it f})');
title('recovered spectrum');
```

AM Modulation and Demodulation

In this exercise, we generate a conventional AM signal with modulation index of $\mu = 1$. Using the same message signal $m_1(t)$, the MATLAB program `ExampleAMdemfilt.m` generates

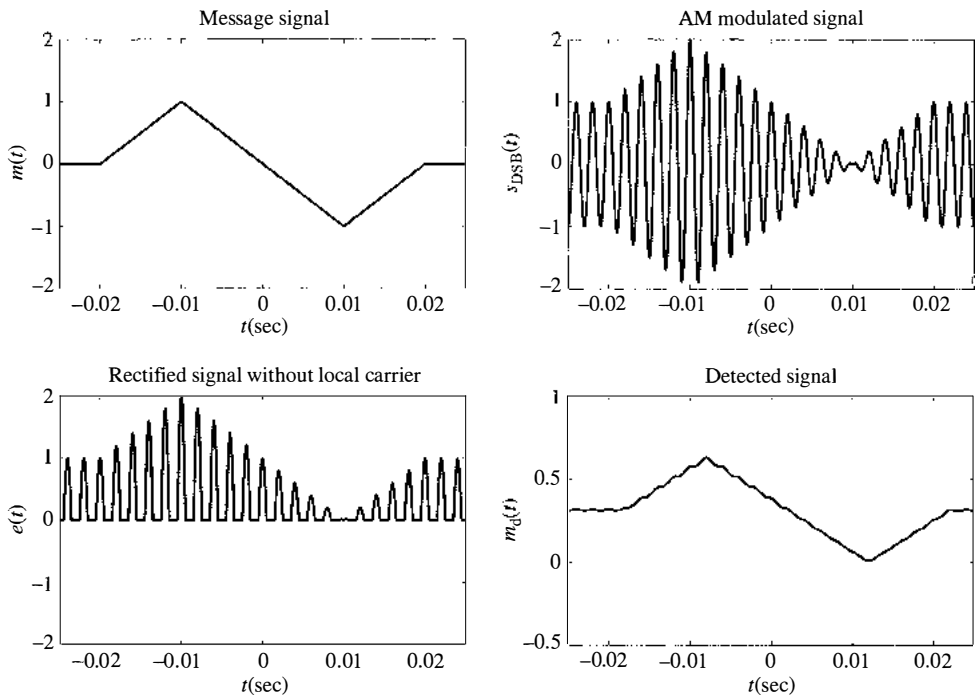


Figure 4.34 Time domain signals in AM modulation and noncoherent demodulation.

the message signal, the corresponding AM signal, the rectified signal in noncoherent demodulation, and the rectified signal after passing through a low-pass filter. The low-pass filter at the demodulator has a bandwidth of 150 Hz. The signals in the time domain are shown in Fig. 4.34, whereas the corresponding frequency domain signals are shown in Fig. 4.35.

Notice the large impulse in the frequency domain of the AM signal. The limited time window means that no ideal impulse is possible and only very large spikes centered at the carrier frequency of ±300 Hz are visible. Finally, because the message signal bandwidth is not strictly band-limited, the relatively low carrier frequency of 300 Hz forces the low-pass filter at the demodulator to truncate some with the message component in the demodulator. Distortion near the sharp corners of the recovered signal is visible.

```
% (ExampleAMdemfilt.m)
% This program uses triangl.m to illustrate AM modulation
% and demodulation

ts=1.e-4;
t=-0.04:ts:0.04;
Ta=0.01;  fc=500;
m_sig=triangl((t+0.01)/0.01)-triangl((t-0.01)/0.01);
Lm_sig=length(m_sig);
Lfft=length(t);  Lfft=2^ceil(log2(Lfft));
M_fre=fftshift(fft(m_sig,Lfft));
freqm=(-Lfft/2:Lfft/2-1)/(Lfft*ts);
B_m=150;    %Bandwidth of the signal is B_m Hz.
h=fir1(40,[B_m*ts]);
```

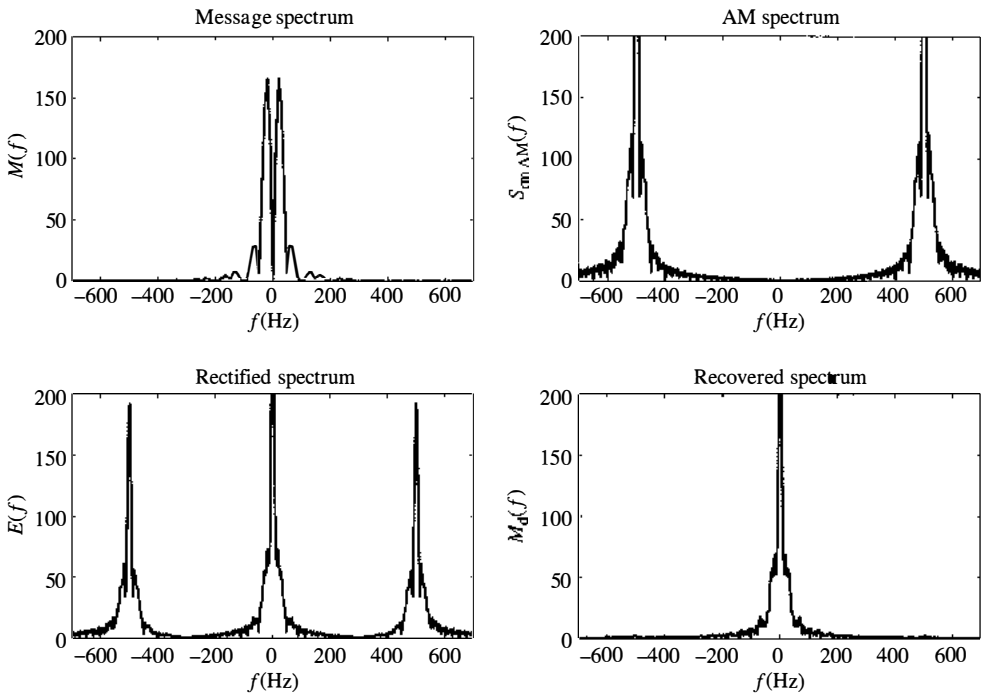


Figure 4.35 Frequency domain signals in AM modulation and noncoherent demodulation.

```
%  AM signal generated by adding a carrier to DSB-SC
s_am=(1+m_sig).*cos(2*pi*fc*t);
Lfft=length(t);  Lfft=2^ceil(log2(Lfft)+1);
S_am=fftshift(fft(s_am,Lfft));
freqs=(-Lfft/2:Lfft/2-1)/(Lfft*ts);

%  Demodulation begins by using a rectifier
s_dem=s_am.*(s_am>0);
S_dem=fftshift(fft(s_dem,Lfft));

%  Using an ideal LPF with bandwidth 150 Hz
s_rec=filter(h,1,s_dem);
S_rec=fftshift(fft(s_rec,Lfft));


Trange=[-0.025 0.025 -2 2];
figure(1)
subplot(221);td1=plot(t,m_sig);
axis(Trange); set(td1,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it m}({\it t})');
title('message signal');
subplot(222);td2=plot(t,s_am);
axis(Trange); set(td2,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it s}_{\rm DSB}({\it t})')
title('AM modulated signal');
```

```
subplot(223);td3=plot(t,s_dem);
axis(Trange); set(td3,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it e}({\it t})')
title('rectified signal without local carrier');
subplot(224);td4=plot(t,s_rec);
Trangelow=[-0.025 0.025 -0.5 1];
axis(Trangelow); set(td4,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it m}_d({\it t})')
title('detected signal');

Frange=[-700 700 0 200];
figure(2)
subplot(221);fd1=plot(freqm,abs(M_fre));
axis(Frange); set(fd1,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it M}({\it f})');
title('message spectrum');
subplot(222);fd2=plot(freqs,abs(S_am));
axis(Frange); set(fd2,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it S}_{rm AM}({\it f})');
title('AM spectrum');
subplot(223);fd3=plot(freqs,abs(S_dem));
axis(Frange); set(fd3,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it E}({\it f})');
title('rectified spectrum');
subplot(224);fd4=plot(freqs,abs(S_rec));
axis(Frange); set(fd4,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it M}_d({\it f})');
title('recovered spectrum');
```

SSB-SC Modulation and Demodulation

To illustrate the SSC-SC modulation and demodulation process, this exercise generates an SSB-SC signal using the same message signal $m_1(t)$ with double triangles. The carrier frequency is still 300 Hz. The MATLAB program `ExampleSSBdemfilt.m` performs this function. Coherent demodulation is applied in which a simple low-pass filter with bandwidth of 150 Hz is used to distill the recovered message signal.

The time domain signals are shown in Fig. 4.36, whereas the corresponding frequency domain signals are shown in Fig. 4.37.

```
% (ExampleSSBdemfilt.m)
% This program uses triangl.m
% to illustrate SSB modulation % and demodulation
clear;clf;

ts=1.e-4;
t=-0.04:ts:0.04;
Ta=0.01;  fc=300;
m_sig=triangl((t+0.01)/0.01)-triangl((t-0.01)/0.01);
Lm_sig=length(m_sig);
```

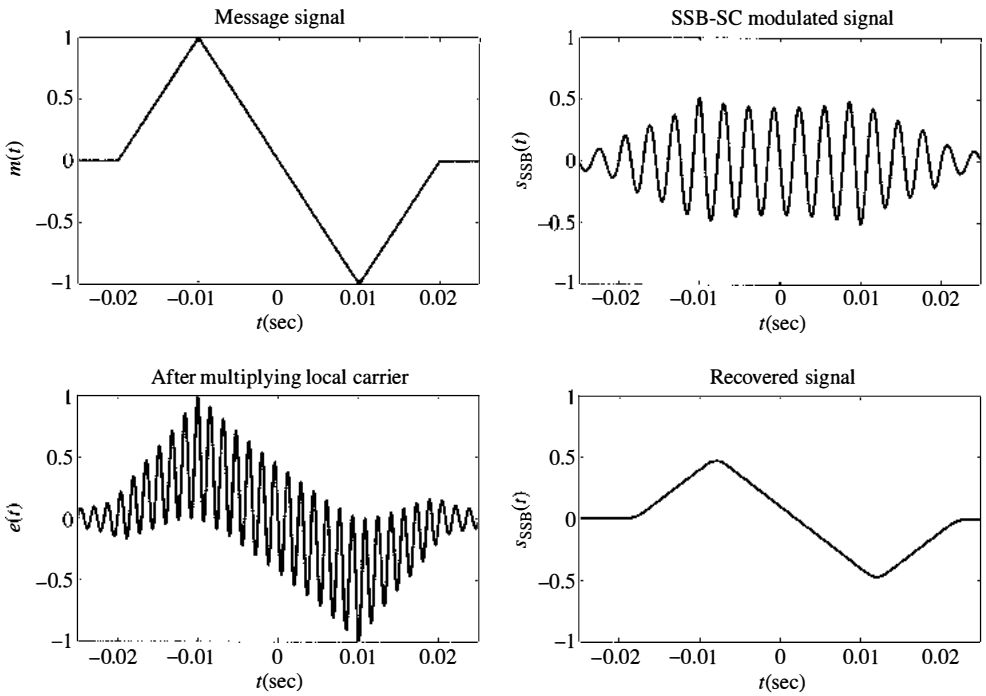


Figure 4.36 Time domain signals during SSB-SC modulation and coherent demodulation.

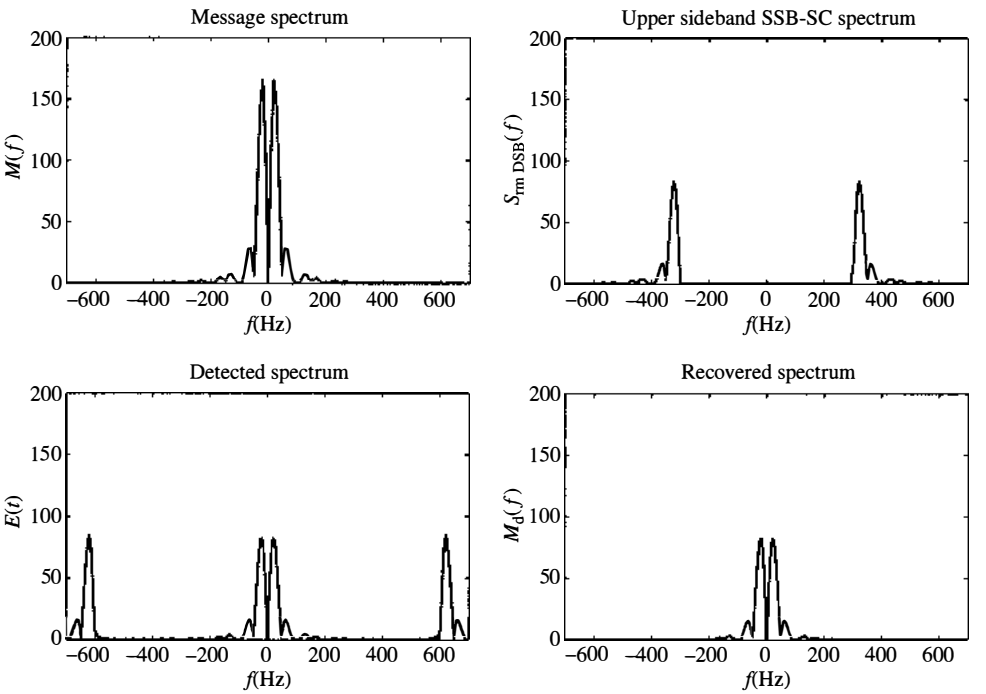


Figure 4.37 Frequency domain signals in SSB-SC modulation and coherent demodulation.

```
Lfft=length(t);  Lfft=2^ceil(log2(Lfft));
M_fre=fftshift(fft(m_sig,Lfft));
freqm=(-Lfft/2:Lfft/2-1)/(Lfft*ts);
B_m=150;    %Bandwidth of the signal is B_m Hz.
h=fir1(40,[B_m*ts]);

s_dsb=m_sig.*cos(2*pi*fc*t);
Lfft=length(t);  Lfft=2^ceil(log2(Lfft)+1);
S_dsb=fftshift(fft(s_dsb,Lfft));
L_lsb=floor(fc*ts*Lfft);
SSBfilt=ones(1,Lfft);
SSBfilt(Lfft/2-L_lsb+1:Lfft/2+L_lsb)=zeros(1,2*L_lsb);
S_ssb=S_dsb.*SSBfilt;
freqs=(-Lfft/2:Lfft/2-1)/(Lfft*ts);
s_ssb=real(ifft(fftshift(S_ssb)));
s_ssb=s_ssb(1:Lm_sig);

%  Demodulation begins by multiplying with the carrier
s_dem=s_ssb.*cos(2*pi*fc*t)*2;
S_dem=fftshift(fft(s_dem,Lfft));

%  Using an ideal LPF with bandwidth 150 Hz
s_rec=filter(h,1,s_dem);
S_rec=fftshift(fft(s_rec,Lfft));

Trange=[-0.025 0.025 -1 1];
figure(1)
subplot(221);td1=plot(t,m_sig);
axis(Trange); set(td1,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it m}({\it t})');
title('message signal');
subplot(222);td2=plot(t,s_ssb);
axis(Trange); set(td2,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it s}_{\rm SSB}({\it t})')
title('SSB-SC modulated signal');
subplot(223);td3=plot(t,s_dem);
axis(Trange); set(td3,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it e}({\it t})')
title('after multiplying local carrier');
subplot(224);td4=plot(t,s_rec);
axis(Trange); set(td4,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it m}_d({\it t})')
title('Recovered signal');

Frange=[-700 700 0 200];
figure(2)
subplot(221);fd1=plot(freqm,abs(M_fre));
axis(Frange); set(fd1,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it M}({\it f})');
title('message spectrum');
```

```
subplot(222);fd2=plot(freqs,abs(S_ssb));
axis(Frange); set(fd2,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it S}_{rm DSB}({\it f})');
title('upper sideband SSB-SC spectrum');
subplot(223);fd3=plot(freqs,abs(S_dem));
axis(Frange); set(fd3,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it E}({\it f})');
title('detector spectrum');
subplot(224);fd4=plot(freqs,abs(S_rec));
axis(Frange); set(fd4,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it M}_d({\it f})');
title('recovered spectrum');
```

QAM Modulation and Demodulation

In this exercise, we will apply QAM to modulate and demodulate two message signals $m_1(t)$ and $m_2(t)$. The carrier frequency stays at 300 Hz, but two signals are simultaneously modulated and detected. The QAM signal is coherently demodulated by multiplying with $\cos 600\pi t$ and $\sin 600\pi t$, respectively, to recover the two message signals. Each signal product is filtered by the same low-pass filter of order 40. The MATLAB program `ExampleQAMdemfilt.m` completes this illustration by showing the time domain signals during the modulation and demodulation of the first signal $m_1(t)$ and the second signal $m_2(t)$. The time domain results for $m_1(t)$ are shown in Fig. 4.38, whereas the frequency domain signals are shown in Fig. 4.39. Additionally, the time domain results for $m_2(t)$ are shown in Fig. 4.40, whereas the frequency domain signals are shown in Fig. 4.41.

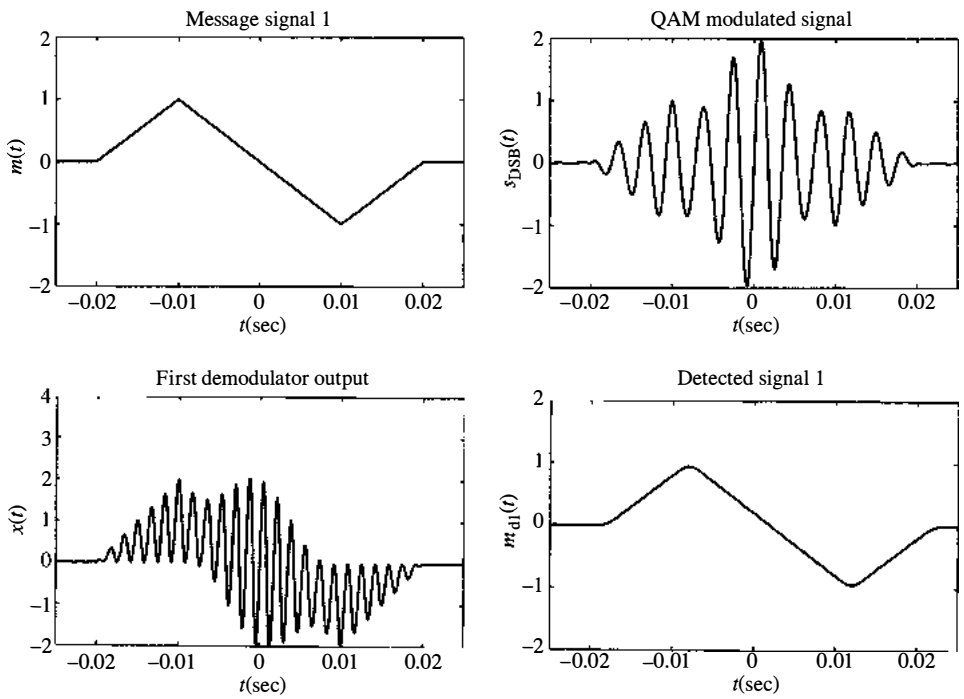


Figure 4.38 Time domain signals during QAM modulation and coherent demodulation for the first message $m_2(t)$.

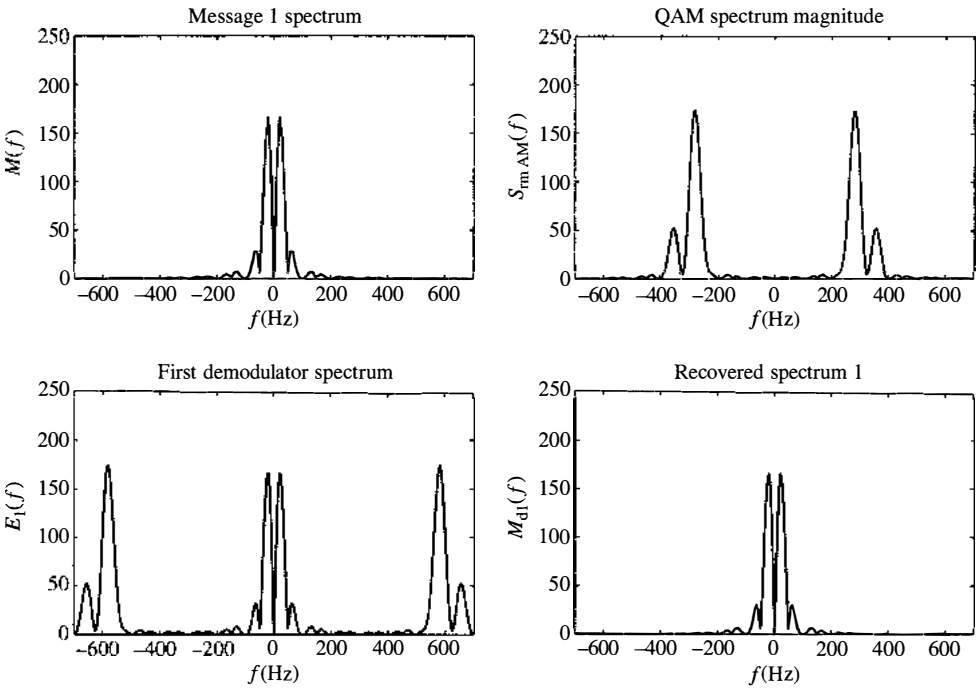


Figure 4.39 Frequency domain signals during QAM modulation and coherent demodulation for the first message $m_2(t)$.

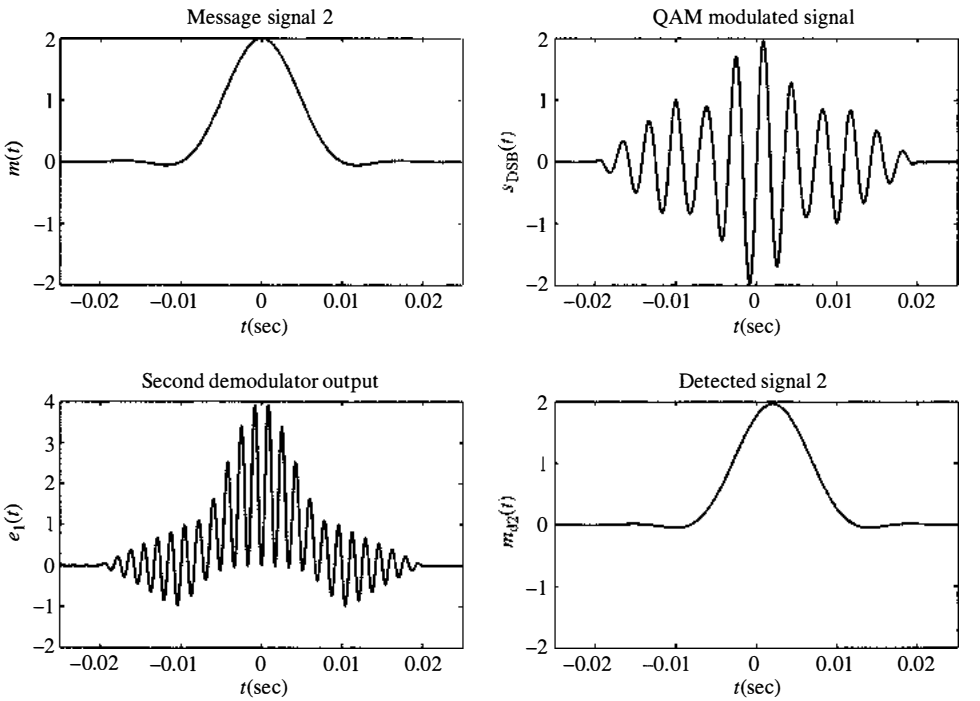


Figure 4.40 Time domain signals during QAM modulation and coherent demodulation for the second message $m_2(t)$.

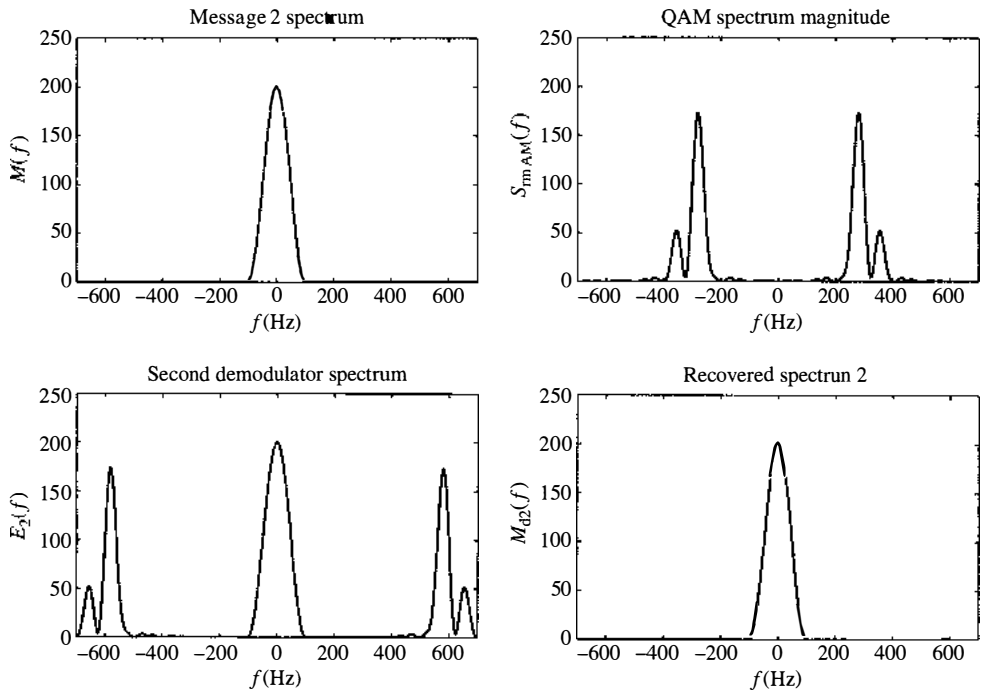


Figure 4.41 Frequency domain signals during QAM modulation and coherent demodulation for the second message $m_2(t)$.

```
% (ExampleQAMdemfilt.m)
% This program uses triangl.m and triplesinc.m
% to illustrate QAM modulation % and demodulation
% of two message signals
clear;clf;
ts=1.e-4;
t=-0.04:ts:0.04;
Ta=0.01;  fc=300;
% Use triangl.m and triplesinc.m to generate
% two message signals of different shapes and spectra
m_sig1=triangl((t+0.01)/0.01)-triangl((t-0.01)/0.01);
m_sig2=triplesinc(t,Ta);
Lm_sig=length(m_sig1);
Lfft=length(t);  Lfft=2^ceil(log2(Lfft));
M1_fre=fftshift(fft(m_sig1,Lfft));
M2_fre=fftshift(fft(m_sig2,Lfft));
freqm=(-Lfft/2:Lfft/2-1)/(Lfft*ts);
%
B_m=150;     %Bandwidth of the signal is B_m Hz.
% Design a simple lowpass filter with bandwidth B_m Hz.
h=fir1(40,[B_m*ts]);

% QAM signal generated by adding a carrier to DSB-SC
s_qam=m_sig1.*cos(2*pi*fc*t)+m_sig2.*sin(2*pi*fc*t);
Lfft=length(t);  Lfft=2^ceil(log2(Lfft)+1);
```

```
S_qam=fftshift(fft(s_qam,Lfft));
freqs=(-Lfft/2:Lfft/2-1)/(Lfft*ts);

%  Demodulation begins by using a rectifier
s_dem1=s_qam.*cos(2*pi*fc*t)*2;
S_dem1=fftshift(fft(s_dem1,Lfft));
%   Demodulate the 2nd signal
s_dem2=s_qam.*sin(2*pi*fc*t)*2;
S_dem2=fftshift(fft(s_dem2,Lfft));
%
%  Using an ideal LPF with bandwidth 150 Hz


s_rec1=filter(h,1,s_dem1);
S_rec1=fftshift(fft(s_rec1,Lfft));
s_rec2=filter(h,1,s_dem2);
S_rec2=fftshift(fft(s_rec2,Lfft));

Trange=[-0.025 0.025 -2 2];
Trange2=[-0.025 0.025 -2 4];
figure(1)
subplot(221);td1=plot(t,m_sig1);
axis(Trange); set(td1,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it m}({\it t})');
title('message signal 1');
subplot(222);td2=plot(t,s_qam);
axis(Trange); set(td2,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it s}_{\rm DSB}({\it t})')
title('QAM modulated signal');
subplot(223);td3=plot(t,s_dem1);
axis(Trange2); set(td3,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it x}({\it t})')
title('first demodulator output');
subplot(224);td4=plot(t,s_rec1);
axis(Trange); set(td4,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it m}_{d1}({\it t})')
title('detected signal 1');

figure(2)
subplot(221);td5=plot(t,m_sig2);
axis(Trange); set(td5,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it m}({\it t})');
title('message signal 2');
subplot(222);td6=plot(t,s_qam);
axis(Trange); set(td6,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it s}_{\rm DSB}({\it t})')
title('QAM modulated signal');
subplot(223);td7=plot(t,s_dem2);
axis(Trange2); set(td7,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it e}_1({\it t})')
```

```
title('second demodulator output');
subplot(224);td8=plot(t,s_rec2);
axis(Trange); set(td8,'Linewidth',1.5);
xlabel('{\it t} (sec)'); ylabel('{\it m}_{d2}({\it t})')
title('detected signal 2');

Frange=[-700 700 0 250];
figure(3)
subplot(221);fd1=plot(freqm,abs(M1_fre));
axis(Frange); set(fd1,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it M}({\it f})');
title('message 1 spectrum');
subplot(222);fd2=plot(freqs,abs(S_qam));
axis(Frange); set(fd2,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it S}_{rm AM}({\it f})');
title('QAM spectrum magnitude');
subplot(223);fd3=plot(freqs,abs(S_dem1));
axis(Frange); set(fd3,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it E}_1({\it f})');
title('first demodulator spectrum');
subplot(224);fd4=plot(freqs,abs(S_rec1));
axis(Frange); set(fd4,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it M}_{d1}({\it f})');
title('recovered spectrum 1');
figure(4)
subplot(221);fd1=plot(freqm,abs(M2_fre));
axis(Frange); set(fd1,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it M}({\it f})');
title('message 2 spectrum');
subplot(222);fd2=plot(freqs,abs(S_qam));
axis(Frange); set(fd2,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it S}_{rm AM}({\it f})');
title('QAM spectrum magnitude');
subplot(223);fd7=plot(freqs,abs(S_dem2));
axis(Frange); set(fd7,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it E}_2({\it f})');
title('second demodulator spectrum');
subplot(224);fd8=plot(freqs,abs(S_rec2));
axis(Frange); set(fd8,'Linewidth',1.5);
xlabel('{\it f} (Hz)'); ylabel('{\it M}_{d2}({\it f})');
title('recovered spectrum 2');
```

REFERENCES

1. Single Sideband Issue, *Proc. IRE*, vol. 44, Dec. 1956.
2. D. K. Weaver Jr., "A Third Method of Generation and Detection of Single Sideband Signals," *Proc. IRE*, vol. 44, pp. 1703–1705, Dec. 1956.
3. Bell Telephone Laboratories, *Transmission Systems for Communication*, 4th ed., Murray Hill, NJ, 1970.

4. R. T. James, "AT&T Facilities and Services," *Proc. IEEE*, vol. 60, pp. 1342–1349, Nov. 1972.
5. W. L. Smith, "Frequency and Time in Communications," *Proc. IEEE*, vol. 60, pp. 589–594, May 1972.
6. B. P. Lathi, B. P., *Linear Systems and Signals,* Oxford University Press, New York, 2000.
7. A. J. Viterbi, *Principles of Coherent Communication*, McGraw-Hill, New York, 1966.
8. F. M. Gardner, *Phaselock Techniques*, 3rd ed., Wiley, Hoboken, NJ, 2005.
9. W. C. Lindsey, *Synchronization Systems in Communication and Control*, Prentice-Hall, Englewood Cliffs, NJ, 1972.
10. J. P. Costas, "Synchronous Communication," *Proc. IRE*, vol. 44, pp. 1713–1718, Dec. 1956.

PROBLEMS

4.2-1 For each of the baseband signals: (i) $m(t) = \cos 1000\pi t$; (ii) $m(t) = 2\cos 1000\pi t + \sin 2000\pi t$; (iii) $m(t) = \cos 1000\pi t \cos 3000\pi t$, do the following.

(a) Sketch the spectrum of $m(t)$.

(b) Sketch the spectrum of the DSB-SC signal $m(t) \cos 10{,}000\pi t$.

(c) Identify the upper sideband (USB) and the lower sideband (LSB) spectra.

(d) Identify the frequencies in the baseband, and the corresponding frequencies in the DSB-SC, USB, and LSB spectra. Explain the nature of frequency shifting in each case.

4.2-2 Repeat Prob. 4.2-1 [parts **(a), (b)**, and **(c)** only] if: **(i)** $m(t) = \text{sinc}\,(100t)$; **(ii)** $m(t) = e^{-|t|}$; **(iii)** $m(t) = e^{-|t-1|}$. Observe that $e^{-|t-1|}$ is $e^{-|t|}$ delayed by 1 second. For the last case you need to consider both the amplitude and the phase spectra.

4.2-3 Repeat Prob. 4.2-1 [parts **(a), (b)**, and **(c)** only] for $m(t) = e^{-|t|}$ if the carrier is $\cos(10,1000t - \pi/4)$.
Hint: Use Eq. (3.37).

4.2-4 You are asked to design a DSB-SC modulator to generate a modulated signal $km(t) \cos(\omega_c t + \theta)$, where $m(t)$ is a signal band-limited to B Hz. Figure P4.2-4 shows a DSB-SC modulator available in the stock room. The carrier generator available generates not $\cos \omega_c t$, but $\cos^3 \omega_c t$. Explain whether you would be able to generate the desired signal using only this equipment. You may use any kind of filter you like.

(a) What kind of filter is required in Fig. P4.2-3?

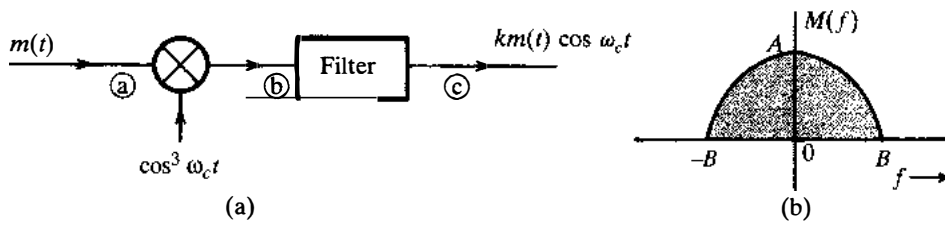
(b) Determine the signal spectra at points b and c, and indicate the frequency bands occupied by these spectra.

(c) What is the minimum usable value of ω_c?

(d) Would this scheme work if the carrier generator output were $\sin^3 \omega_c t$? Explain.

(f) Would this scheme work if the carrier generator output were $\cos^n \omega_c t$ for any integer $n \geq 2$?

Figure P.4.2-4



4.2-5 You are asked to design a DSB-SC modulator to generate a modulated signal $km(t)\cos\omega_c t$ with the carrier frequency $f_c = 300$ kHz ($\omega_c = 2\pi \times 300,000$). The following equipment is available in the stock room: **(i)** a signal generator of frequency 100 kHz; **(ii)** a ring modulator; **(iii)** a bandpass filter tuned to 300 kHz.

(a) Show how you can generate the desired signal.

(b) If the output of the modulator is $k \cdot m(t)\cos\omega_c t$, find k.

4.2-6 Amplitude modulators and demodulators can also be built without using multipliers. In Fig. P4.2-6, the input $\phi(t) = m(t)$, and the amplitude $A \gg |\phi(t)|$. The two diodes are identical, with a resistance of r ohms in the conducting mode and infinite resistance in the cutoff mode. Show that the output $e_o(t)$ is given by

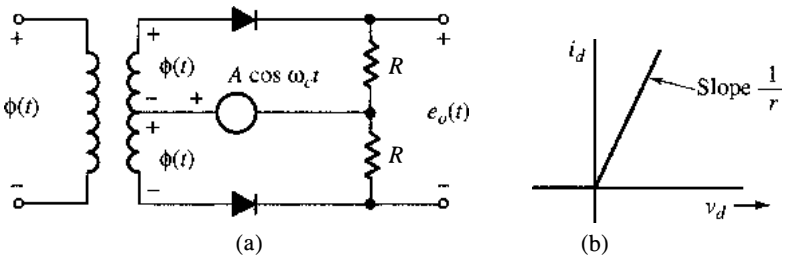
$$e_o(t) = \frac{2R}{R+r}\, w(t)\, m(t)$$

where $w(t)$ is the switching periodic signal shown in Fig. 2.20a with period $2\pi/W_c$ seconds.

(a) Hence, show that this circuit can be used as a DSB-SC modulator.

(b) How would you use this circuit as a synchronous demodulator for DSB-SC signals.

Figure P.4.2-6



4.2-7 In Fig. P4.2-6, if $\phi(t) = \sin(\omega_c t + \theta)$, and the output $e_0(t)$ is passed through a low-pass filter, then show that this circuit can be used as a phase detector, that is, a circuit that measures the phase difference between two sinusoids of the same frequency (ω_c).
Hint: Show that the filter output is a dc signal proportional to $\sin\theta$.

4.2-8 Two signals $m_1(t)$ and $m_2(t)$, both band-limited to 5000 Hz, are to be transmitted simultaneously over a channel by the multiplexing scheme shown in Fig. P4.2-8. The signal at point b is the multiplexed signal, which now modulates a carrier of frequency 20,000 Hz. The modulated signal at point c is transmitted over a channel.

(a) Sketch signal spectra at points a, b, and c.

(b) What must be the bandwidth of the channel?

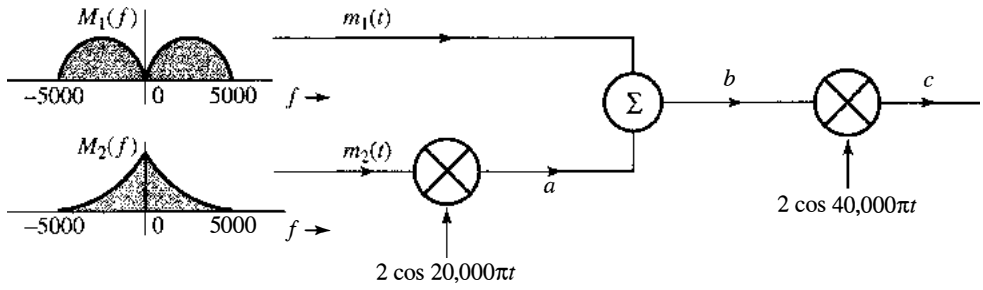
(c) Design a receiver to recover signals $m_1(t)$ and $m_2(t)$ from the modulated signal at point c.

4.2-9 The system shown in Fig. P4.2-9 is used for scrambling audio signals. The output $y(t)$ is the scrambled version of the input $m(t)$.

(a) Find the spectrum of the scrambled signal $y(t)$.

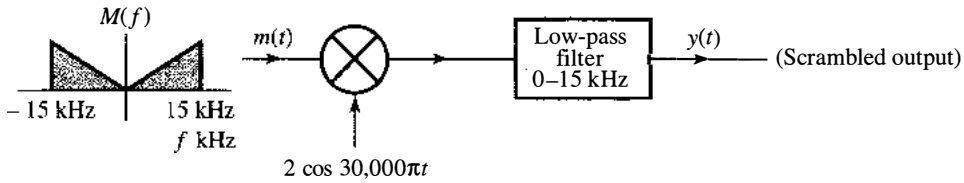
(b) Suggest a method of descrambling $y(t)$ to obtain $m(t)$.

Figure P.4.2-8



A slightly modified version of this scrambler was first used commercially on the 25-mile radiotelephone circuit connecting Los Angeles and Santa Catalina island.

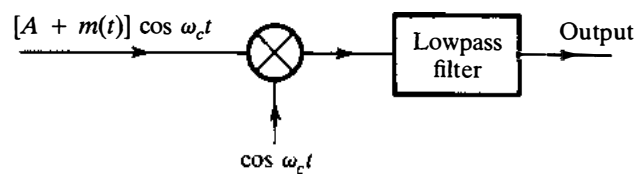
Figure P.4.2-9



4.2-10 A DSB-SC signal is given by $m(t)\cos(2\pi)10^6t$. The carrier frequency of this signal, 1 MHz, is to be changed to 400 kHz. The only equipment available consists of one ring modulator, a bandpass filter centered at the frequency of 400 kHz, and one sine wave generator whose frequency can be varied from 150 to 210 kHz. Show how you can obtain the desired signal $cm(t)\cos(2\pi \times 400 \times 10^3t)$ from $m(t)\cos(2\pi)10^6t$. Determine the value of c.

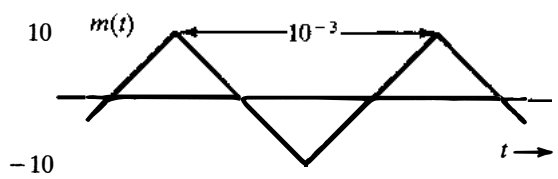
4.3-1 Figure P4.3-1 shows a scheme for coherent (synchronous) demodulation. Show that this scheme can demodulate the AM signal $[A + m(t)]\cos(2\pi f_c t)$ regardless of the value of $A>$

Figure P.4.3-1



4.3-2 Sketch the AM signal $[A + m(t)]\cos(2\pi f_c t)$ for the periodic triangle signal $m(t)$ shown in Fig. P4.3-2 corresponding to the modulation indices **(a)** $\mu = 0.5$; **(b)** $\mu = 1$; **(c)** $\mu = 2$; **(d)** $\mu = \infty$. How do you interpret the case of $\mu = \infty$?

Figure P.4.3-2



4.3-3 For the AM signal with $m(t)$ shown in Fig. P4.3-2 and $\mu = 0.8$:

(a) Find the amplitude and power of the carrier.

(b) Find the sideband power and the power efficiency η.

4.3-4 **(a)** Sketch the DSB-SC signal corresponding to the message signal $m(t) = \cos 2\pi t$.

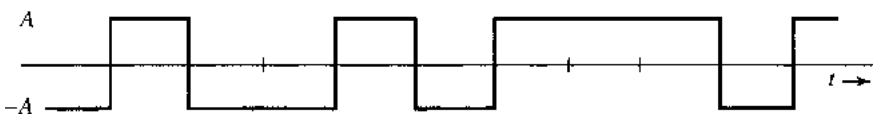
(b) The DSB-SC signal of part **(a)** is applied at the input of an envelope detector. Show that the output of the envelope detector is not $m(t)$, but $|m(t)|$. Show that, in general, if an AM signal $[A + m(t)]\cos \omega_c t$ is envelope-detected, the output is $|A + m(t)|$. Hence, show that the condition for recovering $m(t)$ from the envelope detector is $A + m(t) > 0$ for all t.

4.3-5 Show that any scheme that can be used to generate DSB-SC can also generate AM. Is the converse true? Explain.

4.3-6 Show that any scheme that can be used to demodulate DSB-SC can also demodulate AM. Is the converse true? Explain.

4.3-7 In the text, the power efficiency of AM for a sinusoidal $m(t)$ was found. Carry out a similar analysis when $m(t)$ is a random binary signal as shown in Fig. P4.3-7 and $\mu = 1$. Sketch the AM signal with $\mu = 1$. Find the sideband's power and the total power (power of the AM signal) as well as their ratio (the power efficiency η).

Figure P.4.3-7



4.3-8 In the early days of radio, AM signals were demodulated by a crystal detector followed by a low-pass filter and a dc blocker, as shown in Fig. P4.3-8. Assume a crystal detector to be basically a squaring device. Determine the signals at points a, b, c, and d. Point out the distortion term in the output $y(t)$. Show that if $A \gg |m(t)|$, the distortion is small.

Figure P.4.3-8



4.4-1 In a QAM system (Fig. 4.19), the locally generated carrier has a frequency error $\Delta\omega$ and a phase error δ; that is, the receiver carrier is $\cos [(\omega_c + \Delta\omega)t + \delta]$ or $\sin [(\omega_c + \Delta\omega)t + \delta]$. Show that the output of the upper receiver branch is

$$m_1(t)\cos [(\Delta\omega)t + \delta] - m_2(t)\sin [(\Delta\omega)t + \delta]$$

instead of $m_1(t)$, and the output of the lower receiver branch is

$$m_1(t)\sin [(\Delta\omega)t + \delta] + m_2(t)\cos [(\Delta\omega)t + \delta]$$

instead of $m_2(t)$.

4.4-2 A modulating signal $m(t)$ is given by:

(a) $m(t) = \cos\ 100\pi t + 2\cos\ 300\pi t$

(b) $m(t) = \sin\ 100\pi t \sin\ 500\pi t$

In each case:

(i) Sketch the spectrum of $m(t)$.

(ii) Find and sketch the spectrum of the DSB-SC signal $2m(t)\ \cos\ 1000\pi t$.

(iii) From the spectrum obtained in **(ii)**, suppress the LSB spectrum to obtain the USB spectrum.

(iv) Knowing the USB spectrum in **(ii)**, write the expression $\varphi_{\text{USB}}(t)$ for the USB signal.

(v) Repeat **(iii)** and **(iv)** to obtain the LSB signal $\varphi_{\text{LSB}}(t)$.

4.4-3 For the signals in Prob. 4.4-2, use Eq. (4.20) to determine the time domain expressions $\varphi_{\text{LSB}}(t)$ and $\varphi_{\text{USB}}(t)$ if the carrier frequency $\omega_c = 1000$.

Hint: If $m(t)$ is a sinusoid, its Hilbert transform $m_h(t)$ is the sinusoid $m(t)$ phase-delayed by $\pi/2$ rad.

4.4-4 Find $\varphi_{\text{LSB}}(t)$ and $\varphi_{\text{USB}}(t)$ for the modulating signal $m(t) = \pi B\,\text{sinc}^2\,(2\pi Bt)$ with $B = 2000$ Hz and carrier frequency $f_c = 10,000$ Hz. Follow these steps:

(a) Sketch spectra of $m(t)$ and the corresponding DSB-SC signal $2m(t)\ \cos\ \omega_c t$.

(b) To find the LSB spectrum, suppress the USB in the DSB-SC spectrum found in part **(a)**.

(c) Find the LSB signal $\varphi_{\text{LSB}}(t)$, which is the inverse Fourier transform of the LSB spectrum found in part **(b)**. Follow a similar procedure to find $\varphi_{\text{USB}}(t)$.

4.4-5 If $m_h(t)$ is the Hilbert transform of $m(t)$, then

(a) Show that the Hilbert transform of $m_h(t)$ is $-m(t)$.

(b) Show also that the energies of $m(t)$ and $m_h(t)$ are identical.

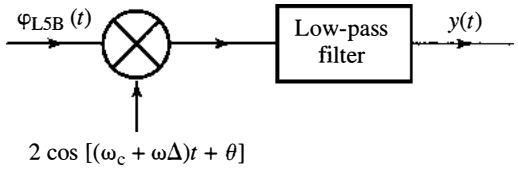
4.4-6 An LSB signal is demodulated coherently, as shown in Fig. P4.4-6. Unfortunately, because of the transmission delay, the received signal carrier is not $2\cos\ \omega_c t$ as sent, but rather, is $2\cos\,[(\omega_c + \Delta\omega)t + \delta]$. The local oscillator is still $\cos\ \omega_c t$. Show the following.

(a) When $\delta = 0$, the output $y(t)$ is the signal $m(t)$ with all its spectral components shifted (offset) by $\Delta\omega$.
Hint: Observe that the output $y(t)$ is identical to the right-hand side of Eq. (4.20a) with ω_c replaced with $\Delta\omega$.

(b) When $\Delta\omega = 0$, the output is the signal $m(t)$ with phases of all its spectral components shifted by δ.
Hint: Show that the output spectrum $Y(f) = M(f)e^{j\delta}$ for $f \geq 0$, and equal to $M(f)e^{-j\delta}$ when $f < 0$.

(c) In each of these cases, explain the nature of distortion.
Hint: For part **(a)**, demodulation consists of shifting an LSB spectrum to the left and right by $\omega_c + \Delta\omega$ and low-pass-filtering the result. For part **(b)**, use the expression (4.20b) for $\varphi_{\text{LSB}}(t)$, multiply it by the local carrier $2\ \cos\ (\omega_c t + \delta)$, and low-pass-filter the result.

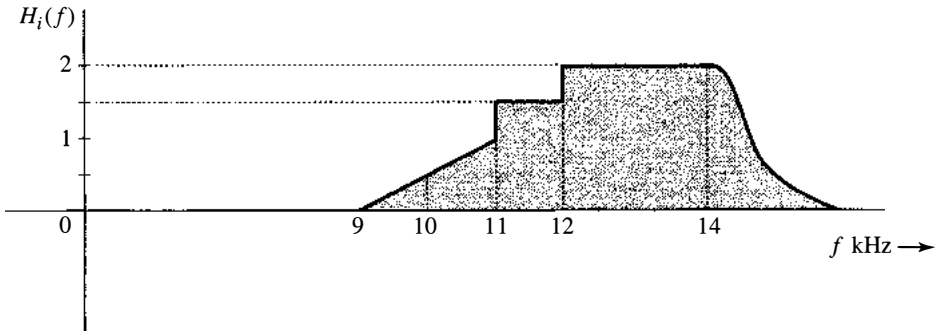
Figure P.4.4-6



4.4-7 A USB signal is generated by using the phase shift method (Fig. 4.17). If the input to this system is $m_h(t)$ instead of $m(t)$, what will be the output? Is this signal still an SSB signal with bandwidth equal to that of $m(t)$? Can this signal be demodulated [to get back $m(t)$]? If so, how?

4.5-1 A vestigial filter $H_i(f)$ shown in the transmitter of Fig. 4.21 has a transfer function as shown in Fig. P4.5-1. The carrier frequency is $f_c = 10$ kHz and the baseband signal bandwidth is 4 kHz. Find the corresponding transfer function of the equalizer filter $H_o(f)$ shown in the receiver of Fig. 4.21.
Hint: Use Eq. (4.25).

Figure P.4.5-1



5 ANGLE MODULATION AND DEMODULATION

As discussed in the previous chapter, a carrier modulation can be achieved by modulating the amplitude, frequency, and phase of a **sinusoidal carrier** of frequency f_c. In that chapter, we focused on various linear amplitude modulation systems and their demodulations. Now we discuss nonlinear frequency modulation (FM) and phase modulation (PM), often collectively known as angle modulation.

5.1 NONLINEAR MODULATION

In AM signals, the amplitude of a carrier is modulated by a signal $m(t)$, and, hence, the information content of $m(t)$ is in the amplitude variations of the carrier. As we have seen, the other two parameters of the carrier sinusoid, namely its frequency and phase, can also be varied in proportion to the message signal as frequency-modulated and phase-modulated signals, respectively. We now describe the essence of frequency modulation (FM) and phase modulation (PM).

False Start

In the 1920s, broadcasting was in its infancy. However, there was an active search for techniques to reduce noise (static). Since the noise power is proportional to the modulated signal bandwidth (sidebands), efforts were focused on finding a modulation scheme that would reduce the bandwidth. More important still, bandwidth reduction also allows more users, and there were rumors of a new method that had been discovered for eliminating sidebands (no sidebands, no bandwidth!). The idea of **frequency modulation (FM)**, where the carrier frequency would be varied in proportion to the message $m(t)$, was quite intriguing. The carrier angular frequency $\omega(t)$ would be varied with time so that $\omega(t) = \omega_c + km(t)$, where k is an arbitrary constant. If the peak amplitude of $m(t)$ is m_p, then the maximum and minimum values of the carrier frequency would be $\omega_c + km_p$ and $\omega_c - km_p$, respectively. Hence, the spectral components would remain within this band with a bandwidth $2km_p$ centered at ω_c. The understanding was that controlling the constant parameter k can control the modulated signal bandwidth. While this is true, there was also the hope that by using an arbitrarily small k, we could make the information bandwidth arbitrarily small. This possibility was seen as a passport to communication heaven. Unfortunately, experimental results showed that the underlying reasoning was seriously wrong. The FM bandwidth, as it turned out, is always greater than (at best equal to)

the AM bandwidth. In some cases, its bandwidth was several times that of AM. Where was the fallacy in the original reasoning? We shall soon find out.

The Concept of Instantaneous Frequency

While AM signals carry a message with their varying amplitude, FM signals can vary the instantaneous frequency in proportion to the modulating signal $m(t)$. This means that the carrier frequency is changing continuously every instant. Prima facie, this does not make much sense, since to define a frequency, we must have a sinusoidal signal at least over one cycle (or a half-cycle or a quarter-cycle) with the same frequency. This problem reminds us of our first encounter with the concept of **instantaneous velocity** in a beginning mechanics course. Until the presentation of derivatives via Leibniz and Newton, we were used to thinking of velocity as being constant over an interval, and we were incapable of even imagining that velocity could vary at each instant. We never forget, however, the wonder and amazement that was caused by the contemplation of derivative and instantaneous velocity when these concepts were first introduced. A similar experience awaits the reader with respect to **instantaneous frequency**.

Let us consider a generalized sinusoidal signal $\varphi(t)$ given by

$$\varphi(t) = A \cos \theta(t) \tag{5.1}$$

where $\theta(t)$ is the **generalized angle** and is a function of t. Figure 5.1 shows a hypothetical case of $\theta(t)$. The generalized angle for a conventional sinusoid $A \cos(\omega_c t + \theta_0)$ is a straight line $\omega_c t + \theta_0$, as shown in Fig. 5.1. A hypothetical case general angle of $\theta(t)$ happens to be tangential to the angle $(\omega_c t + \theta_0)$ at some instant t. The crucial point is that, around t, over a small interval $\Delta t \to 0$, the signal $\varphi(t) = A \cos \theta(t)$ and the sinusoid $A \cos(\omega_c t + \theta_0)$ are identical; that is,

$$\varphi(t) = A \cos(\omega_c t + \theta_0) \qquad t_1 < t < t_2$$

We are certainly justified in saying that over this small interval Δt, the angular frequency of $\varphi(t)$ is ω_c. Because $(\omega_c t + \theta_0)$ is tangential to $\theta(t)$, the angular frequency of $\varphi(t)$ is the slope of its angle $\theta(t)$ over this small interval. We can generalize this concept at **every instant** and define that the instantaneous frequency ω_i at any instant t is the slope of $\theta(t)$ at t. Thus, for

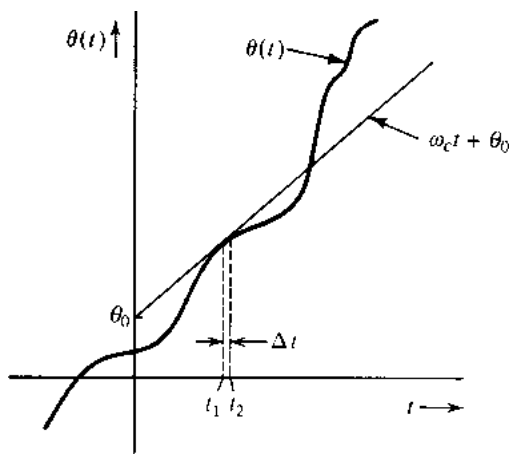


Figure 5.1 Concept of instantaneous frequency.

$\varphi(t)$ in Eq. (5.1), the instantaneous angular frequency and the generalized angle are related via

$$\omega_i(t) = \frac{d\theta}{dt} \tag{5.2a}$$

$$\theta(t) = \int_{-\infty}^{t} \omega_i(\alpha)\, d\alpha \tag{5.2b}$$

Now we can see the possibility of transmitting the information of $m(t)$ by varying the angle θ of a carrier. Such techniques of modulation, where the angle of the carrier is varied in some manner with a modulating signal $m(t)$, are known as **angle modulation** or **exponential modulation**. Two simple possibilities are **phase modulation (PM)** and **frequency modulation (FM)**. In PM, the angle $\theta(t)$ is varied linearly with $m(t)$:

$$\theta(t) = \omega_c t + \theta_0 + k_p m(t)$$

where k_p is a constant and ω_c is the carrier frequency. Assuming $\theta_0 = 0$, without loss of generality,

$$\theta(t) = \omega_c t + k_p m(t) \tag{5.3a}$$

The resulting PM wave is

$$\varphi_{\text{PM}}(t) = A \cos [\omega_c t + k_p m(t)] \tag{5.3b}$$

The instantaneous angular frequency $\omega_i(t)$ in this case is given by

$$\omega_i(t) = \frac{d\theta}{dt} = \omega_c + k_p \dot{m}(t) \tag{5.3c}$$

Hence, in PM, the instantaneous angular frequency ω_i varies linearly with the derivative of the modulating signal. If the instantaneous frequency ω_i is varied linearly with the modulating signal, we have FM. Thus, in FM the instantaneous angular frequency ω_i is

$$\omega_i(t) = \omega_c + k_f m(t) \tag{5.4a}$$

where k_f is a constant. The angle $\theta(t)$ is now

$$\begin{aligned} \theta(t) &= \int_{-\infty}^{t} [\omega_c + k_f m(\alpha)]\, d\alpha \\ &= \omega_c t + k_f \int_{-\infty}^{t} m(\alpha)\, d\alpha \end{aligned}$$

Here we have assumed the constant term in $\theta(t)$ to be zero without loss of generality. The FM wave is

$$\varphi_{\text{FM}}(t) = A \cos \left[\omega_c t + k_f \int_{-\infty}^{t} m(\alpha)\, d\alpha \right] \tag{5.5}$$

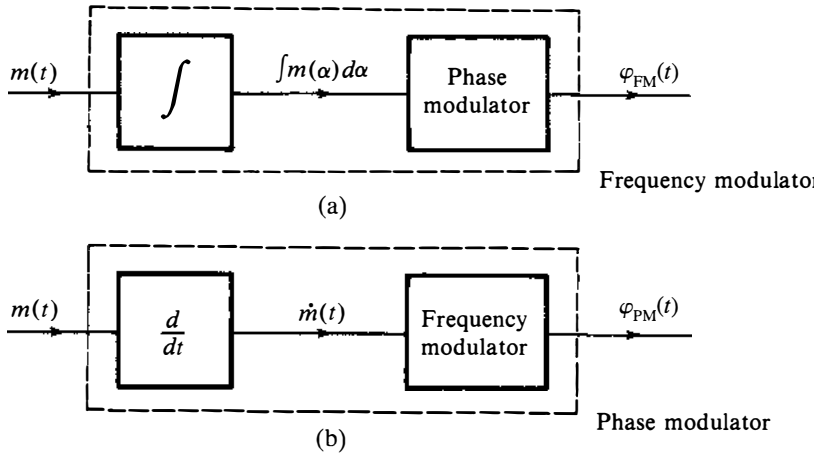


Figure 5.2 Phase and frequency modulation are equivalent and interchangeable.

Relationship between FM and PM

From Eqs. (5.3b) and (5.5), it is apparent that PM and FM not only are very similar but are inseparable. Replacing $m(t)$ in Eq. (5.3b) with $\int m(\alpha)d\alpha$ changes PM into FM. Thus, a signal that is an FM wave corresponding to $m(t)$ is also the PM wave corresponding to $\int m(\alpha)\,d\alpha$ (Fig. 5.2a). Similarly, a PM wave corresponding to $m(t)$ is the FM wave corresponding to $\dot{m}(t)$ (Fig. 5.2b). Therefore, by looking only at an angle-modulated signal $\varphi(t)$, there is no way of telling whether it is FM or PM. In fact, it is meaningless to ask an angle-modulated wave whether it is FM or PM. It is analogous to asking a married man with children whether he is a father or a son. This discussion and Fig. 5.2 also show that we need not separately discuss methods of generation and demodulation of each type of modulation.

Equations (5.3b) and (5.5) show that in both PM and FM the angle of a carrier is varied in proportion to some measure of $m(t)$. In PM, it is directly proportional to $m(t)$, whereas in FM, it is proportional to the integral of $m(t)$. As shown in Fig. 5.2b, a frequency modulator can be directly used to generate an FM signal or the message input $m(t)$ can be processed by a filter (differentiator) with transfer function $H(s) = s$ to generate PM signals. But why should we limit ourselves to these cases? We have an infinite number of possible ways of processing $m(t)$ before FM. If we restrict the choice to a linear operator, then a measure of $m(t)$ can be obtained as the output of an invertible linear (time-invariant) system with transfer function $H(s)$ or impulse response $h(t)$. The generalized angle-modulated carrier $\varphi_{EM}(t)$ can be expressed as

$$\varphi_{EM}(t) = A\cos[\omega_c t + \psi(t)] \tag{5.6a}$$

$$= A\cos\left[\omega_c t + \int_{-\infty}^{t} m(\alpha)\,h(t-\alpha)\,d\alpha\right] \tag{5.6b}$$

As long as $H(s)$ is a reversible operation (or invertible), $m(t)$ can be recovered from $\psi(t)$ by passing it through a system with transfer function $[H(s)]^{-1}$ as shown in Fig. 5.3. Now PM and FM are just two special cases with $h(t) = k_p\delta(t)$ and $h(t) = k_f u(t)$, respectively.

This shows that if we analyze one type of angle modulation (such as FM), we can readily extend those results to any other kind. Historically, the angle modulation concept began with FM, and here in this chapter we shall primarily analyze FM, with occasional discussion of

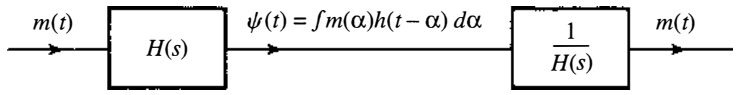


Figure 5.3 Generalized phase modulation by means of the filter $H(s)$ and recovery of the message from the modulated phase through the inverse filter $1/H(s)$.

PM. But this does not mean that FM is superior to other kinds of angle modulation. On the contrary, for most practical signals, PM is superior to FM. Actually, the optimum performance is realized neither by pure PM nor by pure FM, but by something in between.

Power of an Angle-Modulated Wave

Although the instantaneous frequency and phase of an angle-modulated wave can vary with time, the amplitude A remains constant. Hence, the power of an angle-modulated wave (PM or FM) is always $A^2/2$, regardless of the value of k_p or k_f.

Example 5.1 Sketch FM and PM waves for the modulating signal $m(t)$ shown in Fig. 5.4a. The constants k_f and k_p are $2\pi \times 10^5$ and 10π, respectively, and the carrier frequency f_c is 100 MHz.

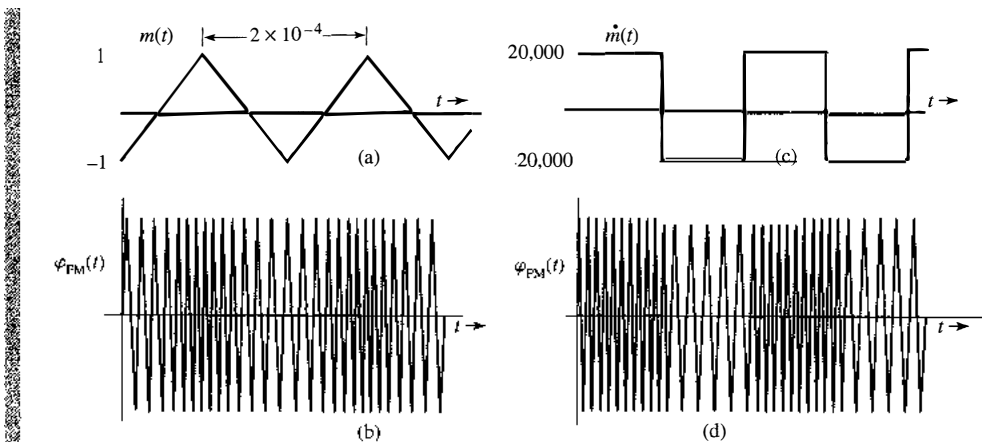


Figure 5.4 FM and PM waveforms.

For FM:

$$\omega_i = \omega_c + k_f m(t)$$

Dividing throughout by 2π, we have the equation in terms of the variable f (frequency in hertz). The instantaneous frequency f_i is

$$\begin{aligned} f_i &= f_c + \frac{k_f}{2\pi} m(t) \\ &= 10^8 + 10^5 m(t) \\ (f_i)_{\min} &= 10^8 + 10^5 [m(t)]_{\min} = 99.9 \text{ MHz} \\ (f_i)_{\max} &= 10^8 + 10^5 [m(t)]_{\max} = 100.1 \text{ MHz} \end{aligned}$$

Because $m(t)$ increases and decreases linearly with time, the instantaneous frequency increases linearly from 99.9 to 100.1 MHz over a half-cycle and decreases linearly from 100.1 to 99.9 MHz over the remaining half-cycle of the modulating signal (Fig. 5.4b).

PM for $m(t)$ is FM for $\dot{m}(t)$. This also follows from Eq. (5.3c).

For PM:

$$\begin{aligned} f_i &= f_c + \frac{k_p}{2\pi} \dot{m}(t) \\ &= 10^8 + 5\, \dot{m}(t) \\ (f_i)_{\min} &= 10^8 + 5 [\dot{m}(t)]_{\min} = 10^8 - 10^5 = 99.9 \text{ MHz} \\ (f_i)_{\max} &= 10^8 + 5 [\dot{m}(t)]_{\max} = 100.1 \text{ MHz} \end{aligned}$$

Because $\dot{m}(t)$ switches back and forth from a value of $-20{,}000$ to 20,000, the carrier frequency switches back and forth from 99.9 to 100.1 MHz every half-cycle of $\dot{m}(t)$, as shown in Fig. 5.4d.

This indirect method of sketching PM [using $\dot{m}(t)$ to frequency-modulate a carrier] works as long as $m(t)$ is a continuous signal. If $m(t)$ is discontinuous, it means that the PM signal has sudden phase changes and, hence, $\dot{m}(t)$ contains impulses. This indirect method fails at *points of the discontinuity*. In such a case, a direct approach should be used at the point of discontinuity to specify the sudden phase changes. This is demonstrated in the next example.

Example 5.2 Sketch FM and PM waves for the digital modulating signal $m(t)$ shown in Fig. 5.5a. The constants k_f and k_p are $2\pi \times 10^5$ and $\pi/2$, respectively, and $f_c = 100$ MHz.

For FM:

$$f_i = f_c + \frac{k_f}{2\pi} m(t) = 10^8 + 10^5 m(t)$$

Because $m(t)$ switches from 1 to -1 and vice versa, the FM wave frequency switches back and forth between 99.9 and 100.1 MHz, as shown in Fig. 5.5b. This scheme of carrier

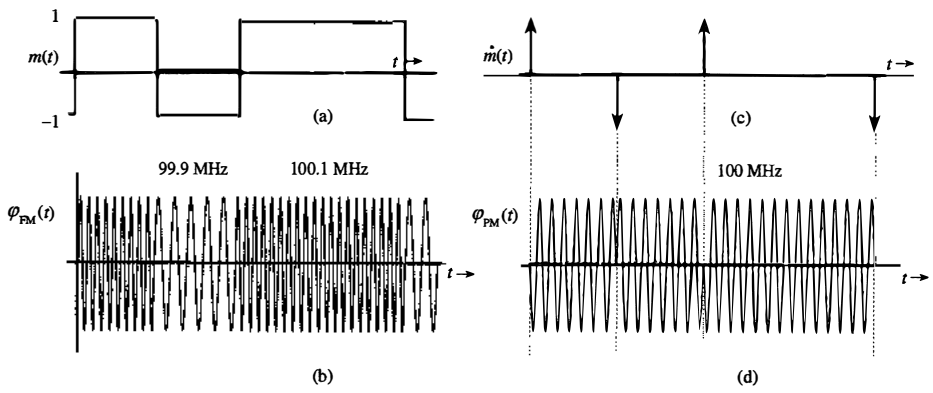


Figure 5.5 FM and PM waveforms.

frequency modulation by a digital signal (Fig. 5.5b) is called **frequency shift keying (FSK)** because information digits are transmitted by keying different frequencies (see Sec. 7.8).

For PM:

$$f_i = f_c + \frac{k_p}{2\pi}\dot{m}(t) = 10^8 + \frac{1}{4}\dot{m}(t)$$

The derivative $\dot{m}(t)$ (Fig. 5.5c) is zero except at points of discontinuity of $m(t)$ where impulses of strength ± 2 are present. This means that the frequency of the PM signal stays the same except at these isolated points of time! It is not immediately apparent how an instantaneous frequency can be changed by an infinite amount and then changed back to the original frequency in zero time. Let us consider the direct approach:

$$\begin{aligned}\varphi_{\text{PM}}(t) &= A\cos[\omega_c t + k_p m(t)] \\ &= A\cos\left[\omega_c t + \frac{\pi}{2}m(t)\right] \\ &= \begin{cases} A\sin\omega_c t & \text{when } m(t) = -1 \\ -A\sin\omega_c t & \text{when } m(t) = 1 \end{cases}\end{aligned}$$

This PM wave is shown in Fig. 5.5d. This scheme of carrier PM by a digital signal is called **phase shift keying (PSK)** because information digits are transmitted by shifting the carrier phase. Note that PSK may also be viewed as a DSB-SC modulation by $m(t)$.

The PM wave $\varphi_{\text{PM}}(t)$ in this case has phase discontinuities at instants where impulses of $\dot{m}(t)$ are located. At these instants, the carrier phase shifts by π instantaneously. A finite phase shift in zero time implies infinite instantaneous frequency at these instants. This agrees with our observation about $\dot{m}(t)$.

The amount of phase discontinuity in $\varphi_{\text{PM}}(t)$ at the instant where $m(t)$ is discontinuous is $k_p m_d$, where m_d is the amount of discontinuity in $m(t)$ at that instant. In the present example, the amplitude of $m(t)$ changes by 2 (from -1 to 1) at the discontinuity. Hence, the phase discontinuity in $\varphi_{\text{PM}}(t)$ is $k_p m_d = (\pi/2) \times 2 = \pi$ rad, which confirms our earlier result.

When $m(t)$ is a digital signal (as in Fig. 5.5a), $\varphi_{\text{PM}}(t)$ shows a phase discontinuity where $m(t)$ has a jump discontinuity. We shall now show that to avoid ambiguity in demodulation, in such a case, the phase deviation $k_p m(t)$ must be restricted to a range $(-\pi, \pi)$. For example, if k_p were $3\pi/2$ in the present example, then

$$\varphi_{\text{PM}}(t) = A \cos\left[\omega_c t + \frac{3\pi}{2} m(t)\right]$$

In this case $\varphi_{\text{PM}}(t) = A \sin \omega_c t$ when $m(t) = 1$ or $-1/3$. This will certainly cause ambiguity at the receiver when $A \sin \omega_c t$ is received. Specifically, the receiver cannot decide the exact value of $m(t)$. Such ambiguity never arises if $k_p m(t)$ is restricted to the range $(-\pi, \pi)$.

What causes this ambiguity? When $m(t)$ has jump discontinuities, the phase of $\varphi_{\text{PM}}(t)$ changes instantaneously. Because a phase $\varphi_o + 2n\pi$ is indistinguishable from the phase φ_o, ambiguities will be inherent in the demodulator unless the phase variations are limited to the range $(-\pi, \pi)$. This means k_p should be small enough to restrict the phase change $k_p m(t)$ to the range $(-\pi, \pi)$.

No such restriction on k_p is required if $m(t)$ is continuous. In this case the phase change is not instantaneous, but gradual over time, and a phase $\varphi_o + 2n\pi$ will exhibit n additional carrier cycles in the case of phase of only φ_o. We can detect the PM wave by using an FM demodulator followed by an integrator (see Prob. 5.4-1). The additional n cycles will be detected by the FM demodulator, and the subsequent integration will yield a phase $2n\pi$. Hence, the phases φ_o and $\varphi_o + 2n\pi$ can be detected without ambiguity. This conclusion can also be verified from Example 5.1, where the maximum phase change $\Delta\varphi = 10\pi$.

Because a band-limited signal cannot have jump discontinuities, we can also say that when $m(t)$ is band-limited, k_p has no restrictions.

5.2 BANDWIDTH OF ANGLE-MODULATED WAVES

Unlike AM, angle modulation is nonlinear and no properties of Fourier transform can be directly applied for its bandwidth analysis. To determine the bandwidth of an FM wave, let us define

$$a(t) = \int_{-\infty}^{t} m(\alpha)\, d\alpha \tag{5.7}$$

and define

$$\hat{\varphi}_{\text{FM}}(t) = A e^{j[\omega_c t + k_f a(t)]} = A e^{jk_f a(t)} e^{j\omega_c t} \tag{5.8a}$$

such that its relationship to the FM signal is

$$\varphi_{\text{FM}}(t) = \text{Re}\left[\hat{\varphi}_{\text{FM}}(t)\right] \tag{5.8b}$$

Expanding the exponential $e^{jk_f a(t)}$ of Eq. (5.8a) in power series yields

$$\hat{\varphi}_{\text{FM}}(t) = A\left[1 + jk_f a(t) - \frac{k_f^2}{2!} a^2(t) + \cdots + j^n \frac{k_f^n}{n!} a^n(t) + \cdots\right] e^{j\omega_c t} \tag{5.9a}$$

and

$$\varphi_{\text{FM}}(t) = \text{Re}\,[\hat{\varphi}_{\text{FM}}(t)]$$
$$= A\left[\cos\,\omega_c t - k_f a(t)\sin\,\omega_c t - \frac{k_f^2}{2!}a^2(t)\cos\,\omega_c t + \frac{k_f^3}{3!}a^3(t)\sin\,\omega_c t + \cdots\right] \quad (5.9b)$$

The modulated wave consists of an unmodulated carrier plus various amplitude-modulated terms, such as $a(t)\sin\,\omega_c t$, $a^2(t)\cos\,\omega_c t$, $a^3(t)\sin\,\omega_c t, \ldots$. The signal $a(t)$ is an integral of $m(t)$. If $M(f)$ is band-limited to B, $A(f)$ is also band-limited* to B. The spectrum of $a^2(t)$ is simply $A(f) * A(f)$ and is band-limited to $2B$. Similarly, the spectrum of $a^n(t)$ is band-limited to nB. Hence, the spectrum consists of an unmodulated carrier plus spectra of $a(t)$, $a^2(t), \ldots, a^n(t), \ldots$, centered at ω_c. Clearly, the modulated wave is not band-limited. It has an infinite bandwidth and is not related to the modulating-signal spectrum in any simple way, as was the case in AM.

Although the bandwidth of an FM wave is theoretically infinite, for practical signals with bounded $|a(t)|$, $|k_f a(t)|$ will remain finite. Because $n!$ increases much faster than $|k_f a(t)|^n$, we have

$$\frac{k_f^n a^n(t)}{n!} \simeq 0 \qquad \text{for large } n$$

Hence, we shall see that most of the modulated-signal power resides in a finite bandwidth. This is the principal foundation of the bandwidth analysis for angle-modulations. There are two distinct possibilities in terms of bandwidths—narrowband FM and wideband FM.

Narrowband Angle Modulation Approximation

Unlike AM, angle modulations are nonlinear. The nonlinear relationship between $a(t)$ and $\varphi(t)$ is evident from the terms involving $a^n(t)$ in Eq. (5.9b). When k_f is very small such that

$$|k_f a(t)| \ll 1$$

then all higher order terms in Eq. (5.9b) are negligible except for the first two. We then have a good approximation

$$\varphi_{\text{FM}}(t) \approx A\left[\cos\,\omega_c t - k_f a(t)\sin\,\omega_c t\right] \quad (5.10)$$

This approximation is a linear modulation that has an expression similar to that of the AM signal with message signal $a(t)$. Because the bandwidth of $a(t)$ is B Hz, the bandwidth of $\varphi_{\text{FM}}(t)$ in Eq. (5.10) is $2B$ Hz according to the frequency-shifting property due to the term $a(t)\sin\,\omega_c t$. For this reason, the FM signal for the case of $|k_f a(t)| \ll 1$ is called **narrowband FM (NBFM)**. Similarly, the **narrowband PM (NBPM)** signal is approximated by

$$\varphi_{\text{PM}}(t) \approx A\left[\cos\,\omega_c t - k_p m(t)\sin\,\omega_c t\right] \quad (5.11)$$

NBPM also has the approximate bandwidth of $2B$.

---

* This is because integration is a linear operation equivalent to passing a signal through a transfer function $1/j2\pi f$. Hence, if $M(f)$ is band-limited to B, $A(f)$ must also be band-limited to B.

A comparison of NBFM [Eq. (5.10)] with AM [Eq. (5.9a)] brings out clearly the similarities and differences between the two types of modulation. Both have the same modulated bandwidth $2B$. The sideband spectrum for FM has a phase shift of $\pi/2$ with respect to the carrier, whereas that of AM is in phase with the carrier. It must be remembered, however, that despite the apparent similarities, the AM and FM signals have very different waveforms. In an AM signal, the oscillation frequency is constant and the amplitude varies with time, whereas in an FM signal, the amplitude stays constant and the frequency varies with time.

Wideband FM (WBFM) Bandwidth Analysis: The Fallacy Exposed

Note that an FM signal is meaningful only if its frequency deviation is large enough. In other words, practical FM chooses the constant k_f large enough that the condition $|k_f a(t)| \ll 1$ is not satisfied. We call FM signals in such cases **wideband FM (WBFM)**. Thus, in analyzing the bandwidth of WBFM, we cannot ignore all the higher order terms in Eq. (5.9b). To begin, we shall take here the route of the pioneers, who by their intuitively simple reasoning came to grief in estimating the FM bandwidth. If we could discover the fallacy in their reasoning, we would have a chance of obtaining a better estimate of the (wideband) FM bandwidth.

Consider a low-pass $m(t)$ with bandwidth B Hz. This signal is well approximated by a staircase signal $\hat{m}(t)$, as shown in Fig. 5.6a. The signal $m(t)$ is now approximated by pulses of constant amplitude. For convenience, each of these pulses will be called a "cell." To ensure

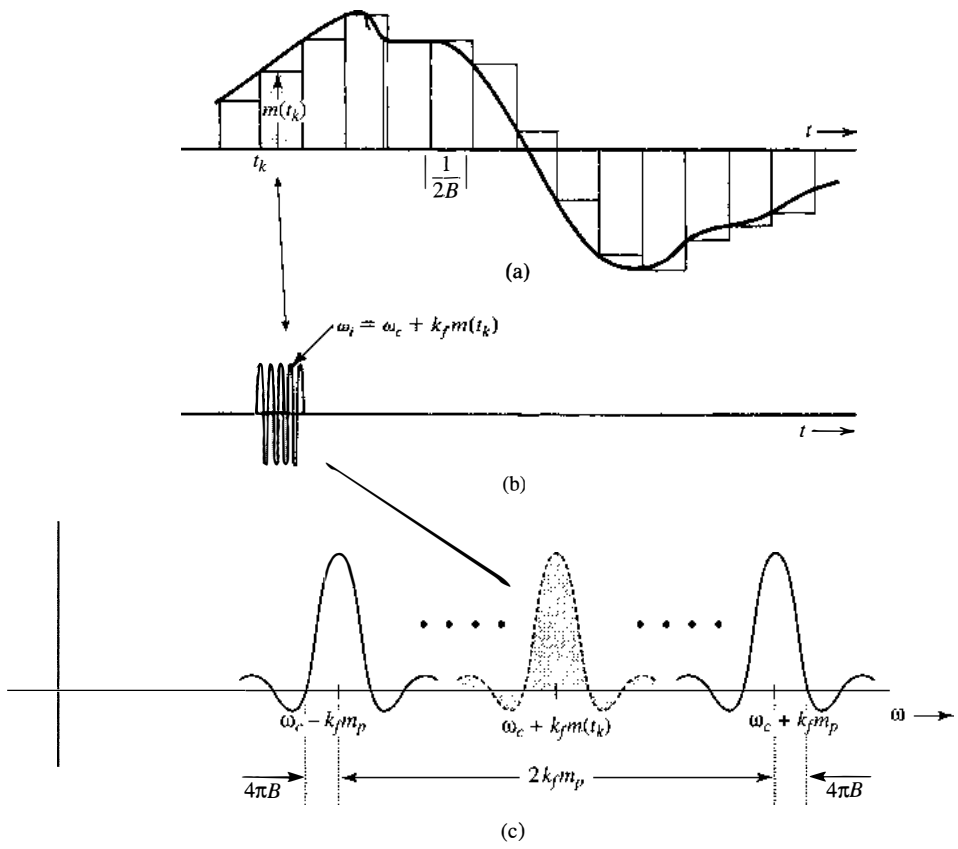


Figure 5.6 Estimation of FM wave bandwidth.

that $\hat{m}(t)$ has all the information of $m(t)$, the cell width in $\hat{m}(t)$ must be no greater than the Nyquist interval of $1/2B$ second according to the sampling theorem (Chapter 6).

It is relatively easier to analyze FM corresponding to $\hat{m}(t)$ because its constant amplitude pulses (cells) of width $T = 1/2B$ second. Consider a typical cell starting at $t = t_k$. This cell has a constant amplitude $m(t_k)$. Hence, the FM signal corresponding to this cell is a sinusoid of frequency $\omega_c + k_f m(t_k)$ and duration $T = 1/2B$, as shown in Fig. 5.6b. The FM signal for $\hat{m}(t)$ consists of a sequence of such constant frequency sinusoidal pulses of duration $T = 1/2B$ corresponding to various cells of $\hat{m}(t)$. The FM spectrum for $\hat{m}(t)$ consists of the sum of the Fourier transforms of these sinusoidal pulses corresponding to all the cells. The Fourier transform of a sinusoidal pulse in Fig. 5.6b (corresponding to the kth cell) is a sinc function shown shaded in Fig. 5.6c see Eq. (3.27a) with $\tau = 1/2B$ and Eq. (3.26) with $f_0 = f_c + k_f m(t_k)/2\pi$:

$$\text{rect}(2Bt)\cos[\omega_c t + k_f m(t_k)t] \Longleftrightarrow \frac{1}{2}\text{sinc}\left[\frac{\omega + \omega_c + k_f m(t_k)}{4B}\right] + \frac{1}{2}\text{sinc}\left[\frac{\omega - \omega_c - k_f m(t_k)}{4B}\right]$$

Note that the spectrum of this pulse is spread out on either side of its center frequency $\omega_c + k_f m(t_k)$ by $4\pi B$ as the main lobe of the sinc function. Figure 5.6c shows the spectra of sinusoidal pulses corresponding to various cells. The minimum and the maximum amplitudes of the cells are $-m_p$ and m_p, respectively. Hence, the minimum and maximum *center* frequencies of the short sinusoidal pulses corresponding to the FM signal for all the cells are $\omega_c - k_f m_p$ and $\omega_c + k_f m_p$, respectively. Consider the sinc main lobe of these frequency responses as significant contribution to the FM bandwidth, as shown in Fig. 5.6c. Hence, the maximum and the minimum significant frequencies in this spectrum are $\omega_c + k_f m_p + 4\pi B$ and $\omega_c - k_f m_p - 4\pi B$, respectively. The FM spectrum bandwidth is approximately

$$B_{\text{FM}} = \frac{1}{2\pi}(2k_f m_p + 8\pi B) = 2\left(\frac{k_f m_p}{2\pi} + 2B\right) \text{ Hz}$$

We can now understand the fallacy in the reasoning of the pioneers. The maximum and minimum carrier frequencies are $\omega_c + k_f m_p$ and $\omega_c - k_f m_p$, respectively. Hence, it was reasoned that the spectral components must also lie in this range, resulting in the FM bandwidth of $2k_f m_p$. The implicit assumption was that a sinusoid of frequency ω has its entire spectrum concentrated at ω. Unfortunately, this is true only of the everlasting sinusoid with $T = \infty$ (because it turns the sinc function into an impulse). For a sinusoid of finite duration T seconds, the spectrum is spread out by the sinc on either side of ω by at least the main lobe width of $2\pi/T$. The pioneers had missed this spreading effect.

For notational convenience, given the deviation of the carrier frequency (in radians per second) by $\pm k_f m_p$, we shall denote the *peak frequency deviation* in hertz by Δf. Thus,

$$\Delta f = k_f \frac{m_{\max} - m_{\min}}{2 \cdot 2\pi} = f = k_f \frac{m_p}{2\pi}$$

The estimated FM bandwidth (in hertz) can then be expressed as

$$B_{\text{FM}} \simeq 2(\Delta f + 2B) \tag{5.12}$$

The bandwidth estimate thus obtained is somewhat higher than the actual value because this is the bandwidth corresponding to the staircase approximation of $m(t)$, not the actual $m(t)$, which is considerably smoother. Hence, the actual FM bandwidth is somewhat smaller than

this value. Based on Fig. 5.6c, it is clear that a better FM bandwidth approximation is between

$$[2\Delta f,\ 2\Delta f + 4B]$$

Therefore, we should readjust our bandwidth estimation. To make this midcourse correction, we observe that for the case of NBFM, k_f is very small. Hence, given a fixed m_p, Δf is very small (in comparison to B) for NBFM. In this case, we can ignore the small Δf term in Eq. (5.12) with the result

$$B_{\text{FM}} \approx 4B$$

But we showed earlier that for narrowband, the FM bandwidth is approximately $2B$ Hz. This indicates that a better bandwidth estimate is

$$B_{\text{FM}} = 2(\Delta f + B) = 2\left(\frac{k_f m_p}{2\pi} + B\right) \tag{5.13}$$

This is precisely the result obtained by Carson,[1] who investigated this problem rigorously for tone modulation [sinusoidal $m(t)$]. This formula goes under the name **Carson's rule** in the literature. Observe that for a truly wideband case, where $\Delta f \gg B$, Eq. (5.13) can be approximated as

$$B_{\text{FM}} \approx 2\Delta f \qquad \Delta f \gg B \tag{5.14}$$

Because $\Delta\omega = k_f m_p$, this formula is precisely what the pioneers had used for FM bandwidth. The only mistake was in thinking that this formula will hold for all cases, especially for the narrowband case, where $\Delta f \ll B$.

We define a deviation ratio β as

$$\beta = \frac{\Delta f}{B} \tag{5.15}$$

Carson's rule can be expressed in terms of the deviation ratio as

$$B_{\text{FM}} = 2B(\beta + 1) \tag{5.16}$$

The deviation ratio controls the amount of modulation and, consequently, plays a role similar to the modulation index in AM. Indeed, for the special case of tone-modulated FM, the deviation ratio β is called the **modulation index.**

Phase Modulation

All the results derived for FM can be directly applied to PM. Thus, for PM, the instantaneous frequency is given by

$$\omega_i = \omega_c + k_p \dot{m}(t)$$

Therefore, the peak frequency deviation Δf is given by

$$\Delta f = k_p \frac{[\dot{m}(t)]_{\max} - [\dot{m}(t)_{\min}]}{2 \cdot 2\pi} \tag{5.17a}$$

If we assume that

$$\dot{m}_p = [\dot{m}(t)]_{\max} = -|\dot{m}(t)_{\min}| \tag{5.17b}$$

then

$$\Delta f = k_p \frac{\dot{m}_p}{2\pi} \tag{5.17c}$$

Therefore,*

$$B_{\text{PM}} = 2(\Delta f + B) \tag{5.18a}$$

$$= 2\left(\frac{k_p \dot{m}_p}{2\pi} + B\right) \tag{5.18b}$$

One very interesting aspect of FM is that $\Delta\omega = k_f m_p$ depends only on the peak value of $m(t)$. It is independent of the spectrum of $m(t)$. On the other hand, in PM, $\Delta\omega = k_p \dot{m}_p$ depends on the peak value of $\dot{m}(t)$. But $\dot{m}(t)$ depends strongly on the spectral composition of $m(t)$. The presence of higher frequency components in $m(t)$ implies rapid time variations, resulting in a higher value of $\dot{m}_p$. Conversely, predominance of lower frequency components will result in a lower value of $\dot{m}_p$. Hence, whereas the FM signal bandwidth [Eq. (5.13)] is practically independent of the spectral shape of $m(t)$, the PM signal bandwidth [Eq. (5.18)] is strongly affected by the spectral shape of $m(t)$. For $m(t)$ with a spectrum concentrated at lower frequencies, B_{PM} will be smaller than when the spectrum of $m(t)$ is concentrated at higher frequencies.

Spectral Analysis of Tone Frequency Modulation

For an FM carrier with a generic message signal $m(t)$, the spectral analysis requires the use of staircase signal approximation. Tone modulation is a special case for which a precise spectral analysis is possible: that is, when $m(t)$ is a sinusoid. We use this special case to verify the FM bandwidth approximation. Let

$$m(t) = \alpha \cos \omega_m t$$

From Eq. (5.7), with the assumption that initially $a(-\infty) = 0$, we have

$$a(t) = \frac{\alpha}{\omega_m} \sin \omega_m t$$

Thus, from Eq. (5.8a), we have

$$\hat{\varphi}_{\text{FM}}(t) = Ae^{(j\omega_c t + k_f \alpha/\omega_m \sin \omega_m t)}$$

Moreover

$$\Delta\omega = k_f m_p = \alpha k_f$$

* Equation (5.17a) can be applied only if $m(t)$ is a continuous function of time. If $m(t)$ has jump discontinuities, its derivative does not exist. In such a case, we should use the direct approach (discussed in Example 5.2) to find $\varphi_{\text{PM}}(t)$ and then determine $\Delta\omega$ from $\varphi_{\text{PM}}(t)$.

and the bandwidth of $m(t)$ is $2\pi B = \omega_m$ rad/s. The deviation ratio (or in this case, the modulation index) is

$$\beta = \frac{\Delta f}{B} = \frac{\Delta\omega}{2\pi B} = \frac{\alpha k_f}{\omega_m}$$

Hence,

$$\begin{aligned}\hat{\varphi}_{\text{FM}}(t) &= Ae^{j(\omega_c t + \beta \sin \omega_m t)} \\ &= Ae^{j\omega_c t}(e^{j\beta \sin \omega_m t}) \end{aligned} \tag{5.19}$$

Note that $e^{j\beta \sin \omega_m t}$ is a periodic signal with period $2\pi/\omega_m$ and can be expanded by the exponential Fourier series, as usual,

$$e^{j\beta \sin \omega_m t} = \sum_{n=-\infty}^{\infty} D_n e^{jn\omega_m t}$$

where

$$D_n = \frac{\omega_m}{2\pi} \int_{-\pi/\omega_m}^{\pi/\omega_m} e^{j\beta \sin \omega_m t} e^{-jn\omega_m t}\, dt = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{j(\beta \sin x - nx)}\, dx$$

The integral on the right-hand side cannot be evaluated in a closed form but must be integrated by expanding the integrand in infinite series. This integral has been extensively tabulated and is denoted by $J_n(\beta)$, the Bessel function of the first kind and the nth order. These functions are plotted in Fig. 5.7a as a function of n for various values of β. Thus,

$$e^{j\beta \sin \omega_m t} = \sum_{n=-\infty}^{\infty} J_n(\beta) e^{jn\omega_m t} \tag{5.20}$$

Substituting Eq. (5.20) into Eq. (5.19), we get

$$\hat{\varphi}_{\text{FM}}(t) = A \sum_{n=-\infty}^{\infty} J_n(\beta) e^{j(\omega_c t + n\omega_m t)}$$

and

$$\varphi_{\text{FM}}(t) = A \sum_{n=-\infty}^{\infty} J_n(\beta) \cos(\omega_c + n\omega_m)t$$

The tone-modulated FM signal has a carrier component and an infinite number of sidebands of frequencies $\omega_c \pm \omega_m$, $\omega_c \pm 2\omega_m, \ldots, \omega_c \pm n\omega_m, \ldots$, as shown in Fig. 5.7b. This is in stark contrast to the DSB-SC spectrum of only one sideband on either side of the carrier frequency. The strength of the nth sideband at $\omega = \omega_c + n\omega_m$ is* $J_n(\beta)$. From the plots of $J_n(\beta)$ in Fig. 5.7a, it can be seen that for a given β, $J_n(\beta)$ decreases with n, and there are only a finite number

* Also $J_{-n}(\beta) = (-1)^n J_n(\beta)$. Hence, the magnitude of the LSB at $\omega = \omega_c - n\omega_m$ is the same as that of the USB at $\omega = \omega_c + n\omega_m$.

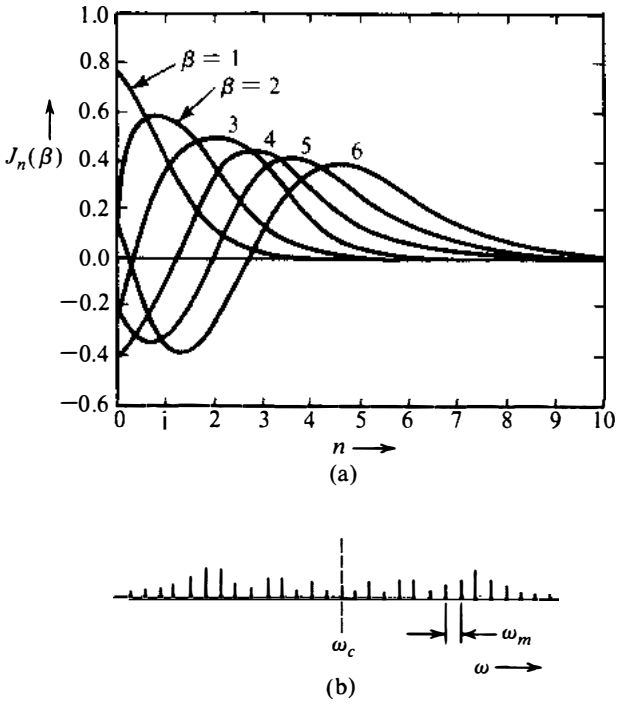


Figure 5.7 (a) Variations of $J_n(\beta)$ as a function of n for various values of β. (b) Tone-modulated FM wave spectrum.

of significant sideband spectral lines. It can be seen from Fig. 5.7a that $J_n(\beta)$ is negligible for $n > \beta + 1$. Hence, the number of significant sideband impulses is $\beta + 1$. The bandwidth of the FM carrier is given by

$$\begin{aligned} B_{\text{FM}} &= 2(\beta + 1)f_m \\ &= 2(\Delta f + B) \end{aligned}$$

which corroborates our previous result [Eqs. (5.13)]. When $\beta \ll 1$ (NBFM), there is only one significant sideband and the bandwidth $B_{\text{FM}} = 2f_m = 2B$. It is important to note that this tone modulation case analysis is a verification, not a proof, of Carson's formula.

In the literature, tone modulation in FM is often discussed in great detail. Since, however, angle modulation is a nonlinear modulation, the results derived for tone modulation may have little connection to practical situations. Indeed, these results are meaningless at best and misleading at worst when generalized to practical signals.* As authors and instructors, we feel that too much emphasis on tone modulation can be misleading. For this reason we have omitted further such discussion here.

The method for finding the spectrum of a tone-modulated FM wave can be used for finding the spectrum of an FM wave when $m(t)$ is a general periodic signal. In this case,

$$\hat{\varphi}_{\text{FM}}(t) = Ae^{j\omega_c t}[e^{jk_f a(t)}]$$

* For instance, based on tone modulation analysis, it is often stated that FM is superior to PM by a factor of 3 in terms of the output SNR. This is in fact untrue for most of the signals encountered in practices.

Because $a(t)$ is a periodic signal, $e^{jk_f a(t)}$ is also a periodic signal, which can be expressed as an exponential Fourier series in the preceding expression. After this, it is relatively straightforward to write $\varphi_{\text{FM}}(t)$ in terms of the carrier and the sidebands.

Example 5.3

(a) Estimate B_{FM} and B_{PM} for the modulating signal $m(t)$ in Fig. 5.4a for $k_f = 2\pi \times 10^5$ and $k_p = 5\pi$. Assume the essential bandwidth of the periodic $m(t)$ as the frequency of its third harmonic.

(b) Repeat the problem if the amplitude of $m(t)$ is doubled [if $m(t)$ is multiplied by 2].

(a) The peak amplitude of $m(t)$ is unity. Hence, $m_p = 1$. We now determine the essential bandwidth B of $m(t)$. It is left as an exercise for the reader to show that the Fourier series for this periodic signal is given by

$$m(t) = \sum_n C_n \cos n\omega_0 t \qquad \omega_0 = \frac{2\pi}{2 \times 10^{-4}} = 10^4\pi$$

where

$$C_n = \begin{cases} \dfrac{8}{\pi^2 n^2} & n \text{ odd} \\ 0 & n \text{ even} \end{cases}$$

It can be seen that the harmonic amplitudes decrease rapidly with n. The third harmonic is only 11% of the fundamental, and the fifth harmonic is only 4% of the fundamental. This means the third and fifth harmonic powers are 1.21 and 0.16%, respectively, of the fundamental component power. Hence, we are justified in assuming the essential bandwidth of $m(t)$ as the frequency of its third harmonic, that is,

$$B = 3 \times \frac{10^4}{2} = 15\,\text{kHz}$$

For FM:

$$\Delta f = \frac{1}{2\pi} k_f m_p = \frac{1}{2\pi}(2\pi \times 10^5)(1) = 100$$

and

$$B_{\text{FM}} = 2(\Delta f + B) = 230\,\text{kHz}$$

Alternatively, the deviation ratio β is given by

$$\beta = \frac{\Delta f}{B} = \frac{100}{15}$$

and

$$B_{\text{FM}} = 2B(\beta + 1) = 30\left(\frac{100}{15} + 1\right) = 230\,\text{kHz}$$

For PM: The peak amplitude of $\dot{m}(t)$ is 20,000 and

$$\Delta f = \frac{1}{2\pi} k_p \dot{m}_p = 50 \text{ kHz}$$

Hence,

$$B_{\text{PM}} = 2(\Delta f + B) = 130 \text{ kHz}$$

Alternately, the deviation ratio β is given by

$$\beta = \frac{\Delta f}{B} = \frac{50}{15}$$

and

$$B_{\text{PM}} = 2B(\beta + 1) = 30\left(\frac{50}{15} + 1\right) = 130 \text{ kHz}.$$

(b) Doubling $m(t)$ doubles its peak value. Hence, $m_p = 2$. But its bandwidth is unchanged so that $B = 15$ kHz.
For FM:

$$\Delta f = \frac{1}{2\pi} k_f m_p = \frac{1}{2\pi}(2\pi \times 10^5)(2) = 200 \text{ kHz}$$

and

$$B_{\text{FM}} = 2(\Delta f + B) = 430 \text{ kHz}$$

Alternately, the deviation ratio β is given by

$$\beta = \frac{\Delta f}{B} = \frac{200}{15}$$

and

$$B_{\text{FM}} = 2B(\beta + 1) = 30\left(\frac{200}{15} + 1\right) = 430 \text{ kHz}$$

For PM: Doubling $m(t)$ doubles its derivative so that now $\dot{m}_p = 40,000$, and

$$\Delta f = \frac{1}{2\pi} k_p \dot{m}_p = 100 \text{ kHz}$$

and

$$B_{\text{PM}} = 2(\Delta f + B) = 230 \text{ kHz}$$

Alternately, the deviation ratio β is given by

$$\beta = \frac{\Delta f}{B} = \frac{100}{15}$$

and

$$B_{\text{PM}} = 2B(\beta + 1) = 30\left(\frac{100}{15} + 1\right) = 230\,\text{kHz}$$

Observe that doubling the signal amplitude [doubling $m(t)$] roughly doubles frequency deviation Δf of both FM and PM waveforms.

Example 5.4 Repeat Example 5.1 if $m(t)$ is time-expanded by a factor of 2: that is, if the period of $m(t)$ is 4×10^{-4}.

Recall that time expansion of a signal by a factor of 2 reduces the signal spectral width (bandwidth) by a factor of 2. We can verify this by observing that the fundamental frequency is now 2.5 kHz, and its third harmonic is 7.5 kHz. Hence, $B = 7.5$ kHz, which is half the previous bandwidth. Moreover, time expansion does not affect the peak amplitude and thus $m_p = 1$. However, $\dot{m}_p$ is halved, that is, $\dot{m}_p = 10,000$.
For FM:

$$\Delta f = \frac{1}{2\pi} k_f m_p = 100\,\text{kHz}$$

$$B_{\text{FM}} = 2(\Delta f + B) = 2(100 + 7.5) = 215\,\text{kHz}$$

For PM:

$$\Delta f = \frac{1}{2\pi} k_p \dot{m}_p = 25\,\text{kHz}$$

$$B_{\text{PM}} = 2(\Delta f + B) = 65\,\text{kHz}$$

Note that time expansion of $m(t)$ has very little effect on the FM bandwidth, but it halves the PM bandwidth. This verifies our observation that the PM spectrum is strongly dependent on the spectrum of $m(t)$.

Example 5.5 An angle-modulated signal with carrier frequency $\omega_c = 2\pi \times 10^5$ is described by the equation

$$\varphi_{\text{EM}}(t) = 10\cos(\omega_c t + 5\,\sin\,3000t + 10\,\sin\,2000\pi t)$$

(a) Find the power of the modulated signal.
(b) Find the frequency deviation Δf.
(c) Find the deviation ratio β.

(d) Find the phase deviation $\Delta\phi$.
(e) Estimate the bandwidth of $\varphi_{\text{EM}}(t)$.

The signal bandwidth is the highest frequency in $m(t)$ (or its derivative). In this case $B = 2000\pi/2\pi = 1000$ Hz.

(a) The carrier amplitude is 10, and the power is

$$P = \frac{10^2}{2} = 50$$

(b) To find the frequency deviation Δf, we find the instantaneous frequency ω_i, given by

$$\omega_i = \frac{d}{dt}\theta(t) = \omega_c + 15{,}000 \cos 3000t + 20{,}000\pi \cos 2000\pi t$$

The carrier deviation is $15{,}000 \cos 3000t + 20{,}000\pi \cos 2000\pi t$. The two sinusoids will add in phase at some point, and the maximum value of this expression is $15{,}000 + 20{,}000\pi$. This is the maximum carrier deviation $\Delta\omega$. Hence,

$$\Delta f = \frac{\Delta\omega}{2\pi} = 12{,}387.32 \text{ Hz}$$

(c) $\beta = \dfrac{\Delta f}{B} = \dfrac{12{,}387.32}{1000} = 12.387$

(d) The angle $\theta(t) = \omega t + (5 \sin 3000t + 10 \sin 2000\pi t)$. The phase deviation is the maximum value of the angle inside the parentheses, and is given by $\Delta\phi = 15$ rad.

(e) $B_{\text{EM}} = 2(\Delta f + B) = 26{,}774.65$ Hz
Observe the generality of this method of estimating the bandwidth of an angle-modulated waveform. We need not know whether it is FM, PM, or some other kind of angle modulation. It is applicable to any angle-modulated signal.

A Historical Note: Edwin H. Armstrong (1890–1954)

Today, nobody doubts that FM has a key place in broadcasting and communication. As recently as the 1960s, however, the FM broadcasting seemed doomed because it was so uneconomical in bandwidth usage.

The history of FM is full of strange ironies. The impetus behind the development of FM was the desire to reduce signal transmission bandwidth. Superficial reasoning showed that it was feasible to reduce the transmission bandwidth by using FM. But the experimental results showed otherwise. The transmission bandwidth of FM was actually larger than that of AM. Careful mathematical analysis by Carson showed that FM indeed required a larger bandwidth than AM. Unfortunately, Carson did not recognize the compensating advantage of FM in its ability to suppress noise. Without much basis, he concluded that FM introduces inherent distortion and has no compensating advantages whatsoever.[1] In a later paper, he continues



Edwin H. Armstrong. (Reproduced with permission from Armstrong Family Archives.)

"In fact, as more and more schemes are analyzed and tested, and as the essential nature of the problem is more clearly perceivable, we are unavoidably forced to the conclusion that static (noise), like the poor, will always be with us."[2] The opinion of one of the most able mathematicians of the day in the communication field, thus, set back the development of FM by more than a decade. The *noise-suppressing advantage* of FM was later proved by Major Edwin H. Armstrong,[3] a brilliant engineer whose contributions to the field of radio systems are comparable to those of Hertz and Marconi. It was largely the work of Armstrong that was responsible for rekindling the interest in FM.

Although Armstrong did not invent the concept, he has been considered the father of modern FM. Born on December 18, 1890, in New York City, Edwin H. Armstrong is widely regarded as one of the foremost contributors to radio electronics of the twentieth century. Armstrong was credited with the invention of the *regenerative circuit* (U.S. Patent 1,113,149 issued in 1912, while he was a junior at Columbia University), the *superheterodyne circuit* (U.S. Patent 1,342,885 issued in 1918, while serving in the U.S. Army stationed in Paris, during World War I), the *super-regenerative circuit* (U.S. Patent 1,424,065, issued in 1922), and the complete FM radio broadcasting system (U.S. Patent 1,941,066, 1933). All are breakthrough contributions to the radio field. *Fortune* magazine in 1939 declared: Wideband frequency modulation is the fourth, and perhaps the greatest, in a line of Armstrong inventions that have made most of modern broadcasting what it is. Major Armstrong is the acknowledged inventor of the regenerative 'feedback' circuit, which brought radio art out of the crystal-detector headphone stage and made the amplification of broadcasting possible; the superheterodyne circuit, which is the basis of practically all modern radio; and the super-regenerative circuit now in wide use in . . . shortwave systems.[4]

Armstrong was the last of the breed of the lone attic inventors. After receiving his FM patents in 1933, he gave his now famous paper (which later appeared in print as in the proceedings of the IRE[3]), accompanied by the first public demonstration of FM broadcasting on November 5, 1935, at the New York section meeting of the Institute of Radio Engineers (IRE, a predecessor of the IEEE). His success in dramatically reducing static noise using FM was not fully embraced by the broadcast establishment, which perceived FM as a threat to its vast commercial investment in AM radio. To establish FM broadcasting, Armstrong fought a

long and costly battle with the radio broadcast establishment, which, abetted by the Federal Communications Commission (FCC), fought tooth and nail to resist FM. Still, by December 1941, 67 commercial FM stations had been authorized with as many as half a million receivers in use and 43 applications were pending. In fact, the Radio Technical Planning Board (RTPB) made its final recommendation during the September 1944 FCC hearing that FM be given 75 channels in the band from 41 to 56 MHz.

Despite the recommendation of the RTPB, which was supposed to be the best advice available from the radio engineering community, strong lobbying for the FCC to shift the FM band persisted, mainly by those who propagated the concern that strong radio interferences in the 40 MHz band might be possible as a result ionospheric reflection. Then in June 1945, the FCC, on the basis of erroneous testimony of a technical expert, abruptly shifted the allocated bandwidth of FM from the 42- to 50-MHz range to the 88- to 108-MHz. This dealt a crippling blow to FM by making obsolete more than half a million receivers and equipment (transmitters, antennas, etc.) that had been built and sold by the FM industry to 50 FM stations since 1941 for the 42 to 50 MHz band. Armstrong fought the decision, and later succeeded in getting the technical expert to admit his error. In spite of all this, the FCC allocations remained unchanged. Armstrong spent the sizable fortune he had made from his inventions in legal struggles. The broadcast giants, which had so strongly resisted FM, turned around and used his inventions without paying him royalties. Armstrong spent much of his time in court in some of the longest, most notable, and acrimonious patent suits of the era.[5] In the end, with his funds depleted, his energy drained, and his family life shattered, a despondent Armstrong committed suicide: (in 1954) he walked out of a window of his thirteenth floor apartment in New York's River House.

Armstrong's widow continued the legal battles and won. By the 1960s, FM was clearly established as the superior radio system,[6] and Edwin H. Armstrong was fully recognized as the inventor of frequency modulation. In 1955 the ITU added him to its roster of great inventors. In 1980 Edwin H. Armstrong was inducted into the U.S. National Inventors Hall of Fame, and his picture was put on a U.S. postage stamp in 1983.[7]

5.3 GENERATING FM WAVES

Basically, there are two ways of generating FM waves: **indirect** and **direct**. We first describe the narrowband FM generator that is utilized in the **indirect FM generation** of wideband angle modulation signals.

NBFM Generation

For NBFM and NBPM signals, we have shown earlier that because $|k_f a(t)| \ll 1$ and $|k_p m(t)| \ll 1$, respectively, the modulated signals can be approximated by

$$\varphi_{\text{NBFM}}(t) \simeq A[\cos \omega_c t - k_f a(t) \sin \omega_c t] \tag{5.21a}$$

$$\varphi_{\text{NBPM}}(t) \simeq A[\cos \omega_c t - k_p m(t) \sin \omega_c t] \tag{5.21b}$$

Both approximations are linear and are similar to the expression of the AM wave. In fact, Eqs. (5.21) suggest a possible method of generating narrowband FM and PM signals by using DSB-SC modulators. The block diagram representation of such systems appears in Fig. 5.8.

It is important to point out that the NBFM generated by Fig. 5.8b has some distortion because of the approximation in Eq. (5.10). The output of this NBFM modulator also has some amplitude variations. A nonlinear device designed to limit the amplitude of a bandpass signal can remove most of this distortion.

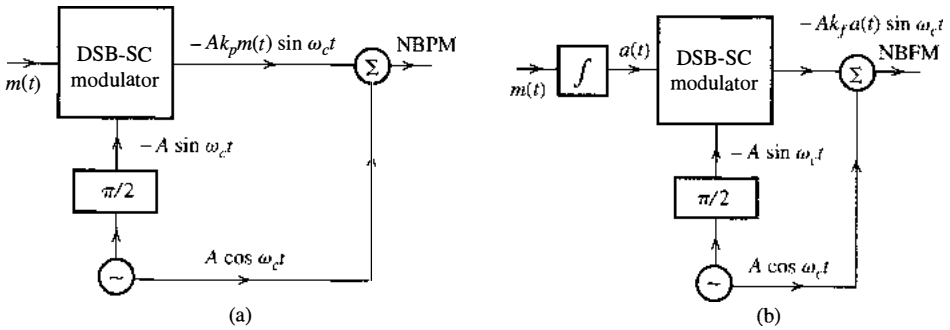


Figure 5.8
(a) Narrowband PM generator. (b) Narrowband FM signal generator.

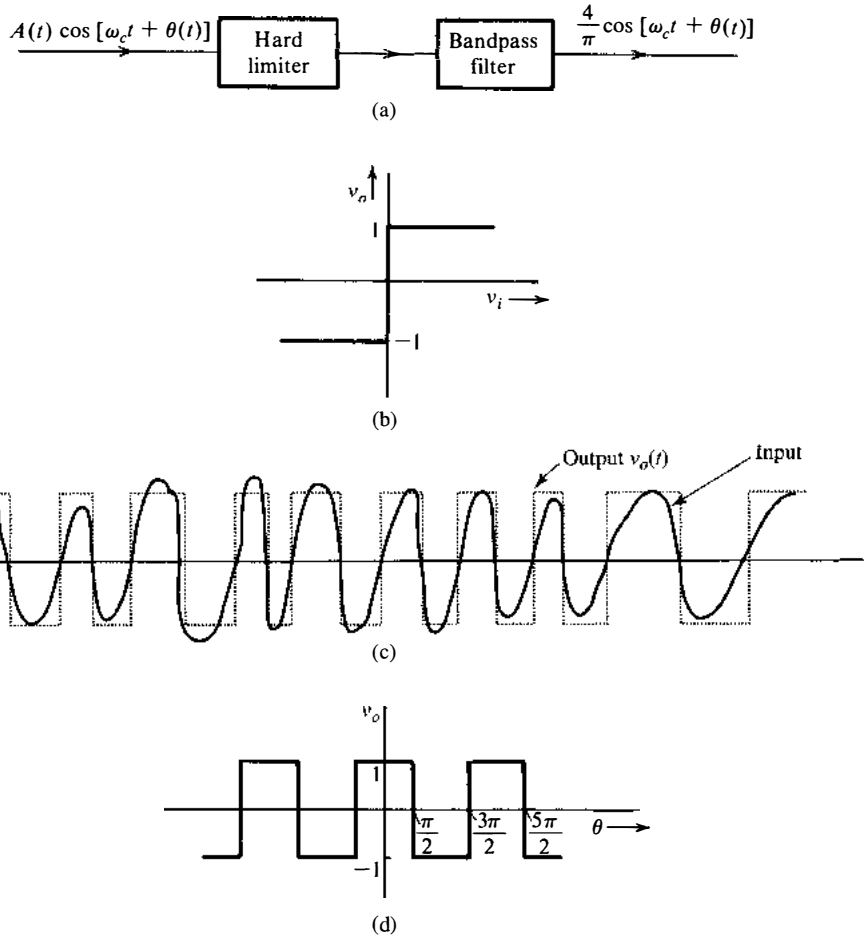


Figure 5.9
(a) Hard limiter and bandpass filter used to remove amplitude variations in FM wave. (b) Hard limiter input-output characteristic. (c) Hard limiter input and the corresponding output. (d) Hard limiter output as a function of θ.

Bandpass Limiter

The amplitude variations of an angle-modulated carrier can be eliminated by what is known as a **bandpass limiter**, which consists of a hard limiter followed by a bandpass filter (Fig. 5.9a). The input-output characteristic of a hard limiter is shown in Fig. 5.9b. Observe that the bandpass limiter output to a sinusoid will be a square wave of unit amplitude regardless of the incoming sinusoidal amplitude. Moreover, the zero crossings of the incoming sinusoid are preserved

in the output because when the input is zero, the output is also zero (Fig. 5.9b). Thus an angle-modulated sinusoidal input $v_i(t) = A(t) \cos \theta(t)$ results in a constant amplitude, angle-modulated square wave $v_o(t)$, as shown in Fig. 5.9c. As we have seen, such a nonlinear operation preserves the angle modulation information. When $v_o(t)$ is passed through a bandpass filter centered at ω_c, the output is a angle-modulated wave, of constant amplitude. To show this, consider the incoming angle-modulated wave

$$v_i(t) = A(t) \cos \theta(t)$$

where

$$\theta(t) = \omega_c t + k_f \int_{-\infty}^{t} m(\alpha)\, d\alpha$$

The output $v_o(t)$ of the hard limiter is $+1$ or -1, depending on whether $v_i(t) = A(t) \cos \theta(t)$ is positive or negative (Fig. 5.9c). Because $A(t) \geq 0$, $v_o(t)$ can be expressed as a function of θ:

$$v_o(\theta) = \begin{cases} +1 & \cos \theta > 0 \\ -1 & \cos \theta < 0 \end{cases}$$

Hence, v_o as a function of θ is a periodic square wave function with period 2π (Fig. 5.9d), which can be expanded by a Fourier series (Chapter 2)

$$v_o(\theta) = \frac{4}{\pi}\left(\cos \theta - \frac{1}{3} \cos 3\theta + \frac{1}{5} \cos 5\theta + \cdots\right)$$

At any instant t, $\theta = \omega_c t + k_f \int m(\alpha)\, d\alpha$. Hence, the output v_o as a function of time is given by

$$\begin{aligned} v_o[\theta(t)] &= v_o\left[\omega_c t + k_f \int m(\alpha)\, d\alpha\right] \\ &= \frac{4}{\pi}\left\{\cos\left[\omega_c t + k_f \int m(\alpha)\, d\alpha\right] - \frac{1}{3} \cos 3\left[\omega_c t + k_f \int m(\alpha)\, d\alpha\right]\right. \\ &\quad \left. + \frac{1}{5} \cos 5\left[\omega_c t + k_f \int m(\alpha)\, d\alpha\right] \cdots\right\} \end{aligned}$$

The output, therefore, has the original FM wave plus frequency-multiplied FM waves with multiplication factors of 3, 5, 7, ... We can pass the output of the hard limiter through a bandpass filter with a center frequency ω_c and a bandwidth B_{FM}, as shown in Fig. 5.9a. The filter output $e_o(t)$ is the desired angle-modulated carrier with a constant amplitude,

$$e_o(t) = \frac{4}{\pi} \cos\left[\omega_c(t) + k_f \int m(\alpha) d\alpha\right]$$

Although we derived these results for FM, this applies to PM (angle modulation in general) as well. The bandpass filter not only maintains the constant amplitude of the angle-modulated carrier but also partially suppresses the channel noise when the noise is small.[8]

Indirect Method of Armstrong

In Armstrong's indirect method, NBFM is generated as shown in Fig. 5.8b [or Eq. (5.10)]. The NBFM is then converted to WBFM by using additional **frequency multipliers**.

A frequency multiplier can be realized by a nonlinear device followed by a bandpass filer. First consider a nonlinear device whose output signal $y(t)$ to an input $x(t)$ is given by

$$y(t) = a_2 x^2(t)$$

If an FM signal passes through this device, then the output signal will be

$$\begin{aligned} y(t) &= a_2 \cos^2\left[\omega_c t + k_f \int m(\alpha)\, d\alpha\right] \\ &= 0.5a_2 + 0.5a_2 \cos\left[2\omega_c t + 2k_f \int m(\alpha)\, d\alpha\right] \end{aligned} \tag{5.22}$$

Thus, a bandpass filter centered at $2\omega_c$ would recover an FM signal with twice the original instantaneous frequency. To generalize, a nonlinear device may have the characteristic of

$$y(t) = a_0 + a_1 x(t) + a_2 x^2(t) + \cdots + a_n x^n(t) \tag{5.23}$$

If $x(t) = A \cos\left[\omega_c t + k_f \int m(\alpha)\, d\alpha\right]$, then by using trigonometric identities, we can readily show that $y(t)$ is of the form

$$\begin{aligned} y(t) = c_o + c_1 \cos\left[\omega_c t + k_f \int m(\alpha)\, d\alpha\right] + c_2 \cos\left[2\omega_c t + 2k_f \int m(\alpha)\, d\alpha\right] \\ + \cdots + c_n \cos\left[n\omega_c t + nk_f \int m(\alpha)\, d\alpha\right] \end{aligned} \tag{5.24}$$

Hence, the output will have spectra at ω_c, $2\omega_c, \ldots,$ $n\omega_c$, with frequency deviations Δf, $2\Delta f, \ldots, n\Delta f$, respectively. Each one of these components is an FM signal separated from the others. Thus, a bandpass filter centering at $n\omega_c$ can recover an FM signal whose instantaneous frequency has been multiplied by a factor of n. These devices, consisting of nonlinearity and bandpass filters, are known as **frequency multipliers**. In fact, a frequency multiplier can increase both the carrier frequency and the frequency deviation by an integer n. Thus, if we want a twelfth-fold increase in the frequency deviation, we can use a twelfth-order nonlinear device or two second-order and one third-order devices in cascade. The output has a bandpass filter centered at $12\omega_c$, so that it selects only the appropriate term, whose carrier frequency as well as the frequency deviation Δf are 12 times the original values.

This forms the basis of the Armstrong indirect frequency modulator. First, generate an NBFM approximately. Then multiply the NBFM frequency and limit its amplitude variation. Generally, we require to increase Δf by a very large factor n. This increases the carrier frequency also by n. Such a large increase in the carrier frequency may not be needed. In this case we can apply frequency mixing (see Example 4.2, Fig. 4.7) to shift down the carrier frequency to the desired value.

A simplified diagram of a commercial FM transmitter using Armstrong's method is shown in Fig. 5.10. The final output is required to have a carrier frequency of 91.2 MHz and $\Delta f = 75$ kHz. We begin with NBFM with a carrier frequency $f_{c_1} = 200$ kHz generated by a crystal oscillator. This frequency is chosen because it is easy to construct stable crystal oscillators as well as balanced modulators at this frequency. To maintain $\beta \ll 1$, as required in NBPM, the

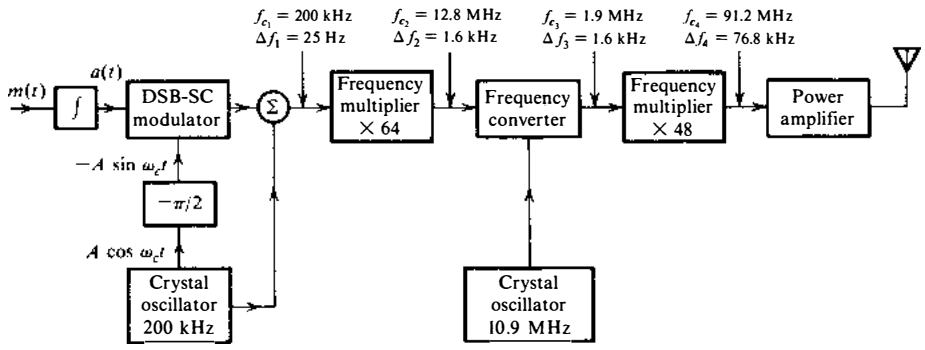


Figure 5.10 Block diagram of the Armstrong indirect FM transmitter.

deviation Δf is chosen to be 25 Hz. For tone modulation, $\beta = \Delta f / f_m$. The baseband spectrum (required for high-fidelity purposes) ranges from 50 Hz to 15 kHz. The choice of $\Delta f = 25$ Hz is reasonable because it gives $\beta = 0.5$ for the worst possible case ($f_m = 50$).

To achieve $\Delta f = 75$ kHz, we need a multiplication of 75,000/25 = 3000. This can be done by two multiplier stages, of 64 and 48, as shown in Fig. 5.10, giving a total multiplication of $64 \times 48 = 3072$, and $\Delta f = 76.8$ kHz.* The multiplication is effected by using frequency doublers and triplers in cascade, as needed. Thus, a multiplication of 64 can be obtained by six doublers in cascade, and a multiplication of 48 can be obtained by four doublers and a tripler in cascade. Multiplication of $f_c = 200$ kHz by 3072, however, would yield a final carrier of about 600 MHz. This problem is solved by using a frequency translation, or conversion, after the first multiplier (Fig. 5.10). The first multiplication by 64 results in the carrier frequency $f_{c_2} = 200\,\text{kHz} \times 64 = 12.8$ MHz, and the carrier deviation $\Delta f_2 = 25 \times 64 = 1.6$ kHz. We now use a frequency converter (or mixer) with carrier frequency 10.9 MHz to shift the entire spectrum. This results in a new carrier frequency $f_{c_3} = 12.8 - 10.9 = 1.9$ MHz. The frequency converter shifts the entire spectrum without altering Δf. Hence, $\Delta f_3 = 1.6$ kHz. Further multiplication, by 48, yields $f_{c_4} = 1.9 \times 48 = 91.2$ MHz and $\Delta f_4 = 1.6 \times 48 = 76.8$ kHz.

This scheme has an advantage of frequency stability, but it suffers from inherent noise caused by excessive multiplication and distortion at lower modulating frequencies, where $\Delta f / f_m$ is not small enough.

Example 5.6 Discuss the nature of distortion inherent in the Armstrong indirect FM generator.

Two kinds of distortion arise in this scheme: amplitude distortion and frequency distortion. The NBFM wave is given by [Eq. (5.10)]

$$\begin{aligned}\varphi_{\text{FM}}(t) &= A[\cos \omega_c t - k_f a(t) \sin \omega_c t] \\ &= AE(t)\cos[\omega_c t + \theta(t)]\end{aligned}$$

where

$$E(t) = \sqrt{1 + k_f^2 a^2(t)} \qquad \text{and} \qquad \theta(t) = \tan^{-1}[k_f a(t)]$$

---

* If we wish Δf to be exactly 75 kHz instead of 76.8 kHz, we must reduce the narrowband Δf from 25 Hz to 25(75/76.8) = 24.41 Hz.

Amplitude distortion occurs because the amplitude $AE(t)$ of the modulated waveform is not constant. This is not a serious problem because amplitude variations can be eliminated by a bandpass limiter, as discussed earlier in the section (see also Fig. 5.9). Ideally, $\theta(t)$ should be $k_f a(t)$. Instead, the phase $\theta(t)$ in the preceding equation is

$$\theta(t) = \tan^{-1}[k_f a(t)]$$

and the instantaneous frequency $\omega_i(t)$ is

$$\begin{aligned}\omega_i(t) = \dot{\theta}(t) &= \frac{k_f \dot{a}(t)}{1 + k_f^2 a^2(t)} \\ &= \frac{k_f m(t)}{1 + k_f^2 a^2(t)} \\ &= k_f m(t)[1 - k_f^2 a^2(t) + k_f^4 a^4(t) - \cdots]\end{aligned}$$

Ideally, the instantaneous frequency should be $k_f m(t)$. The remaining terms in this equation are the distortion.

Let us investigate the effect of this distortion in tone modulation where $m(t) = \alpha \cos \omega_m t$, $a(t) = \alpha \sin \omega_m t/\omega_m$, and the modulation index $\beta = \alpha k_f/\omega_m$:

$$\omega_i(t) = \beta\omega_m \cos \omega_m t(1 - \beta^2 \sin^2 \omega_m t + \beta^4 \sin^4 \omega_m t - \cdots)$$

It is evident from this equation that the scheme has odd-harmonic distortion, the most important term being the third harmonic. Ignoring the remaining terms, this equation becomes

$$\begin{aligned}\omega_i(t) &\simeq \beta\omega_m \cos \omega_m t(1 - \beta^2 \sin^2 \omega_m t) \\ &= \underbrace{\beta\omega_m \left(1 - \frac{\beta^2}{4}\right) \cos \omega_m t}_{\text{desired}} + \underbrace{\frac{\beta^3\omega_m}{4} \cos 3\omega_m t}_{\text{distortion}}\end{aligned}$$

The ratio of the third-harmonic distortion to the desired signal can be found for the generator in Fig. 5.10. For the NBFM stage,

$$\beta B = \Delta f_1 = 25 \text{ Hz}$$

Hence, the worst possible case occurs at the lower modulation frequency. For example, if the tone frequency is only 50 Hz, then $\beta = 0.5$. In this case the third-harmonic distortion is 1/15, or 6.67%.

Direct Generation

In a voltage-controlled oscillator (VCO), the frequency is controlled by an external voltage. The oscillation frequency varies linearly with the control voltage. We can generate an FM wave by using the modulating signal $m(t)$ as a control signal. This gives

$$\omega_i(t) = \omega_c + k_f m(t)$$

One can construct a VCO using an operational amplifier and a hysteretic comparator,[9] (such as a Schmitt trigger circuit). Another way of accomplishing the same goal is to vary one of the reactive parameters (C or L) of the resonant circuit of an oscillator. A *reverse-biased semiconductor diode* acts as a capacitor whose capacitance varies with the bias voltage. The capacitance of these diodes, known under several trade names (e.g., Varicap, Varactor, Voltacap), can be approximated as a linear function of the bias voltage $m(t)$ over a limited range. In Hartley or Colpitt oscillators, for instance, the frequency of oscillation is given by

$$\omega_0 = \frac{1}{\sqrt{LC}}$$

If the capacitance C is varied by the modulating signal $m(t)$, that is, if

$$C = C_0 - km(t)$$

then

$$\begin{aligned}\omega_0 &= \frac{1}{\sqrt{LC_0\left[1 - \dfrac{km(t)}{C_0}\right]}} \\ &= \frac{1}{\sqrt{LC_0}\left[1 - \dfrac{km(t)}{C_0}\right]^{1/2}} \\ &\approx \frac{1}{\sqrt{LC_0}}\left[1 + \frac{km(t)}{2C_0}\right] \qquad \frac{km(t)}{C_0} \ll 1\end{aligned}$$

Here we have applied the Taylor series approximation

$$(1 + x)^n \approx 1 + nx \qquad |x| \ll 1$$

with $n = 1/2$. Thus,

$$\begin{aligned}\omega_0 &= \omega_c\left[1 + \frac{km(t)}{2C_0}\right] \quad \text{where} \quad \omega_c = \frac{1}{\sqrt{LC_0}} \\ &= \omega_c + k_f m(t) \qquad \text{with} \quad k_f = \frac{k\omega_c}{2C_0}\end{aligned}$$

Because $C = C_0 - km(t)$, the maximum capacitance deviation is

$$\Delta C = km_p = \frac{2k_f C_0 m_p}{\omega_c}$$

Hence,

$$\frac{\Delta C}{C_0} = \frac{2k_f m_p}{\omega_c} = \frac{2\Delta f}{f_c}$$

In practice, $\Delta f / f_c$ is usually small, and, hence, ΔC is a small fraction of C_0, which helps limit the harmonic distortion that arises because of the approximation used in this derivation.

We may also generate direct FM by using a saturable core reactor, where the inductance of a coil is varied by a current through a second coil (also wound around the same core). This results in a variable inductor whose inductance is proportional to the current in the second coil.

Direct FM generation generally produces sufficient frequency deviation and requires little frequency multiplication. But this method has poor frequency stability. In practice, feedback is used to stabilize the frequency. The output frequency is compared with a constant frequency generated by a stable crystal oscillator. An error signal (error in frequency) is detected and fed back to the oscillator to correct the error.

Features of Angle Modulation

FM (like angle modulation in general) has a number of unique features that recommend it for various radio systems. The transmission bandwidth of AM systems cannot be changed. Because of this, AM systems do not have the feature of exchanging signal power for transmission bandwidth. Pulse-coded modulation (PCM) systems (Chapter 6) have such a feature, and so do angle-modulated systems. In angle modulation, the transmission bandwidth can be adjusted by adjusting Δf. For angle-modulated systems, the SNR is roughly proportional to the square of the transmission bandwidth B_T. In PCM, the SNR varies exponentially with B_T and is, therefore, superior to angle modulation.

Example 5.7 Design an Armstrong indirect FM modulator to generate an FM signal with carrier frequency 97.3 MHz and $\Delta f = 10.24$ kHz. A NBFM generator of $f_{c_1} = 20$ kHz and $\Delta f = 5$ Hz is available. Only frequency doublers can be used as multipliers. Additionally, a local oscillator (LO) with adjustable frequency between 400 and 500 kHz is readily available for frequency mixing.

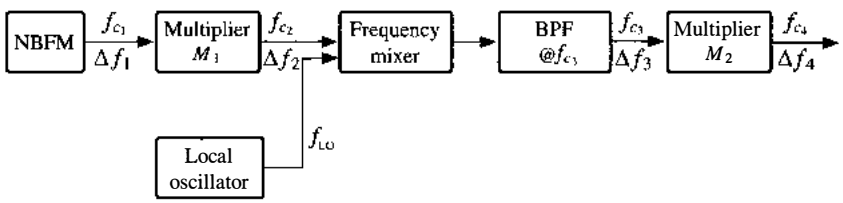


Figure 5.11 Designing an Armstrong indirect modulator.

The modulator is shown in Fig. 5.11. We need to determine M_1, M_2, and f_{LO}. First, the NBFM generator generates

$$f_{c_1} = 20,000 \qquad \text{and} \qquad \Delta f_1 = 5$$

The final WBFM should have

$$f_{c_4} = 97.3 \times 10^6 \qquad \Delta f_4 = 10,240$$

We first find the total factor of frequency multiplication needed as

$$M_1 \cdot M_2 = \frac{\Delta f_4}{\Delta f_1} = 2048 = 2^{11} \tag{5.25}$$

Because only frequency doublers can be used, we have three equations:

$$M_1 = 2^{n_1}$$
$$M_2 = 2^{n_2}$$
$$n_1 + n_2 = 11$$

It is also clear that

$$f_{c_2} = 2^{n_1} f_{c_1} \qquad \text{and} \qquad f_{c_4} = 2^{n_2} f_{c_3}$$

To find f_{LO}, there are three possible relationships:

$$f_{c_3} = f_{c_2} \pm f_{\text{LO}} \qquad \text{and} \qquad f_{c_3} = f_{\text{LO}} - f_{c_2}$$

Each should be tested to determine the one that will fall in

$$400,000 \le f_{\text{LO}} \le 500,000$$

(a) First, we test $f_{c_3} = f_{c_2} - f_{\text{LO}}$. This case leads to

$$\begin{aligned} 97.3 \times 10^6 &= 2^{n_2}\left(2^{n_1} f_{c_1} - f_{\text{LO}}\right) \\ &= 2^{n_1+n_2} f_{c_1} - 2^{n_2} f_{\text{LO}} \\ &= 2^{11} 20 \times 10^3 - 2^{n_2} f_{\text{LO}} \end{aligned}$$

Thus, we have

$$f_{\text{LO}} = 2^{-n_2}\left(4.096 \times 10^7 - 9.73 \times 10^7\right) < 0$$

This is outside the local oscillator frequency range.

(b) Next, we test $f_{c_3} = f_{c_2} + f_{\text{LO}}$. This case leads to

$$\begin{aligned} 97.3 \times 10^6 &= 2^{n_2}\left(2^{n_1} f_{c_1} + f_{\text{LO}}\right) \\ &= 2^{11} 20 \times 10^3 + 2^{n_2} f_{\text{LO}} \end{aligned}$$

Thus, we have

$$f_{\text{LO}} = 2^{-n_2}\left(5.634 \times 10^7\right)$$

If $n_2 = 7$, then $f_{\text{LO}} = 440$ kHz, which is within the realizable range of the local oscillator.

(c) If we choose $f_{c_3} = f_{\text{LO}} - f_{c_2}$, then we have

$$\begin{aligned} 97.3 \times 10^6 &= f_{\text{LO}} - 2^{n_2} 2^{n_1} f_{c_1} \\ &= 2^{n_2} f_{\text{LO}} - 2^{11}(20 \times 10^3) \end{aligned}$$

Thus, we have

$$f_{LO} = 2^{-n_2}\left(13.826 \times 10^7\right)$$

No integer n_2 will lead to a realizable f_{LO}.

Thus, the final design is $M_1 = 16$, $M_2 = 128$, and $f_{LO} = 440$ kHz.

5.4 DEMODULATION OF FM SIGNALS

The information in an FM signal resides in the instantaneous frequency $\omega_i = \omega_c + k_f m(t)$. Hence, a frequency-selective network with a transfer function of the form $|H(f)| = 2a\pi f + b$ over the FM band would yield an output proportional to the instantaneous frequency (Fig. 5.12a).* There are several possible circuits with such characteristics. The simplest among them is an ideal differentiator with the transfer function $j2\pi f$.

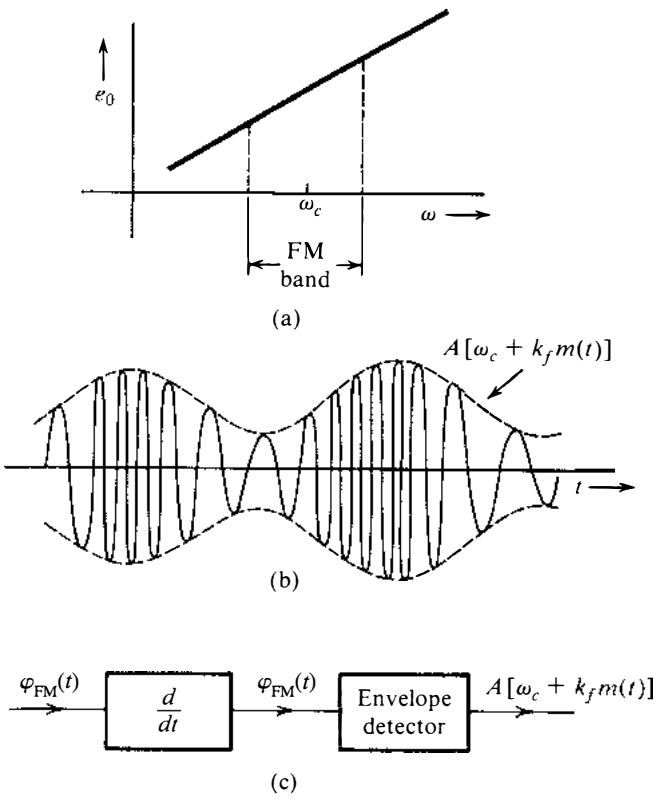


Figure 5.12 (a) FM demodulator frequency response. (b) Output of a differentiator to the input FM wave. (c) FM demodulation by direct differentiation.

* Provided the variations of ω_i are slow in comparison to the time constant of the network.

If we apply $\varphi_{FM}(t)$ to an ideal differentiator, the output is

$$\dot{\varphi}_{FM}(t) = \frac{d}{dt}\left\{A\cos\left[\omega_c t + k_f \int_{-\infty}^{t} m(\alpha)\, d\alpha\right]\right\}$$
$$= A\left[\omega_c + k_f m(t)\right] \sin\left[\omega_c t + k_f \int_{-\infty}^{t} m(\alpha)\, d(\alpha) - \pi\right] \qquad (5.26)$$

Both the amplitude and the frequency of the signal $\dot{\varphi}_{FM}(t)$ are modulated (Fig. 5.12b), the envelope being $A[\omega_c + k_f m(t)]$. Because $\Delta\omega = k_f m_p < \omega_c$, we have $\omega_c + k_f m(t) > 0$ for all t, and $m(t)$ can be obtained by envelope detection of $\dot{\varphi}_{FM}(t)$ (Fig. 5.12c).

The amplitude A of the incoming FM carrier must be constant. If the amplitude A were not constant, but a function of time, there would be an additional term containing dA/dt on the right-hand side of Eq. (5.26). Even if this term were neglected, the envelope of $\dot{\varphi}_{FM}(t)$ would be $A(t)[\omega_c + k_f m(t)]$, and the envelope-detector output would be proportional to $m(t)A(t)$, still leading to distortions. Hence, it is essential to maintain A constant. Several factors, such as channel noise and fading, cause A to vary. This variation in A should be suppressed via the bandpass limiter (discussed earlier in Sec. 5.3) before the signal is applied to the FM detector.

Practical Frequency Demodulators

The differentiator is only one way to convert frequency variation of FM signals into amplitude variation that subsequently can be detected by means of envelope detectors. One can use an operational amplifier differentiator at the FM receiver. On the other hand, the role of the differentiator can be replaced by any linear system whose frequency response contains a linear segment of positive slope. By approximating the ideal linear slope in Fig. 5.12a, this method is known as **slope detection.**

One simple device would be an RC high-pass filter of Fig. 5.13. The RC frequency response is simply

$$H(f) = \frac{j2\pi fRC}{1 + j2\pi fRC} \approx j2\pi fRC \qquad \text{if} \quad 2\pi fRC \ll 1$$

Thus, if the parameter RC is be very small such that its product with the carrier frequency $\omega_c RC \ll 1$, the RC filter approximates a differentiator.

Similarly, a simple tuned RLC circuit followed by an envelope detector can also serve as a frequency detector because its frequency response $|H(f)|$ below the resonance frequency $\omega_o = 1/\sqrt{LC}$ approximates a linear slope. Thus, such a receiver design requires that

$$\omega_c < \omega_o = \frac{1}{\sqrt{LC}}$$

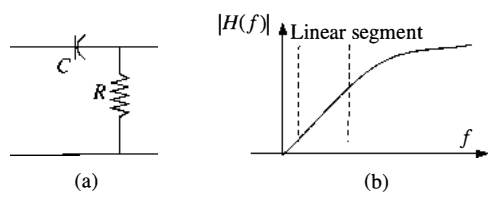


Figure 5.13
(a) RC high-pass filter.
(b) Segment of positive slope in amplitude response.

Because the operation is on the slope of $|H(f)|$, this method is also called **slope detection**. Since, however, the slope of $|H(f)|$ is linear over only a small band, there is considerable distortion in the output. This fault can be partially corrected by a **balanced discriminator** formed by two slope detectors. Another balanced demodulator, the **ratio detector**, also widely used in the past, offers better protection against carrier amplitude variations than does the discriminator. For many years ratio detectors were standard in almost all FM receivers.[10]

Zero-crossing detectors are also used because of advances in digital integrated circuits. The first step is to use the amplitude limiter of Fig. 5.9a to generate the rectangular pulse output of Fig. 5.9c. The resulting rectangular pulse train of varying width can then be applied to trigger a digital counter. These are the **frequency counters** designed to measure the instantaneous frequency from the number of zero crossings. The rate of zero crossings is equal to the instantaneous frequency of the input signal.

FM Demodulation via PLL

Consider a PLL that is in lock with input signal $\sin[\omega_c t + \theta_i(t)]$ and output error signal $e_o(t)$. When the input signal is an FM signal,

$$\theta_i(t) = k_f \int_{-\infty}^{t} m(\alpha)\, d\alpha + \frac{\pi}{2} \tag{5.27}$$

then,

$$\theta_o(t) = k_f \int_{-\infty}^{t} m(\alpha)\, d\alpha + 0.5\pi - \theta_e(t)$$

With PLL in lock we can assume a small frequency error $\dot{\theta}_e(t) \approx 0$. Thus, the loop filter output signal is

$$e_o(t) = \frac{1}{c}\dot{\theta}_o(t) = \frac{1}{c}\frac{d}{dt}\left[k_f \int_{-\infty}^{t} m(\alpha)\, d\alpha + 0.5\pi - \theta_e(t)\right] \simeq \frac{k_f}{c} m(t) \tag{5.28}$$

Thus, the PLL acts as an FM demodulator. If the incoming signal is a PM wave, then $e_o(t) = k_p \dot{m}(t)/c$. In this case we need to integrate $e_o(t)$ to obtain the desired signal $m(t)$.

To more precisely analyze PLL behavior as an FM demodulator, we consider the case of a small error (linear model of the PLL) with $H(s) = 1$. For this case, feedback analysis of the small-error PLL in Chapter 4 becomes

$$\Theta_o(s) = \frac{AK\,H(s)}{s + AK\,H(s)}\Theta_i(s) = \frac{AK}{s + AK}\Theta_i(s)$$

If $E_o(s)$ and $M(s)$ are Laplace transforms of $e_o(t)$ and $m(t)$, respectively, then from Eqs. (5.27) and (5.28) we have

$$\Theta_i(s) = \frac{k_f M(s)}{s} \qquad \text{and} \qquad s\Theta_o(s) = cE_o(s)$$

Hence,

$$E_o(s) = \left(\frac{k_f}{c}\right)\frac{AK}{s+AK}M(s)$$

Thus, the PLL output $e_o(t)$ is a distorted version of $m(t)$ and is equivalent to the output of a single-pole circuit (such as a simple RC circuit) with transfer function $k_f AK/c(s + AK)$ to which $m(t)$ as the input. To reduce distortion, we must choose AK well above the bandwidth of $m(t)$, so that $e_o(t) \simeq k_f m(t)/c$.

In the presence of small noise, the behavior of the PLL is comparable to that of a frequency discriminator. The advantage of the PLL over a frequency discriminator appears only when the noise is large.

5.5 EFFECTS OF NONLINEAR DISTORTION AND INTERFERENCE

Immunity of Angle Modulation to Nonlinearities

A very useful feature of angle modulation is its constant amplitude, which makes it less susceptible to nonlinearities. Consider, for instance, an amplifier with second-order nonlinear distortion whose input $x(t)$ and output $y(t)$ are related by

$$y(t) = a_0 + a_1 x(t) + a_2 x^2(t) + \cdots + a_n x^n(t)$$

Clearly, the first term is the desired signal amplification term, while the remaining terms are the unwanted nonlinear distortion. For the angle modulated signal

$$x(t) = A\cos[\omega_c t + \psi(t)]$$

trigonometric identities can be applied to rewrite the nonideal system output $y(t)$ as

$$\begin{aligned} y(t) = c_o + c_1 \cos[\omega_c t + \psi(t)] + c_2 \cos[2\omega_c t + 2\psi(t)] \\ + \cdots + c_n \cos[n\omega_c t + n\psi(t)] \end{aligned}$$

Because sufficiently large ω_c makes each component of $y(t)$ separable in frequency domain, a bandpass filter centered at ω_c with bandwidth equaling to B_{FM} (or B_{PM}) can extract the desired FM signal component $c_1 \cos[\omega_c t + \psi(t)]$ without any distortion. This shows that angle-modulated signals are immune to nonlinear distortions.

A similar nonlinearity in AM not only causes unwanted modulation with carrier frequencies $n\omega_c$ but also causes distortion of the desired signal. For instance, if a DSB-SC signal $m(t)\cos\omega_c t$ passes through a nonlinearity $y(t) = a\,x(t) + b\,x^3(t)$, the output is

$$\begin{aligned} y(t) &= a\,m(t)\cos\omega_c t + b\,m^3(t)\cos^3\omega_c t \\ &= \left[a\,m(t) + \frac{3b}{4}m^3(t)\right]\cos\omega_c t + \frac{b}{4}m^3(t)\cos 3\omega_c t \end{aligned}$$

Passing this signal through a bandpass filter still yields $[a\,m(t)+(3b/4)m^3(t)]\cos\omega_c t$. Observe the distortion component $(3b/4)m^3(t)$ present along with the desired signal $a\,m(t)$.

Immunity from nonlinearity is the primary reason for the use of angle modulation in microwave radio relay systems, where power levels are high. This requires highly efficient nonlinear class C amplifiers. In addition, the constant amplitude of FM gives it a kind of immunity to rapid fading. The effect of amplitude variations caused by rapid fading can be eliminated by using automatic gain control and bandpass limiting. These advantages made FM attractive as the technology behind the first-generation (1G) cellular phone system.

The same advantages of FM also make it attractive for microwave radio relay systems. In the legacy analog long-haul telephone systems, several channels are multiplexed by means of SSB signals to form L-carrier signals. The multiplexed signals are frequency-modulated and transmitted over a microwave radio relay system with many links in tandem. In this application, however, FM is used not to reduce noise effects but to realize other advantages of constant amplitude, and, hence, NBFM rather than WBFM is used.

Interference Effect

Angle modulation is also less vulnerable than AM to small-signal interference from adjacent channels.

Let us consider the simple case of the interference of an unmodulated carrier $A \cos \omega_c t$ with another sinusoid $I \cos (\omega_c + \omega)t$. The received signal $r(t)$ is

$$\begin{aligned} r(t) &= A \cos \omega_c t + I \cos (\omega_c + \omega)t \\ &= (A + I \cos \omega t) \cos \omega_c t - I \sin \omega t \sin \omega_c t \\ &= E_r(t) \cos [\omega_c t + \psi_d(t)] \end{aligned}$$

where

$$\psi_d(t) = \tan^{-1} \frac{I \sin \omega t}{A + I \cos \omega t}$$

When the interfering signal is small in comparison to the carrier ($I \ll A$),

$$\psi_d(t) \simeq \frac{I}{A} \sin \omega t \tag{5.29}$$

The phase of $E_r(t) \cos [\omega_c t + \psi_d(t)]$ is $\psi_d(t)$, and its instantaneous frequency is $\omega_c + \dot{\psi}_d(t)$. If the signal $E_r(t) \cos [\omega_c t + \psi_d(t)]$ is applied to an ideal phase demodulator, the output $y_d(t)$ would be $\psi_d(t)$. Similarly, the output $y_d(t)$ of an ideal frequency demodulator would be $\dot{\psi}_d(t)$. Hence,

$$y_d(t) = \frac{I}{A} \sin \omega t \qquad \text{for PM} \tag{5.30}$$

$$y_d(t) = \frac{I\omega}{A} \cos \omega t \qquad \text{for FM} \tag{5.31}$$

Observe that in either case, the interference output is inversely proportional to the carrier amplitude A. Thus, the larger the carrier amplitude A, the smaller the interference effect. This behavior is very different from that in AM signals, where the interference output is independent

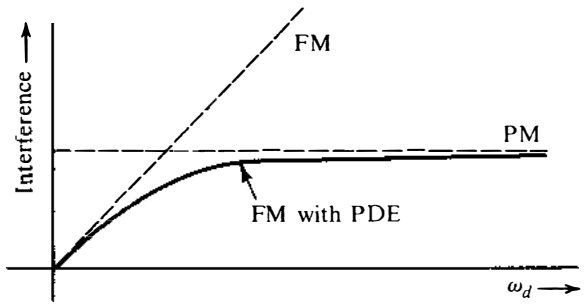


Figure 5.14 Effect of interference in PM, FM, and FM with preemphasis-deemphasis (PDE).

of the carrier amplitude.* Hence, angle-modulated systems are much better than AM systems at suppressing weak interference ($I \ll A$).

Because of the suppression of weak interference in FM, we observe what is known as the **capture effect** when listening to FM radios. For two transmitters with carrier frequency separation less than the audio range, instead of getting interference, we observe that the stronger carrier effectively suppresses (captures) the weaker carrier. Subjective tests show that an interference level as low as 35 dB in the audio signals can cause objectionable effects. Hence, in AM, the interference level should be kept below 35 dB. On the other hand, for FM, because of the capture effect, the interference level need only be below 6 dB.

The interference amplitude (I/A for PM and $I\omega/A$ for FM) vs. ω at the receiver output is shown in Fig. 5.14. The interference amplitude is constant for all ω in PM but increases linearly with ω in FM.†

Interference due to Channel Noise

The channel noise acts as interference in an angle-modulated signal. We shall consider the most common form of noise, white noise, which has a constant power spectral density. Such a noise may be considered as a sum of sinusoids of all frequencies in the band. All components have the same amplitudes (because of uniform density). This means I is constant for all ω, and the amplitude spectrum of the interference at the receiver output is as shown in Fig. 5.14. The interference amplitude spectrum is constant for PM, and increases linearly with ω for FM.

Preemphasis and Deemphasis in FM Broadcasting

Figure 5.14 shows that in FM, the interference (the noise) increases linearly with frequency, and the noise power in the receiver output is concentrated at higher frequencies. A glance at Fig. 4.18b shows that the PSD of an audio signal $m(t)$ is concentrated at lower frequencies below 2.1 kHz. Thus, the noise PSD is concentrated at higher frequencies, where $m(t)$ is

---

* For instance, an AM signal with an interfering sinusoid $I \cos (\omega_c + \omega)t$ is given by

$$\begin{aligned} r(t) &= [A + m(t)] \cos \omega_c t + I \cos (\omega_c + \omega)t \\ &= [A + m(t) + I \cos \omega t] \cos \omega_c t - I \sin \omega t \sin \omega_c t \end{aligned}$$

The envelope of this signal is

$$E(t) = \{[A + m(t) + I \cos \omega t]^2 + I^2 \sin^2 \omega t\}^{1/2} \approx A + m(t) + I \cos \omega t \qquad I \ll A$$

Thus the interference signal at the envelope detector output is $I \cos \omega t$, which is independent of the carrier amplitude A. We obtain the same result when synchronous demodulation is used. We come to a similar conclusion for AM-SC systems.

† The results in Eqs. (5.30) and (5.31) can be readily extended to more than one interfering sinusoid. The system behaves linearly for multiple interfering sinusoids provided their amplitudes are very small in comparison to the carrier amplitude.

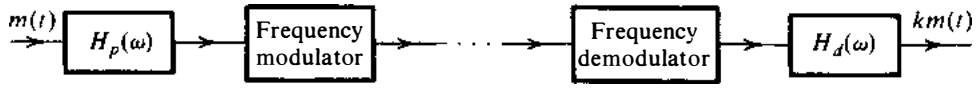


Figure 5.15 Preemphasis-deemphasis in an FM system.

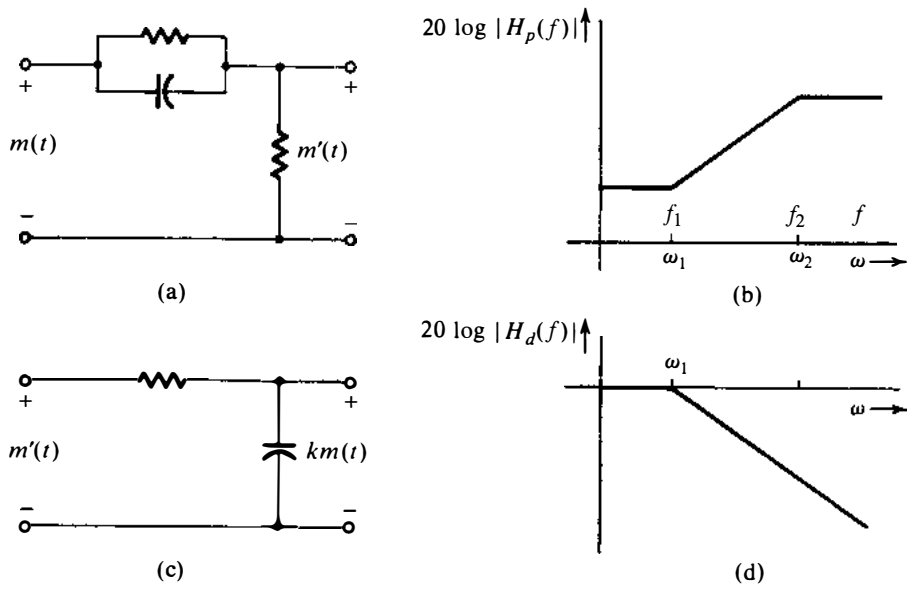


Figure 5.16 (a) Preemphasis filter and (b) its frequency response. (c) Deemphasis filter and (d) its frequency response.

the weakest. This may seem like a disaster. But actually, in this very situation there is a hidden opportunity to reduce noise greatly. The process, shown in Fig. 5.15, works as follows. At the transmitter, the weaker high-frequency components (beyond 2.1 kHz) of the audio signal $m(t)$ are boosted before modulation by a **preemphasis** filter of transfer function $H_p(f)$. At the receiver, the demodulator output is passed through a **deemphasis** filter of transfer function $H_d(f) = 1/H_p(f)$. Thus, the deemphasis filter undoes the preemphasis by attenuating (deemphasizing) the higher frequency components (beyond 2.1 kHz), and thereby restores the original signal $m(t)$. The noise, however, enters at the channel, and therefore has not been preemphasized (boosted). However, it passes through the deemphasis filter, which attenuates its higher frequency components, where most of the noise power is concentrated (Fig. 5.14). Thus, the process of preemphasis-deemphasis (PDE) leaves the desired signal untouched but reduces the noise power considerably.

Preemphasis and Deemphasis Filters

Figure 5.14 provides an opportunity to preemphasis. The FM has smaller interference than PM at lower frequencies, while the opposite is true at higher frequencies. If we can make our system behave like FM at lower frequencies and behave like PM at higher frequencies, we will have the best of both worlds. This is accomplished by a system used in commercial broadcasting (Fig. 5.15) with the preemphasis (before modulation) and deemphasis (after demodulation) filters $H_p(f)$ and $H_d(f)$ shown in Fig. 5.16. The frequency f_1 is 2.1 kHz, and f_2 is typically 30 kHz or more (well beyond audio range), so that f_2 does not even enter into the picture. These filters can be realized by simple *RC* circuits (Fig. 5.16). The choice of $f_1 = 2.1$ kHz was apparently made on an experimental basis. It was found that this choice of f_1 maintained

the same peak amplitude m_p with or without preemphasis.[11] This satisfied the constraint of a fixed transmission bandwidth.

The preemphasis transfer function is

$$H_p(f) = K\frac{j2\pi f + \omega_1}{j2\pi f + \omega_2} \tag{5.32a}$$

where K, the gain, is set at a value of ω_2/ω_1. Thus,

$$H_p(f) = \left(\frac{\omega_2}{\omega_1}\right)\frac{j2\pi f + \omega_1}{j2\pi f + \omega_2} \tag{5.32b}$$

For $2\pi f \ll \omega_1$,

$$H_p(f) \simeq 1 \tag{5.32c}$$

For frequencies $\omega_1 \ll 2\pi f \ll \omega_2$,

$$H_p(f) \simeq \frac{j2\pi f}{\omega_1} \tag{5.32d}$$

Thus, the preemphasizer acts as a differentiator at intermediate frequencies (2.1–15 kHz), which effectively makes the scheme PM over these frequencies. This means that FM with PDE is FM over the modulating-signal frequency range of 0 to 2.1 kHz and is nearly PM over the range of 2.1 to 15 kHz, as desired.

The deemphasis filter $H_d(f)$ is given by

$$H_d(f) = \frac{\omega_1}{j2\pi f + \omega_1}$$

Note that for $2\pi f \ll \omega_2$, $H_p(f) \simeq (j2\pi f + \omega_1)/\omega_1$. Hence, $H_p(f)H_d(f) \simeq 1$ over the baseband of 0 to 15 kHz.

For historical and practical reasons, optimum PDE filters are not used in practice. It can be shown that the PDE enhances the SNR by 13.27 dB (a power ratio of 21.25).

The side benefit of PDE is improvement in the interference characteristics. Because the interference (from unwanted signals and the neighboring stations) enters after the transmitter stage, it undergoes only the deemphasis operation, not the boosting, or preemphasis. Hence, the interference amplitudes for frequencies beyond 2.1 kHz undergo attenuation that is roughly linear with frequency.

The PDE method of noise reduction is not limited to FM broadcast. It is also used in audiotape recording and in (analog) phonograph recording, where the hissing noise is also concentrated at the high-frequency end. A sharp, hissing sound is caused by irregularities in the recording material. The **Dolby noise reduction** systems for audiotapes operates on the same principle, although the Dolby-A system is somewhat more elaborate. In the Dolby-B and Dolby-C systems, the band is divided into two subbands (below and above 3 kHz instead of

2.1 kHz). In the Dolby-A system, designed for commercial use, the bands are divided into four subbands (below 80 Hz, 80–3 kHz, 3–9 kHz, and above 9 kHz). The amount of preemphasis is optimized for each band.

We could also use PDE in AM broadcasting to improve the output SNR. In practice, however, this is not done for several reasons. First, the output noise amplitude in AM is constant with frequency and does not increase linearly as in FM. Hence, the deemphasis does not yield such a dramatic improvement in AM as it does in FM. Second, introduction of PDE would necessitate modifications of receivers already in use. Third, increasing high-frequency component amplitudes (preemphasis) would increase interference with adjacent stations (no such problem arises in FM). Moreover, an increase in the frequency deviation ratio β at high frequencies would make detector design more difficult.

5.6 SUPERHETERODYNE ANALOG AM/FM RECEIVERS

The radio receiver used in broadcast AM and FM systems, is called the **superheterodyne** receiver (Fig. 5.17). It consists of an RF (radio-frequency) section, a frequency converter (Example 4.2), an intermediate-frequency (IF) amplifier, an envelope detector, and an audio amplifier.

The RF section consists basically of a tunable filter and an amplifier that picks up the desired station by tuning the filter to the right frequency band. The next section, the frequency mixer (converter), translates the carrier from ω_c to a fixed IF frequency of ω_{IF} (see Example 4.2 for frequency conversion). For this purpose, the receiver uses a local oscillator whose frequency f_{LO} is exactly f_{IF} above the incoming carrier frequency f_c; that is,

$$f_{LO} = f_c + f_{IF}$$

The simultaneous tuning of the local oscillator and the RF tunable filter is done by one joint knob. Tuning capacitors in both circuits are ganged together and are designed so that the tuning

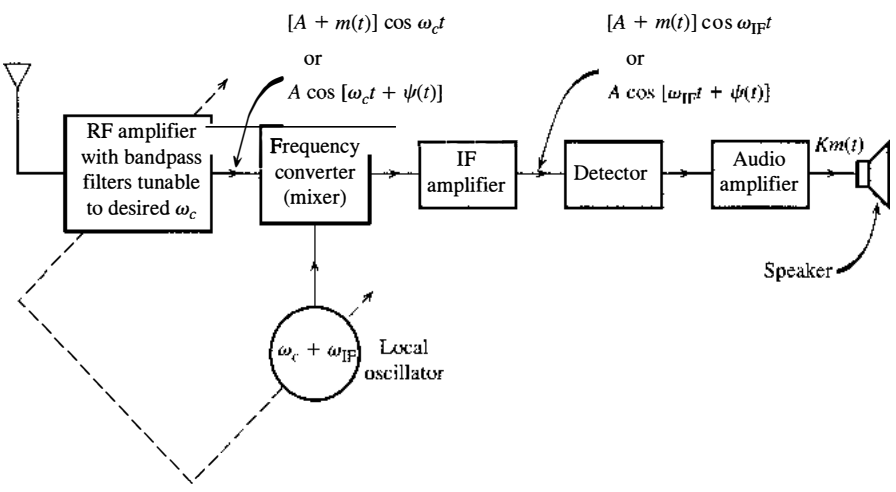


Figure 5.17 Superheterodyne receiver.

frequency of the local oscillator is always f_{IF} Hz above the tuning frequency f_c of the RF filter. This means every station that is tuned in is translated to a fixed carrier frequency of f_{IF} Hz by the frequency converter for subsequent processing at IF.

This superheterodyne receiver structure is broadly utilized in most broadcast systems. The intermediate frequencies are chosen to be 455 kHz (AM radio), 10.7 MHz (FM radio), and 38 MHz (TV reception).

As discovered by Armstrong for AM signals, the translation of all stations to a fixed intermediate frequency ($f_{IF} = 455$ kHz for AM) allows us to obtain adequate selectivity. It is difficult to design precise bandpass filters of bandwidth 10 kHz (the modulated audio spectrum) if the center frequency f_c is very high. This is particularly true in the case of tunable filters. Hence, the RF filter cannot provide adequate selectivity against adjacent channels. But when this signal is translated to an IF frequency by a converter, it is further amplified by an IF amplifier (usually a three-stage amplifier), which does have good selectivity. This is because the IF frequency is reasonably low; moreover, its center frequency is fixed and factory-tuned. Hence, the IF section can effectively suppress adjacent-channel interference because of its high selectivity. It also amplifies the signal for envelope detection.

In reality, the entire selectivity is practically realized in the IF section; the RF section plays a negligible role. The main function of the RF section is image frequency suppression. As observed in Example 4.2, the output of the mixer, or converter, consists of components of the difference between the incoming (f_c) and the local oscillator frequencies (f_{LO}) (i.e., $f_{IF} = |f_{LO} - f_c|$). Now, consider the AM example. If the incoming carrier frequency $f_c = 1000$ kHz, then $f_{LO} = f_c + f_{RF} = 1000 + 455 = 1455$ kHz. But another carrier, with $f_c' = 1455 + 455 = 1910$ kHz, will also be picked up because the difference $f_c' - f_{LO}$ is also 455 kHz. The station at 1910 kHz is said to be the **image** of the station of 1000 kHz. AM stations that are $2f_{IF} = 910$ kHz apart are called **image stations** and both would appear simultaneously at the IF output, were it not for the RF filter at receiver input. The RF filter may provide poor selectivity against adjacent stations separated by 10 kHz, but it can provide reasonable selectivity against a station separated by 910 kHz. Thus, when we wish to tune in a station at 1000 kHz, the RF filter, tuned to 1000 kHz, provides adequate suppression of the image station at 1910 kHz.

The receiver (Fig. 5.17) converts the incoming carrier frequency to the IF by using a local oscillator of frequency f_{LO} higher than the incoming carrier frequency and, hence, is called a superheterodyne receiver. We pick f_{LO} higher than f_c because this leads to a smaller tuning ratio of the maximum to minimum tuning frequency for the local oscillator. The AM broadcast-band frequencies range from 530 to 1710 kHz. The superheterodyne f_{LO} ranges from 1005 to 2055 kHz (ratio of 2.045), whereas the subheterodyne range of f_{LO} would be 95 to 1145 kHz (ratio of 12.05). It is much easier to design an oscillator that is tunable over a smaller frequency ratio.

The importance of the superheterodyne principle in radio and television broadcasting cannot be overstressed. In the early days (before 1919), the entire selectivity against adjacent stations was realized in the RF filter. Because this filter often had poor selectivity, it was necessary to use several stages (several resonant circuits) in cascade for adequate selectivity. In the earlier receivers each filter was tuned individually. It was very time-consuming and cumbersome to tune in a station by bringing all resonant circuits into synchronism. This task was made easier as variable capacitors were ganged together by mounting them on the same shaft rotated by one knob. But variable capacitors are bulky, and there is a limit to the number that can be ganged together. These factors, in turn, limited the selectivity available from receivers. Consequently, adjacent carrier frequencies had to be separated widely, resulting in fewer frequency bands. It was the superheterodyne receiver that made it possible to accommodate many more radio stations.

5.7 FM BROADCASTING SYSTEM

The FCC has assigned a frequency range of 88 to 108 MHz for FM broadcasting, with a separation of 200 kHz between adjacent stations and a peak frequency deviation $\Delta f = 75$ kHz.

A monophonic FM receiver is identical to the superheterodyne AM receiver in Fig. 5.17, except that the intermediate frequency is 10.7 MHz and the envelope detector is replaced by a PLL or a frequency discriminator followed by a deemphasizer.

Earlier FM broadcasts were monophonic. Stereophonic FM broadcasting, in which two audio signals, L (left microphone) and R (right microphone), are used for a more natural effect, was proposed later. The FCC ruled that the stereophonic system had to be compatible with the original monophonic system. This meant that the older monophonic receivers should be able to receive the signal $L + R$, and the total transmission bandwidth for the two signals (L and R) should still be 200 kHz, with $\Delta f = 75$ kHz for the two combined signals. This would ensure that the older receivers could continue to receive monophonic as well as stereophonic broadcasts, although the stereo effect would be absent.

A transmitter and a receiver for a stereo broadcast are shown in Fig. 5.18a and c. At the transmitter, the two signals L and R are added and subtracted to obtain $L + R$ and $L - R$. These signals are preemphasized. The preemphasized signal $(L - R)'$ DSB-SC modulates a carrier of 38 kHz obtained by doubling the frequency of a 19-kHz signal that is used as a pilot. The signal $(L + R)'$ is used directly. All three signals (the third being the pilot) form a composite baseband signal $m(t)$ (Fig. 5.18b),

$$m(t) = (L + R)' + (L - R)' \cos \omega_c t + \alpha \cos \frac{\omega_c t}{2} \tag{5.33}$$

The reason for using a pilot of 19 kHz rather than 38 kHz is that it is easier to separate the pilot at 19 kHz because there are no signal components within 4 kHz of that frequency.

The receiver operation (Fig. 5.18c) is self-explanatory. A monophonic receiver consists of only the upper branch of the stereo receiver and, hence, receives only $L + R$. This is of course the complete audio signal without the stereo effect. Hence, the system is compatible. The pilot is extracted, and (after doubling its frequency) it is used to demodulate coherently the signal $(L - R)' \cos \omega_c t$.

An interesting aspect of stereo transmission is that the peak amplitude of the composite signal $m(t)$ in Eq. (5.33) is practically the same as that of the monophonic signal (if we ignore the pilot), and, hence, Δf—which is proportional to the peak signal amplitude for stereophonic transmission—remains practically the same as for the monophonic case. This can be explained by the so-called **interleaving** effect as follows.

The L' and R' signals are very similar in general. Hence, we can assume their peak amplitudes to be equal to A_p. Under the worst possible conditions, L' and R' will reach their peaks at the same time, yielding [Eq. (5.33)]

$$|m(t)|_{\max} = 2A_p + \alpha$$

In the monophonic case, the peak amplitude of the baseband signal $(L+R)'$ is $2A_p$. Hence, the peak amplitudes in the two cases differ only by α, the pilot amplitude. To account for this, the peak sound amplitude in the stereo case is reduced to 90% of its full value. This amounts to a reduction in the signal power by a ratio of $(0.9)^2 = 0.81$, or 1 dB. Thus, the effective SNR is reduced by 1 dB because of the inclusion of the pilot.

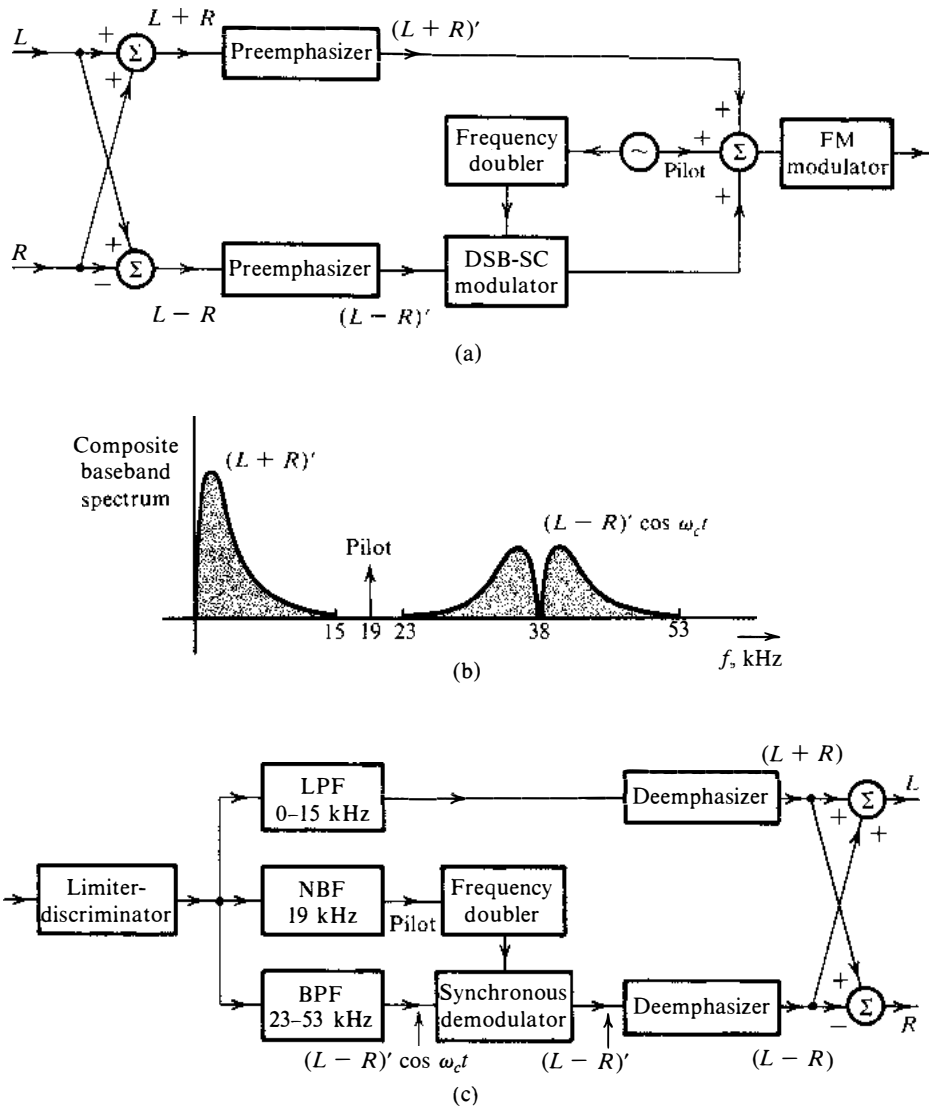


Figure 5.18 (a) FM stereo transmitter. (b) Spectrum of a baseband stereo signal. (c) FM stereo receiver.

5.8 MATLAB EXERCISES

In this section, we use MATLAB to build an FM modulation and demodulation example. The MATLAB program is given by `ExampleFM.m`. Once again use apply the same message signal $m_2(t)$. The FM coefficient is $k_f = 80$ and the PM coefficient is $k_p = \pi$. The carrier frequency remains 300 Hz. The resulting FM and PM signals in the time domain are shown in Fig. 5.19. The corresponding frequency responses are also shown in Fig. 5.19. The frequency domain responses clearly show the much higher bandwidths of the FM and PM signals when compared with amplitude modulations.

```
% (ExampleFM.m)
% This program uses triangl.m to illustrate frequency modulation
% and demodulation
```

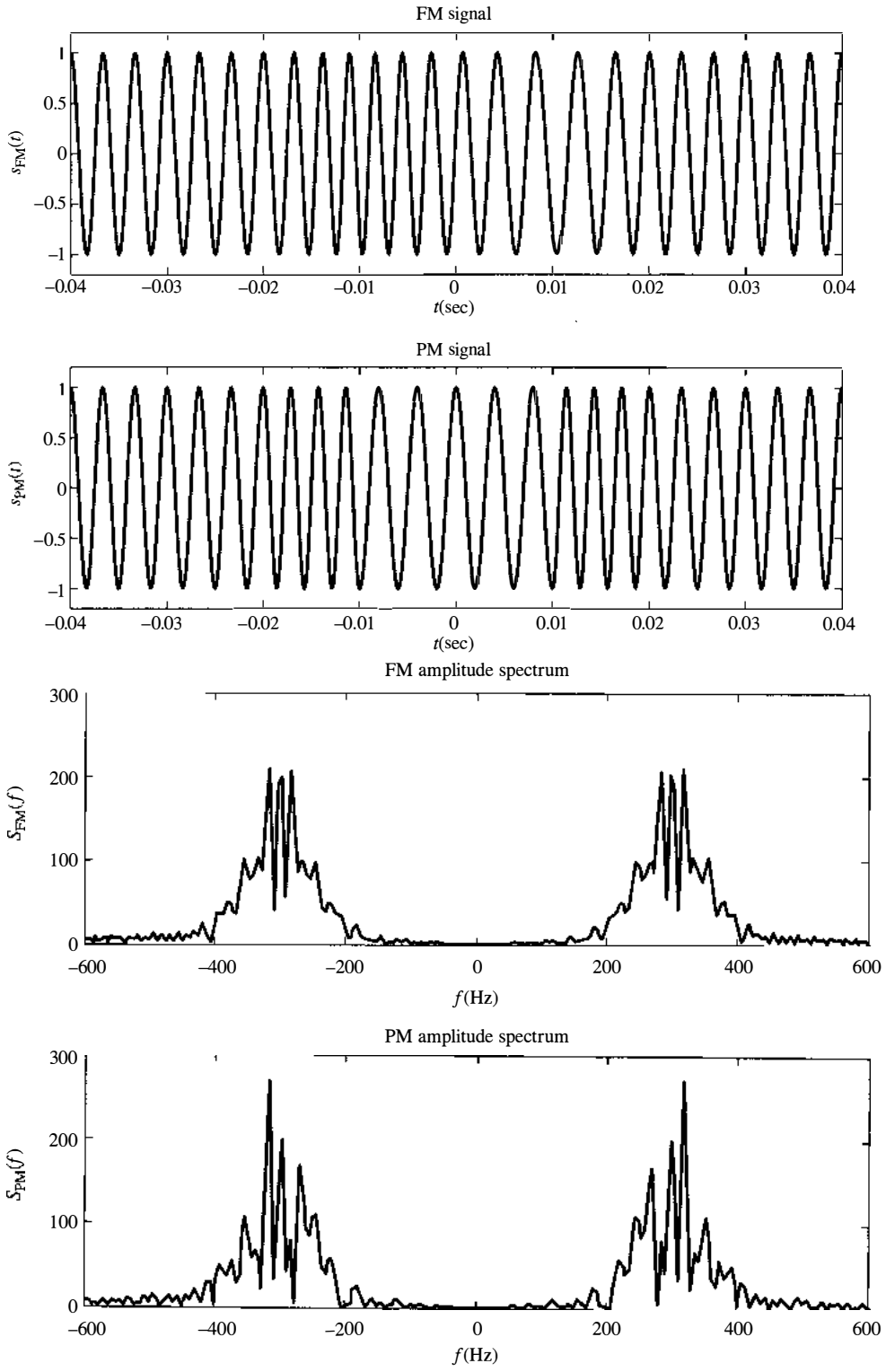


Figure 5.19 FM and PM signals in the time and frequency domains.

```
ts=1.e-4;

t=-0.04:ts:0.04;
Ta=0.01;
m_sig=triangl((t+0.01)/Ta)-triangl((t-0.01)/Ta);
Lfft=length(t);  Lfft=2^ceil(log2(Lfft));
M_fre=fftshift(fft(m_sig,Lfft));
freqm=(-Lfft/2:Lfft/2-1)/(Lfft*ts);
B_m=100;    %Bandwidth of the signal is B_m Hz.
% Design a simple lowpass filter with bandwidth B_m Hz.
h=fir1(80,[B_m*ts]);
%
kf=160*pi;
m_intg=kf*ts*cumsum(m_sig);
s_fm=cos(2*pi*300*t+m_intg);
s_pm=cos(2*pi*300*t+pi*m_sig);
Lfft=length(t);  Lfft=2^ceil(log2(Lfft)+1);
S_fm=fftshift(fft(s_fm,Lfft));
S_pm=fftshift(fft(s_pm,Lfft));
freqs=(-Lfft/2:Lfft/2-1)/(Lfft*ts);

s_fmdem=diff([s_fm(1) s_fm])/ts/kf;
s_fmrec=s_fmdem.*(s_fmdem>0);
s_dec=filter(h,1,s_fmrec);


%   Demodulation
%   Using an ideal LPF with bandwidth 200 Hz

Trange1=[-0.04 0.04 -1.2 1.2];

figure(1)
subplot(211);m1=plot(t,m_sig);
axis(Trange1); set(m1,'Linewidth',2);
xlabel('{\it t} (sec)'); ylabel('{\it m}({\it t})');
title('Message signal');
subplot(212);m2=plot(t,s_dec);
set(m2,'Linewidth',2);
xlabel('{\it t} (sec)'); ylabel('{\it m}_d({\it t})')
title('demodulated FM signal');

figure(2)
subplot(211);td1=plot(t,s_fm);
axis(Trange1); set(td1,'Linewidth',2);
xlabel('{\it t} (sec)'); ylabel('{\it s}_{\rm FM}({\it t})')
title('FM signal');
subplot(212);td2=plot(t,s_pm);
axis(Trange1); set(td2,'Linewidth',2);
xlabel('{\it t} (sec)'); ylabel('{\it s}_{\rm PM}({\it t})')
```

```
title('PM signal');

figure(3)
subplot(211);fp1=plot(t,s_fmdem);
set(fp1,'Linewidth',2);
xlabel('{\it t} (sec)'); ylabel('{\it d s}_{\rm FM}({\it t})/dt')
title('FM derivative');
subplot(212);fp2=plot(t,s_fmrec);
set(fp2,'Linewidth',2);
xlabel('{\it t} (sec)');
title('rectified FM derivative');

Frange=[-600 600 0 300];
figure(4)
subplot(211);fd1=plot(freqs,abs(S_fm));
axis(Frange); set(fd1,'Linewidth',2);
xlabel('{\it f} (Hz)'); ylabel('{\it S}_{\rm FM}({\it f})')
title('FM amplitude spectrum');
subplot(212);fd2=plot(freqs,abs(S_pm));
axis(Frange); set(fd2,'Linewidth',2);
xlabel('{\it f} (Hz)'); ylabel('{\it S}_{\rm PM}({\it f})')
title('PM amplitude spectrum');
```

To obtain the demodulation results (Fig. 5.20), a differentiator is first applied to change the frequency-modulated signal into an amplitude- and frequency-modulated signal (Fig. 5.20).

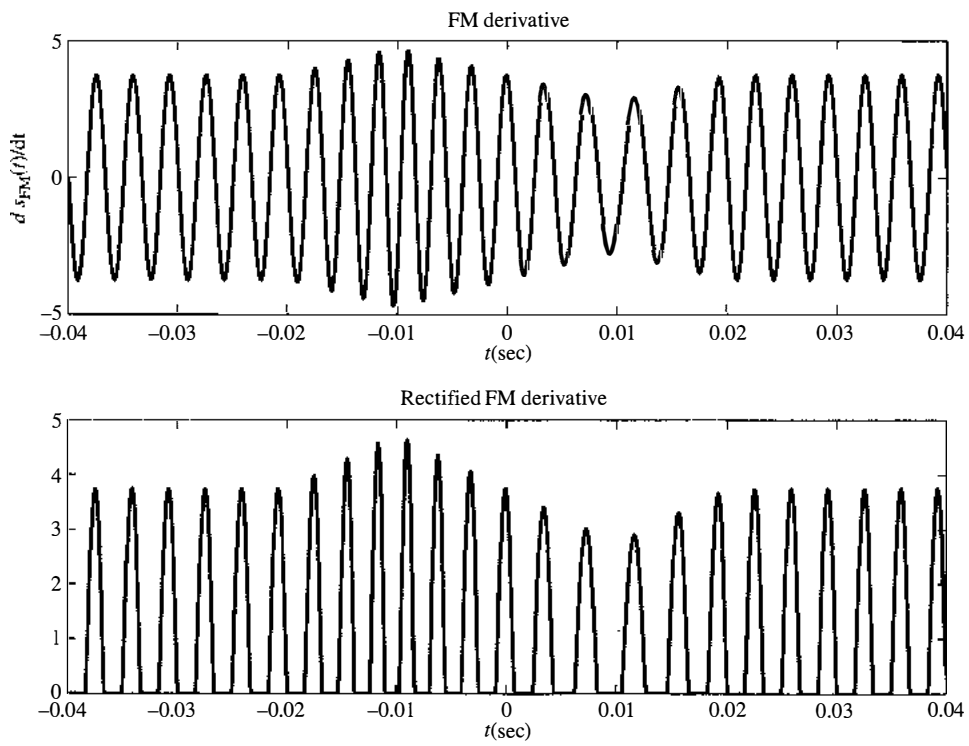


Figure 5.20 Signals at the demodulator: (a) after differentiator; (b) after rectifier.

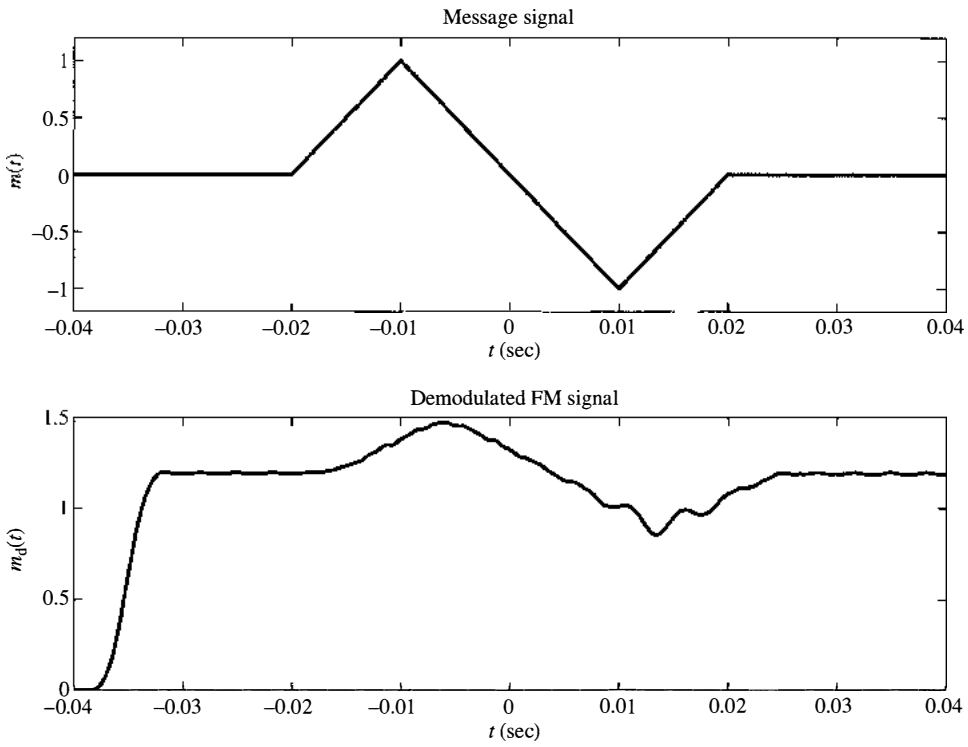


Figure 5.21 FM modulation and demodulation: (a) original message; (b) recovered signal.

Upon applying the rectifier for envelope detection, we see that the message signal follows closely to the envelope variation of the rectifier output.

Finally, the rectifier output signal is passed through a low-pass filter with bandwidth 100 Hz. We used the finite impulse response low-pass filter of order 80 this time because of the tighter filter constraint in this example. The FM detector output is then compared with the original message signal in Fig. 5.21.

The FM demodulation results clearly show some noticeable distortions. First, the higher order low-pass filter has a much longer response time and delay. Second, the distortion during the negative half of the message is more severe because the rectifier generates very few cycles of the half-sinusoid. This happens because when the message signal is negative, the instantaneous frequency of the FM signal is low. Because we used a carrier frequency of only 300 Hz, the effect of low instantaneous frequency is much more pronounced. If a practical carrier frequency of 100 MHz were applied, this kind of distortion would be completely negligible.

REFERENCES

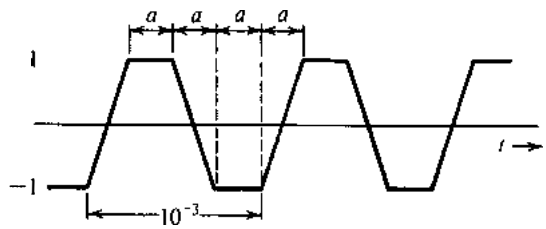
1. J. Carson, "Notes on the Theory of Modulation," *Proc. IRE*, vol. 10, pp. 57–64, Feb. 1922.
2. J. Carson, "Reduction of Atmospheric Disturbances," *Proc. IRE*, vol. 16, July 1928.
3. E. H. Armstrong, "A Method of Reducing Disturbances in Radio Signaling by a System of Frequency Modulation," *Proc. IRE*, vol. 24, pp. 689–740, May 1936.
4. "A Revolution in Radio," *Fortune*, vol. 20, p. 116, Oct. 1939.
5. L. Lessing, *Man of High Fidelity: Edwin Howard Armstrong*, Lippincott, Philadelphia, 1956.

6. H. R. Slotten, "'Rainbow in the Sky': FM Radio Technical Superiority, and Regulatory Decision Making," Society for the History of Technology, 1996.
7. J. E. Brittain, "Electrical Engineering Hall of Fame—Edwin H. Armstrong," *Proc. IEEE*, vol 92, pp. 575–578, Mar. 2004.
8. W. B. Davenport, Jr., "Signal-to-Noise Ratios in Bandpass Limiters," *J. Appl. Phys.*, vol. 24, pp. 720–727, June 1953.
9. D. H. Sheingold, ed., *Nonlinear Circuits Handbook*, Analog Devices, Inc., Norwood, MA, 1974.
10. H. L. Krauss, C. W. Bostian, and F. H. Raab, *Solid-State Radio Engineering*, Wiley, New York, 1980.
11. 12. L. B. Arguimbau and R. B. Adler, *Vacuum Tube Circuits and Transistors*, Wiley, New York, 1964, p. 466.

PROBLEMS

5.1-1 Sketch $\varphi_{FM}(t)$ and $\varphi_{PM}(t)$ for the modulating signal $m(t)$ shown in Fig. P5.1-1, given $\omega_c = 10^8$, $k_f = 10^5$, and $k_p = 25$.

Figure P.5.1-1

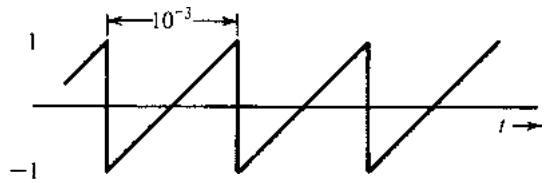


5.1-2 A baseband signal $m(t)$ is the periodic sawtooth signal shown in Fig. P5.1-2.

(a) Sketch $\varphi_{FM}(t)$ and $\varphi_{PM}(t)$ for this signal $m(t)$ if $\omega_c = 2\pi \times 10^6$, $k_f = 2000\pi$, and $k_p = \pi/2$.

(b) Show that the PM signal is equivalent to a PM signal modulated by a rectangular periodic message. Explain why it is necessary to use $k_p < \pi$ in this case. [Note that the PM signal has a constant frequency but has phase discontinuities corresponding to the discontinuities of $m(t)$.]

Figure P.5.1-2



5.1-3 Over an interval $|t| \leq 1$, an angle-modulated signal is given by

$$\varphi_{EM}(t) = 10 \cos 13,000\pi t$$

It is known that the carrier frequency $\omega_c = 10,000\pi$.

(a) If this were a PM signal with $k_p = 1000$, determine $m(t)$ over the interval $|t| \leq 1$.

(b) If this were an FM signal with $k_f = 1000$, determine $m(t)$ over the interval $|t| \leq 1$.

5.2-1 For a message signal

$$m(t) = 2\cos 100t + 18\cos 2000\pi t$$

(a) Write expressions (do not sketch) for $\varphi_{PM}(t)$ and $\varphi_{FM}(t)$ when $A = 10$, $\omega_c = 10^6$, $k_f = 1000\pi$, and $k_p = 1$. For determining $\varphi_{FM}(t)$, use the indefinite integral of $m(t)$; that is, take the value of the integral at $t = -\infty$ to be 0.

(b) Estimate the bandwidths of $\varphi_{FM}(t)$ and $\varphi_{PM}(t)$.

5.2-2 An angle-modulated signal with carrier frequency $\omega_c = 2\pi \times 10^6$ is described by the equation

$$\varphi_{EM}(t) = 10\cos(\omega_c t + 0.1\sin 2000\pi t)$$

(a) Find the power of the modulated signal.

(b) Find the frequency deviation Δf.

(c) Find the phase deviation $\Delta\phi$.

(d) Estimate the bandwidth of $\varphi_{EM}(t)$.

5.2-3 Repeat Prob. 5.2-2 if

$$\varphi_{EM}(t) = 5\cos(\omega_c t + 20\sin 1000\pi t + 10\sin 2000\pi t)$$

5.2-4 Estimate the bandwidth for $\varphi_{PM}(t)$ and $\varphi_{FM}(t)$ in Prob. 5.1-1. Assume the bandwidth of $m(t)$ in Fig. P5.1-1 to be the third-harmonic frequency of $m(t)$.

5.2-5 Estimate the bandwidth for $\varphi_{PM}(t)$ and $\varphi_{FM}(t)$ in Prob. 5.1-2. Assume the bandwidth of $m(t)$ in Fig. P5.1-1 to be the fifth-harmonic frequency of $m(t)$.

5.2-6 Given $m(t) = \sin 2000\pi t$, $k_f = 200{,}000\pi$, and $k_p = 10$.

(a) Estimate the bandwidths of $\varphi_{FM}(t)$ and $\varphi_{PM}(t)$.

(b) Repeat part **(a)** if the message signal amplitude is doubled.

(c) Repeat part **(a)** if the message signal frequency is doubled.

(d) Comment on the sensitivity of FM and PM bandwidths to the spectrum of $m(t)$.

5.2-7 Given $m(t) = e^{-t^2}$, $f_c = 10^4$ Hz, $k_f = 6000\pi$, and $k_p = 8000\pi$:

(a) Find Δf, the frequency deviation for FM and PM.

(b) Estimate the bandwidths of the FM and PM waves.
Hint: Find $M(f)$ and find its 3 dB bandwidth ($B \ll \Delta f$).

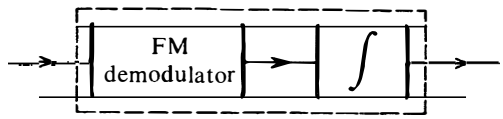
5.3-1 Design (only the block diagram) an Armstrong indirect FM modulator to generate an FM carrier with a carrier frequency of 98.1 MHz and $\Delta f = 75$ kHz. A narrowband FM generator is available at a carrier frequency of 100 kHz and a frequency deviation $\Delta f = 10$ Hz. The stock room also has an oscillator with an adjustable frequency in the range of 10 to 11 MHz. There are also plenty of frequency doublers, triplers, and quintuplers.

5.3-2 Design (only the block diagram) an Armstrong indirect FM modulator to generate an FM carrier with a carrier frequency of 96 MHz and $\Delta f = 20$ kHz. A narrowband FM generator with $f_c = 200$ kHz and adjustable Δf in the range of 9 to 10 Hz is available. The stock room also has an oscillator with adjustable frequency in the range of 9 to 10 MHz. There is a bandpass filter with any center frequency, and only frequency doublers are available.

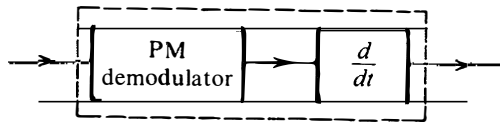
5.4-1 **(a)** Show that when $m(t)$ has no jump discontinuities, an FM demodulator followed by an integrator (Fig. P5.4-1a) forms a PM demodulator. Explain why it is necessary for the FM demodulator to remove any dc offset before the integrator.

(b) Show that a PM demodulator followed by a differentiator (Fig. P5.4-1b) serves as an FM demodulator even if $m(t)$ has jump discontinuities or if the PM demodulator output has dc offset.

Figure P.5.4-1



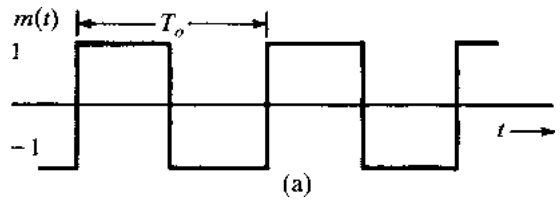
(a) PM demodulator



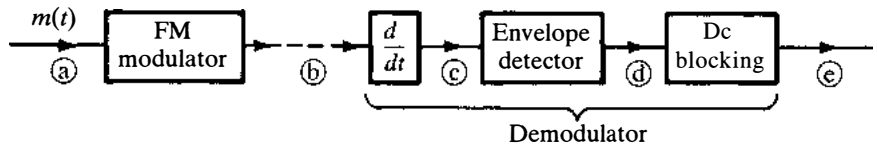
(b) FM demodulator

5.4-2 A periodic square wave $m(t)$ (Fig. P5.4-2a) frequency-modulates a carrier of frequency $f_c = 10$ kHz with $\Delta f = 1$ kHz. The carrier amplitude is A. The resulting FM signal is demodulated, as shown in Fig. P5.4-2b by the method discussed in Sec. 5.4 (Fig. 5.12). Sketch the waveforms at points b, c, d, and e.

Figure P.5.4-2



(a)



(b)

5.4-3 Use small-error PLL analysis to show that a first-order loop [$H(s) = 1$] cannot track an incoming signal whose instantaneous frequency is varying linearly with time [$\theta_i(t) = kt^2$]. This signal can be tracked within a constant phase if $H(s) = (s + a)/s$. It can be tracked with a zero phase error if $H(s) = (s^2 + as + b)/s^2$.

5.6-1 A transmitter transmits an AM signal with a carrier frequency of 1500 kHz. When an inexpensive radio receiver (which has a poor selectivity in its RF-stage bandpass filter) is tuned to 1500 kHz, the signal is heard loud and clear. This same signal is also heard (not as well) at another dial setting. State, with reasons, at what frequency you will hear this station. The IF frequency is 455 kHz.

5.6-2 Consider a superheterodyne FM receiver designed to receive the frequency band of 1 to 30 MHz with an IF frequency 8 MHz. What is the range of frequencies generated by the local oscillator for this receiver? An incoming signal with a carrier frequency of 10 MHz is received at the 10 MHz setting. At this setting of the receiver, we also get interference from a signal with some other carrier frequency if the receiver RF-stage bandpass filter has poor selectivity. What is the carrier frequency of the interfering signal?

6 SAMPLING AND ANALOG-TO-DIGITAL CONVERSION

As briefly discussed in Chapter 1, analog signals can be digitized through sampling and quantization. This analog-to-digital (A/D) conversion sets the foundation of modern digital communication systems. In the A/D converter, the sampling rate must be large enough to permit the analog signal to be reconstructed from the samples with sufficient accuracy. The **sampling theorem**, which is the basis for determining the proper (lossless) sampling rate for a given signal, has played a huge role in signal processing, communication theory, and A/D circuit design.

6.1 SAMPLING THEOREM

We first show that a signal $g(t)$ whose spectrum is band-limited to B Hz, that is,

$$G(f) = 0 \qquad \text{for } |f| > B$$

can be reconstructed exactly (without any error) from its discrete time samples taken uniformly at a rate of R samples per second. The condition is that $R > 2B$. In other words, the minimum sampling frequency for perfect signal recovery is $f_s = 2B$ Hz.

To prove the sampling theorem, consider a signal $g(t)$ (Fig. 6.1a) whose spectrum is band-limited to B Hz (Fig. 6.1b).* For convenience, spectra are shown as functions of f as well as of ω. Sampling $g(t)$ at a rate of f_s Hz means that we take f_s uniform samples per second. This uniform sampling can be accomplished by multiplying $g(t)$ by an impulse train $\delta_{T_s}(t)$ of Fig. 6.1c, consisting of unit impulses repeating periodically every T_s seconds, where $T_s = 1/f_s$. This results in the sampled signal $\overline{g}(t)$ shown in Fig. 6.1d. The sampled signal consists of impulses spaced every T_s seconds (the sampling interval). The nth impulse, located at $t = nT_s$, has a strength $g(nT_s)$ which is the value of $g(t)$ at $t = nT_s$. Thus, the relationship between the

* The spectrum $G(f)$ in Fig. 6.1b is shown as real, for convenience. Our arguments are valid for complex $G(f)$.

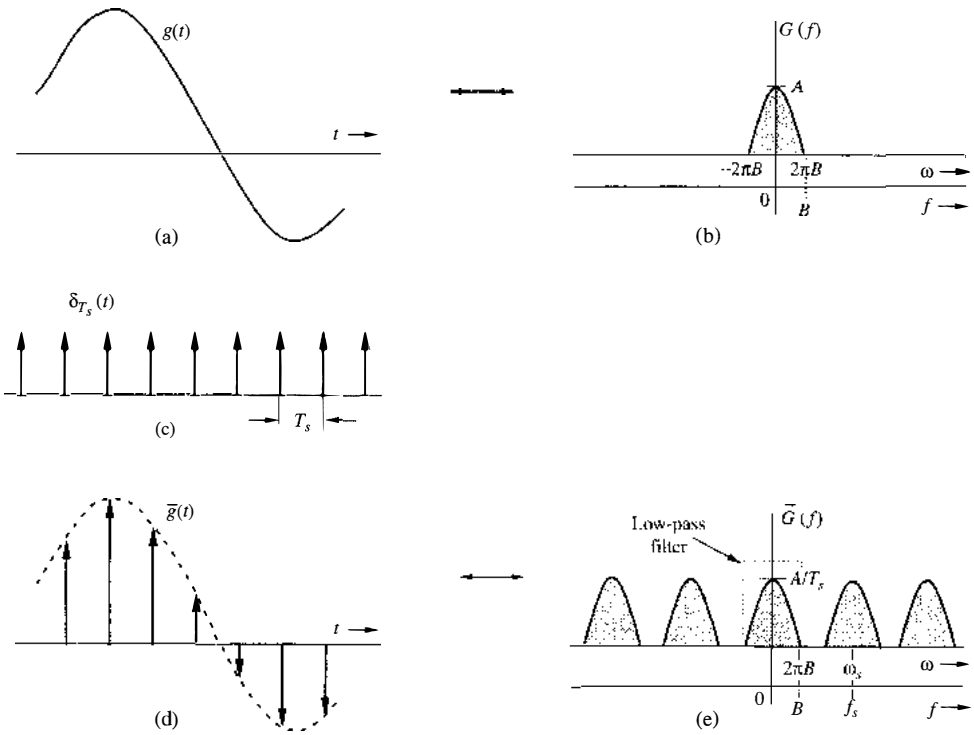


Figure 6.1 Sampled signal and its Fourier spectra.

sampled signal $\overline{g}(t)$ and the original analog signal $g(t)$ is

$$\overline{g}(t) = g(t)\delta_{T_s}(t) = \sum_n g(nT_s)\delta(t - nT_s) \tag{6.1}$$

Because the impulse train $\delta_{T_s}(t)$ is a periodic signal of period T_s, it can be expressed as an exponential Fourier series, already found in Example 3.11 as

$$\delta_{T_s}(t) = \frac{1}{T_s}\sum_{n=-\infty}^{\infty} e^{jn\omega_s t} \qquad \omega_s = \frac{2\pi}{T_s} = 2\pi f_s \tag{6.2}$$

Therefore,

$$\begin{aligned}\overline{g}(t) &= g(t)\delta_{T_s}(t) \\ &= \frac{1}{T_s}\sum_{n=-\infty}^{\infty} g(t)e^{jn2\pi f_s t}\end{aligned} \tag{6.3}$$

To find $\overline{G}(f)$, the Fourier transform of $\overline{g}(t)$, we take the Fourier transform of the summation in Eq. (6.3). Based on the frequency-shifting property, the transform of the *nth* term is shifted

by nf_s. Therefore,

$$\overline{G}(f) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} G(f - nf_s) \tag{6.4}$$

This means that the spectrum $\overline{G}(f)$ consists of $G(f)$, scaled by a constant $1/T_s$, repeating periodically with period $f_s = 1/T_s$ Hz, as shown in Fig. 6.1e.

After uniform sampling that generates a set of signal samples $\{g(kT_s)\}$, the vital question becomes: **Can $g(t)$ be reconstructed from $\overline{g}(t)$ without any loss or distortion?** If we are to reconstruct $g(t)$ from $\overline{g}(t)$, equivalently in the frequency domain we should be able to recover $G(f)$ from $\overline{G}(f)$. Graphically from Fig. 6.1, perfect recovery is possible if there is no overlap among the replicas in $\overline{G}(f)$. Figure 6.1e clearly shows that this requires

$$f_s > 2B \tag{6.5}$$

Also, the sampling interval $T_s = 1/f_s$. Therefore,

$$T_s < \frac{1}{2B} \tag{6.6}$$

Thus, as long as the sampling frequency f_s is greater than twice the signal bandwidth B (in hertz), $\overline{G}(f)$ will consist of nonoverlapping repetitions of $G(f)$. When this is true, Fig. 6.1e shows that $g(t)$ can be recovered from its samples $\overline{g}(t)$ by passing the sampled signal $\overline{g}(t)$ through an ideal low-pass filter of bandwidth B Hz. The minimum sampling rate $f_s = 2B$ required to recover $g(t)$ from its samples $\overline{g}(t)$ is called the **Nyquist rate** for $g(t)$, and the corresponding sampling interval $T_s = 1/2B$ is called the **Nyquist interval** for the low-pass signal $g(t)$.*

We need to stress one important point regarding the possibility of $f_s = 2B$ and a particular class of low-pass signals. For a general signal spectrum, we have proved that the sampling rate $f_s > 2B$. However, if the spectrum $G(f)$ has no impulse (or its derivatives) at the highest frequency B, then the overlap is still zero as long as the sampling rate is greater than or equal to the Nyquist rate, that is,

$$f_s \geq 2B$$

If, on the other hand, $G(f)$ contains an impulse at the highest frequency $\pm B$, then the equality must be removed or else overlap will occur. In such case, the sampling rate f_s must be greater than $2B$ Hz. A well-known example is a sinusoid $g(t) = \sin 2\pi B(t - t_0)$. This signal is band-limited to B Hz, but all its samples are zero when uniformly taken at a rate $f_s = 2B$ (starting at $t = t_0$), and $g(t)$ cannot be recovered from its Nyquist samples. Thus, for sinusoids, the condition of $f_s > 2B$ must be satisfied.

6.1.1 Signal Reconstruction from Uniform Samples

The process of reconstructing a continuous time signal $g(t)$ from its samples is also known as **interpolation**. In Fig. 6.1, we used a constructive proof to show that a signal $g(t)$ band-limited

* The theorem stated here (and proved subsequently) applies to low-pass signals. A bandpass signal whose spectrum exists over a frequency band $f_c - B/2 < |f| < f_c + B/2$ has a bandwidth B Hz. Such a signal is also uniquely determined by samples taken at above the Nyquist frequency $2B$. The sampling theorem is generally more complex in such case. It uses two interlaced uniform sampling trains, each at half the overall sampling rate $R_s > B$. See, for example, the Refs. 1 and 2.

to B Hz can be reconstructed (interpolated) exactly from its samples. This means not only that uniform sampling at above the Nyquist rate preserves all the signal information, but also that simply passing the sampled signal through an ideal low-pass filter of bandwidth B Hz will reconstruct the original message. As seen from Eq. (6.3), the sampled signal contains a component $(1/T_s)g(t)$, and to recover $g(t)$ [or $G(f)$], the sampled signal

$$\bar{g}(t) = \sum g(nT_s)\delta(t - nT_s)$$

must be sent through an ideal low-pass filter of bandwidth B Hz and gain T_s. Such an ideal filter response has the transfer function

$$H(f) = T_s \Pi\left(\frac{\omega}{4\pi B}\right) = T_s \Pi\left(\frac{f}{2B}\right) \tag{6.7}$$

Ideal Reconstruction

To recover the analog signal from its uniform samples, the ideal interpolation filter transfer function found in Eq. (6.7) is shown in Fig. 6.2a. The impulse response of this filter, the inverse Fourier transform of $H(f)$, is

$$h(t) = 2BT_s \operatorname{sinc}(2\pi Bt) \tag{6.8}$$

Assuming the use of Nyquist sampling rate, that is, $2BT_s = 1$, then

$$h(t) = \operatorname{sinc}(2\pi Bt) \tag{6.9}$$

This $h(t)$ is shown in Fig. 6.2b. Observe the very interesting fact that $h(t) = 0$ at all Nyquist sampling instants ($t = \pm n/2B$) except $t = 0$. When the sampled signal $\bar{g}(t)$ is applied at the input of this filter, the output is $g(t)$. Each sample in $\bar{g}(t)$, being an impulse, generates a sinc pulse of height equal to the strength of the sample, as shown in Fig. 6.2c. The process is

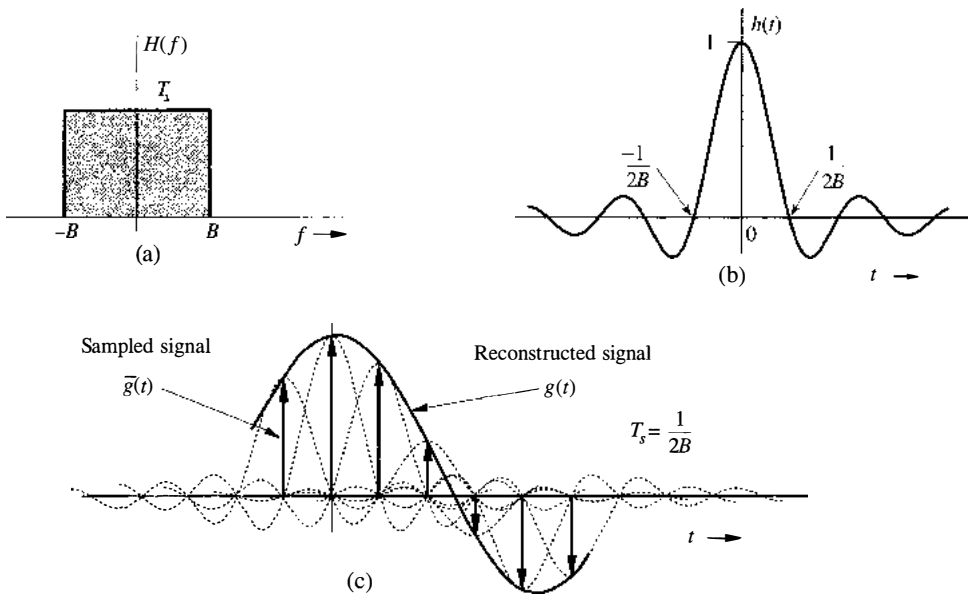


Figure 6.2 Ideal interpolation.

identical to that shown in Fig. 6.6, except that $h(t)$ is a sinc pulse instead of a rectangular pulse. Addition of the sinc pulses generated by all the samples results in $g(t)$. The kth sample of the input $\overline{g}(t)$ is the impulse $g(kT_s)\delta(t-kT_s)$; the filter output of this impulse is $g(kT_s)h(t-kT_s)$. Hence, the filter output to $\overline{g}(t)$, which is $g(t)$, can now be expressed as a sum,

$$g(t) = \sum_k g(kT_s)h(t-kT_s)$$

$$= \sum_k g(kT_s)\ \text{sinc}\ [2\pi B(t-kT_s)] \tag{6.10a}$$

$$= \sum_k g(kT_s)\ \text{sinc}\ (2\pi Bt - k\pi) \tag{6.10b}$$

Equation (6.10) is the **interpolation formula**, which yields values of $g(t)$ between samples as a weighted sum of all the sample values.

Example 6.1 Find a signal $g(t)$ that is band-limited to B Hz and whose samples are

$$g(0) = 1 \quad \text{and} \quad g(\pm T_s) = g(\pm 2T_s) = g(\pm 3T_s) = \cdots = 0$$

where the sampling interval T_s is the Nyquist interval for $g(t)$, that is, $T_s = 1/2B$.

We use the interpolation formula (6.10b) to construct $g(t)$ from its samples. Since all but one of the Nyquist samples are zero, only one term (corresponding to $k = 0$) in the summation on the right-hand side of Eq. (6.10b) survives. Thus,

$$g(t) = \text{sinc}\,(2\pi Bt) \tag{6.11}$$

This signal is shown in Fig. 6.3. Observe that this is the only signal that has a bandwidth B Hz and sample values $g(0) = 1$ and $g(nT_s) = 0\,(n \neq 0)$. No other signal satisfies these conditions.

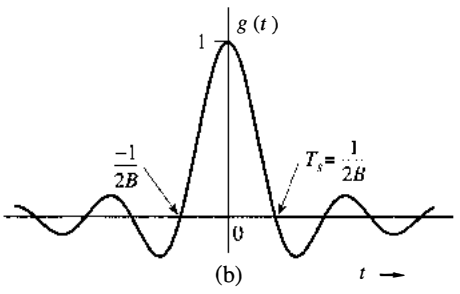


Figure 6.3 Signal reconstructed from the Nyquist samples in Example 6.1.

Practical Signal Reconstruction (Interpolation)

We established in Sec. 3.5 that the ideal low-pass filter is noncausal and unrealizable. This can be equivalently seen from the infinitely long nature of the sinc reconstruction pulse used in the ideal reconstruction of Eq. (6.10). For practical application of signal reconstruction (e.g., a

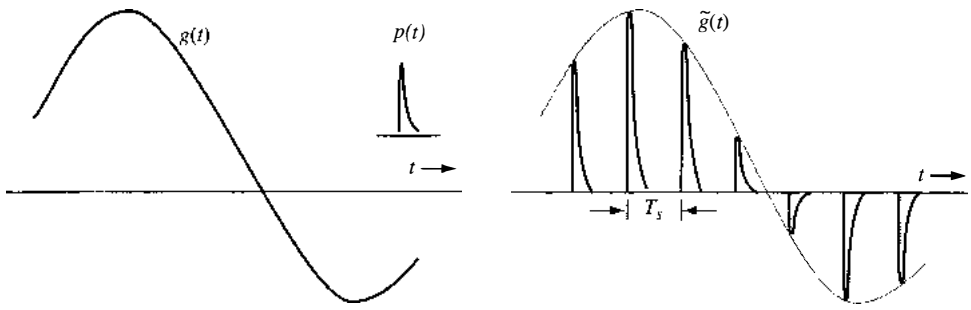


Figure 6.4 Practical reconstruction (interpolation) pulse.

CD player), we need to implement realizable signal reconstruction systems from the uniform signal samples.

For practical implementation, this reconstruction pulse $p(t)$ must be easy to generate. For example, we may apply the reconstruction pulse $p(t)$ as shown in Fig. 6.4. However, we must first use the nonideal interpolation pulse $p(t)$ to analyze the accuracy of the reconstructed signal. Let us denote the new signal from reconstruction as

$$\tilde{g}(t) \triangleq \sum_n g(nT_s)p(t - nT_s) \tag{6.12}$$

To determine its relation to the original analog signal $g(t)$, we can see from the properties of convolution and Eq.(6.1) that

$$\begin{aligned}\tilde{g}(t) &= \sum_n g(nT_s)p(t - nT_s) = p(t) * \left[\sum_n g(nT_s)\delta(t - nT_s)\right] \\ &= p(t) * \bar{g}(t)\end{aligned} \tag{6.13a}$$

In the frequency domain, the relationship between the reconstruction and the original analog signal can rely on Eq. (6.4)

$$\tilde{G}(f) = P(f)\frac{1}{T_s}\sum_n G(f - nf_s) \tag{6.13b}$$

This means that the reconstructed signal $\tilde{g}(t)$ using pulse $p(t)$ consists of multiple replicas of $G(f)$ shifted to the frequency center nf_s and filtered by $P(f)$. To fully recover $g(t)$, further filtering of $\tilde{g}(t)$ becomes necessary. Such filters are often referred to as equalizers.

Denote the equalizer transfer function as $E(f)$. Distortionless reconstruction requires that

$$\begin{aligned}G(f) &= E(f)\tilde{G}(f) \\ &= E(f)P(f)\frac{1}{T_s}\sum_n G(f - nf_s)\end{aligned}$$

This relationship clearly illustrates that the equalizer must remove all the shifted replicas $G(f - nf_s)$ in the summation except for the low-pass term with $n = 0$, that is,

$$E(f)P(f) = 0 \quad |f| > f_s - B \tag{6.14a}$$

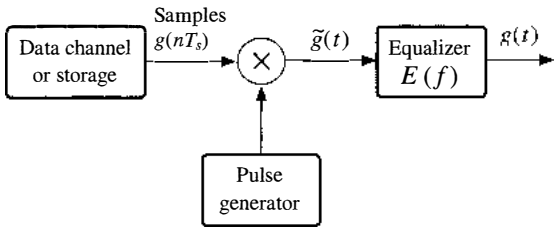


Figure 6.5 Practical signal reconstruction.

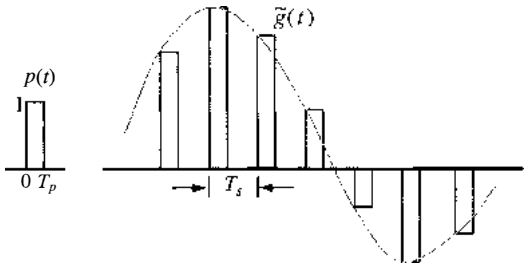


Figure 6.6 Simple interpolation by means of simple rectangular pulses.

Additionally, distortionless reconstruction requires that

$$E(f)P(f) = T_s \quad |f| < B \tag{6.14b}$$

The equalizer filter $E(f)$ must be low-pass in nature to stop all frequency content above $f_s - B$ Hz, and it should be the inverse of $P(f)$ within the signal bandwidth of B Hz. Figure 6.5 demonstrates the diagram of a practical signal reconstruction system utilizing such an equalizer.

Let us now consider a very simple interpolating pulse generator that generates short (zero-order hold) pulses. As shown in Fig. 6.6,

$$p(t) = \Pi\left(\frac{t - 0.5T_p}{T_p}\right)$$

This is a gate pulse of unit height with pulse duration T_p. The reconstruction will first generate

$$\tilde{g}(t) = \sum_n g(nT_s)\,\Pi\left(\frac{t - nT_s - 0.5T_p}{T_p}\right)$$

The transfer function of filter $P(f)$ is the Fourier transform of $\Pi(t/T_p)$ shifted by $0.5T_p$:

$$P(f) = T_p\,\text{sinc}\left(\pi f T_p\right) e^{-j\pi f T_p} \tag{6.15}$$

As a result, the equalizer frequency response should satisfy

$$E(f) = \begin{cases} T_s/P(f) & |f| \le B \\ \text{Flexible} & B < |f| < (1/T_s - B) \\ 0 & |f| \ge (1/T_s - B) \end{cases}$$

It is important for us to ascertain that the equalizer passband response is realizable. First of all, we can add another time delay to the reconstruction such that

$$E(f) = T_s \cdot \frac{\pi f}{\sin(\pi f T_p)} e^{-j2\pi f t_0} \qquad |f| \le B \tag{6.16}$$

For the passband gain of $E(f)$ to be well defined, it is imperative for us to choose a short pulse width T_p such that

$$\frac{\sin(\pi f T_p)}{\pi f} \neq 0 \quad |f| \le B$$

This means that the equalizer $E(f)$ does not need to achieve infinite gain. Otherwise the equalizer would become unrealizable. Equivalently, this requires that

$$T_p < 1/B$$

Hence, as long as the rectangular reconstruction pulse width is shorter than $1/B$, it may be possible to design an analog equalizer filter to recover the original analog signal $g(t)$ from the nonideal reconstruction pulse train. Of course, this is a requirement for a rectangular reconstruction pulse generator. In practice, T_p can be chosen very small, to yield the following equalizer passband response:

$$E(f) = T_s \cdot \frac{\pi f}{\sin(\pi f T_p)} \approx \frac{T_s}{T_p} \qquad |f| \le B \tag{6.17}$$

This means that very little distortion remains when very short rectangular pulses are used in signal reconstruction. Such cases make the design of the equalizer either unnecessary or very simple. An illustrative example is given as a MATLAB exercise in Sec. 6.9.

We can improve on the zero-order-hold filter by using the **first-order-hold** filter, which results in a linear interpolation instead of the staircase interpolation. The linear interpolator, whose impulse response is a triangle pulse $\Delta(t/2T_s)$, results in an interpolation in which successive sample tops are connected by straight-line segments (Prob. 6.1-7).

6.1.2 Practical Issues in Signal Sampling and Reconstruction

Realizability of Reconstruction Filters

If a signal is sampled at the Nyquist rate $f_s = 2B$ Hz, the spectrum $\overline{G}(f)$ consists of repetitions of $G(f)$ without any gap between successive cycles, as shown in Fig. 6.7a. To recover $g(t)$ from $\overline{g}(t)$, we need to pass the sampled signal $\overline{g}(t)$ through an ideal low-pass filter (dotted area in Fig. 6.7a). As seen in Sec. 3.5, such a filter is unrealizable in practice; it can be closely approximated only with infinite time delay in the response. This means that we can recover the signal $g(t)$ from its samples with infinite time delay.

A practical solution to this problem is to sample the signal at a rate higher than the Nyquist rate ($f_s > 2B$ or $\omega_s > 4\pi B$). This yields $\overline{G}(f)$, consisting of repetitions of $G(f)$ with a finite band gap between successive cycles, as shown in Fig. 6.7b. We can now recover $G(f)$ from $\overline{G}(f)$ [or from $\widetilde{G}(f)$] by using a low-pass filter with a gradual cutoff characteristic (dotted area in Fig. 6.7b). But even in this case, the filter gain is required to be zero beyond the first cycle

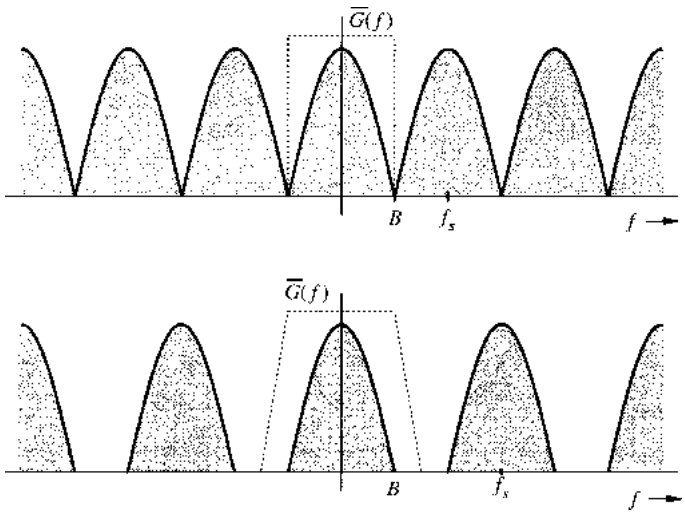


Figure 6.7 Spectra of a sampled signal: (a) at the Nyquist rate; (b) above the Nyquist rate.

of $G(f)$ (Fig. 6.7b). According to the Paley-Wiener criterion, it is impossible to realize even this filter. The only advantage in this case is that the required filter can be better approximated with a smaller time delay. This shows that it is impossible in practice to recover a band-limited signal $g(t)$ exactly from its samples, even if the sampling rate is higher than the Nyquist rate. However, as the sampling rate increases, the recovered signal approaches the desired signal more closely.

The Treachery of Aliasing

There is another fundamental practical difficulty in reconstructing a signal from its samples. The sampling theorem was proved on the assumption that the signal $g(t)$ is band-limited. *All practical signals are time-limited;* that is, they are of finite duration or width. We can demonstrate (Prob. 6.1-8) that a signal cannot be time-limited and band-limited simultaneously. A time-limited signal cannot be band-limited, and vice versa (but a signal can be simultaneously non-time-limited and non-band-limited). Clearly, all practical signals, which are necessarily time-limited, are non-band-limited, as shown in Fig. 6.8a; they have infinite bandwidth, and the spectrum $\overline{G}(f)$ consists of overlapping cycles of $G(f)$ repeating every f_s Hz (the sampling frequency), as illustrated in Fig. 6.8b. Because of the infinite bandwidth in this case, the spectral overlap is unavoidable, regardless of the sampling rate. Sampling at a higher rate reduces but does not eliminate overlapping between repeating spectral cycles. Because of the overlapping tails, $\overline{G}(f)$ no longer has complete information about $G(f)$, and it is no longer possible, even theoretically, to recover $g(t)$ exactly from the sampled signal $\overline{g}(t)$. If the sampled signal is passed through an ideal low-pass filter of cutoff frequency $f_s/2$ Hz, the output is not $G(f)$ but $G_a(f)$ (Fig. 6.8c), which is a version of $G(f)$ distorted as a result of two separate causes:

1. The loss of the tail of $G(f)$ beyond $|f| > f_s/2$ Hz.
2. The reappearance of this tail inverted or folded back onto the spectrum.

Note that the spectra cross at frequency $f_s/2 = 1/2T$ Hz, which is called the *folding* frequency. The spectrum may be viewed as if the lost tail is folding back onto itself at the folding frequency. For instance, a component of frequency $(f_s/2) + f_z$ shows up as, or "impersonates," a component of lower frequency $(f_s/2) - f_z$ in the reconstructed signal. Thus, the components of frequencies above $f_s/2$ reappear as components of frequencies below $f_s/2$. This tail inversion,

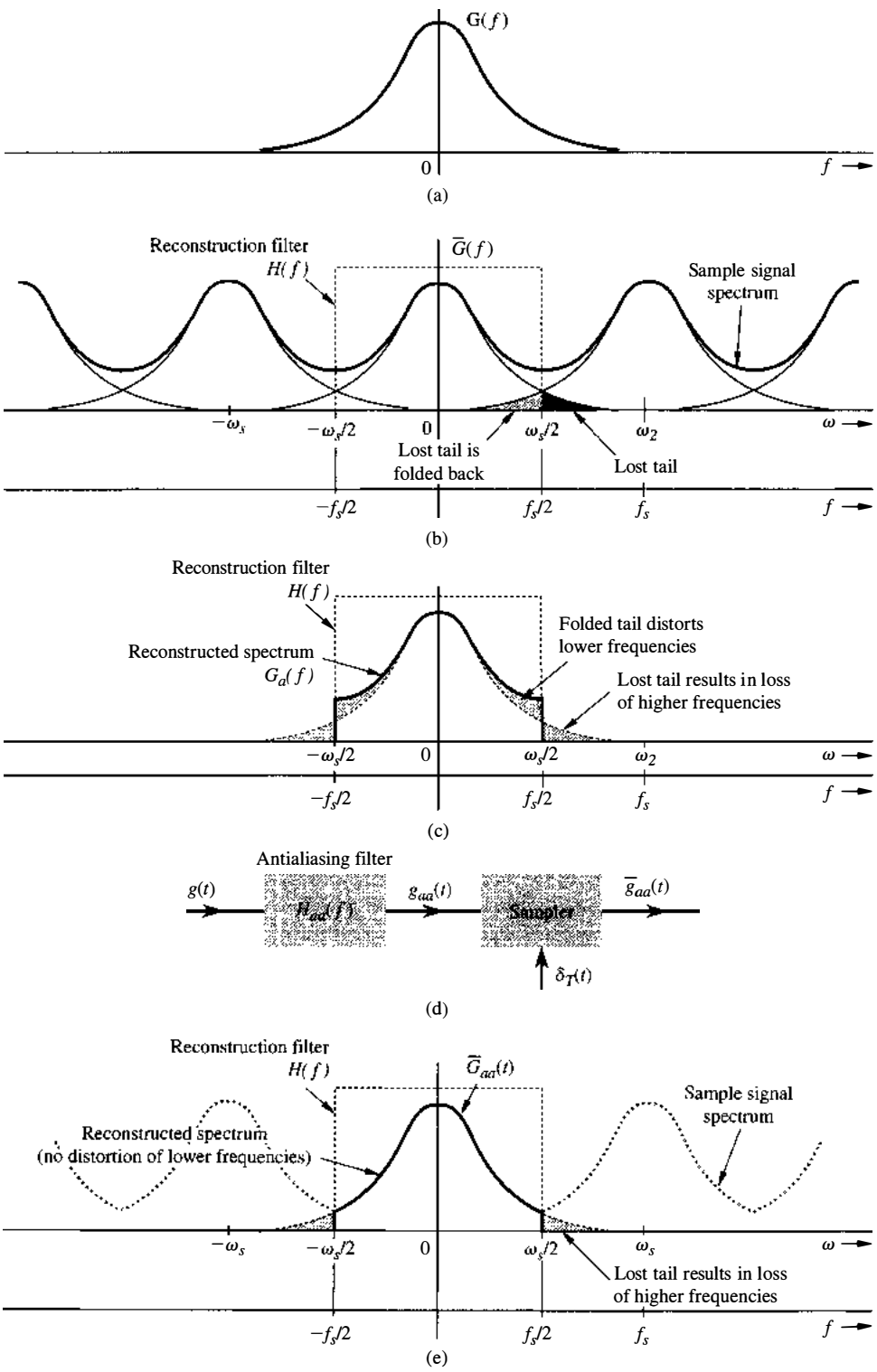


Figure 6.8
Aliasing effect.
(a) Spectrum of a practical signal $g(t)$.
(b) Spectrum of sampled $g(t)$.
(c) Reconstructed signal spectrum.
(d) Sampling scheme using antialiasing filter.
(e) Sampled signal spectrum (dotted) and the reconstructed signal spectrum (solid) when antialiasing filter is used.

known as *spectral folding* or *aliasing,* is shown shaded in Fig. 6.8b and also in Fig. 6.8c. In the process of aliasing, not only are we losing all the components of frequencies above the folding frequency $f_s/2$ Hz, but these very components reappear (aliased) as lower frequency components in Fig. 6.8b or c. Such aliasing destroys the integrity of the frequency components below the folding frequency $f_s/2$, as depicted in Fig. 6.8c.

The problem of aliasing is analogous to that of an army when a certain platoon has secretly defected to the enemy side but remains nominally loyal to their army. The army is in double jeopardy. First, it has lost the defecting platoon as an effective fighting force. In addition, during actual fighting, the army will have to contend with sabotage caused by the defectors and will have to use loyal platoon to neutralize the defectors. Thus, the army has lost two platoons to nonproductive activity.

Defectors Eliminated: The Antialiasing Filter

If you were the commander of the betrayed army, the solution to the problem would be obvious. As soon as you got wind of the defection, you would incapacitate, by whatever means, the defecting platoon. By taking this action *before the fighting begins*, you lose only one (the defecting)* platoon. This is a partial solution to the double jeopardy of betrayal and sabotage, a solution that partly rectifies the problem and cuts the losses in half.

We follow exactly the same procedure. The potential defectors are all the frequency components beyond the folding frequency $f_s/2 = 1/2T$ Hz. We should eliminate (suppress) these components from $g(t)$ *before sampling* $g(t)$. Such suppression of higher frequencies can be accomplished by an ideal low-pass filter of cutoff $f_s/2$ Hz, as shown in Fig. 6.8d. This is called the *antialiasing filter*. Figure 6.8d also shows that antialiasing filtering is performed before sampling. Figure 6.8e shows the sampled signal spectrum and the reconstructed signal $G_{aa}(f)$ when the antialiasing scheme is used. An antialiasing filter essentially band-limits the signal $g(t)$ to $f_s/2$ Hz. This way, we lose only the components beyond the folding frequency $f_s/2$ Hz. These suppressed components now cannot reappear, corrupting the components of frequencies below the folding frequency. Clearly, use of an antialiasing filter results in the reconstructed signal spectrum $G_{aa}(f) = G(f)$ for $|f| < f_s/2$. Thus, although we lost the spectrum beyond $f_s/2$ Hz, the spectrum for all the frequencies below $f_s/2$ remains intact. The effective aliasing distortion is cut in half owing to elimination of folding. We stress again that the antialiasing operation must be performed *before the signal is sampled.*

An antialiasing filter also helps to reduce noise. Noise, generally, has a wideband spectrum, and without antialiasing, the aliasing phenomenon itself will cause the noise components outside the desired signal band to appear in the signal band. Antialiasing suppresses the entire noise spectrum beyond frequency $f_s/2$.

The antialiasing filter, being an ideal filter, is unrealizable. In practice we use a steep-cutoff filter, which leaves a sharply attenuated residual spectrum beyond the folding frequency $f_s/2$.

Sampling Forces Non-Band-Limited Signals to Appear Band-Limited

Figure 6.8b shows the spectrum of a signal $\overline{g}(t)$ consists of overlapping cycles of $G(f)$. This means that $\overline{g}(t)$ are sub-Nyquist samples of $g(t)$. However, we may also view the spectrum in Fig. 6.8b as the spectrum $G_a(f)$ (Fig. 6.8c), repeating periodically every f_s Hz without overlap. The spectrum $G_a(f)$ is band-limited to $f_s/2$ Hz. Hence, these (sub-Nyquist) samples of $g(t)$

* Figure 6.8b shows that from the infinite number of repeating cycles, only the neighboring spectral cycles overlap. This is a somewhat simplified picture. In reality, all the cycles overlap and interact with every other cycle because of the infinite width of all practical signal spectra. Fortunately, all practical spectra also must decay at higher frequencies. This results in an insignificant amount of interference from cycles other than the immediate neighbors. When such an assumption is not justified, aliasing computations become little more involved.

are actually the Nyquist samples for signal $g_a(t)$. In conclusion, sampling a non-band-limited signal $g(t)$ at a rate f_s Hz makes the samples appear to be the Nyquist samples of some signal $g_a(t)$, band-limited to $f_s/2$ Hz. In other words, sampling makes a non-band-limited signal appear to be a band-limited signal $g_a(t)$ with bandwidth $f_s/2$ Hz. A similar conclusion applies if $g(t)$ is band-limited but sampled at a sub-Nyquist rate.

6.1.3 Maximum Information Rate: Two Pieces of Information per Second per Hertz

A knowledge of the maximum rate at which information can be transmitted over a channel of bandwidth B Hz is of fundamental importance in digital communication. We now derive one of the basic relationships in communication, which states that *a maximum of* $2B$ *independent pieces of information per second can be transmitted, error free, over a noiseless channel of bandwidth* B *Hz.* The result follows from the sampling theorem.

First, the sampling theorem shows that a low-pass signal of bandwidth B Hz can be fully recovered from samples uniformly taken at the rate of $2B$ samples per second. Conversely, we need to show that any sequence of independent data at the rate of $2B$ Hz can come from uniform samples of a low-pass signal with bandwidth B. Moreover, we can construct this low-pass signal from the independent data sequence.

Suppose a sequence of independent data samples is denoted as $\{g_n\}$. Its rate is $2B$ samples per second. Then there always exists a (not necessarily band-limited) signal $g(t)$ such that

$$g_n = g(nT_s) \qquad T_s = \frac{1}{2B}$$

In Figure 6.9a we illustrate again the effect of sampling the non-band-limited signal $g(t)$ at sampling rate $f_s = 2B$ Hz. Because of aliasing, the ideal sampled signal

$$\begin{aligned}\overline{g}(t) &= \sum_n g(nT_s)\delta(t - nT_s) \\ &= \sum_n g_a(nT_s)\delta(t - nT_s)\end{aligned}$$

where $g_a(t)$ is the aliased low-pass signal whose samples $g_a(nT_s)$ equal to the samples of $g(nT_s)$. In other words, sub-Nyquist sampling of a signal $g(t)$ generates samples that can be equally well obtained by Nyquist sampling of a band-limited signal $g_a(t)$. Thus, through Figure 6.9, we demonstrate that sampling $g(t)$ and $g_a(t)$ at the rate of $2B$ Hz will generate the same independent information sequence $\{g_n\}$:

$$g_n = g(nT_s) = g_a(nT_s) \qquad T_s = \frac{1}{2B} \tag{6.18}$$

Also from the sampling theorem, a low-pass signal $g_a(t)$ with bandwidth B can be reconstructed from its uniform samples [Eq. (6.10)]

$$g_a(t) = \sum_n g_n \text{ sinc } (2\pi Bt - k\pi)$$

Assuming no noise, this signal can be transmitted over a distortionless channel of bandwidth B Hz, error free. At the receiver, the data sequence $\{g_n\}$ can be recovered from the Nyquist samples of the distortionless channel output $g_a(t)$ as the desired information data.

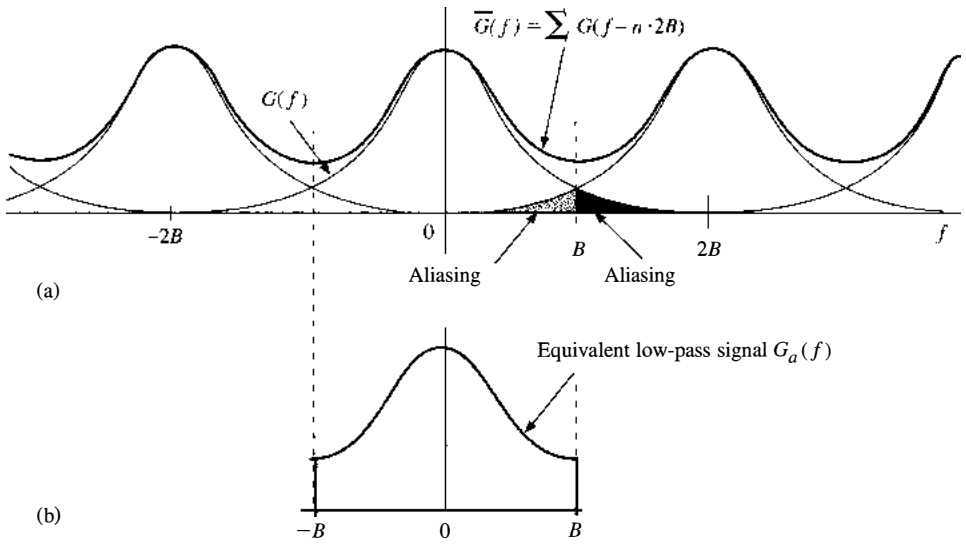


Figure 6.9
(a) Non-band-limited signal spectrum and its sampled spectrum $\overline{G}(f)$. (b) Equivalent low-pass signal spectrum $G_a(f)$ constructed from uniform samples of $g(t)$ at sampling rate $2B$.

This theoretical rate of communication assumes a noise-free channel. In practice, channel noise is unavoidable, and consequently, this rate will cause some detection errors. In Chapter 14, we shall present the Shannon capacity which determines the theoretical error-free communication rate in the presence of noise.

6.1.4 Nonideal Practical Sampling Analysis

Thus far, we have mainly focused on ideal uniform sampling that can use an ideal impulse sampling pulse train to precisely extract the signal value $g(kT_s)$ at the precise instant of $t = kT_s$. In practice, no physical device can carry out such a task. Consequently, we need to consider the more practical implementation of sampling. This analysis is important to the better understanding of errors that typically occur during practical A/D conversion and their effects on signal reconstruction.

Practical samplers take each signal sample over a short time interval T_p around $t = kT_s$. In other words, every T_s seconds, the sampling device takes a short snapshot of duration T_p from the signal $g(t)$ being sampled. This is just like taking a sequence of still photographs of a sprinter during an 100-meter Olympic race. Much like a regular camera that generates a still picture by averaging the picture scene over the window T_p, the practical sampler would generate a sample value at $t = kT_s$ by averaging the values of signal $g(t)$ over the window T_p, that is,

$$g_1(kT_s) = \frac{1}{T_p} \int_{-T_p/2}^{T_p/2} g(kT_s + t)\, dt \tag{6.19a}$$

Depending on the actual device, this averaging may be weighted by a device-dependent averaging function $q(t)$ such that

$$g_1(kT_s) = \frac{1}{T_p} \int_{-T_p/2}^{T_p/2} q(t) g(kT_s + t)\, dt \tag{6.19b}$$

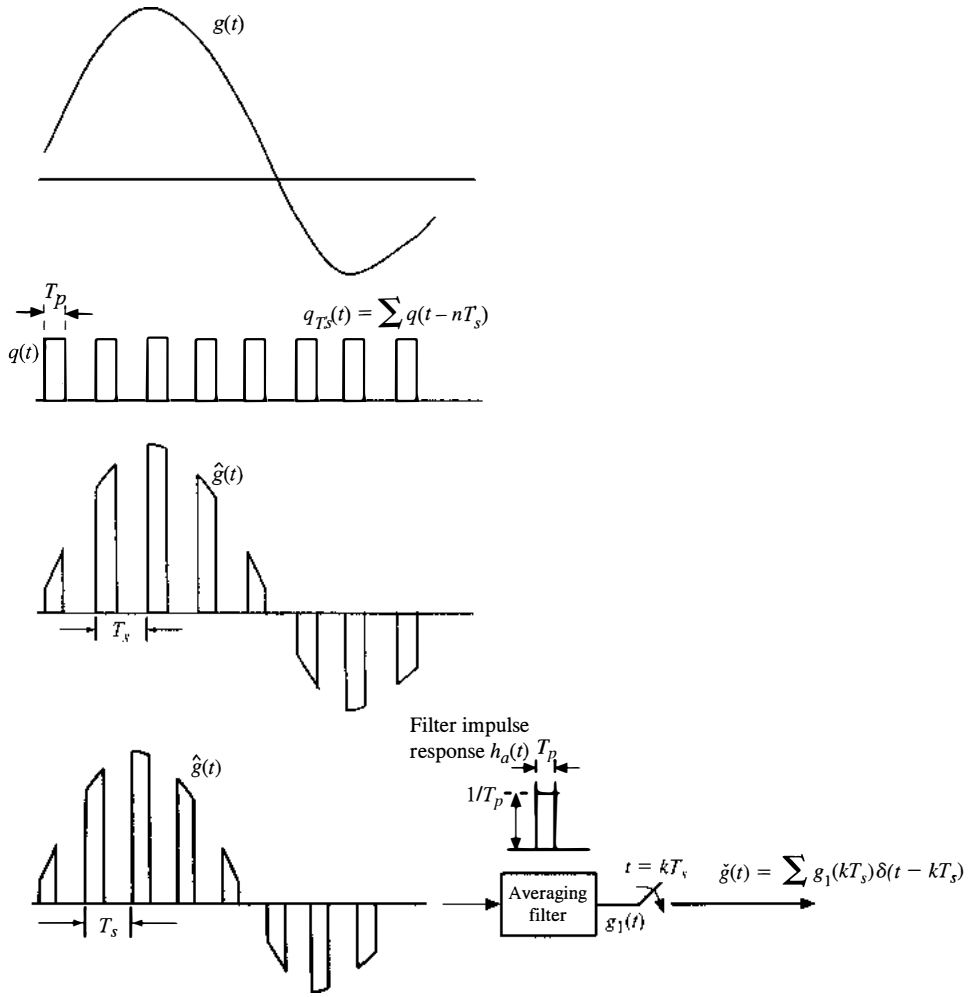


Figure 6.10 Illustration of practical sampling.

Thus we have used the camera analogy to establish that practical samplers in fact generate sampled signal of the form

$$\breve{g}(t) = \sum g_1(kT_s)\delta(t - kT_s) \tag{6.20}$$

We will now show the relationship between the practically sampled signal $\breve{g}(t)$ and the original low-pass analog signal $g(t)$ in the frequency domain.

We will use Fig. 6.10 to illustrate the relationship between $\breve{g}(t)$ and $g(t)$ for the special case of uniform weighting. This means that

$$q(t) = \begin{cases} 1 & |t| \leq 0.5T_p \\ 0 & |t| > 0.5T_p \end{cases}$$

As shown in Fig. 6.10, $g_1(t)$ can be equivalently obtained by first using "natural gating" to generate the signal *snapshots*

$$\widehat{g}(t) = g(t) \cdot q_{T_s}(t) \tag{6.21}$$

where

$$q_{T_s}(t) = \sum_{n=-\infty}^{\infty} q(t - nT_s)$$

Figure 6.10b illustrates the snapshot signal $\widehat{g}(t)$. We can then define an averaging filter with impulse response

$$h_a(t) = \begin{cases} \dfrac{1}{T_p} & -\dfrac{T_p}{2} \le t < \dfrac{T_p}{2} \\ 0 & \text{elsewhere} \end{cases}$$

or transfer function

$$H_a(f) = \text{sinc}\,(\pi f T_p)$$

Sending the naturally gated snapshot signal $\widehat{g}(t)$ into the averaging filter generates the output signal

$$g_1(t) = h_a(t) * \widehat{g}(t)$$

As illustrated in Fig. 6.10c, the practical sampler generate a sampled signal $\check{g}(t)$ by sampling the averaging filter output $g_1(kT_s)$. Thus we have used Fig. 6.10c to establish the equivalent process of taking snapshots, averaging, and sampling in generating practical samples of $g(t)$. Now we can examine the frequency domain relationships to analyze the distortion generated by practical samplers.

In the following analysis, we will consider a general weighting function $q(t)$ whose only constraint is that

$$q(t) = 0, \qquad t \notin (-0.5T_p,\ 0.5T_p)$$

To begin, note that $q_{T_s}(t)$ is periodic. Therefore, its Fourier series can be written as

$$q_{T_s}(t) = \sum_{n=-\infty}^{\infty} Q_n e^{jn\omega_s t}$$

where

$$Q_n = \frac{1}{T_s} \int_{0.5T_p}^{0.5T_p} q(t) e^{-jn\omega_s t}\, dt$$

Thus, the averaging filter output signal is

$$\begin{aligned} g_1(t) &= h_a(t) * \left[g(t) q_{T_s}(t) \right] \\ &= h_a(t) * \sum_{n=-\infty}^{\infty} Q_n g(t) e^{jn\omega_s t} \end{aligned} \tag{6.22}$$

In the frequency domain, we have

$$G_1(f) = H(f) \sum_{n=-\infty}^{\infty} Q_n G(f - nf_s)$$

$$= \text{sinc}\,(\pi f T_p) \sum_{n=-\infty}^{\infty} Q_n G(f - nf_s) \tag{6.23}$$

Because

$$\check{g}(t) = \sum_k g_1(kT_s)\delta(t - kT_s)$$

we can apply the sampling theorem to show that

$$\check{G}(f) = \frac{1}{T_s} \sum_m G_1(f + mf_s)$$

$$= \frac{1}{T_s} \sum_m \text{sinc}\left[\frac{(2\pi f + m2\pi f_s)T_p}{2}\right] \sum_n Q_n G(f + mf_s - nf_s)$$

$$= \sum_\ell \left(\frac{1}{T_s} \sum_n Q_n \,\text{sinc}\left[(\pi f + (n + \ell)\pi f_s)T_p\right]\right) G(f + \ell f_s) \tag{6.24}$$

The last equality came from the change of the summation index $\ell = m - n$.

We can define frequency responses

$$F_\ell(f) = \frac{1}{T_s} \sum_n Q_n \,\text{sinc}\left[(\pi f + (n + \ell)\pi f_s)T_p\right]$$

This definition allows us to conveniently write

$$\check{G}(f) = \sum_\ell F_\ell(f) G_1(f + \ell f_s) \tag{6.25}$$

For the low-pass signal $G(f)$ with bandwidth B Hz, applying an ideal low-pass (interpolation) filter will generate a distorted signal

$$F_0(f)G(f) \tag{6.26a}$$

in which

$$F_0(f) = \frac{1}{T_s} \sum_n Q_n \,\text{sinc}\left[\pi(f + nf_s)T_p\right] \tag{6.26b}$$

It can be seen from Eqs. (6.25) and (6.26) that the practically sampled signal already contains a known distortion $F_0(f)$.

Moreover, the use of a practical reconstruction pulse $p(t)$ as in Eq. (6.12) will generate additional distortion. Let us reconstruct $g(t)$ by using the practical samples to generate

$$\breve{g}(t) = \sum_n g_1(nT_s)p(t - nT_s)$$

Then from Eq. (6.13) we obtain the relationship between the spectra of the reconstruction and the original message $G(f)$ as

$$\breve{G}(f) = P(f) \sum_n F_n(f)G(f + nf_s) \tag{6.27}$$

Since $G(f)$ has bandwidth B Hz, we will need to design a new equalizer with transfer function $E(f)$ such that the reconstruction is distortionless within the bandwidth B, that is,

$$E(f)P(f)F_0(f) = \begin{cases} 1 & |f| < B \\ \text{Flexible} & B < |f| < f_s - B \\ 0 & |f| > f_s - B \end{cases} \tag{6.28}$$

This single equalizer can be designed to compensate for two sources of distortion: nonideal sampling effect in $F_0(f)$ and nonideal reconstruction effect in $P(f)$. The equalizer design is made practically possible because both distortions are known in advance.

6.1.5 Some Applications of the Sampling Theorem

The sampling theorem is very important in signal analysis, processing, and transmission because it allows us to replace a continuous time signal by a discrete sequence of numbers. Processing a continuous time signal is therefore equivalent to processing a discrete sequence of numbers. This leads us directly into the area of digital filtering. In the field of communication, the transmission of a continuous time message reduces to the transmission of a sequence of numbers. This opens doors to many new techniques of communicating continuous time signals by pulse trains. The continuous time signal $g(t)$ is sampled, and sample values are used to modify certain parameters of a periodic pulse train. We may vary the amplitudes (Fig. 6.11b), widths (Fig. 6.11c), or positions (Fig. 6.11d) of the pulses in proportion to the sample values of the signal $g(t)$. Accordingly, we can have **pulse amplitude modulation** (PAM), **pulse width modulation** (PWM), or **pulse position modulation** (PPM). The most important form of pulse modulation today is **pulse code modulation** (PCM), introduced in Sec. 1.2. In all these cases, instead of transmitting $g(t)$, we transmit the corresponding pulse-modulated signal. At the receiver, we read the information of the pulse-modulated signal and reconstruct the analog signal $g(t)$.

One advantage of using pulse modulation is that it permits the simultaneous transmission of several signals on a time-sharing basis—**time division multiplexing (TDM)**. Because a pulse-modulated signal occupies only a part of the channel time, we can transmit several pulse-modulated signals on the same channel by interweaving them. Figure 6.12 shows the TDM of two PAM signals. In this manner we can multiplex several signals on the same channel by reducing pulse widths.

Another method of transmitting several baseband signals simultaneously is frequency division multiplexing (FDM), briefly discussed in Chapter 4. In FDM, various signals are multiplexed by sharing the channel bandwidth. The spectrum of each message is shifted to a specific band not occupied by any other signal. The information of various signals is located in nonoverlapping frequency bands of the channel. In a way, TDM and FDM are duals of each other.

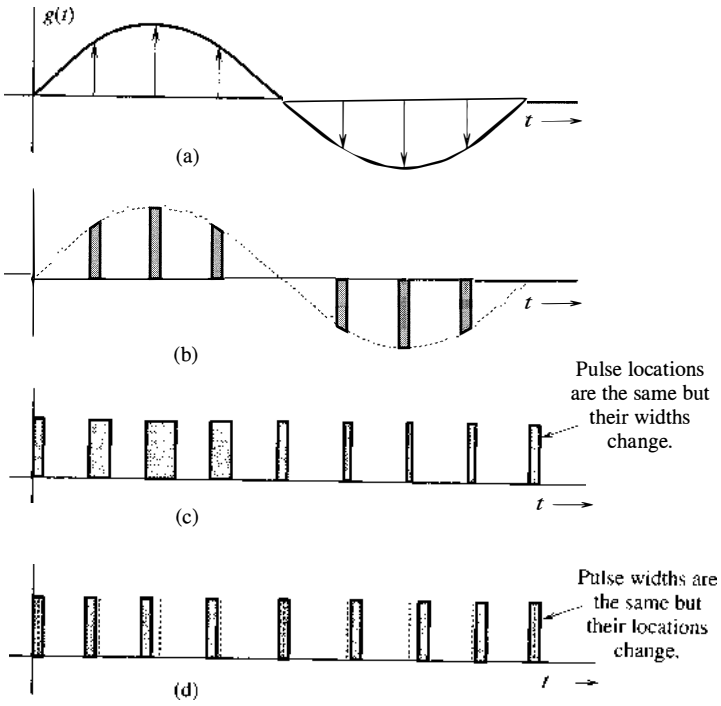


Figure 6.11 Pulse-modulated signals. (a) The unmodulated signal. (b) The PAM signal. (c) The PWM (PDM) signal. (d) The PPM signal.

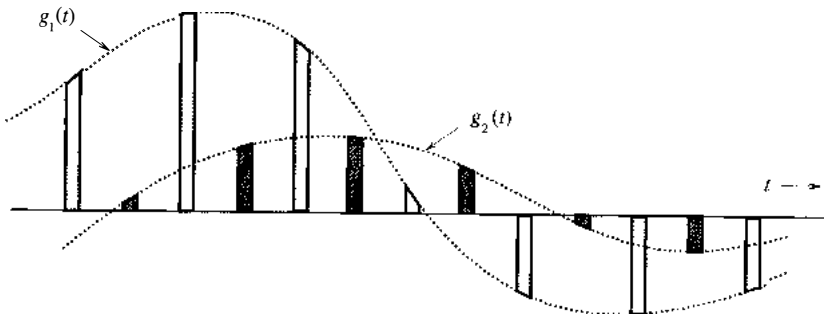


Figure 6.12 Time division multiplexing of two signals.

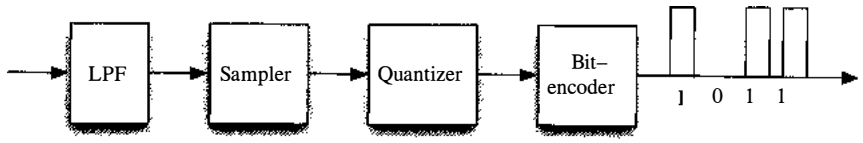


Figure 6.13 PCM system diagram.

6.2 PULSE CODE MODULATION (PCM)

PCM is the most useful and widely used of all the pulse modulations mentioned. As shown in Fig. 6.13, PCM basically is a tool for converting an analog signal into a digital signal (A/D conversion). An **analog** signal is characterized by an amplitude that can take on any value over a continuous range. This means that it can take on an infinite number of values. On the other hand, **digital** signal amplitude can take on only a finite number of values. An analog signal can

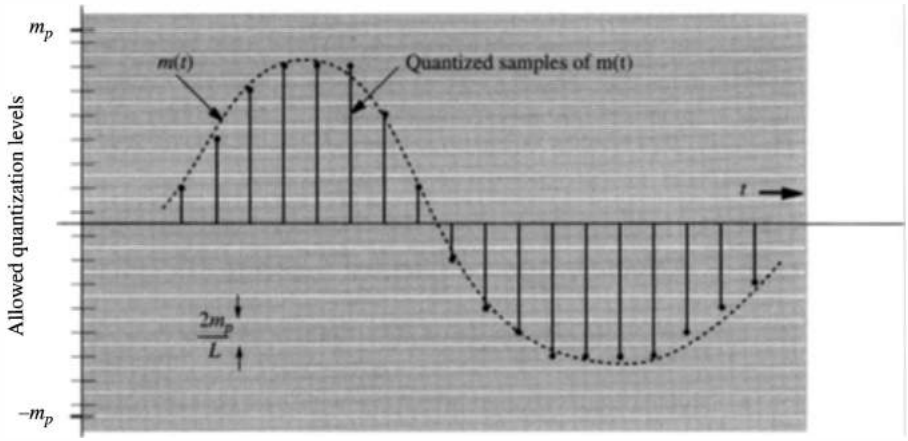


Figure 6.14 Quantization of a sampled analog signal.

be converted into a digital signal by means of sampling and **quantizing**, that is, rounding off its value to one of the closest permissible numbers (or **quantized levels**), as shown in Fig. 6.14. The amplitudes of the analog signal $m(t)$ lie in the range $(-m_p, m_p)$, which is partitioned into L subintervals, each of magnitude $\Delta v = 2m_p/L$. Next, each sample amplitude is approximated by the midpoint value of the subinterval in which the sample falls (see Fig. 6.14 for $L = 16$). Each sample is now approximated to one of the L numbers. Thus, the signal is digitized, with quantized samples taking on any one of the L values. Such a signal is known as an **L-ary digital signal**.

From practical viewpoint, a binary digital signal (a signal that can take on only two values) is very desirable because of its simplicity, economy, and ease of engineering. We can convert an L-ary signal into a binary signal by using pulse coding. Such a coding for the case of $L = 16$ was shown in Fig. 1.5. This code, formed by binary representation of the 16 decimal digits from 0 to 15, is known as the **natural binary code (NBC)**. Other possible ways of assigning a binary code will be discussed later. Each of the 16 levels to be transmitted is assigned one binary code of four digits. The analog signal $m(t)$ is now converted to a (binary) digital signal. A **bi**nary digi**t** is called a **bit** for convenience. This contraction of "binary digit" to "bit" has become an industry standard abbreviation and is used throughout the book.

Thus, each sample in this example is encoded by four bits. To transmit this binary data, we need to assign a distinct pulse shape to each of the two bits. One possible way is to assign a negative pulse to a binary **0** and a positive pulse to a binary **1** (Fig. 1.5) so that each sample is now transmitted by a group of four binary pulses (pulse code). The resulting signal is a binary signal.

The audio signal bandwidth is about 15 kHz. However, for speech, subjective tests show that signal articulation (intelligibility) is not affected if all the components above 3400 Hz are suppressed.*[3] Since the objective in telephone communication is intelligibility rather than high fidelity, the components above 3400 Hz are eliminated by a low-pass filter. The resulting signal is then sampled at a rate of 8000 samples per second (8 kHz). This rate is intentionally kept higher than the Nyquist sampling rate of 6.8 kHz so that realizable filters can be applied for signal reconstruction. Each sample is finally quantized into 256 levels ($L = 256$), which requires a group of eight binary pulses to encode each sample ($2^8 = 256$). Thus, a telephone signal requires $8 \times 8000 = 64,000$ binary pulses per second.

* Components below 300 Hz may also be suppressed without affecting the articulation.

The compact disc (CD) is a more recent application of PCM. This is a high-fidelity situation requiring the audio signal bandwidth to be 20 kHz. Although the Nyquist sampling rate is only 40 kHz, the actual sampling rate of 44.1 kHz is used for the reason mentioned earlier. The signal is quantized into a rather large number ($L = 65,536$) of quantization levels, each of which is represented by 16 bits to reduce the quantizing error. The binary-coded samples (1.4 million bit/s) are then recorded on the compact disc.

6.2.1 Advantages of Digital Communication

Here are some of the advantages of digital communication over analog communication.

1. Digital communication, which can withstand channel noise and distortion much better than analog as long as the noise and the distortion are within limits, is more rugged than analog communication. With analog messages, on the other hand, any distortion or noise, no matter how small, will distort the received signal.

2. The greatest advantage of digital communication over analog communication, however, is the viability of regenerative repeaters in the former. In an analog communication system, a message signal becomes progressively weaker as it travels along the channel, whereas the cumulative channel noise and the signal distortion grow progressively stronger. Ultimately the signal is overwhelmed by noise and distortion. Amplification offers little help because it enhances the signal and the noise by the same proportion. Consequently, the distance over which an analog message can be transmitted is limited by the initial transmission power. For digital communications, a long transmission path may also lead to overwhelming noise and interferences. The trick, however, is to set up repeater stations along the transmission path at distances short enough to be able to detect signal pulses before the noise and distortion have a chance to accumulate sufficiently. At each repeater station the pulses are detected, and new, clean pulses are transmitted to the next repeater station, which, in turn, duplicates the same process. If the noise and distortion are within limits (which is possible because of the closely spaced repeaters), pulses can be detected correctly.* This way the digital messages can be transmitted over longer distances with greater reliability. The most significant error in PCM comes from quantizing. This error can be reduced as much as desired by increasing the number of quantizing levels, the price of which is paid in an increased bandwidth of the transmission medium (channel).

3. Digital hardware implementation is flexible and permits the use of microprocessors, digital switching, and large-scale integrated circuits.

4. Digital signals can be coded to yield extremely low error rates and high fidelity as well as for privacy.

5. It is easier and more efficient to multiplex several digital signals.

6. Digital communication is inherently more efficient than analog in exchanging SNR for bandwidth.

7. Digital signal storage is relatively easy and inexpensive. It also has the ability to search and select information from distant electronic database.

8. Reproduction with digital messages can be extremely reliable without deterioration. Analog messages such as photocopies and films, for example, lose quality at each successive stage of reproduction and must be transported physically from one distant place to another, often at relatively high cost.

---

* The error in pulse detection can be made negligible.

9. The cost of digital hardware continues to halve every two or three years, while performance or capacity doubles over the same time period. And there is no end in sight yet to this breathtaking and relentless exponential progress in digital technology. As a result, digital technologies today dominate in any given area of communication or storage technologies.

A Historical Note

The ancient Indian writer Pingala applied what turns out to be advanced mathematical concepts for describing prosody, and in doing so presented the first known description of a binary numeral system, possibly as early as the eighth century BCE.[6] Others, like R. Hall in Mathematics of Poetry place him later, circa 200 BCE. Gottfried Wilhelm Leibniz (1646–1716) was the first mathematician in the West to work out systematically the binary representation (using 1s and 0s) for any number. He felt a spiritual significance in this discovery, believing that **1**, representing unity, was clearly a symbol for God, while **0** represented nothingness. He reasoned that if all numbers can be represented merely by the use of **1** and **0**, this surely proves that God created the universe out of nothing!

6.2.2 Quantizing

As mentioned earlier, digital signals come from a variety of sources. Some sources such as computers are inherently digital. Some sources are analog, but are converted into digital form by a variety of techniques such as PCM and delta modulation (DM), which will now be analyzed. The rest of this section provides quantitative discussion of PCM and its various aspects, such as quantizing, encoding, synchronizing, the required transmission bandwidth and SNR.

For quantization, we limit the amplitude of the message signal $m(t)$ to the range $(-m_p, m_p)$, as shown in Fig. 6.14. Note that m_p is not necessarily the peak amplitude of $m(t)$. The amplitudes of $m(t)$ beyond $\pm m_p$ are simply chopped off. Thus, m_p is not a parameter of the signal $m(t)$; rather, it is the limit of the quantizer. The amplitude range $(-m_p, m_p)$ is divided into L uniformly spaced intervals, each of width $\Delta v = 2m_p/L$. A sample value is approximated by the midpoint of the interval in which it lies (Fig. 6.14). The quantized samples are coded and transmitted as binary pulses. At the receiver some pulses may be detected incorrectly. Hence, there are two sources of error in this scheme: *quantization error* and *pulse detection error*. In almost all practical schemes, the pulse detection error is quite small compared to the quantization error and can be ignored. In the present analysis, therefore, we shall assume that the error in the received signal is caused exclusively by quantization.

If $m(kT_s)$ is the kth sample of the signal $m(t)$, and if $\hat{m}(kT_s)$ is the corresponding quantized sample, then from the interpolation formula in Eq. (6.10),

$$m(t) = \sum_k m(kT_s) \text{ sinc } (2\pi Bt - k\pi)$$

and

$$\hat{m}(t) = \sum_k \hat{m}(kT_s) \text{ sinc } (2\pi Bt - k\pi)$$

where $\hat{m}(t)$ is the signal reconstructed from quantized samples. The distortion component $q(t)$ in the reconstructed signal is $q(t) = \hat{m}(t) - m(t)$. Thus,

$$\begin{aligned} q(t) &= \sum_k [\hat{m}(kT_s) - m(kT_s)] \operatorname{sinc}(2\pi Bt - k\pi) \\ &= \sum_k q(kT_s) \operatorname{sinc}(2\pi Bt - k\pi) \end{aligned}$$

where $q(kT_s)$ is the quantization error in the kth sample. The signal $q(t)$ is the undesired signal, and, hence, acts as noise, known as **quantization noise**. To calculate the power, or the mean square value of $q(t)$, we have

$$\begin{aligned} \overbrace{q^2(t)} &= \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} q^2(t)\, dt \\ &= \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} \left[\sum_k q(kT_s) \operatorname{sinc}(2\pi Bt - k\pi) \right]^2 dt \end{aligned} \tag{6.29a}$$

We can show that (see Prob. 3.7-4) the signals $\operatorname{sinc}(2\pi Bt - m\pi)$ and $\operatorname{sinc}(2\pi Bt - n\pi)$ are orthogonal, that is,

$$\int_{-\infty}^{\infty} \operatorname{sinc}(2\pi Bt - m\pi) \operatorname{sinc}(2\pi Bt - n\pi)\, dt = \begin{cases} 0 & m \neq n \\ \dfrac{1}{2B} & m = n \end{cases} \tag{6.29b}$$

Because of this result, the integrals of the cross-product terms on the right-hand side of Eq. (6.29a) vanish, and we obtain

$$\begin{aligned} \overbrace{q^2(t)} &= \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} \sum_k q^2(kT_s) \operatorname{sinc}^2(2\pi Bt - k\pi)\, dt \\ &= \lim_{T\to\infty} \frac{1}{T} \sum_k q^2(kT_s) \int_{-T/2}^{T/2} \operatorname{sinc}^2(2\pi Bt - k\pi)\, dt \end{aligned}$$

From the orthogonality relationship (6.29b), it follows that

$$\overbrace{q^2(t)} = \lim_{T\to\infty} \frac{1}{2BT} \sum_k q^2(kT_s) \tag{6.30}$$

Because the sampling rate is $2B$, the total number of samples over the averaging interval T is $2BT$. Hence, the right-hand side of Eq. (6.30) represents the average, or the mean of the square of the quantization error. The quantum levels are separated by $\Delta v = 2m_p/L$. Since a sample value is approximated by the midpoint of the subinterval (of height Δv) in which the sample falls, the maximum quantization error is $\pm\Delta v/2$. Thus, the quantization error lies in the range $(-\Delta v/2,\ \Delta v/2)$, where

$$\Delta v = \frac{2m_p}{L} \tag{6.31}$$

Assuming that the error is equally likely to lie anywhere in the range $(-\Delta v/2,\ \Delta v/2)$, the mean square quantizing error $\overbrace{q^2}$ is given by*

$$\overbrace{q^2} = \frac{1}{\Delta v}\int_{-\Delta v/2}^{\Delta v/2} q^2\, dq$$

$$= \frac{(\Delta v)^2}{12} \tag{6.32}$$

$$= \frac{m_p^2}{3L^2} \tag{6.33}$$

Because $\overbrace{q^2(t)}$ is the mean square value or power of the quantization noise, we shall denote it by N_q,

$$N_q = \overbrace{q^2(t)} = \frac{m_p^2}{3L^2}$$

Assuming that the pulse detection error at the receiver is negligible, the reconstructed signal $\hat{m}(t)$ at the receiver output is

$$\hat{m}(t) = m(t) + q(t)$$

The desired signal at the output is $m(t)$, and the (quantization) noise is $q(t)$. Since the power of the message signal $m(t)$ is $\overbrace{m^2(t)}$, then

$$S_o = \overbrace{m^2(t)}$$

$$N_o = N_q = \frac{m_p^2}{3L^2}$$

and

$$\frac{S_o}{N_o} = 3L^2\frac{\overbrace{m^2(t)}}{m_p^2} \tag{6.34}$$

In this equation, m_p is the peak amplitude value that a quantizer can accept, and is therefore a parameter of the quantizer. This means S_o/N_o, the SNR, is a linear function of the message signal power $\overbrace{m^2(t)}$ (see Fig. 6.18 with $\mu = 0$).

---

* Those who are familiar with the theory of probability can derive this result directly by noting that the probability density of the quantization error q is $1/(2m_p/L) = L/2m_p$ over the range $|q| \le m_p/L$ and is zero elsewhere. Hence,

$$\overbrace{q^2} = \int_{-m_p/L}^{m_p/L} q^2 p(q)\, dq = \int_{-m_p/L}^{m_p/L} \frac{L}{2m_p} q^2\, dq = \frac{m_p^2}{3L^2}$$

6.2.3 Principle of Progressive Taxation: Nonuniform Quantization

Recall that S_o/N_o, the SNR, is an indication of the quality of the received signal. Ideally we would like to have a constant SNR (the same quality) for all values of the message signal power $m^2(t)$. Unfortunately, the SNR is directly proportional to the signal power $m^2(t)$, which varies from speaker to speaker by as much as 40 dB (a power ratio of 10^4). The signal power can also vary because of the different lengths of the connecting circuits. This indicates that the SNR in Eq. (6.34) can vary widely, depending on the speaker and the length of the circuit. Even for the same speaker, the quality of the received signal will deteriorate markedly when the person speaks softly. Statistically, it is found that smaller amplitudes predominate in speech and larger amplitudes are much less frequent. This means the SNR will be low most of the time.

The root of this difficulty lies in the fact that the quantizing steps are of uniform value $\Delta v = 2m_p/L$. The quantization noise $N_q = (\Delta v)^2/12$ [Eq. (6.32)] is directly proportional to the square of the step size. The problem can be solved by using smaller steps for smaller amplitudes (nonuniform quantizing), as shown in Fig. 6.15a. The same result is obtained by first compressing signal samples and then using a uniform quantization. The input-output characteristics of a compressor are shown in Fig. 6.15b. The horizontal axis is the normalized input signal (i.e., the input signal amplitude m divided by the signal peak value m_p). The vertical axis is the output signal y. The compressor maps input signal increments Δm into larger increments Δy for small input signals, and vice versa for large input signals. Hence, a given interval Δm contains a larger number of steps (or smaller step size) when m is small. The quantization noise is lower for smaller input signal power. An approximately logarithmic compression characteristic yields a quantization noise nearly proportional to the signal power $m^2(t)$, thus making the SNR practically independent of the input signal power over a large dynamic range[5] (see later Fig. 6.18). This approach of equalizing the SNR appears similar to the use of progressive income tax to equalize incomes. The loud talkers and stronger signals are penalized with higher noise steps Δv to compensate the soft talkers and weaker signals.

Among several choices, two compression laws have been accepted as desirable standards by the ITU-T:[6] the μ-law used in North America and Japan, and the A-law used in Europe and the rest of the world and on international routes. Both the μ-law and the A-law curves have odd symmetry about the vertical axis. The μ-law (for positive amplitudes) is given by

$$y = \frac{1}{\ln(1+\mu)} \ln\left(1 + \frac{\mu m}{m_p}\right) \qquad 0 \le \frac{m}{m_p} \le 1 \tag{6.35a}$$

The A-law (for positive amplitudes) is

$$y = \begin{cases} \dfrac{A}{1+\ln A}\left(\dfrac{m}{m_p}\right) & 0 \le \dfrac{m}{m_p} \le \dfrac{1}{A} \\[2ex] \dfrac{1}{1+\ln A}\left(1 + \ln \dfrac{A m}{m_p}\right) & \dfrac{1}{A} \le \dfrac{m}{m_p} \le 1 \end{cases} \tag{6.35b}$$

These characteristics are shown in Fig. 6.16.

The compression parameter μ (or A) determines the degree of compression. To obtain a nearly constant S_o/N_o over a dynamic range of for input signal power 40 dB, μ should be greater than 100. Early North American channel banks and other digital terminals used a value of $\mu = 100$, which yielded the best results for 7-bit (128-level) encoding. An optimum value

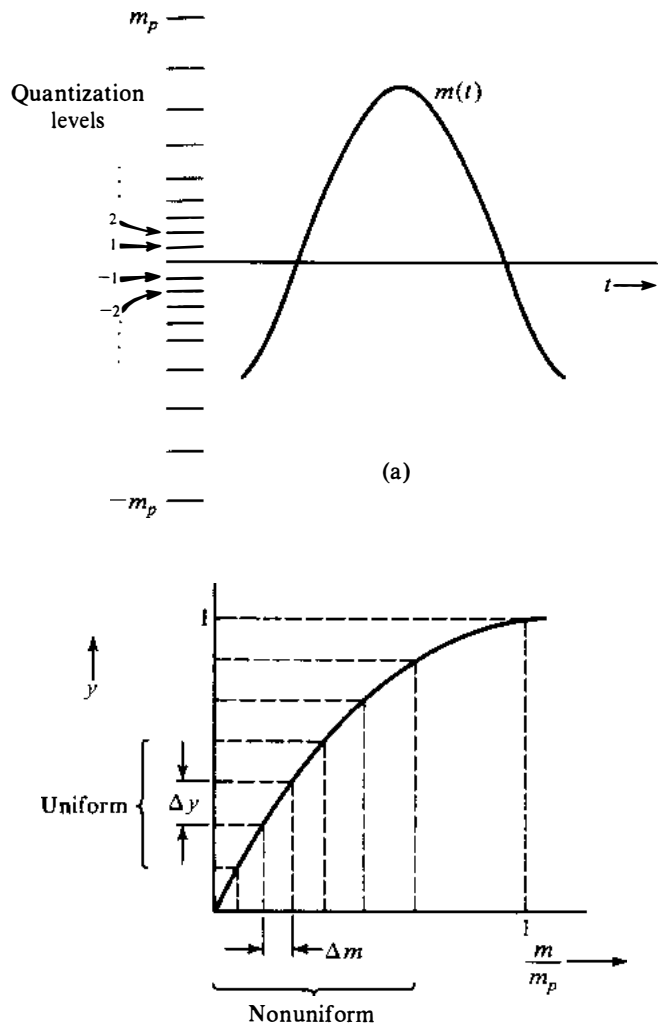


Figure 6.15 Nonuniform quantization.

of $\mu = 255$ has been used for all North American 8-bit (256-level) digital terminals, and the earlier value of μ is now almost extinct. For the A-law, a value of $A = 87.6$ gives comparable results and has been standardized by the ITU-T.[6]

The compressed samples must be restored to their original values at the receiver by using an expander with a characteristic complementary to that of the compressor. The compressor and the expander together are called the **compandor**. Figure 6.17 describes the use of compressor and expander along with a uniform quantizer to achieve nonuniform quantization.

Generally speaking, time compression of a signal increases its bandwidth. But in PCM, we are compressing not the signal $m(t)$ in time but its sample values. Because neither the time scale not the number of samples changes, the problem of bandwidth increase does not arise here. It happens that when a μ-law compandor is used, the output SNR is

$$\frac{S_o}{N_o} \simeq \frac{3L^2}{[\ln (1+\mu)]^2} \qquad \mu^2 \gg \frac{m_p^2}{m^2(t)} \tag{6.36}$$

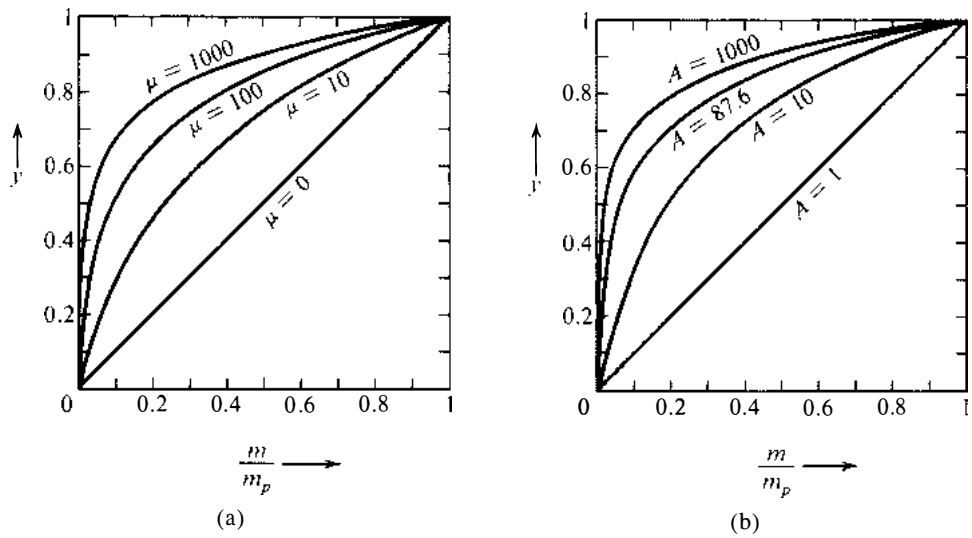


Figure 6.16 (a) μ-Law characteristic. (b) A-Law characteristic.

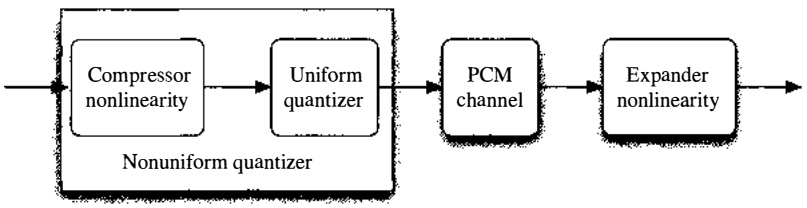


Figure 6.17 Utilization of compressor and expander for nonuniform quantization.

The output SNR for the cases of $\mu = 255$ and $\mu = 0$ (uniform quantization) as a function of $m^2(t)$ (the message signal power) is shown in Fig. 6.18.

The Compandor

A logarithmic compressor can be realized by a semiconductor diode, because the V–I characteristic of such a diode is of the desired form in the first quadrant:

$$V = \frac{KT}{q} \ln\left(1 + \frac{I}{I_s}\right)$$

Two matched diodes in parallel with opposite polarity provide the approximate characteristic in the first and third quadrants (ignoring the saturation current). In practice, adjustable resistors are placed in series with each diode and a third variable resistor is added in parallel. By adjusting various resistors, the resulting characteristic is made to fit a finite number of points (usually seven) on the ideal characteristics.

An alternative approach is to use a piecewise linear approximation to the logarithmic characteristics. A 15-segmented approximation (Fig. 6.19) to the eighth bit ($L = 256$) with $\mu = 255$ law is widely used in the D2 channel bank that is used in conjunction with the T1 carrier system. The segmented approximation is only marginally inferior in terms of SNR.[8] The piecewise linear approximation has almost universally replaced earlier logarithmic approximations to the true $\mu = 255$ characteristic and is the method of choice in North American standards.

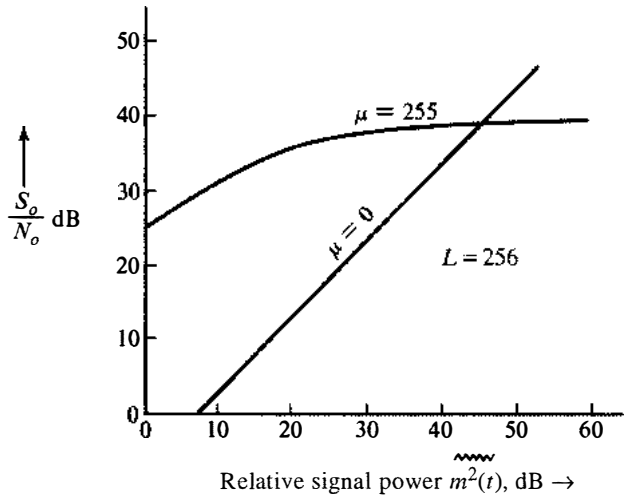


Figure 6.18
Ratio of signal to quantization noise in PCM with and without compression.

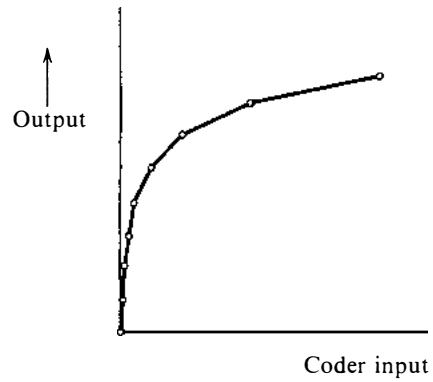


Figure 6.19
Piecewise linear compressor characteristic.

Though a true $\mu = 255$ compressor working with a $\mu = 255$ expander will be superior to similar piecewise linear devices, a digital terminal device exhibiting the true characteristic in today's network must work end-to-end against other network elements that use the piecewise linear approximation. Such a combination of differing characteristics is inferior to either of the characteristics obtained when the compressor and the expander operate using the same compression law.

In the standard audio file format used by Sun, Unix and Java, the audio in "au" files can be pulse-code-modulated or compressed with the ITU-T G.711 standard through either the μ-law or the A-law.[6] The μ-law compressor ($\mu = 255$) converts 14-bit signed linear PCM samples to logarithmic 8-bit samples, leading to storage saving. The A-law compressor ($A = 87.6$) converts 13-bit signed linear PCM samples to logarithmic 8-bit samples. In both cases, sampling at the rate of 8000 Hz, a G.77 encoder thus creates from audio signals bit streams at 64 kilobits per second (kbit/s). Since the A-law and the μ-law are mutually compatible, audio recoded into "au" files can be decoded in either format. It should be noted that the Microsoft WAV audio format also has compression options that use μ-law and A-law.

The PCM Encoder

The multiplexed PAM output is applied at the input of the encoder, which quantizes and encodes each sample into a group of n binary digits. A variety of encoders is available.[7, 10] We shall discuss here the **digit-at-a-time** encoder, which makes n sequential comparisons to generate an n-bit codeword. The sample is compared with a voltage obtained by a combination of reference voltages proportional to 2^7, 2^6, $2^5, \ldots,$ 2^0. The reference voltages are conveniently generated by a bank of resistors R, $2R$, 2^2R, $\ldots$, 2^7R.

The encoding involves answering successive questions, beginning with whether the sample is in the upper or lower half of the allowed range. The first code digit **1** or **0** is generated, depending on whether the sample is in the upper or the lower half of the range. In the second step, another digit **1** or **0** is generated, depending on whether the sample is in the upper or the lower half of the subinterval in which it has been located. This process continues until the last binary digit in the code has been generated.

Decoding is the inverse of encoding. In this case, each of the n digits is applied to a resistor of different value. The kth digit is applied to a resistor 2^kR. The currents in all the resistors are added. The sum is proportional to the quantized sample value. For example, a binary code word **10010110** will give a current proportional to $2^7 + 0 + 0 + 2^4 + 0 + 2^2 + 2^1 + 0 = 150$. This completes the D/A conversion.

6.2.4 Transmission Bandwidth and the Output SNR

For a binary PCM, we assign a distinct group of n binary digits (bits) to each of the L quantization levels. Because a sequence of n binary digits can be arranged in 2^n distinct patterns,

$$L = 2^n \qquad \text{or} \qquad n = \log_2 L \tag{6.37}$$

each quantized sample is, thus, encoded into n bits. Because a signal $m(t)$ band-limited to B Hz requires a minimum of $2B$ samples per second, we require a total of $2nB$ bit/s, that is, $2nB$ pieces of information per second. Because a unit bandwidth (1 Hz) can transmit a maximum of two pieces of information per second (Sec. 6.1.3), we require a minimum channel of bandwidth B_T Hz, given by

$$B_T = nB \text{ Hz} \tag{6.38}$$

This is the theoretical minimum transmission bandwidth required to transmit the PCM signal. In Secs. 7.2 and 7.3, we shall see that for practical reasons we may use a transmission bandwidth higher than this minimum.

Example 6.2 A signal $m(t)$ band-limited to 3 kHz is sampled at a rate $33\frac{1}{3}\%$ higher than the Nyquist rate. The maximum acceptable error in the sample amplitude (the maximum quantization error) is 0.5% of the peak amplitude m_p. The quantized samples are binary coded. Find the minimum bandwidth of a channel required to transmit the encoded binary signal. If 24 such signals are time-division-multiplexed, determine the minimum transmission bandwidth required to transmit the multiplexed signal.

The Nyquist sampling rate is $R_N = 2 \times 3000 = 6000$ Hz (samples per second). The actual sampling rate is $R_A = 6000 \times (1\frac{1}{3}) = 8000$ Hz.

The quantization step is Δv, and the maximum quantization error is $\pm\Delta v/2$.

Therefore, from Eq. (6.31),

$$\frac{\Delta v}{2} = \frac{m_p}{L} = \frac{0.5}{100} m_p \Longrightarrow L = 200$$

For binary coding, L must be a power of 2. Hence, the next higher value of L that is a power of 2 is $L = 256$.

From Eq. (6.37), we need $n = \log_2 256 = 8$ bits per sample. We require to transmit a total of $C = 8 \times 8000 = 64,000$ bit/s. Because we can transmit up to 2 bit/s per hertz of bandwidth, we require a minimum transmission bandwidth $B_T = C/2 = 32$ kHz.

The multiplexed signal has a total of $C_M = 24 \times 64,000 = 1.536$ Mbit/s, which requires a minimum of $1.536/2 = 0.768$ MHz of transmission bandwidth.

Exponential Increase of the Output SNR

From Eq. (6.37), $L^2 = 2^{2n}$, and the output SNR in Eq. (6.34) or Eq. (6.36) can be expressed as

$$\frac{S_o}{N_o} = c(2)^{2n} \tag{6.39}$$

where

$$c = \begin{cases} \dfrac{3\ \overbrace{m^2(t)}}{m_p^2} & \text{[uncompressed case, in Eq. (6.34)]} \\ \dfrac{3}{[\ln(1+\mu)]^2} & \text{[compressed case, in Eq. (6.36)]} \end{cases}$$

Substitution of Eq. (6.38) into Eq. (6.39) yields

$$\frac{S_o}{N_o} = c(2)^{2B_T/B} \tag{6.40}$$

From Eq. (6.40) we observe that the SNR increases exponentially with the transmission bandwidth B_T. This trade of SNR for bandwidth is attractive and comes close to the upper theoretical limit. A small increase in bandwidth yields a large benefit in terms of SNR. This relationship is clearly seen by using the decibel scale to rewrite Eq. (6.39) as

$$\begin{aligned} \left(\frac{S_o}{N_o}\right)_{\text{dB}} &= 10\log_{10}\left(\frac{S_o}{N_o}\right) \\ &= 10\log_{10}[c(2)^{2n}] \\ &= 10\log_{10} c + 2n\log_{10} 2 \\ &= (\alpha + 6n) \quad \text{dB} \end{aligned} \tag{6.41}$$

where $\alpha = 10 \log_{10} c$. This shows that increasing n by 1 (increasing one bit in the codeword) quadruples the output SNR (a 6 dB increase). Thus, if we increase n from 8 to 9, the SNR quadruples, but the transmission bandwidth increases only from 32 kHz to 36 kHz (an increase of only 12.5%). This shows that in PCM, SNR can be controlled by transmission bandwidth. We shall see later that frequency and phase modulation also do this. But it requires a doubling of the bandwidth to quadruple the SNR. In this respect, PCM is strikingly superior to FM or PM.

Example 6.3 A signal $m(t)$ of bandwidth $B = 4$ kHz is transmitted using a binary companded PCM with $\mu = 100$. Compare the case of $L = 64$ with the case of $L = 256$ from the point of view of transmission bandwidth and the output SNR.

For $L = 64$, $n = 6$, and the transmission bandwidth is $nB = 24$ kHz,

$$\frac{S_o}{N_o} = (\alpha + 36) \text{ dB}$$

$$\alpha = 10 \log \frac{3}{[\ln (101)]^2} = -8.51$$

Hence,

$$\frac{S_o}{N_o} = 27.49 \text{ dB}$$

For $L = 256$, $n = 8$, and the transmission bandwidth is 32 kHz,

$$\frac{S_o}{N_o} = \alpha + 6n = 39.49 \text{ dB}$$

The difference between the two SNRs is 12 dB, which is a ratio of 16. Thus, the SNR for $L = 256$ is 16 times the SNR for $L = 64$. The former requires just about 33% more bandwidth compared to the latter.

Comments on Logarithmic Units

Logarithmic units and logarithmic scales are very convenient when a variable has a large dynamic range. Such is the case with frequency variables or SNRs. A logarithmic unit for the power ratio is the decibel (dB), defined as $10 \log_{10}$ (power ratio). Thus, an SNR is x dB, where

$$x = 10 \log_{10} \frac{S}{N}$$

We use the same unit to express power gain or loss over a certain transmission medium. For instance, if over a certain cable the signal power is attenuated by a factor of 15, the cable gain is

$$G = 10 \log_{10} \frac{1}{15} = -11.76 \text{ dB}$$

or the cable attenuation (loss) is 11.76 dB.

Although the decibel is a measure of power ratios, it is often used as a measure of power itself. For instance, "100 watt" may be considered to be a power ratio of 100 with respect to 1-watt power, and is expressed in units of dBW as

$$P_{\text{dBW}} = 10 \log_{10} 100 = 20 \text{ dBW}$$

Thus, 100-watt power is 20 dBW. Similarly, power measured with respect to 1 mW power is dBm. For instance, 100-watt power is

$$P_{\text{dBm}} = 10 \log \frac{100\,\text{W}}{1\,\text{mW}} = 50\,\text{dBm}$$

6.3 DIGITAL TELEPHONY: PCM IN T1 CARRIER SYSTEMS

A Historical Note

Because of the unavailability of suitable switching devices, more than 20 years elapsed between the invention of PCM and its implementation. Vacuum tubes, used before the invention of the transistor, were not only bulky, but they were poor switches and dissipated a lot of heat. Systems having vacuum tubes as switches were large, rather unreliable, and tended to overheat. PCM was just waiting for the invention of the transistor, which happens to be a small device that consumes little power and is a nearly ideal switch.

Coincidentally, at about the time the transistor was invented, the demand for telephone service had become so heavy that the existing system was overloaded, particularly in large cities. It was not easy to install new underground cables because space available under the streets in many cities was already occupied by other services (water, gas, sewer, etc.). Moreover, digging up streets and causing many dislocations was not very attractive. An attempt was made on a limited scale to increase the capacity by frequency-division-multiplexing several voice channels through amplitude modulation. Unfortunately, the cables were primarily designed for the audio voice range (0–4 kHz) and suffered severely from noise. Furthermore, cross talk between pairs of channels on the same cable was unacceptable at high frequencies. Ironically, PCM—requiring a bandwidth several times larger than that required for FDM signals—offered the solution. This is because digital systems with closely spaced regenerative repeaters can work satisfactorily on noisy lines that give poor high-frequency performance.[9] The repeaters, spaced approximately 6000 feet apart, clean up the signal and regenerate new pulses before the pulses get too distorted and noisy. This is the history of the Bell System's T1 carrier system.[3, 10] A pair of wires that used to transmit one audio signal of bandwidth 4 kHz is now used to transmit 24 time-division-multiplexed PCM telephone signals with a total bandwidth of 1.544 MHz.

T1 Time Division Multiplexing

A schematic of a T1 carrier system is shown in Fig. 6.20a. All 24 channels are sampled in a sequence. The sampler output represents a time-division-multiplexed PAM signal. The multiplexed PAM signal is now applied to the input of an encoder that quantizes each sample and encodes it into eight binary pulses—a binary codeword* (see Fig. 6.20b). The signal, now converted to digital form, is sent over the transmission medium. Regenerative repeaters spaced approximately 6000 feet apart detect the pulses and retransmit new pulses. At the receiver, the decoder converts the binary pulses into samples (decoding). The samples are then demultiplexed (i.e., distributed to each of the 24 channels). The desired audio signal is reconstructed by passing the samples through a low-pass filter in each channel.

* In an earlier version, each sample was encoded by seven bits. An additional bit was added for signaling.

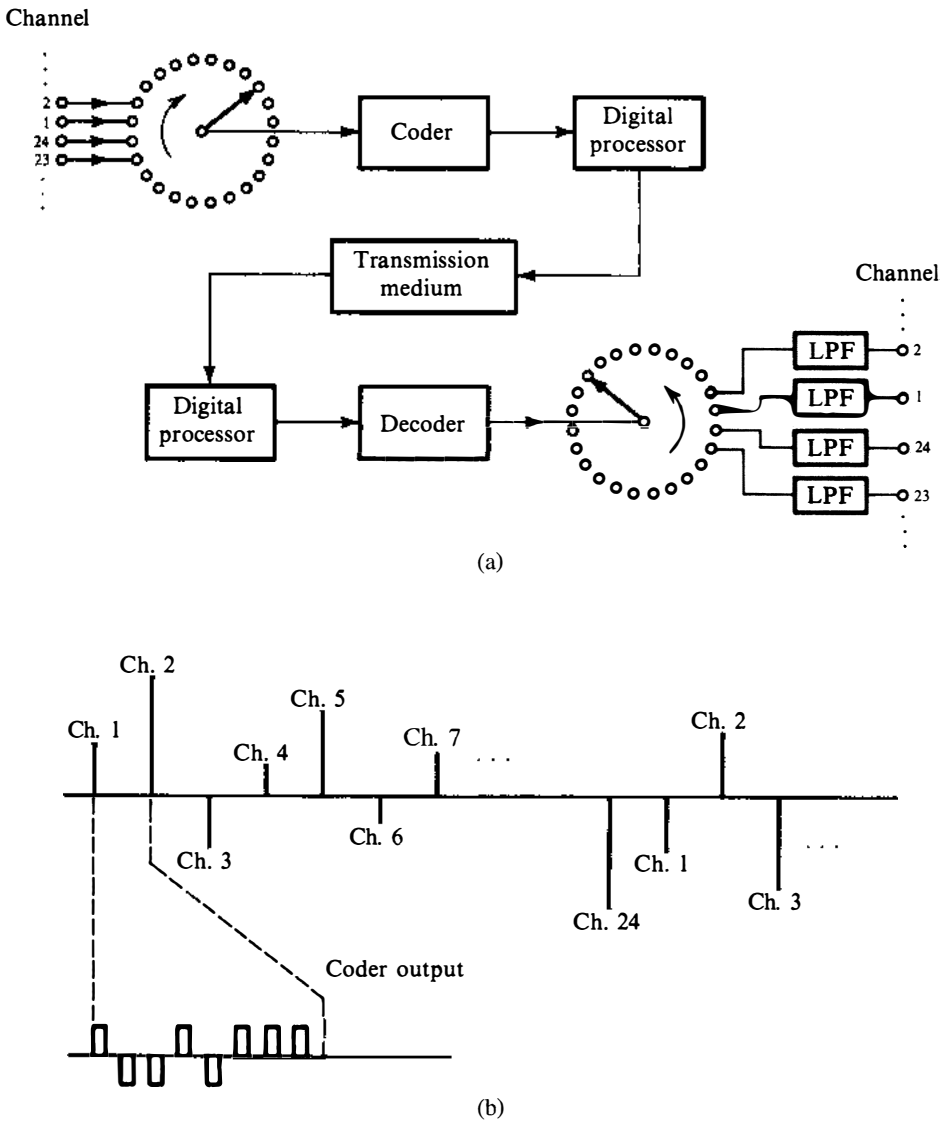


Figure 6.20 T1 carrier system.

The commutators in Fig. 6.20 are not mechanical but are high-speed electronic switching circuits. Several schemes are available for this purpose.[11] Sampling is done by electronic gates (such as a bridge diode circuit, as shown in Fig. 4.5a) opened periodically by narrow pulses of 2 μs duration. The 1.544 Mbit/s signal of the T1 system, called **digital signal level 1 (DS1)**, is used further to multiplex into progressively higher level signals DS2, DS3, and DS4, as described next, in Sec. 6.4

After the Bell System introduced the T1 carrier system in the United States, dozens of variations were proposed or adopted elsewhere before the ITU-T standardized its 30-channel PCM system with a rate of 2.048 Mbit/s (in contrast to T1, with 24 channels and 1.544 Mbit/s). The 30-channel system is used all over the world, except in North America and Japan. Because of the widespread adoption of the T1 carrier system in the United States and Japan before the ITU-T standardization, the two standards continue to be used in different parts of the world, with appropriate interfaces in international connections.

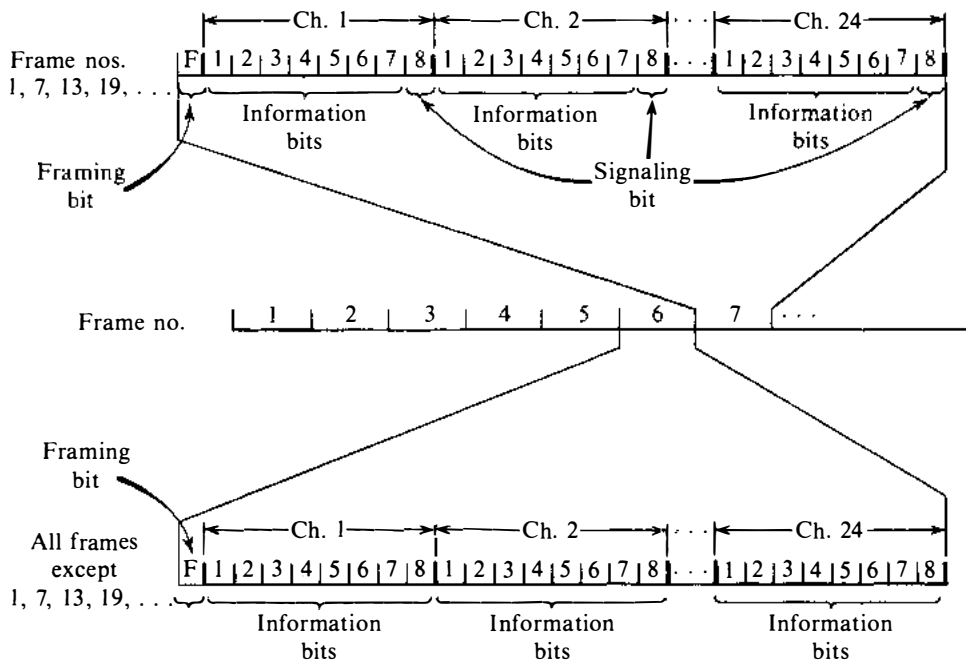


Figure 6.21
T1 system signaling format.

Synchronizing and Signaling

Binary codewords corresponding to samples of each of the 24 channels are multiplexed in a sequence, as shown in Fig. 6.21. A segment containing one codeword (corresponding to one sample) from each of the 24 channels is called a **frame**. Each frame has $24 \times 8 = 192$ information bits. Because the sampling rate is 8000 samples per second, each frame takes 125 μs. To separate information bits correctly at the receiver, it is necessary to be sure where each frame begins. Therefore, a **framing bit** is added at the beginning of each frame. This makes a total of 193 bits per frame. Framing bits are chosen so that a sequence of framing bits, one at the beginning of each frame, forms a special pattern that is unlikely to be formed in a speech signal.

The sequence formed by the first bit from each frame is examined by the logic of the receiving terminal. If this sequence does not follow the given code pattern (framing bit pattern), a synchronization loss is detected, and the next position is examined to determine whether it is actually the framing bit. It takes about 0.4 to 6 ms to detect and about 50 ms (in the worst possible case) to reframe.

In addition to information and framing bits, we need to transmit signaling bits corresponding to dialing pulses, as well as telephone on-hook/off-hook signals. When channels developed by this system are used to transmit signals between telephone switching systems, the switches must be able to communicate with each other to use the channels effectively. Since all eight bits are now used for transmission instead of the seven bits used in the earlier version,* the signaling channel provided by the eighth bit is no longer available. Since only a rather low-speed signaling channel is required, rather than create extra time slots for this information, we use one information bit (the least significant bit) of every sixth sample of a signal

* In the earlier version of T1, quantizing levels $L = 128$ required only seven information bits. The eighth bit was used for signaling.

to transmit this information. This means that every sixth sample of each voice signal will have a possible error corresponding to the least significant digit. Every sixth frame, therefore, has $7 \times 24 = 168$ information bits, 24 signaling bits, and 1 framing bit. In all the remaining frames, there are 192 information bits and 1 framing bit. This technique is called $\mathbf{7\frac{5}{6}}$ bit encoding, and the signaling channel so derived is called **robbed-bit signaling**. The slight SNR degradation suffered by impairing one out of six frames is considered to be an acceptable penalty. The signaling bits for each signal occur at a rate of $8000/6 = 1333$ bit/s. The frame format is shown in Fig. 6.21.

The older seven-bit framing format required only that frame boundaries be identified so that the channels could be located in the bit stream. When signaling is superimposed on the channels in every sixth frame, it is necessary to identify, at the receiver, which frames are the signaling frames. A new framing structure, called the **superframe**, was developed to take care of this. The framing bits are transmitted at 8 kbit/s as before and occupy the first bit of each frame. The framing bits form a special pattern, which repeats in 12 frames: **100011011100**. The pattern thus allows the identification of frame boundaries as before, but also allows the determination of the locations of the sixth and twelfth frames within the superframe. Note that the superframe described here is 12 frames in length. Since two bits per superframe are available for signaling for each channel, it is possible to provide four-state signaling for a channel by using the four possible patterns of the two signaling bits: **00, 01, 10,** and **11**. Although most switch-to-switch applications in the telephone network require only two-state signaling, three- and four-state signaling techniques are used in certain special applications.

Advances in digital electronics and in coding theory have made it unnecessary to use the full 8 kbit/s of the framing channel in a DS1 signal to perform the framing task. A new superframe structure, called the **extended superframe (ESF)** format, was introduced during the 1970s to take advantage of the reduced framing bandwidth requirement. An ESF is 24 frames in length and carries signaling bits in the eighth bit of each channel in frames 6, 12, 18, and 24. Sixteen-state signaling is thus possible and is sometimes used although, as with the superframe format, most applications require only two-state signaling.

The 8 kbit/s overhead (framing) capacity of the ESF signal is divided into three channels: 2 kbit/s for framing, 2 kbit/s for a cyclic redundancy check (CRC-6) error detection channel, and 4 kbit/s for a data channel. The highly reliable error checking provided by the CRC-6 pattern and the use of the data channel to transport information on signal performance as received by the distant terminal make ESF much more attractive to service providers than the older superframe format. More discussions on CRC error detection can be found in Chapter 14.

The 2 kbit/s framing channel of the ESF format carries the repetitive pattern **001011** . . . , a pattern that repeats in 24 frames and is much less vulnerable to counterfeiting than the patterns associated with the earlier formats.

For various reasons, including the development of intelligent network-switching nodes, the function of signaling is being transferred out from the channels that carry the messages or data signals to separate signaling networks called **common channel interoffice signaling (CCIS)** systems. The universal deployment of such systems will significantly decrease the importance of robbed-bit signaling, and all eight bits of each message (or sample) will be transmitted in most applications.

The Conference on European Postal and Telegraph Administration (CEPT) has standardized a PCM with 256 time slots per frame. Each frame has $30 \times 8 = 240$ information bits, corresponding to 30 speech channels (with eight bits each). The remaining 16 bits per frame are used for frame synchronization and signaling. Therefore, although the bit rate is 2.048 Mbit/s, corresponding to 32 voice channels, only 30 voice channels are transmitted.

6.4 DIGITAL MULTIPLEXING

Several low-bit-rate signals can be multiplexed, or combined, to form one high-bit-rate signal, to be transmitted over a high-frequency medium. Because the medium is time-shared by various incoming signals, this is a case of TDM (time division multiplexing). The signals from various incoming channels, or tributaries, may be as diverse as a digitized voice signal (PCM), a computer output, telemetry data, and a digital facsimile. The bit rates of various tributaries need not be the same.

To begin with, consider the case of all tributaries with identical bit rates. Multiplexing can be done on a bit-by-bit basis (known as bit or **digit interleaving**) as shown in Fig. 6.22a, or on a word-by-word basis (known as byte or **word interleaving**). Figure 6.22b shows the interleaving of words, formed by four bits. The North American digital hierarchy uses bit interleaving (except at the lowest level), where bits are taken one at a time from the various signals to be multiplexed. Byte interleaving, used in building the DS1 signal and SONET-formatted signals, involves inserting bytes in succession from the channels to be multiplexed.

The T1 carrier, discussed in Sec. 6.3, uses eight-bit word interleaving. When the bit rates of incoming channels are not identical, the high-bit-rate channel is allocated proportionately more slots. Four-channel multiplexing consists of three channels B, C, and D of identical bit rate R and one channel (channel A) with a bit rate of $3R$. (Fig. 6.22c,d). Similar results can be attained by combining words of different lengths. It is evident that the minimum length of the multiplex frame must be a multiple of the lowest common multiple of the incoming channel bit rates, and, hence, this type of scheme is practical only when some fairly simple relationship exists among these rates. The case of completely asynchronous channels is discussed later.

At the receiving terminal, the incoming digit stream must be divided and distributed to the appropriate output channel. For this purpose, the receiving terminal must be able to correctly identify each bit. This requires the receiving system to uniquely synchronize in time with the beginning of each frame, with each slot in a frame, and with each bit within a slot. This is accomplished by adding framing and synchronization bits to the data bits. These bits are part of the so-called **overhead bits.**

6.4.1 Signal Format

Figure 6.23 illustrates a typical format, that of the DM1/2 multiplexer. We have here bit-by-bit interleaving of four channels each at a rate of 1.544 Mbit/s. The main frame (multiframe) consists of four subframes. Each subframe has six overhead bits: for example the subframe 1 (first line in Fig. 6.23) has overhead bits M_0, C_A, F_0, C_A, C_A, and F_1. In between these overhead bits are 48 interleaved data bits from the four channels (12 data bits from each channel). We begin with overhead bit M_0, followed by 48 multiplexed data bits, then add a second overhead bit C_A followed by the next 48 multiplexed bits, and so on. Thus, there are a total of $48 \times 6 \times 4 = 1152$ data bits and $6 \times 4 = 24$ overhead bits making a total 1176 bits/frame. The efficiency is $1152/1176 \simeq 98\%$. The overhead bits with subscript 0 are always **0** and those with subscript 1 are always **1**. Thus, M_0, F_0 are all **0**s and M_1 and F_1 are all **1**s. The F digits are periodic **010101** ... and provide the main framing pattern, which the multiplexer uses to synchronize on the frame. After locking onto this pattern, the demultiplexer searches for the **0111** pattern formed by overhead bits $M_0M_1M_1M_1$. This further identifies the four subframes, each corresponding to a line in Fig. 6.23. It is possible, although unlikely, that signal bits will also have a pattern **101010**. . . . The receiver could lock onto this

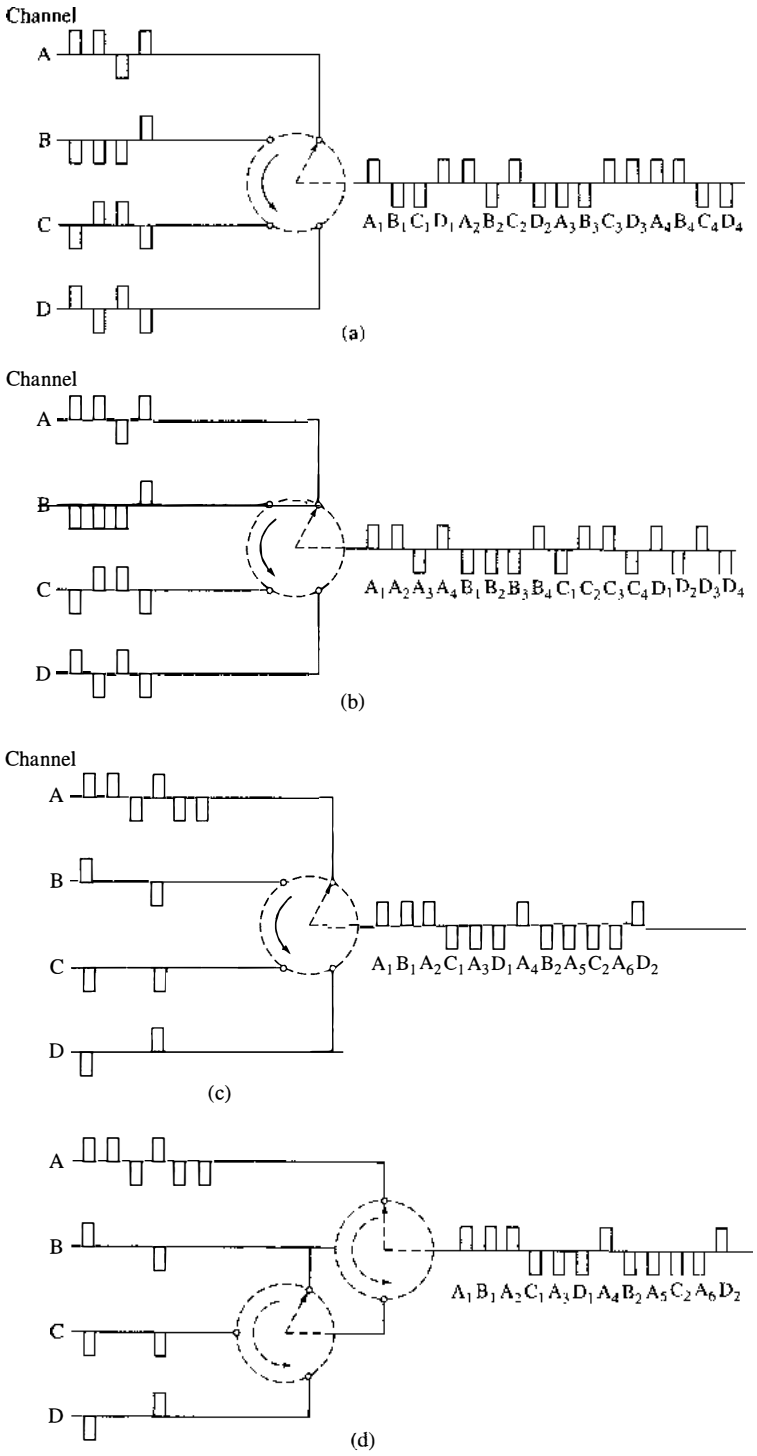


Figure 6.22 Time division multiplexing of digital signals: (a) digit interleaving; (b) word (or byte) interleaving; (c) interleaving channel having different bit rate; (d) alternate scheme for (c).

Figure 6.23 DM1/2 multiplexer format.

M_0	[48]	C_A	[48]	F_0	[48]	C_A	[48]	C_A	[48]	F_1	[48]
M_1	[48]	C_B	[48]	F_0	[48]	C_B	[48]	C_B	[48]	F_1	[48]
M_1	[48]	C_C	[48]	F_0	[48]	C_C	[48]	C_C	[48]	F_1	[48]
M_1	[48]	C_D	[48]	F_0	[48]	C_D	[48]	C_D	[48]	F_1	[48]

wrong sequence. The presence of $M_0M_1M_1M_1$ provides verification of the genuine $F_0F_1F_0F_1$ sequence. The C bits are used to transmit additional information about bit stuffing, as discussed later.

In the majority of cases, not all incoming channels are active all the time: some transmit data, and some are idle. This means the system is underutilized. We can, therefore, accept more input channels to take advantage of the inactivity, at any given time, of at least one channel. This obviously involves much more complicated switching operations, and also rather careful system planning. In any random traffic situation we cannot guarantee that the number of transmission channels demanded will not exceed the number available; but by taking account of the statistics of the signal sources, it is possible to ensure an acceptably low probability of this occurring. Multiplex structures of this type have been developed for satellite systems and are known as **time division multiple-access (TDMA) systems**.

In TDMA systems employed for telephony, the design parameters are chosen so that any overload condition lasts only a fraction of a second, which leads to acceptable performance for speech communication. For other types of data and telegraphy, transmission delays are unimportant. Hence, in overload condition, the incoming data can be stored and transmitted later.

6.4.2 Asynchronous Channels and Bit Stuffing

In the preceding discussion, we assumed synchronization between all the incoming channels and the multiplexer. This is difficult even when all the channels are nominally at the same rate. For example, consider a 1000 km coaxial cable carrying 2×10^8 pulses per second. Assuming the nominal propagation speed in the cable to be 2×10^8 m/s, it takes 1/200 second of transit time and 1 million pulses will be in transit. If the cable temperature increases by 1°F, the propagation velocity will increase by about 0.01%. This will cause the pulses in transit to arrive sooner, thus producing a temporary increase in the rate of pulses received. Because the extra pulses cannot be accommodated in the multiplexer, they must be temporarily stored at the receiver. If the cable temperature drops, the rate of received pulses will drop, and the multiplexer will have vacant slots with no data. These slots need to be stuffed with dummy digits (**pulse stuffing**).

DS1 signals in the North American network are often generated by crystal oscillators in individual channel banks or other digital terminal equipment. Although the oscillators are quite stable, they will not oscillate at exactly the same frequency, leading to another cause of asynchronicity in the network.

This shows that even in synchronously multiplexed systems, the data are rarely received at a synchronous rate. We always need a storage (known as an **elastic store**) and pulse stuffing (also known as **justification**) to accommodate such an situation. Obviously, this method of an elastic store and pulse stuffing will work even when the channels are asynchronous.

Three variants of the pulse stuffing scheme exist: (1) positive pulse stuffing, (2) negative pulse stuffing, and (3) positive/negative pulse stuffing. In positive pulse stuffing, the multiplexer

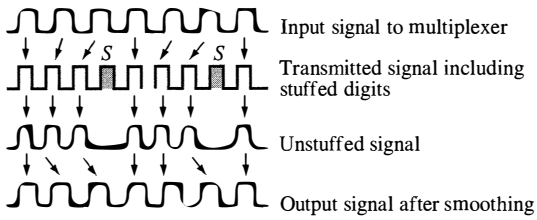


Figure 6.24
Pulse stuffing.

rate is higher than that required to accommodate all incoming tributaries at their maximum rate. Hence, the time slots in the multiplexed signal will become available at a rate exceeding that of the incoming data so that the tributary data will tend to lag (Fig. 6.24). At some stage, the system will decide that this lag has become great enough to require pulse stuffing. The information about the stuffed-pulse positions is transmitted through overhead bits. From the overhead bits, the receiver knows the stuffed-pulse position and eliminates that pulse.

Negative pulse stuffing is a complement of positive pulse stuffing. The time slots in the multiplexed signal now appear at a slightly slower rate than those of the tributaries, and thus the multiplexed signal cannot accommodate all the tributary pulses. Information about any left-out pulse and its position is transmitted through overhead bits. The positive/negative pulse stuffing is a combination of the first two schemes. The nominal rate of the multiplexer is equal to the nominal rate required to accommodate all incoming channels. Hence, we may need positive pulse stuffing at some times and negative stuffing at others. All this information is sent through overhead bits.

The C digits in Fig. 6.23 are used to transmit stuffing information. Only one stuffed bit per input channel is allowed per frame. This is sufficient to accommodate expected variations in the input signal rate. The bits C_A convey information about stuffing in channel A, bits C_B convey information about stuffing in channel B, and so on. The insertion of any stuffed pulse in any one subframe is denoted by setting all the three Cs in that line to **1**. No stuffing is indicated by using **0**s for all the three Cs. If a bit has been stuffed, the stuffed bit is the first information bit associated with the immediate channel following the F_1 bit, that is, the first such bit in the last 48-bit sequence in that subframe. For the first subframe, the stuffed bit will immediately follow the F_1 bit. For the second subframe, the stuffed bit will be the second bit following the F_1 bit, and so on.

6.4.3 Plesiochronous (almost Synchronous) Digital Hierarchy

We now present the digital hierarchy developed by the Bell System and currently included in the ANSI standards for telecommunications (Fig. 6.25). The North American hierarchy is implemented in North America and Japan.

Two major classes of multiplexers are used in practice. The first category is used for combining low-data-rate channels. It multiplexes channels of rates of up to 9600 bit/s into a signal of data rate of up to 64 kbit/s. The multiplexed signal, called **"digital signal level 0" (DS0)** in the North American hierarchy, is eventually transmitted over a voice-grade channel. The second class of multiplexers is at a much higher bit rate.

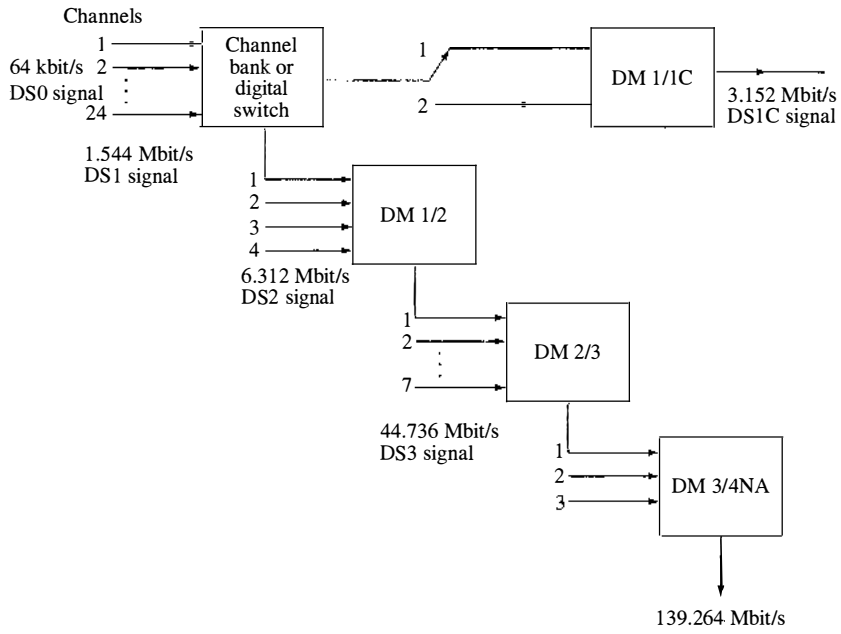


Figure 6.25 North American digital hierarchy (AT&T system).

There are four orders, or levels, of multiplexing. The first level is the **T1 multiplexer** or **channel bank**, consisting of 24 channels of 64 kbit/s each. The output of this multiplexer is a **DS1 (digital level 1)** signal at a rate of 1.544 Mbit/s. Four DS1 signals are multiplexed by a DM1/2 multiplexer to yield a DS2 signal at a rate 6.312 Mbit/s. Seven DS2 signals are multiplexed by a DM2/3 multiplexer to yield a DS3 signal at a rate of 44.736 Mbit/s. Finally, three DS3 signals are multiplexed by a DM3/4NA multiplexer to yield a DS4NA signal at a rate 139.264 Mbit/s. There is also a lower rate multiplexing hierarchy, known as the **digital data system (DDS)**, which provides standards for multiplexing digital signals with rates as low as 2.4 kbit/s into a DS0 signal for transmission through the network.

The inputs to a T1 multiplexer need not be restricted only to digitized voice channels alone. Any digital signal of 64 kbit/s of appropriate format can be transmitted. The case of the higher levels is similar. For example, all the incoming channels of the DM1/2 multiplexer need not be DS1 signals obtained by multiplexing 24 channels of 64 kbit/s each. Some of them may be 1.544 Mbit/s digital signals of appropriate format, and so on.

In Europe and many other parts of the world, another hierarchy, recommended by the ITU as an standard, has been adopted. This hierarchy, based on multiplexing 30 telephone channels of 64 kbit/s (E-0 channels) into an E-1 carrier at 2.048 Mbit/s (30 channels) is shown in Fig. 6.26. Starting from the base level of E-1, four lower level lines form one higher level line progressively, generating an E-2 line with data throughput of 8.448 Mbit/s, an E-3 line with data throughput of 34.368 Mbit/s, an E-4 line line with data throughput of 139.264 Mbit/s, and an E-5 line with data throughput of 565.148 Mbit/s. Because different networks must be able to interface with one another across the three different systems (North American, Japanese, and other) in the world, Fig. 6.26 demonstrates the relative relationship and the points of their common interface.

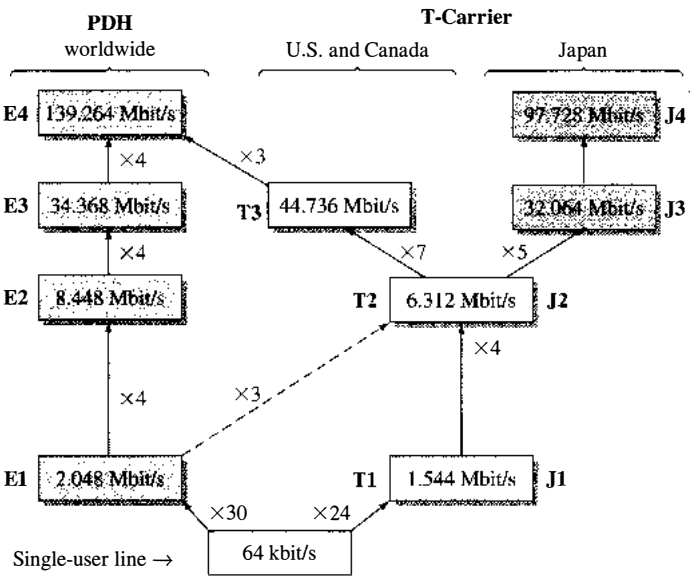


Figure 6.26 Plesiochronous digital hierarchy (PDH) according to ITU-T Recommendation G.704.

6.5 DIFFERENTIAL PULSE CODE MODULATION (DPCM)

PCM is not a very efficient system because it generates so many bits and requires so much bandwidth to transmit. Many different ideas have been proposed to improve the encoding efficiency of A/D conversion. In general, these ideas exploit the characteristics of the source signals. DPCM is one such scheme.

In analog messages we can make a good guess about a sample value from knowledge of past sample values. In other words, the sample values are not independent, and generally there is a great deal of redundancy in the Nyquist samples. Proper exploitation of this redundancy leads to encoding a signal with fewer bits. Consider a simple scheme; instead of transmitting the sample values, we transmit the difference between the successive sample values. Thus, if $m[k]$ is the kth sample, instead of transmitting $m[k]$, we transmit the difference $d[k] = m[k]-m[k-1]$. At the receiver, knowing $d[k]$ and several previous sample value $m[k-1]$, we can reconstruct $m[k]$. Thus, from knowledge of the difference $d[k]$, we can reconstruct $m[k]$ iteratively at the receiver. Now, the difference between successive samples is generally much smaller than the sample values. Thus, the peak amplitude m_p of the transmitted values is reduced considerably. Because the quantization interval $\Delta v = m_p/L$, for a given L (or n), this reduces the quantization interval Δv, thus reducing the quantization noise, which is given by $\Delta v^2/12$. This means that for a given n (or transmission bandwidth), we can increase the SNR, or for a given SNR, we can reduce n (or transmission bandwidth).

We can improve upon this scheme by estimating (predicting) the value of the kth sample $m[k]$ from a knowledge of several previous sample values. If this estimate is $\hat{m}[k]$, then we transmit the difference (prediction error) $d[k] = m[k]-\hat{m}[k]$. At the receiver also, we determine the estimate $\hat{m}[k]$ from the previous sample values, and then generate $m[k]$ by adding the received $d[k]$ to the estimate $\hat{m}[k]$. Thus, we reconstruct the samples at the receiver iteratively. If our prediction is worth its salt, the predicted (estimated) value $\hat{m}[k]$ will be close to $m[k]$, and their difference (prediction error) $d[k]$ will be even smaller than the difference between the successive samples. Consequently, this scheme, known as the **differential PCM (DPCM)**,

is superior to the naive prediction described in the preceding paragraph, which is a special case of DPCM, where the estimate of a sample value is taken as the previous sample value, that is, $\hat{m}[k] = m[k-1]$.

Spirits of Taylor, Maclaurin, and Wiener

Before describing DPCM, we shall briefly discuss the approach to signal prediction (estimation). To the uninitiated, future prediction seems like mysterious stuff, fit only for psychics, wizards, mediums, and the like, who can summon help from the spirit world. Electrical engineers appear to be hopelessly outclassed in this pursuit. Not quite so! We can also summon the spirits of Taylor, Maclaurin, Wiener, and the like to help us. What is more, unlike Shakespeare's spirits, our spirits come when called.* Consider, for example, a signal $m(t)$, which has derivatives of all orders at t. Using the Taylor series for this signal, we can express $m(t+T_s)$ as

$$m(t+T_s) = m(t) + T_s\dot{m}(t) + \frac{T_s^2}{2!}\ddot{m}(t) + \frac{T_s^3}{3!}\dddot{m}(t) + \cdots \tag{6.42a}$$

$$\approx m(t) + T_s\dot{m}(t) \qquad \text{for small } T_s \tag{6.42b}$$

Equation (6.42a) shows that from a knowledge of the signal and its derivatives at instant t, we can predict a future signal value at $t+T_s$. In fact, even if we know just the first derivative, we can still predict this value approximately, as shown in Eq. (6.42b). Let us denote the kth sample of $m(t)$ by $m[k]$, that is, $m(kT_s) = m[k]$, and $m(kT_s \pm T_s) = m[k \pm 1]$, and so on. Setting $t = kT_s$ in Eq. (6.42b), and recognizing that $\dot{m}(kT_s) \approx [m(kT_s) - m(kT_s - T_s)]/T_s$, we obtain

$$\begin{aligned} m[k+1] &\approx m[k] + T_s\left[\frac{m[k]-m[k-1]}{T_s}\right] \\ &= 2m[k] - m[k-1] \end{aligned}$$

This shows that we can find a crude prediction of the $(k+1)$th sample from the two previous samples. The approximation in Eq. (6.42b) improves as we add more terms in the series on the right-hand side. To determine the higher order derivatives in the series, we require more samples in the past. The larger the number of past samples we use, the better will be the prediction. Thus, in general, we can express the prediction formula as

$$m[k] \approx a_1 m[k-1] + a_2 m[k-2] + \cdots + a_N m[k-N] \tag{6.43}$$

The right-hand side is $\hat{m}[k]$, the predicted value of $m[k]$. Thus,

$$\hat{m}[k] = a_1 m[k-1] + a_2 m[k-2] + \cdots + a_N m[k-N] \tag{6.44}$$

This is the equation of an Nth-order predictor. Larger N would result in better prediction in general. The output of this filter (predictor) is $\hat{m}[k]$, the predicted value of $m[k]$. The input consists of the previous samples $m[k-1]$, $m[k-2]$, ..., $m[k-N]$, although it is customary to say that the input is $m[k]$ and the output is $\hat{m}[k]$. Observe that this equation reduces to

* From Shakespeare, Henry IV, Part 1, Act III, Scene 1:
Glendower: *I can call the spirits from vasty deep.*
Hotspur: *Why, so can I, or so can any man;*
But will they come when you do call for them?

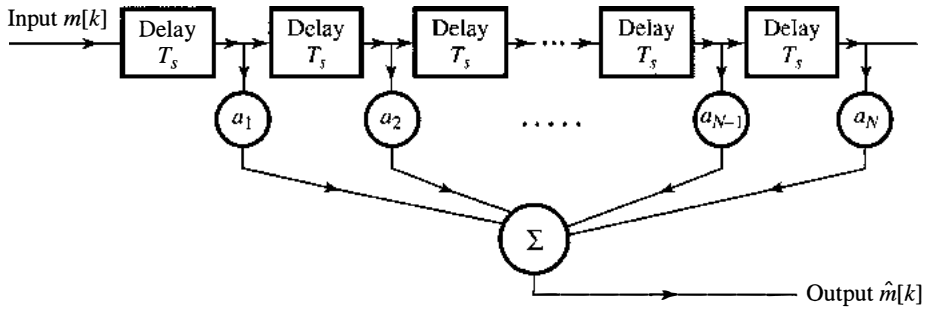


Figure 6.27 Transversal filter (tapped delay line) used as a linear predictor.

$\hat{m}[k] = m[k-1]$ in the case of the first-order prediction. It follows from Eq. (6.42b), where we retain only the first term on the right-hand side. This means that $a_1 = 1$, and the first-order predictor is a simple time delay.

We have outlined here a very simple procedure for predictor design. In a more sophisticated approach, discussed in Sec. 8.5, where we use the minimum mean squared error criterion for best prediction, the **prediction coefficients** a_j in Eq. (6.44) are determined from the statistical correlation between various samples. The predictor described in Eq. (6.44) is called a *linear predictor*. It is basically a transversal filter (a tapped delay line), where the tap gains are set equal to the prediction coefficients, as shown in Fig. 6.27.

Analysis of DPCM

As mentioned earlier, in DPCM we transmit not the present sample $m[k]$, but $d[k]$ (the difference between $m[k]$ and its predicted value $\hat{m}[k]$). At the receiver, we generate $\hat{m}[k]$ from the past sample values to which the received $d[k]$ is added to generate $m[k]$. There is, however, one difficulty associated with this scheme. At the receiver, instead of the past samples $m[k-1]$, $m[k-2]$, ..., as well as $d[k]$, we have their quantized versions $m_q[k-1]$, $m_q[k-2]$, ... Hence, we cannot determine $\hat{m}[k]$. We can determine only $\hat{m}_q[k]$, the estimate of the quantized sample $m_q[k]$, in terms of the quantized samples $m_q[k-1]$, $m_q[k-2]$, ... This will increase the error in reconstruction. In such a case, a better strategy is to determine $\hat{m}_q[k]$, the estimate of $m_q[k]$ (instead of $m[k]$), at the transmitter also from the quantized samples $m_q[k-1]$, $m_q[k-2]$, ... The difference $d[k] = m[k] - \hat{m}_q[k]$ is now transmitted via PCM. At the receiver, we can generate $\hat{m}_q[k]$, and from the received $d[k]$, we can reconstruct $m_q[k]$.

Figure 6.28a shows a DPCM transmitter. We shall soon show that the predictor input is $m_q[k]$. Naturally, its output is $\hat{m}_q[k]$, the predicted value of $m_q[k]$. The difference

$$d[k] = m[k] - \hat{m}_q[k] \tag{6.45}$$

is quantized to yield

$$d_q[k] = d[k] + q[k] \tag{6.46}$$

where $q[k]$ is the quantization error. The predictor output $\hat{m}_q[k]$ is fed back to its input so that the predictor input $m_q[k]$ is

$$\begin{aligned} m_q[k] &= \hat{m}_q[k] + d_q[k] \\ &= m[k] - d[k] + d_q[k] \\ &= m[k] + q[k] \end{aligned} \tag{6.47}$$

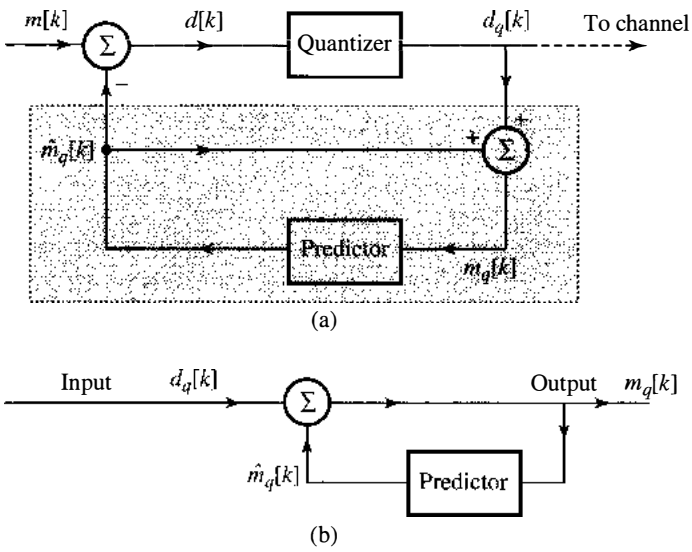


Figure 6.28
DPCM system:
(a) transmitter;
(b) receiver.

This shows that $m_q[k]$ is a quantized version of $m[k]$. The predictor input is indeed $m_q[k]$, as assumed. The quantized signal $d_q[k]$ is now transmitted over the channel. The receiver shown in Fig. 6.28b is identical to the shaded portion of the transmitter. The inputs in both cases are also the same, namely, $d_q[k]$. Therefore, the predictor output must be $\hat{m}_q[k]$ (the same as the predictor output at the transmitter). Hence, the receiver output (which is the predictor input) is also the same, viz., $m_q[k] = m[k] + q[k]$, as found in Eq. (6.47). This shows that we are able to receive the desired signal $m[k]$ plus the quantization noise $q[k]$. This is the quantization noise associated with the difference signal $d[k]$, which is generally much smaller than $m[k]$. The received samples $m_q[k]$ are decoded and passed through a low-pass filter for D/A conversion.

SNR Improvement

To determine the improvement in DPCM over PCM, let m_p and d_p be the peak amplitudes of $m(t)$ and $d(t)$, respectively. If we use the same value of L in both cases, the quantization step Δv in DPCM is reduced by the factor d_p/m_p. Because the quantization noise power is $(\Delta v)^2/12$, the quantization noise in DPCM is reduced by the factor $(m_p/d_p)^2$, and the SNR is increased by the same factor. Moreover, the signal power is proportional to its peak value squared (assuming other statistical properties invariant). Therefore, G_p (SNR improvement due to prediction) is at least

$$G_p = \frac{P_m}{P_d}$$

where P_m and P_d are the powers of $m(t)$ and $d(t)$, respectively. In terms of decibel units, this means that the SNR increases by $10 \log_{10}(P_m/P_d)$ dB. Therefore, Eq. (6.41) applies to DPCM also with a value of α that is higher by $10 \log_{10}(P_m/P_d)$ dB. In Example 8.24, a second-order predictor processor for speech signals is analyzed. For this case, the SNR improvement is found to be 5.6 dB. In practice, the SNR improvement may be as high as 25 dB in such cases as short-term voiced speech spectra and in the spectra of low-activity images.[12] Alternately, for the same SNR, the bit rate for DPCM could be lower than that for PCM by 3 to 4 bits per sample. Thus, telephone systems using DPCM can often operate at 32 or even 24 kbit/s.

6.6 ADAPTIVE DIFFERENTIAL PCM (ADPCM)

Adaptive DPCM (ADPCM) can further improve the efficiency of DPCM encoding by incorporating an adaptive quantizer at the encoder. Figure 6.29 illustrates the basic configuration of ADPCM. For practical reasons, the number of quantization level L is fixed. When a fixed quantization step Δv is applied, either the quantization error is too large because Δv is too big or the quantizer cannot cover the necessary signal range when Δv is too small. Therefore, it would be better for the quantization step Δv to be adaptive so that Δv is large or small depending on whether the prediction error for quantizing is large or small.

It is important to note that the quantized prediction error $d_q[k]$ can be a good indicator of the prediction error size. For example, when the quantized prediction error samples vary close to the largest positive value (or the largest negative value), it indicates that the prediction error is large and Δv needs to grow. Conversely, if the quantized samples oscillate near zero, then the prediction error is small and Δv needs to decrease. It is important that both the modulator and the receiver have access to the same quantized samples. Hence, the adaptive quantizer and the receiver reconstruction can apply the same algorithm to adjust the Δv identically.

Compared with DPCM, ADPCM can further compress the number of bits needed for a signal waveform. For example, it is very common in practice for an 8-bit PCM sequence to be encoded into a 4-bit ADPCM sequence at the same sampling rate. This easily represents a 2:1 bandwidth or storage reduction with virtually no loss.

ADPCM encoder has many practical applications. The ITU-T standard G.726 specifies an ADPCM speech coder and decoder (called **codec**) for speech signal samples at 8 kHz.[7] The G.726 ADPCM predictor uses an eighth-order predictor. For different quality levels, G.726 specifies four different ADPCM rates at 16, 24, 32, and 40 kbit/s. They correspond to four different bit sizes for each speech sample at 2 bits, 3 bits, 4 bits, and 5 bits, respectively, or equivalently, quantization levels of 4, 8, 16, and 32, respectively.

The most common ADPCM speech encoders use 32 kbit/s. In practice, there are multiple variations of ADPCM speech codec. In addition to the ITU-T G.726 specification,[7] these include the OKI ADPCM codec, the Microsoft ADPCM codec supported by WAVE players, and the Interactive Multimedia Association (IMA) ADPCM, also known as the DVI ADPCM. The 32 kbit/s ITU-T G.726 ADPCM speech codec is widely used in the DECT (digital enhanced cordless telecommunications) system, which itself is widely used for residential and business cordless phone communications. Designed for short-range use as an access mechanism to the main networks, DECT offers cordless voice, fax, data, and multimedia communications. DECT is now in use in over 100 countries worldwide. Another major user of the 32 kbit/s ADPCM codec is the Personal Handy-phone System (or PHS), also marketed as the Personal Access System (PAS) and known as Xiaolingtong in China.

PHS is a mobile network system similar to a cellular network, operating in the 1880 to 1930 MHz frequency band, used mainly in Japan, China, Taiwan, and elsewhere in Asia. Originally developed by the NTT Laboratory in Japan in 1989, PHS is much simpler to implement and

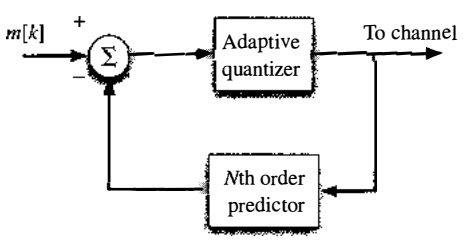


Figure 6.29 ADPCM encoder uses an adaptive quantizer controlled only by the encoder output bits.

deploy. Unlike cellular networks, PHS phones and base stations are low-power, short-range facilities. The service is often pejoratively called the "poor man's cellular" because of its limited range and poor roaming ability. PHS first saw limited deployment (NTT-Personal, DDI-Pocket, and ASTEL) in Japan in 1995 but has since nearly disappeared. Surprisingly, PHS has seen a resurgence in markets like China, Taiwan, Vietnam, Bangladesh, Nigeria, Mali, Tanzania, and Honduras, where its low cost of deployment and hardware costs offset the system's disadvantages. In China alone, there was an explosive expansion of subscribers, reaching nearly 80 million in 2006.

6.7 DELTA MODULATION

Sample correlation used in DPCM is further exploited in **delta modulation (DM)** by oversampling (typically four times the Nyquist rate) the baseband signal. This increases the correlation between adjacent samples, which results in a small prediction error that can be encoded using only one bit ($L = 2$). Thus, DM is basically a 1-bit DPCM, that is, a DPCM that uses only two levels ($L = 2$) for quantization of $m[k] - \hat{m}_q[k]$. In comparison to PCM (and DPCM), it is a very simple and inexpensive method of A/D conversion. A 1-bit codeword in DM makes word framing unnecessary at the transmitter and the receiver. This strategy allows us to use fewer bits per sample for encoding a baseband signal.

In DM, we use a first-order predictor, which, as seen earlier, is just a time delay of T_s (the sampling interval). Thus, the DM transmitter (modulator) and receiver (demodulator) are identical to those of the DPCM in Fig. 6.28, with a time delay for the predictor, as shown in Fig. 6.30, from which we can write

$$m_q[k] = m_q[k-1] + d_q[k] \tag{6.48}$$

Hence,

$$m_q[k-1] = m_q[k-2] + d_q[k-1]$$

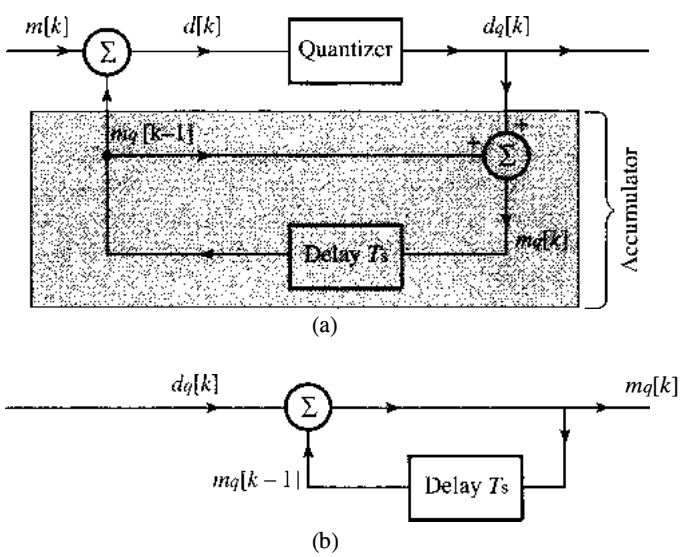


Figure 6.30 Delta modulation is a special case of DPCM.

Substituting this equation into Eq. (6.48) yields

$$m_q[k] = m_q[k-2] + d_q[k] + d_q[k-1]$$

Proceeding iteratively in this manner, and assuming zero initial condition, that is, $m_q[0] = 0$, we write

$$m_q[k] = \sum_{m=0}^{k} d_q[m] \tag{6.49}$$

This shows that the receiver (demodulator) is just an accumulator (adder). If the output $d_q[k]$ is represented by impulses, then the accumulator (receiver) may be realized by an integrator because its output is the sum of the strengths of the input impulses (sum of the areas under the impulses). We may also replace with an integrator the feedback portion of the modulator (which is identical to the demodulator). The demodulator output is $m_q[k]$, which when passed through a low-pass filter yields the desired signal reconstructed from the quantized samples.

Figure 6.31 shows a practical implementation of the delta modulator and demodulator. As discussed earlier, the first-order predictor is replaced by a low-cost integrator circuit (such as an RC integrator). The modulator (Fig. 6.31a) consists of a comparator and a sampler in the direct path and an integrator-amplifier in the feedback path. Let us see how this delta modulator works.

The analog signal $m(t)$ is compared with the feedback signal (which serves as a predicted signal) $\hat{m}_q(t)$. The error signal $d(t) = m(t) - \hat{m}_q(t)$ is applied to a comparator. If $d(t)$ is positive, the comparator output is a constant signal of amplitude E, and if $d(t)$ is negative, the comparator output is $-E$. Thus, the difference is a binary signal ($L = 2$) that is needed to generate a 1-bit DPCM. The comparator output is sampled by a sampler at a rate of f_s samples per second, where f_s is typically much higher than the Nyquist rate. The sampler thus produces a train of narrow pulses $d_q[k]$ (to simulate impulses) with a positive pulse when $m(t) > \hat{m}_q(t)$ and a negative pulse when $m(t) < \hat{m}_q(t)$. Note that each sample is coded by a single binary pulse (1-bit DPCM), as required. The pulse train $d_q[k]$ is the delta-modulated pulse train (Fig. 6.31d). The modulated signal $d_q[k]$ is amplified and integrated in the feedback path to generate $\hat{m}_q(t)$ (Fig. 6.31c), which tries to follow $m(t)$.

To understand how this works, we note that each pulse in $d_q[k]$ at the input of the integrator gives rise to a step function (positive or negative, depending on the pulse polarity) in $\hat{m}_q(t)$. If, for example, $m(t) > \hat{m}_q(t)$, a positive pulse is generated in $d_q[k]$, which gives rise to a positive step in $\hat{m}_q(t)$, trying to equalize $\hat{m}_q(t)$ to $m(t)$ in small steps at every sampling instant, as shown in Fig. 6.31c. It can be seen that $\hat{m}_q(t)$ is a kind of staircase approximation of $m(t)$. When $\hat{m}_q(t)$ is passed through a low-pass filter, the coarseness of the staircase in $\hat{m}_q(t)$ is eliminated, and we get a smoother and better approximation to $m(t)$. The demodulator at the receiver consists of an amplifier-integrator (identical to that in the feedback path of the modulator) followed by a low-pass filter (Fig. 6.31b).

DM Transmits the Derivative of $m(t)$

In PCM, the analog signal samples are quantized in L levels, and this information is transmitted by n pulses per sample ($n = \log_2 L$). A little reflection shows that in DM, the modulated signal carries information not about the signal samples but about the difference between successive samples. If the difference is positive or negative, a positive or a negative pulse (respectively) is generated in the modulated signal $d_q[k]$. Basically, therefore, DM carries the information about the derivative of $m(t)$, hence, the name "delta modulation." This can also be seen from

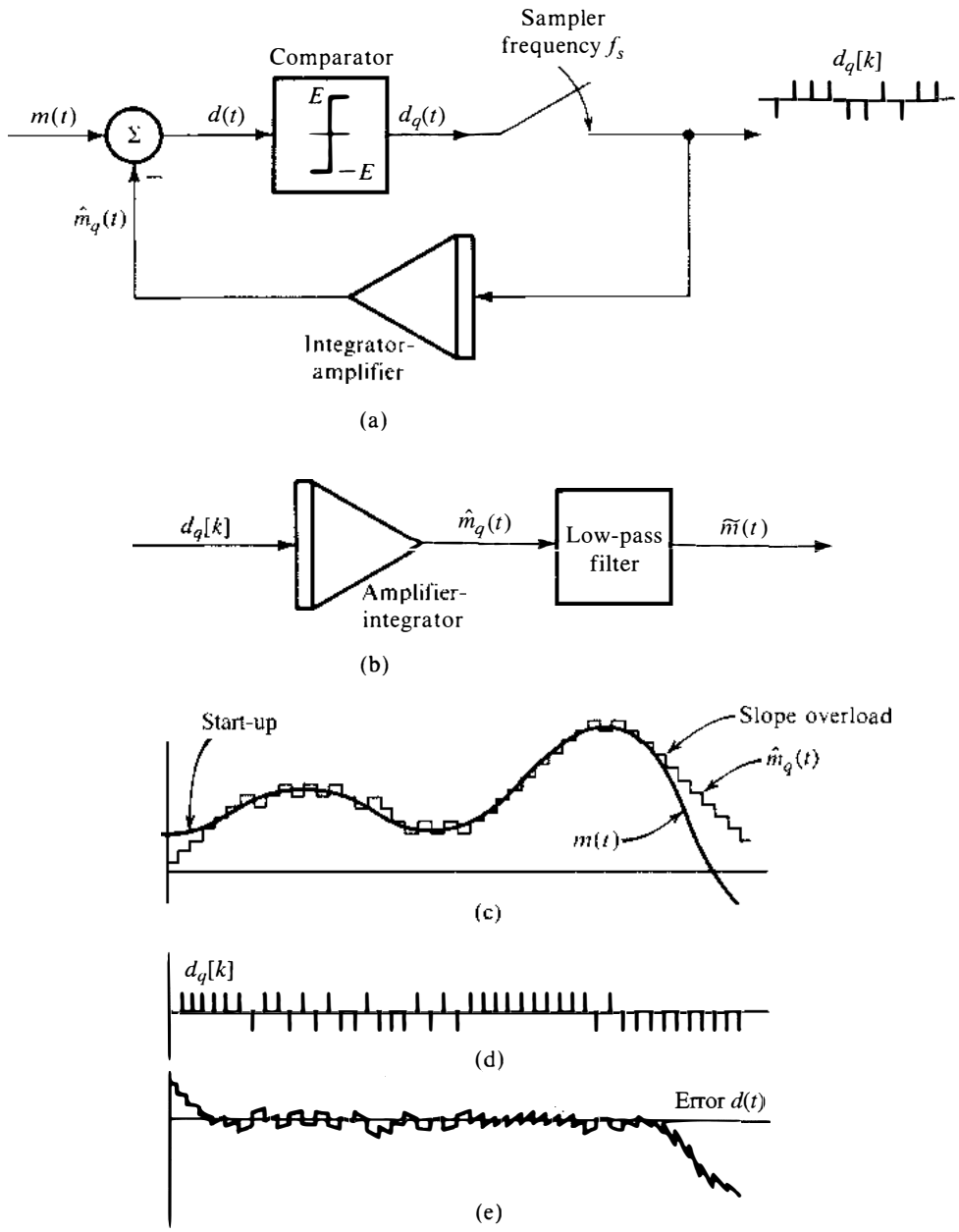


Figure 6.31
Delta modulation: (a) and (b) delta demodulators; (c) message signal versus integrator output signal (d) delta-modulated pulse trains; (e) modulation errors.

the fact that integration of the delta-modulated signal yields $\hat{m}_q(t)$, which is an approximation of $m(t)$.

In PCM, the information of each quantized sample is transmitted by an n-bit code word, whereas in DM the information of the difference between successive samples is transmitted by a 1-bit code word.

Threshold of Coding and Overloading

Threshold and overloading effects can be clearly seen in Fig. 6.31c. Variations in $m(t)$ smaller than the step value (threshold of coding) are lost in DM. Moreover, if $m(t)$ changes too fast,

that is, if $\dot{m}(t)$ is too high, $\hat{m}_q(t)$ cannot follow $m(t)$, and overloading occurs. This is the so-called **slope overload**, which gives rise to the slope overload noise. This noise is one of the basic limiting factors in the performance of DM. We should expect slope overload rather than amplitude overload in DM, because DM basically carries the information about $\dot{m}(t)$. The granular nature of the output signal gives rise to the granular noise similar to the quantization noise. The slope overload noise can be reduced by increasing E (the step size). This unfortunately increases the granular noise. There is an optimum value of E, which yields the best compromise giving the minimum overall noise. This optimum value of E depends on the sampling frequency f_s and the nature of the signal.[12]

The slope overload occurs when $\hat{m}_q(t)$ cannot follow $m(t)$. During the sampling interval T_s, $\hat{m}_q(t)$ is capable of changing by E, where E is the height of the step. Hence, the maximum slope that $\hat{m}_q(t)$ can follow is E/T_s, or Ef_s, where f_s is the sampling frequency. Hence, no overload occurs if

$$|\dot{m}(t)| < Ef_s$$

Consider the case of tone modulation (meaning a sinusoidal message):

$$m(t) = A \cos \omega t$$

The condition for no overload is

$$|\dot{m}(t)|_{\max} = \omega A < Ef_s \tag{6.50}$$

Hence, the maximum amplitude $A_{\max}$ of this signal that can be tolerated without overload is given by

$$A_{\max} = \frac{Ef_s}{\omega} \tag{6.51}$$

The overload amplitude of the modulating signal is inversely proportional to the frequency ω. For higher modulating frequencies, the overload occurs for smaller amplitudes. For voice signals, which contain all frequency components up to (say) 4 kHz, calculating $A_{\max}$ by using $\omega = 2\pi \times 4000$ in Eq. (6.51) will give an overly conservative value. It has been shown by de Jager[13] that $A_{\max}$ for voice signals can be calculated by using $\omega_r \simeq 2\pi \times 800$ in Eq. (6.51),

$$[A_{\max}]_{\text{voice}} \simeq \frac{Ef_s}{\omega_r} \tag{6.52}$$

Thus, the maximum voice signal amplitude $A_{\max}$ that can be used without causing slope overload in DM is the same as the maximum amplitude of a sinusoidal signal of reference frequency f_r ($f_r \simeq 800$ Hz) that can be used without causing slope overload in the same system.

Fortunately, the voice spectrum (as well as the television video signal) also decays with frequency and closely follows the overload characteristics (curve c, Fig. 6.32). For this reason, DM is well suited for voice (and television) signals. Actually, the voice signal spectrum (curve b) decreases as $1/\omega$ up to 2000 Hz, and beyond this frequency, it decreases as $1/\omega^2$. If we had used a double integration in the feedback circuit instead of a single integration, $A_{\max}$ in Eq. (6.51) would be proportional to $1/\omega^2$. Hence, a better match between the voice spectrum and the overload characteristics is achieved by using a single integration up to 2000 Hz and a double integration beyond 2000 Hz. Such a circuit (the double integration) responds fast

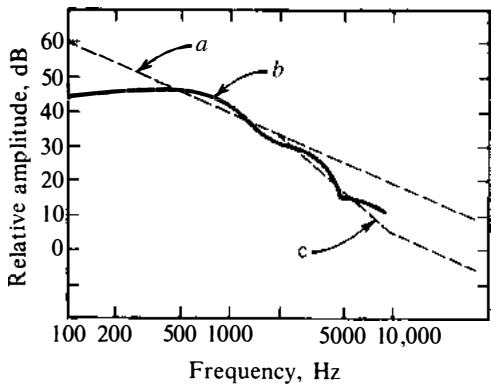


Figure 6.32
Voice signal spectrum.

but has a tendency to instability, which can be reduced by using some low-order prediction along with double integration. A double integrator can be built by placing in cascade two low-pass RC integrators with time constants $R_1C_1 = 1/200\pi$ and $R_2C_2 = 1/4000\pi$, respectively. This results in single integration from 100 to 2000 Hz and double integration beyond 2000 Hz.

Sigma-Delta Modulation

While discussing the threshold of coding and overloading, we illustrated that the essence of the conventional DM is to encode and transmit the derivative of the analog message signal. Hence, the receiver of DM requires an integrator as shown in Fig. 6.31 and also, equivalently, in Fig. 6.33a. Since signal transmission inevitably is subjected to channel noise, such noise will be integrated and will accumulate at the receiver output, which is a highly undesirable phenomenon that is a major drawback of DM.

To overcome this critical drawback of DM, a small modification can be made. First, we can view the overall DM system consisting of the transmitter and the receiver as approximately distortionless and linear. Thus, one of its serial components, the receiver integrator $1/s$, may be moved to the front of the transmitter (encoder) without affecting the overall modulator and demodulator response, as shown in Fig. 6.33b. Finally, the two integrators can be merged into a single one after the subtractor, as shown in Fig. 6.33c. This modified system is known as the sigma-delta modulation (Σ-ΔM) .

As we found in the study of preemphasis and deemphasis filters in FM, because channel noise and the message signal do not follow the same route, the order of serial components in the overall modulation-demodulation system can have different effects on the SNR. The seemingly minor move of the integrator $1/s$ in fact has several major advantages:

- The channel noise no longer accumulates at the demodulator.
- The important low-frequency content of the message $m(t)$ is preemphasized by the integrator $1/j\omega$. This helps many practical signals (such as speech) whose low-frequency components are more important.
- The integrator effectively smooths the signal for encoding (Fig. 6.33b). Hence, overloading becomes less likely.
- The low-pass nature of the integrator increases the correlation between successive samples, leading to smaller encoding error.
- The demodulator is simplified.

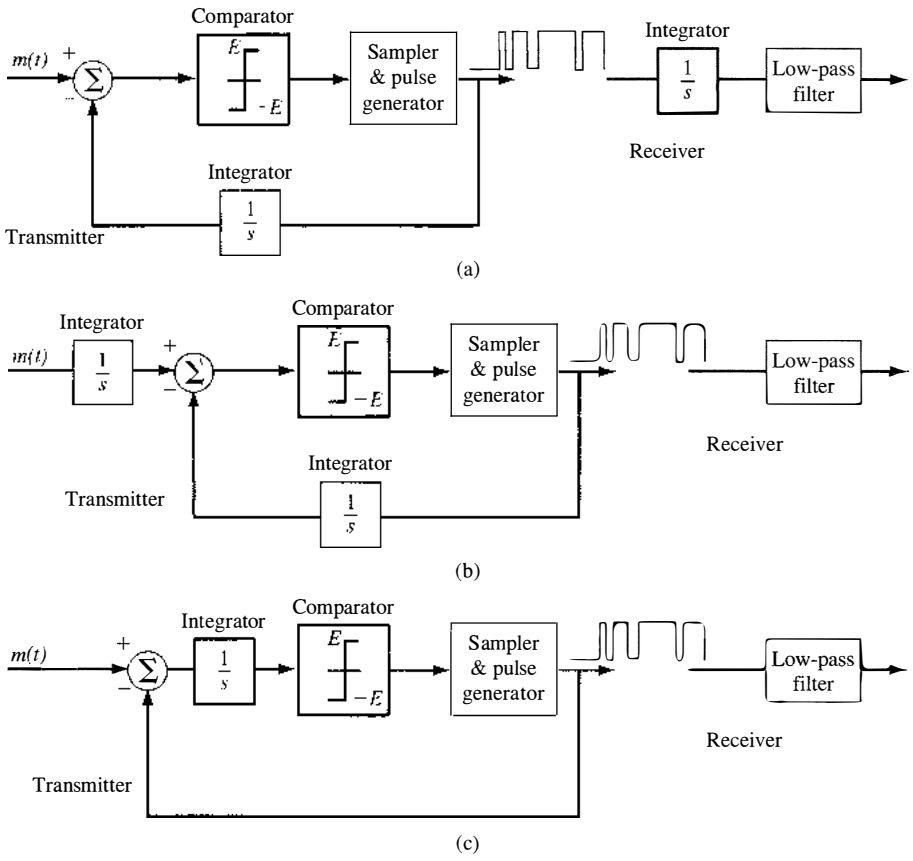


Figure 6.33
(a) Conventional delta modulator. (b) Σ-Δ modulator. (c) Simpler Σ-Δ modulator.

Adaptive Delta Modulation (ADM)

The DM discussed so far suffers from one serious disadvantage. The dynamic range of amplitudes is too small because of the threshold and overload effects discussed earlier. To address this problem, some type of signal compression is necessary. In DM, a suitable method appears to be the adaptation of the step value E according to the level of the input signal derivative. For example, in Fig. 6.31, when the signal $m(t)$ is falling rapidly, slope overload occurs. If we can increase the step size during this period, the overload could be avoided. On the other hand, if the slope of $m(t)$ is small, a reduction of step size will reduce the threshold level as well as the granular noise. The slope overload causes $d_q[k]$ to have several pulses of the same polarity in succession. This calls for increased step size. Similarly, pulses in $d_q[k]$ alternating continuously in polarity indicates small-amplitude variations, requiring a reduction in step size. In ADM we detect such pulse patterns and automatically adjust the step size.[14] This results in a much larger dynamic range for DM.

6.8 VOCODERS AND VIDEO COMPRESSION

PCM, DPCM, ADPCM, DM, and Σ-ΔM are all examples of what are known as waveform source encoders. Basically, waveform encoders do not take into consideration how the signals for digitization are generated. Hence, the amount of compression achievable by waveform encoders is highly limited by the degree of correlation between successive signal samples.

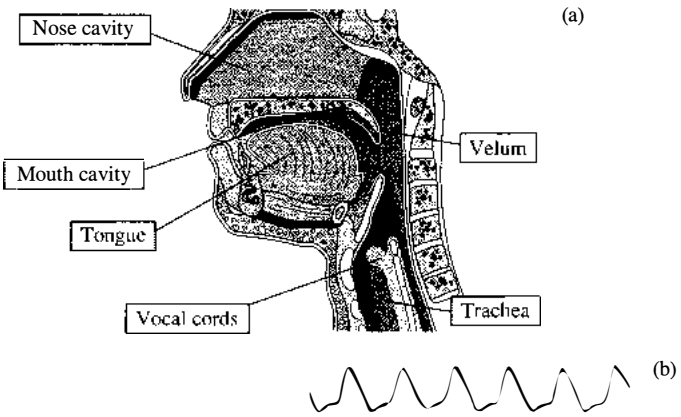


Figure 6.34 (a) The human speech production mechanism. (b) Typical pressure impulses.

For a low-pass source signal with finite bandwidth B Hz, even if we apply the minimum Nyquist sampling rate $2B$ Hz and 1-bit encoding, the bit rate cannot be lower than $2B$ bit/s. There have been many successful methods introduced to drastically reduce the source coding rates of speech and video signals, very important to our daily communication needs. Unlike waveform encoders, the most successful speech and video encoders are based on the human physiological models involved in speech generation and in video perception. Here we describe the basic principles of the linear prediction voice coders (known as vocoders) and the video compression method proposed by the Moving Picture Experts Group (MPEG).

6.8.1 Linear Prediction Coding Vocoders

Voice Models and Model-Based Vocoders

Linear prediction coding (LPC) vocoders are model-based systems. The model, in turn, is based on a good understanding of the human voice mechanism. Fig. 6.34a provides a cross-sectional illustration of the human speech apparatus. Briefly, human speech is produced by the joint interaction of lungs, vocal cords, and the articulation tract, consisting of the mouth and the nose cavity. Based on this physiological speech model, human voices can be divided into *voiced* and the *unvoiced* sound categories. Voiced sounds are those made while the vocal cords are vibrating. Put a finger on your Adam's apple* while speaking, and you can feel the vibration the vocal cords when you pronounce all the vowels and some consonants, such as *g* as in *gut*, *b* as in *but*, and *n* as in *nut*. Unvoiced sounds are made while the vocal cords are not vibrating. Several consonants such as *k*, *p*, and *t* are unvoiced. Examples of unvoiced sounds include *h* in *hut*, *c* in *cut*, and *p* in *put*.

For the production of voiced sounds, the lungs expel air through the epiglottis, causing the vocal cords to vibrate. The vibrating vocal cords interrupt the airstream and produce a quasi-periodic pressure wave consisting of impulses. The pressure wave impulses are commonly called pitch impulses, and the frequency of the pressure signal is the pitch frequency or fundamental frequency as shown in Fig. 6.34b. This is the part of the voice signal that defines the speech tone. Speech that is uttered in a constant pitch frequency sounds monotonous. In ordinary cases, the pitch frequency of a speaker varies almost constantly, often from syllable to syllable.

* The slight projection at the front of the throat formed by the largest cartilage of the larynx, usually more prominent in men than in women.

For voiced sound, the pitch impulses stimulate the air in the vocal tract (mouth and nasal cavities). For unvoiced sounds, the excitation comes directly from the air flow. Extensive studies[15–17] have shown that for unvoiced sounds, the excitation to the vocal tract is more like a broadband noise. When cavities in the vocal tract resonate under excitation, they radiate a sound wave, which is the speech signal. Both cavities form resonators with characteristic resonance frequencies (formant frequencies). Changing the shape (hence the resonant characteristics) of the mouth cavity allows different sounds to be pronounced. Amazingly, this (vocal) articulation tract can be approximately modeled by a simple linear digital filter with an all-pole transfer function

$$H(z) = \frac{g}{A(z)} = g \cdot \left(1 - \sum_{i=1}^{p} a_i z^{-i}\right)^{-1}$$

where g is a gain factor and $A(z)$ is known as the prediction filter, much like the feedback filter used in DPCM and ADPCM. One can view the function of the vocal articulation apparatus as a spectral shaping filter $H(z)$.

LPC Models

Based on this human speech model, a voice encoding approach different from waveform coding can be established. Instead of sending actual signal samples, the model-based vocoders *analyze* the voice signals segment by segment to determine the best-fitting speech model parameters. As shown in Fig. 6.35, after speech analysis, the transmitter sends the necessary speech model parameters (formants) for each voice segment to the receiver. The receiver then uses the parameters for the speech model to set up a voice synthesizer to regenerate the respective voice segments. In other words, what a user hears at the receiver actually consists of signals reproduced by an artificial voice *synthesizing machine*!

In the analysis of a sampled voice segment (consisting of multiple samples), the pitch analysis will first determine whether the speech is a voiced or an unvoiced piece. If the signal is classified as "voiced," the pitch analyzer will estimate pitch frequency (or equivalently the pitch period). In addition, the LPC analyzer will estimate the all-pole filter coefficients in $A(z)$. Because the linear prediction error indicates how well the linear prediction filter fits the voice samples, the LPC analyzer can determine the optimum filter coefficients by minimizing the mean square error (MSE) of the linear prediction error.[18, 19]

Directly transmitting the linear prediction (LP) filter parameters is unsound because the filter is very sensitive to parameter errors due to quantization and channel noises. Worse yet, the LP filter may even become unstable because of small coefficient errors. In practice, the stability of this all-pole linear prediction (LP) filter can be ensured by utilizing the modular lattice filter structure through the well-known Levinson-Durbin algorithm.[20, 21] Lattice filter parameters, known as reflection coefficients $\{r_k\}$, are less sensitive to quantization errors and

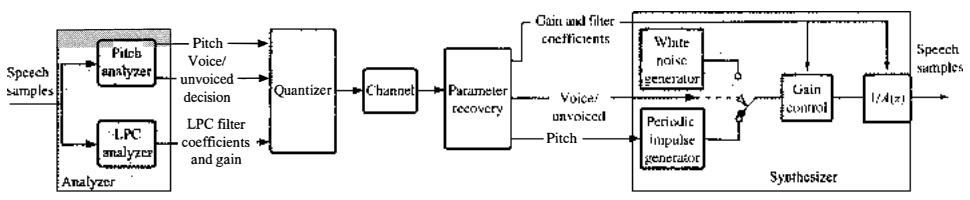


Figure 6.35 Analysis and synthesis of voice signals in an LPC encoder and decoder.

TABLE 6.1
Quantization Bit Allocation in LPC-10 Vocoder

Pitch Period	Voiced/Unvoiced	Gain g	10 LP Filter Parameters, bits/coefficient				
			$r_1 - r_4$	$r_5 - r_8$	r_9	r_{10}	
			5 bits	4 bits	3 bits	2 bits	Voiced
6 bits	1 bit	5 bits	5 bits		*Not used*		Unvoiced

noise. Transmission is further improved by sending their log-area ratios (LAR), defined as

$$o_k \triangleq \log \frac{1 + r_k}{1 - r_k}$$

or by sending intermediate values from the Levinson-Durbin recursion known as the partial reflection coefficients (PARCOR). Another practical approach is to find the equivalent *line spectral pairs (LSP)* as representation of the LPC filter coefficients for transmission over channels. LSP has the advantage of low sensitivity to quantization noise.[22, 23] As long as the pth-order all-pole LP filter is stable, it can be represented by p real-valued, line spectral frequencies. In every representation, however, a pth-order synthesizer filter can be obtained by the LPC decoder from the quantization of p real-valued coefficients. In general 8 to 14 LP parameters are sufficient for vocal tract representation.

We can now use a special LPC example to illustrate the code efficiency of such model-based vocoders. In the so-called LPC-10 vocoder,* the speech is sampled at 8 kHz. 180 samples (22.5 ms) form an LPC frame for transmission.[24] The bits per speech frame are allocated to quantize the pitch period, the voiced/unvoiced flag, the filter gain, and the 10 filter coefficients, according to Table 6.1. Thus, each frame requires between 32 (unvoiced) and 53 (voiced) bits. Adding frame control bits results an average coded stream of 54 bits per speech frame, or an overall rate of 2400 bit/s.[24] Based on subjective tests, this rather minimal LPC-10 codec has low mean opinion score (MOS) but does provide highly intelligible speech connections. LPC-10 is part of the FS-1015, a low-rate secure telephony codec standard developed by the U.S. Department of Defense in 1984. A later enhancement to LPC-10 is known as the LPC-10(e).

Compared with the 64 kbit/s PCM or the 32 kbit/s ADPCM waveform codec, LPC vocoders are much more efficient and can achieve speech code rates below 9.6 kbit/s. The 2.4 kbit/s LPC-10 example can provide speech digitization at a rate much lower than even the speech waveform sampling rate of 8 kHz. The loss of speech quality is a natural trade-off. To better understand the difference between waveform vocoders and the model-based vocoders such as LPC, we can use the analogy of a food delivery service. Imagine a family living Alaska that wishes to order a nice meal from a famous restaurant in New York City. For practical reasons, the restaurant would have to send prepared dishes uncooked and frozen; then the family would follow the cooking directions. The food would probably taste fine, but the meal would be missing the finesse of the original chef. This option is like speech transmission via PCM. The receiver has the basic ingredients but must tolerate the quantization error (manifested by the lack of the chef's cooking finesse). To reduce transportation weight, another option is for the family to order the critical ingredients only. The heavier but common ingredients (such as rice and potatoes) can be acquired locally. This approach is like DPCM or ADPCM, in which only the unpredictable part of the voice is transmitted. Finally, the family can simply go online to

* So-called because it uses order $p = 10$. The idea is to allocate two parameters for each possible formant frequency peak.

order the chef's recipe. All the ingredients are purchased locally and the cooking is also done locally. The Alaskan family can satisfy their gourmet craving without receiving a single food item form New York! Clearly, the last scenario captures the idea of model-based vocoders. LPC vocoders essentially deliver the recipe (i.e., the LPC parameters) for voice synthesis at the receiver end.

Practical High-Quality LP Vocoders

The simple dual-state LPC synthesis of Fig. 6.35 describes no more than the basic idea behind model-based voice codecs. The quality of LP vocoders has been greatly improved by a number of more elaborate codecs in practice. By adding a few bits, these LP-based vocoders attempt to improve the speech quality in two ways: by encoding the residual prediction error and by enhancing the excitation signal.

The most successful methods belong to the class known as code-excited linear prediction (CELP) vocoders. CELP vocoders use a codebook, a table of typical LP error (or residue) signals, which is set up a priori by designers. At the transmitter, the analyzer compares the actual prediction residue to all the entries in the codebook, chooses the entry that is the closest match, and just adds the address (code) for that entry to the bits for transmission. The synthesizer receives this code, retrieves the corresponding residue from the codebook, and uses it to modify the synthesizing output. For CELP to work well, the codebook must be big enough, requiring more transmission bits. The FS-1016 vocoder is an improvement over FS-1015 and provides good quality, natural-sounding speech at 4.8 kbit/s.[25] More modern variants include the RPE-LTP (regular pulse excitation, long-term prediction) LPC codec used in GSM cellular systems, the algebraic CELP (ACELP), the relaxed CELP (RCELP), the Qualcomm CELP (QCELP) in CDMA cellular phones, and vector-sum excited linear prediction (VSELP). Their data rates range from as low as 1.2 kbit/s to 13 kbit/s (full-rate GSM). These vocoders form the basis of many modern cellular vocoders, voice over Internet Protocol (VoIP), and other ITU-T G-series standards.

Video Compression

For video and television to go digital we face a tremendous the challenge. Because of the high video bandwidth (approximately 4.2 MHz), use of direct sampling and quantization leads to an uncompressed digital video signal of roughly 150 Mbit/s. Thus, the modest compression afforded by techniques such as ADPCM and subband coding[26, 27] is insufficient. The key to video compression, as it turns out, has to do with human visual perception.

A great deal of research and development has resulted in methods to drastically reduce the digital bandwidth required for video transmission. Early compression techniques compressed video signals to approximately 45 Mbit/s (DS3). For the emerging video delivery technologies of HFC, ADSL, HDTV, and so on, however, much greater compression was required. MPEG approached this problem and developed new compression techniques, which provide network or VCR quality video at much greater levels of compression. MPEG is a joint effort of the International Standards Organizations (ISO), the International Electrotechnical Committee (IEC), and the American National Standards Institute (ANSI) X3L3 Committee.[28, 29] MPEG has a very informative website that provides extensive information on MPEG and JPEG technologies and standards (http://www.mpeg.org/index.html/). MPEG also has an industrial forum promoting the organization's products (http://www.m4if.org/).

The concept of digital video compression is based on the fact that, on the average, a relatively small number of pixels change from frame to frame. Hence, if only the changes are transmitted, the transmission bandwidth can be reduced significantly. Digitizing allows the noise-free recovery of analog signals and improves the picture quality at the receiver.

Compression reduces the bandwidth required for transmission and the amount of storage for a video program and, hence, expands channel capacity. Without compression, a 2-hour digitized NTSC video program would require roughly 100 gigabytes of storage, far exceeding the capacity of any DVD disc.

There are three primary MPEG standards in use:

MPEG-1:	Used for VCR-quality video and storage on video CD (or VCD) at a data rate of 1.5 Mbit/s. These VCDs were quite popular throughout Asia (except Japan). MPEG-1 decoders are available on most computers. VCD is also a very popular format for karaoke.
MPEG-2:	Supports diverse video coding applications for transmissions ranging in quality from VCR to high-definition TV (HDTV), depending on data rate. It offers 50:1 compression of raw video. MPEG-2 is a highly popular format used in DVD, HDTV, terrestrial digital video broadcasting (DVB-T), and digital video broadcasting by satellite (DVB-S).
MPEG-4	Provides multimedia (audio, visual, or audiovisual) content streaming over different bandwidths including internet. MPEG-4 is supported by Microsoft Windows Media Player, Real Networks, and Apple's Quicktime and iPod. MPEG-4 recently converged with an ITU-T standard known as H.264, to be discussed later.

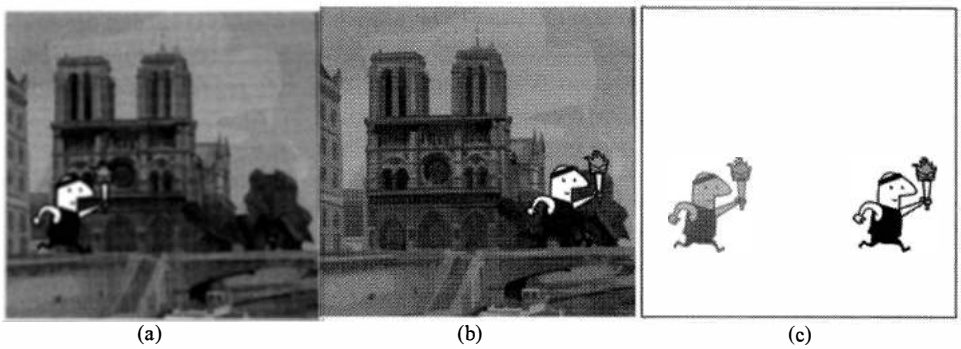
The power of video compression is staggering. By comparison, NTSC broadcast television in digital form requires 45 to 120 Mbit/s, whereas MPEG-2 requires 1.5 to 15 Mbit/s. On the other hand HDTV would require 800 Mbit/s uncompressed which, under MPEG-2 compression, will transmit at 19.39 Mbit/s.

There are two types of MPEG compression, which eliminate redundancies in the audiovisual signals that are not perceptible by the listener or the viewer:

1. Video
   - Temporal or *interframe* compression by predicting interframe motion and removing interframe redundancy
   - Spatial or intraframe compression, which forms a block identifier for a group of pixels having the same characteristics (color, intensity, etc.) for each frame. Only the block identifier is transmitted.
2. Audio, which uses a psychoacoustic model of masking effects.

The basis for video compression is to remove redundancy in the video signal stream. As an example of interframe redundancy, consider Fig. 6.36a and b. In Fig. 6.36a the runner is in position *A* and in Fig. 6.36b he is in position *B*. Note that the background (cathedral, buildings, and bridge) remains essentially unchanged from frame to frame. Figure 6.36c represents the nonredundant information for transmission; that is, the change between the two frames. The runner image on the left represents the blocks of frame 1 that are replaced by background in frame 2. The runner image on the right represents the blocks of frame 1 that replace the background in frame 2.

Video compression starts with an encoder, which converts the analog video signal from the video camera to a digital format on a pixel-by-pixel basis. Each video frame is divided into 8×8 pixel blocks, which are analyzed by the encoder to determine which blocks must be transmitted, that is, which blocks have significant changes from frame to frame. This process takes place in two stages:



(a) (b) (c)

Figure 6.36
(a) Frame 1.
(b) Frame 2.
(c) Information transferred between frames 1 and 2.

1. Motion estimation and compensation. Here a motion estimator identifies the areas or groups of blocks from a preceding frame that match corresponding areas in the current frame and sends the magnitude and direction of the displacement to a predictor in the decoder. The frame difference information is called the residual.
2. Transforming the residual on a block-by-block basis into more compact form.

The encoded residual signal is transformed into a more compact form by means of a discrete cosine transform (DCT) (see Sec. 6.5.2 in Haskel et al.,[28]), which uses a numerical value to represent each pixel and normalizes that value for more efficient transmission. The DCT is of the form

$$F(j,k) = \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} f(n,m)\cos\left[\frac{(2n+1)j\pi}{2N}\right]\cos\left[\frac{(2m+1)k\pi}{2N}\right]$$

where $f(n, m)$ is the value assigned to the block in the (n, m) position. The inverse transform is

$$f(n,m) = \frac{1}{N^2}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} F(j,k)\cos\left[\frac{(2n+1)j\pi}{2N}\right]\cos\left[\frac{(2m+1)k\pi}{2N}\right]$$

The DCT is typically multiplied, for an 8×8 block, by the expression $C(j)C(k)/4$, where

$$C(x) = \begin{cases} \dfrac{1}{\sqrt{2}} & \text{for } x = 0 \\ 1 & \text{otherwise} \end{cases}$$

Tables 6.2 and 6.3 depict the pixel block values before and after the DCT. One can notice from Table 6.3 that there are relatively few meaningful elements, that is, elements with significant values relative to the values centered about the 0, 0 position. Because of this, most of the matrix values may be assumed to be zero, and, upon inverse transformation, the original values are quite accurately reproduced. This process reduces the amount of data that must be transmitted greatly, perhaps by a factor of 8 to 10 on the average. Note that the size of the transmitted residual may be that of an individual block or, at the other extreme, that of the entire picture.

The transformed matrix values of a block (Table 6.4) are normalized so that most of the values in the block matrix are less than 1. Then the resulting normalized matrix is quantized to

TABLE 6.2
8 × 8 Pixel Block Residual

	n							
	158	158	158	163	161	161	162	162
	157	157	157	162	163	161	162	162
	157	157	157	160	161	161	161	161
	155	155	155	162	162	161	160	159
m	159	159	159	160	160	162	161	159
	156	156	156	158	163	160	155	150
	156	156	156	159	156	153	151	144
	155	155	155	155	153	149	144	139

TABLE 6.3
Transformed 8 × 8 Pixel Block Residual DCT Coefficients

	i							
	1259.6	1.0	−12.1	5.2	2.1	1.7	−2.7	−1.3
	22.6	−17.5	6.2	−3.2	2.9	−0.1	−0.4	−1.2
	−10.9	9.3	−1.6	−1.5	0.2	0.9	−0.6	0.1
	7.1	−1.9	−0.2	1.5	−0.9	−0.1	0.0	0.3
k	−0.6	0.8	1.5	−1.6	−0.1	0.7	0.6	−1.3
	−1.8	−0.2	−1.6	−0.3	0.8	1.5	−1.0	−1.0
	−1.3	0.4	−0.3	1.5	−0.5	−1.7	1.1	0.8
	2.6	1.6	3.8	−1.8	−1.9	1.2	0.6	−0.4

TABLE 6.4
Normalized and Quantized Residual DCT Coefficients

	jn							
	21	0	−1	0	0	0	0	0
	2	−1	0	0	0	0	0	0
	−1	1	0	0	0	0	0	0
k	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

obtain Table 6.4. Normalization is accomplished by a dynamic matrix of multiplicative values, which are applied element by element to the transformed matrix. The normalized matrix of Table 6.4 is the block information transmitted to the decoder. The denormalized matrix pictured in Table 6.5 and the reconstructed (inverse-transformed) residual in Table 6.6 are determined by the decoder. The transformation proceeds in a zigzag pattern, as illustrated in Fig. 6.37.

MPEG approaches the motion estimation and compensation to remove temporal (frame-to-frame) redundancy in a unique way. MPEG uses three types of frame, the intraframe or I-frame (sometimes called the independently coded or intracoded frame), the predicted (predictive) or

TABLE 6.5
Denormalized DCT Coefficients

	j/n							
	1260	0	−12	0	0	0	0	0
	23	−18	0	0	0	0	0	0
	−11	10	0	0	0	0	0	0
k	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

TABLE 6.6
Inverse DCT Coefficients Reconstructed Residual

	n							
	158	158	158	163	161	161	162	162
	157	157	157	162	163	161	162	162
	157	157	157	160	161	161	161	161
m	155	155	155	162	162	161	160	159
	159	159	159	160	160	162	161	159
	156	156	156	158	163	160	155	150
	156	156	156	159	156	153	151	144
	155	155	155	155	153	149	144	139

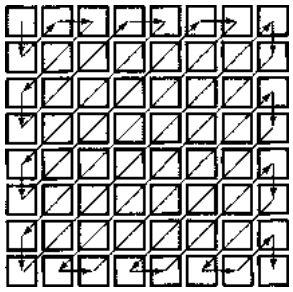


Figure 6.37 Zigzag DCT coefficient scanning pattern.

P-frame, and the bidirectionally predictive frame or B-frame. The P-frames are predicted from the I-frames. The B-frames are bidirectionally predicted from either past or future frames. An I-frame and one or more P-frames and B-frames make up the basic MPEG processing pattern, called a group of pictures (GOP). Most of the frames in an MPEG compressed image are B-frames. The I-frame provides the initial reference for the frame differences to start the MPEG encoding process. Note that the bidirectional aspect of the procedure introduces a delay in the transmission of the frames. This is because the GOP is transmitted as a unit and, hence, transmission cannot start until the GOP is complete (Fig. 6.38). The details of the procedure are beyond the scope of this text. There are many easily accessible books that cover this subject in detail. In addition, one may find numerous references to MPEG compression and HDTV on the internet.

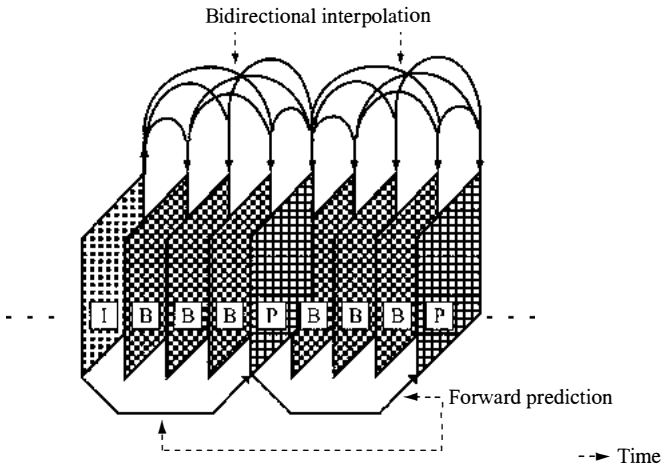


Figure 6.38
MPEG temporal frame structure.

Other Video Compression Standards

We should mention that in addition to MPEG, there is a parallel attempt by ITU-T to standardize video coding. These standards apply similar concepts for video compression. Today, the well-known ITU-T video compression standards are the H.26x series, including H.261, H.263, and H.264. H.261 was developed for transmission of video at a rate of multiples of 64 kbit/s in applications such as videophone and videoconferencing. Similar to MPEG compression, H.261 uses motion-compensated temporal prediction.

H.263 was designed for very low bit rate coding applications, such as videoconferencing. It uses block motion-compensated DCT structure for encoding.[30] Based on H.261, H.263 is better optimized for coding at low bit rates and achieves much higher efficiency than H.261 encoding. Flash Video, a highly popular format for video sharing on many web engines such as YouTube and MySpace, uses a close variant of the H.263 codec called the Sorenson Spark codec.

In fact, H.264 represents a recent convergence between ITU-T and MPEG and is a joint effort of the two groups. Also known as MPEG-4 Part 10, H.264 typically outperforms MPEG-2 by cutting the data rate nearly in half. This versatile standard supports video applications over multiple levels of bandwidth and quality, including, mobile phone service at 50 to 60 kbit/s, Internet/standard definition video at 1 to 2 Mbit/s, and high-definition video at 5 to 8 Mbit/s. H.264 is also supported in many other products and applications including iPod, direct broadcasting satellite TV, some regional terrestrial digital TV, Mac OS X (Tiger), and Sony's Playstation Portable.

A Note on High-Definition Television (HDTV)

Utilizing MPEG-2 for video compression, high-definition television (HDTV) is one of the advanced television (ATV) functions along with 525-line compressed video for direct broadcast satellite (DBS) or cable. The concept of HDTV appeared in the late 1970s. Early development work was performed primarily in Japan based on an analog system. In the mid-1980s it became apparent that the bandwidth requirements of an analog system would be excessive, and work began on a digital system that could utilize the 6 MHz bandwidth of NTSC television. In the early 1990s seven digital systems were proposed, but testing indicated that none would be highly satisfactory. Therefore, in 1993 the FCC suggested the formation of an industrial "Grand Alliance" (GA) to develop a common HDTV standard. In December 1997, Standard

A/53 for broadcast transmission, proposed by the Advanced Television Systems Committee (ATSC), was finalized by the FCC in the United States.

The GA HDTV standard is based on a 16:9 aspect ratio (motion picture aspect ratio) rather than the 4:3 aspect ratio of NTSC television. HDTV uses MPEG-2 compression at 19.39 Mbit/s and a digital modulation format called 8-VSB (vestigial sideband), which uses an eight-amplitude-level symbol to represent 3 bits of information. Transmission is in 207-byte blocks, which include 20 parity bytes for Reed-Solomon forward error correction. The remaining 187-byte packet format is a subset of the MPEG-2 protocol and includes headers for timing, switching, and other transmission control.

The Advanced Television Systems Group, the successor to the Grand Alliance, has been developing standards and recommended practices for HDTV. These are found, along with a great deal of other information, on their website: http://www.atsc.org/.

6.9 MATLAB EXERCISES

In the MATLAB exercises of this section, we provide examples of signal sampling, signal reconstruction from samples, uniform quantization, pulse-coded modulation (PCM), and delta modulation (DM).

Sampling and Reconstruction of Lowpass Signals

In the sampling example, we first construct a signal $g(t)$ with two sinusoidal components of 1-second duration; their frequencies are 1 and 3 Hz. Note, however, that when the signal duration is infinite, the bandwidth of $g(t)$ would be 3 Hz. However, the finite duration of the signal implies that the actual signal is not band-limited, although most of the signal content stays within a bandwidth of 5 Hz. For this reason, we select a sampling frequency of 50 Hz, much higher than the minimum Nyquist frequency of 6 Hz. The MATLAB program, `Exsample.m`, implements sampling and signal reconstruction. Figure 6.39 illustrates the original signal, its uniform samples at the 50 Hz sampling rate, and the frequency response of the sampled signal. In accordance with our analysis of Section 6.1, the spectrum of the sampled signal $g_T(t)$ consists of the original signal spectrum periodically repeated every 50 Hz.

```
% (Exsample.m)
% Example of sampling, quantization, and zero-order hold
clear;clf;
td=0.002;       %original sampling rate 500 Hz
t=[0:td:1.];    %time interval of 1 second
xsig=sin(2*pi*t)-sin(6*pi*t);   % 1Hz+3Hz sinusoids
Lsig=length(xsig);

ts=0.02;        %new sampling rate = 50Hz.
Nfactor=ts/td;
% send the signal through a 16-level uniform quantizer
[s_out,sq_out,sqh_out,Delta,SQNR]=sampandquant(xsig,16,td,ts);
%   receive 3 signals:
%       1. sampled signal s_out
%       2. sampled and quantized signal sq_out
%       3. sampled, quantized, and zero-order hold signal sqh_out
%
```

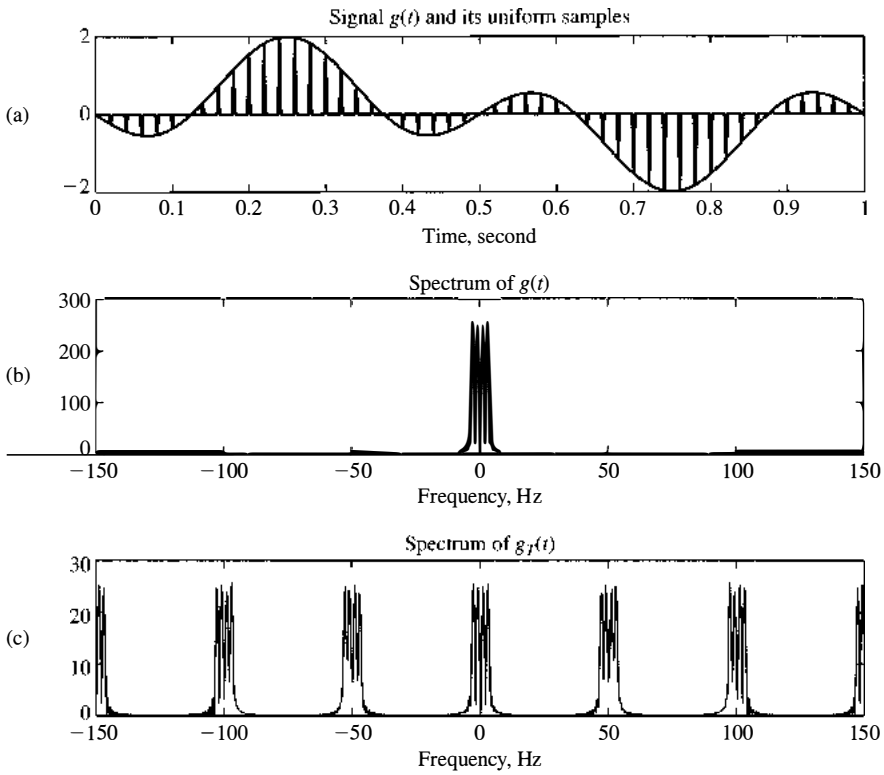


Figure 6.39
The relationship between the original signal and the ideal uniformly sampled signal in the time (a) and frequency (b, c) domains.

```
%   calculate the Fourier transforms
 Lfft=2^ceil(log2(Lsig)+1);
 Fmax=1/(2*td);
 Faxis=linspace(-Fmax,Fmax,Lfft);
 Xsig=fftshift(fft(xsig,Lfft));
 S_out=fftshift(fft(s_out,Lfft));
%   Examples of sampling and reconstruction using
%      a) ideal impulse train through LPF
%      b) flat top pulse reconstruction through LPF
%   plot the original signal and the sample signals in time
%   and frequency domain
figure(1);
subplot(311); sfig1a=plot(t,xsig,'k');
hold on; sfig1b=plot(t,s_out(1:Lsig),'b'); hold off;
set(sfig1a,'Linewidth',2); set(sfig1b,'Linewidth',2.);
xlabel('time (sec)');
title('Signal {\it g}({\it t}) and its uniform samples');
subplot(312); sfig1c=plot(Faxis,abs(Xsig));
xlabel('frequency (Hz)');
axis([-150 150 0 300])
set(sfig1c,'Linewidth',1);  title('Spectrum of {\it g}({\it t})');
subplot(313); sfig1d=plot(Faxis,abs(S_out));
xlabel('frequency (Hz)');
axis([-150 150 0 300/Nfactor])
```

```
set(sfig1c,'Linewidth',1);  title('Spectrum of {\it g}_T({\it t})');
%   calculate the reconstructed signal from ideal sampling and
%   ideal LPF
%   Maximum LPF bandwidth equals to BW=floor((Lfft/Nfactor)/2);
 BW=10;              %Bandwidth is no larger than 10Hz.
 H_lpf=zeros(1,Lfft);H_lpf(Lfft/2-BW:Lfft/2+BW-1)=1;  %ideal LPF
 S_recv=Nfactor*S_out.*H_lpf;             % ideal filtering
 s_recv=real(ifft(fftshift(S_recv)));     % reconstructed f-domain
 s_recv=s_recv(1:Lsig);                   % reconstructed t-domain
%   plot the ideally reconstructed signal in time
%   and frequency domain
figure(2)
subplot(211); sfig2a=plot(Faxis,abs(S_recv));
xlabel('frequency (Hz)');
axis([-150 150 0 300]);
title('Spectrum of ideal filtering (reconstruction)');
subplot(212); sfig2b=plot(t,xsig,'k-.',t,s_recv(1:Lsig),'b');
legend('original signal','reconstructed signal');
xlabel('time (sec)');
title('original signal versus ideally reconstructed signal');
set(sfig2b,'Linewidth',2);
%   non-ideal reconstruction
 ZOH=ones(1,Nfactor);
 s_ni=kron(downsample(s_out,Nfactor),ZOH);
 S_ni=fftshift(fft(s_ni,Lfft));
 S_recv2=S_ni.*H_lpf;             % ideal filtering
 s_recv2=real(ifft(fftshift(S_recv2)));     % reconstructed f-domain
 s_recv2=s_recv2(1:Lsig);                   % reconstructed t-domain
%   plot the ideally reconstructed signal in time
%   and frequency domain
figure(3)
subplot(211); sfig3a=plot(t,xsig,'b',t,s_ni(1:Lsig),'b');
xlabel('time (sec)');
title('original signal versus flat-top reconstruction');
subplot(212); sfig3b=plot(t,xsig,'b',t,s_recv2(1:Lsig),'b--');
legend('original signal','LPF reconstruction');
xlabel('time (sec)');
set(sfig3a,'Linewidth',2); set(sfig3b,'Linewidth',2);
title('original and flat-top reconstruction after LPF');
```

To construct the original signal $g(t)$ from the impulse sampling train $g_T(t)$, we applied an ideal low-pass filter with bandwidth 10 Hz in the frequency domain. This corresponds to the interpolation using the ideal sinc function as shown in Sec. 6.1.1. The resulting spectrum, as shown in Fig. 6.40, is nearly identical to the original message spectrum of $g(t)$. Moreover, the time domain signal waveforms are also compared in Fig. 6.40 and show near perfect match.

In our last exercise in sampling and reconstruction, given in the same program, we use a simple rectangular pulse of width T_s (sampling period) to reconstruct the original signal from the samples (Fig. 6.41). A low-pass filter is applied on the rectangular reconstruction and also shown in Fig. 6.41. It is clear from comparison to the original source signal that the

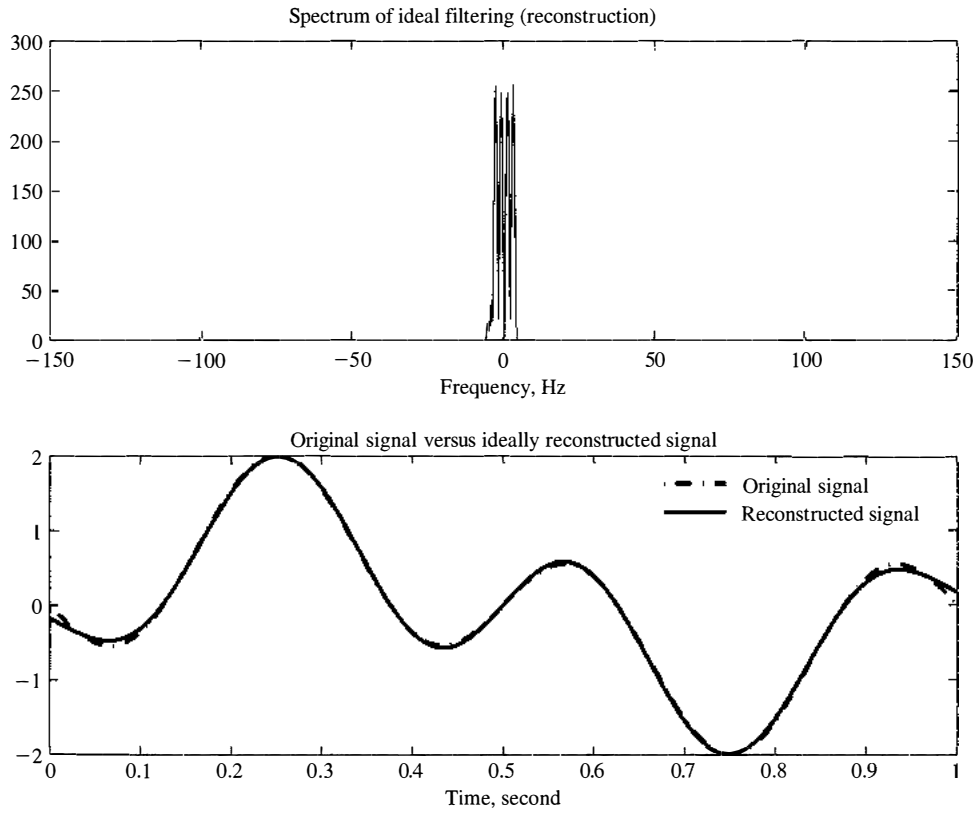


Figure 6.40 Reconstructed signal spectrum and waveform from applying the ideal impulse sampling and ideal low-pass filter reconstruction.

recovered signal is still very close to the original signal $g(t)$. This is because we have chosen a high sampling rate such that $T_p = T_s$ is so small that the approximation of Eq. (6.17) holds. Certainly, based on our analysis, by applying the low-pass equalization filter of Eq. (6.16), the reconstruction error can be greatly reduced.

PCM Illustration

The uniform quantization of an analog signal using L quantization levels can be implemented by the MATLAB function `uniquan.m`.

```
% (uniquan.m)
function [q_out,Delta,SQNR]=uniquan(sig_in,L)
%   Usage
%   [q_out,Delta,SQNR]=uniquan(sig_in,L)
%   L   -    number of uniform quantization levels
%   sig_in -    input signal vector
%   Function outputs:
%             q_out - quantized output
%             Delta - quantization interval
%             SQNR  - actual signal to quantization noise ratio
sig_pmax=max(sig_in);   % finding the positive peak
sig_nmax=min(sig_in);   % finding the negative peak
Delta=(sig_pmax-sig_nmax)/L;    % quantization interval
```

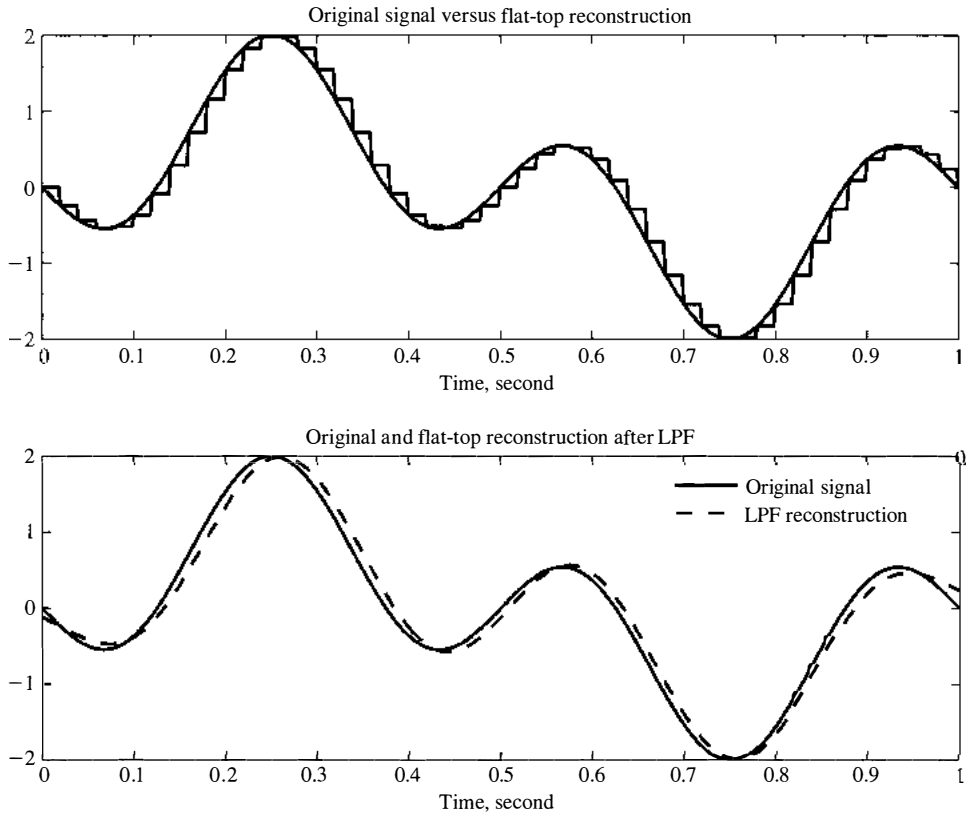


Figure 6.41 Reconstructed signal spectrum and waveform from applying the simple rectangular reconstruction pulse (Fig. 6.6) followed by LPF without equalization.

```
q_level=sig_nmax+Delta/2:Delta:sig_pmax-Delta/2;   % define Q-levels
L_sig=length(sig_in);   % find signal length
sigp=(sig_in-sig_nmax)/Delta+1/2;   % convert into 1/2 to L+1/2 range
qindex=round(sigp);     % round to 1, 2, ... L levels
qindex=min(qindex,L);   % eleminate L+1 as a rare possibility
q_out=q_level(qindex);  % use index vector to generate output
SQNR=20*log10(norm(sig_in)/norm(sig_in-q_out)); %actual SQNR value
end
```

The function `sampandquant.m` executes both sampling and uniform quantization simultaneously. The sampling period `ts` is needed, along with the number `L` of quantization levels, to generate the sampled output `s_out`, the sampled and quantized output `sq_out`, and the signal after sampling, quantizing, and zero-order-hold `sqh_out`.

```
% (sampandquant.m)
function [s_out,sq_out,sqh_out,Delta,SQNR]=sampandquant(sig_in,L,td,ts)
%   Usage
%       [s_out,sq_out,sqh_out,Delta,SQNR]=sampandquant(sig_in,L,td,fs)
%   L   -    number of uniform quantization levels
%   sig_in -    input signal vector
%   td  -    original signal sampling period of sig_in
```

```
%   ts  -    new sampling period
% NOTE:  td*fs must be a positive integer;
%   Function outputs:
%            s_out  - sampled output
%            sq_out - sample-and-quantized output
%            sqh_out- sample,quantize,and hold output
%             Delta - quantization interval
%             SQNR  - actual signal to quantization noise ratio

if (rem(ts/td,1)==0)
nfac=round(ts/td);
p_zoh=ones(1,nfac);
s_out=downsample(sig_in,nfac);
[sq_out,Delta,SQNR]=uniquan(s_out,L);
s_out=upsample(s_out,nfac);
sqh_out=kron(sq_out,p_zoh);
sq_out=upsample(sq_out,nfac);
else
    warning('Error! ts/td is not an integer!');
    s_out=[];sq_out=[];sqh_out=[];Delta=[];SQNR=[];
end
end
```

The MATLAB program `ExPCM.m` provides a numerical example that uses these two MATLAB functions to generate PCM signals.

```
% (ExPCM.m)
% Example of sampling, quantization, and zero-order hold
clear;clf;
td=0.002;       %original sampling rate 500 Hz
t=[0:td:1.];    %time interval of 1 second
xsig=sin(2*pi*t)-sin(6*pi*t);   % 1Hz+3Hz sinusoids
Lsig=length(xsig);
Lfft=2^ceil(log2(Lsig)+1);
Xsig=fftshift(fft(xsig,Lfft));
Fmax=1/(2*td);
Faxis=linspace(-Fmax,Fmax,Lfft);
ts=0.02;        %new sampling rate = 50Hz.
Nfact=ts/td;
% send the signal through a 16-level uniform quantizer
[s_out,sq_out,sqh_out1,Delta,SQNR]=sampandquant(xsig,16,td,ts);
%   obtained the PCM signal which is
%       - sampled, quantized, and zero-order hold signal sqh_out
%   plot the original signal and the PCM signal in time domain
figure(1);
subplot(211);sfig1=plot(t,xsig,'k',t,sqh_out1(1:Lsig),'b');
set(sfig1,'Linewidth',2);
title('Signal {\it g}({\it t}) and its 16 level PCM signal')
```

```
xlabel('time (sec.)');
% send the signal through a 16-level uniform quantizer
[s_out,sq_out,sqh_out2,Delta,SQNR]=sampandquant(xsig,4,td,ts);
%   obtained the PCM signal which is
%       - sampled, quantized, and zero-order hold signal sqh_out
%   plot the original signal and the PCM signal in time domain
subplot(212);sfig2=plot(t,xsig,'k',t,sqh_out2(1:Lsig),'b');
set(sfig2,'Linewidth',2);
title('Signal {\it g}({\it t}) and its 4 level PCM signal')
xlabel('time (sec.)');

 Lfft=2^ceil(log2(Lsig)+1);
 Fmax=1/(2*td);
 Faxis=linspace(-Fmax,Fmax,Lfft);
 SQH1=fftshift(fft(sqh_out1,Lfft));
 SQH2=fftshift(fft(sqh_out2,Lfft));
% Now use LPF to filter the two PCM signals
 BW=10;              %Bandwidth is no larger than 10Hz.
 H_lpf=zeros(1,Lfft);H_lpf(Lfft/2-BW:Lfft/2+BW-1)=1;  %ideal LPF
 S1_recv=SQH1.*H_lpf;             % ideal filtering
 s_recv1=real(ifft(fftshift(S1_recv)));     % reconstructed f-domain
 s_recv1=s_recv1(1:Lsig);                   % reconstructed t-domain
  S2_recv=SQH2.*H_lpf;             % ideal filtering
 s_recv2=real(ifft(fftshift(S2_recv)));     % reconstructed f-domain
 s_recv2=s_recv2(1:Lsig);                   % reconstructed t-domain
 % Plot the filtered signals against the original signal
figure(2)
 subplot(211);sfig3=plot(t,xsig,'b-',t,s_recv1,'b-.');
 legend('original','recovered')
set(sfig3,'Linewidth',2);
title('Signal {\it g}({\it t}) and filtered 16-level PCM signal')
xlabel('time (sec.)');
subplot(212);sfig4=plot(t,xsig,'b-',t,s_recv2(1:Lsig),'b-.');
 legend('original','recovered')
set(sfig4,'Linewidth',2);
title('Signal {\it g}({\it t}) and filtered 4-level PCM signal')
xlabel('time (sec.)');
```

In the first example, we maintain the 50 Hz sampling frequency and utilize $L = 16$ uniform quantization levels. The resulting PCM signal is shown in Fig. 6.42. This PCM signal can be low-pass-filtered at the receiver and compared against the original message signal, as shown in Fig. 6.43. The recovered signal is seen to be very close to the original signal $g(t)$.

To illustrate the effect of quantization, we next apply $L = 4$ PCM quantization levels. The resulting PCM signal is again shown in Fig. 6.42. The corresponding signal recovery is given in Fig. 6.43. It is very clear that smaller number of quantization levels ($L = 4$) leads to much larger approximation error.

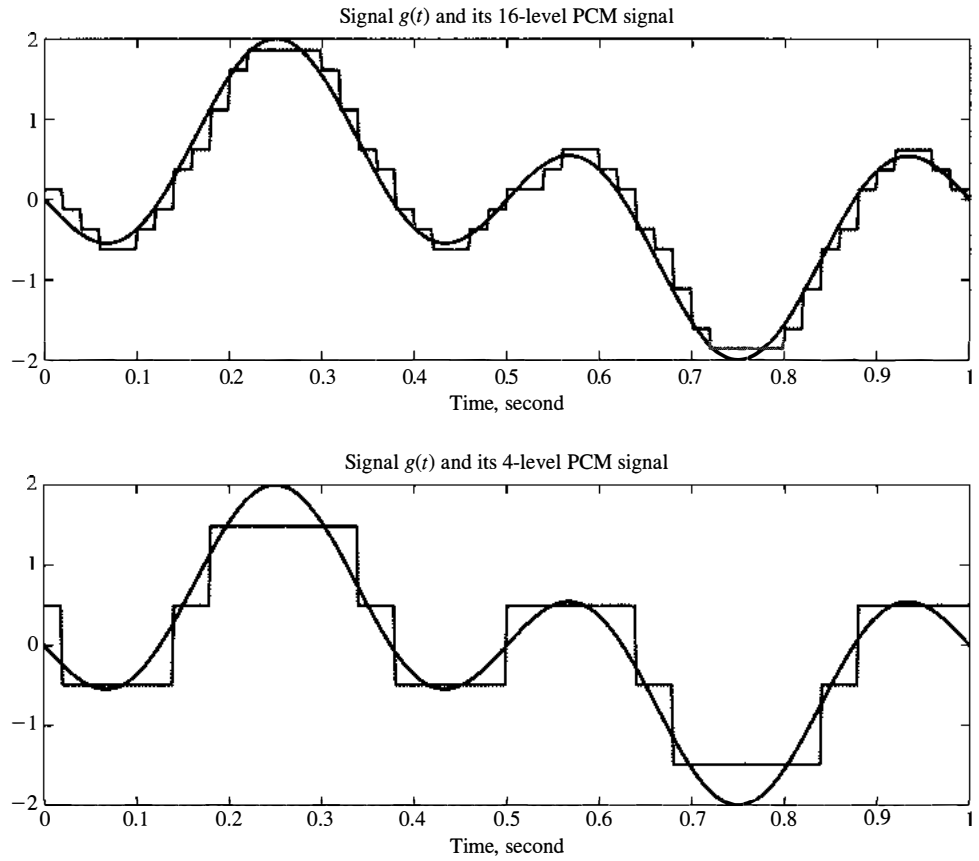


Figure 6.42 Original signal and the PCM signals with different numbers of quantization levels.

Delta Modulation

Instead of applying PCM, we illustrate the practical effect of step size selection Δ in the design of DM encoder. The basic function to implement DM is given in `deltamod.m`.

```
% (deltamod.m)
function s_DMout= deltamod(sig_in,Delta,td,ts)
%   Usage
%       s_DMout = deltamod(xsig,Delta,td,ts))
%   Delta  -    DM stepsize
%   sig_in -    input signal vector
%   td  -    original signal sampling period of sig_in
%   ts  -    new sampling period
% NOTE:  td*fs must be a positive integer;
%   Function outputs:
%           s_DMout  - DM sampled output
if (rem(ts/td,1)==0)
nfac=round(ts/td);
p_zoh=ones(1,nfac);
s_down=downsample(sig_in,nfac);
Num_it=length(s_down);
```

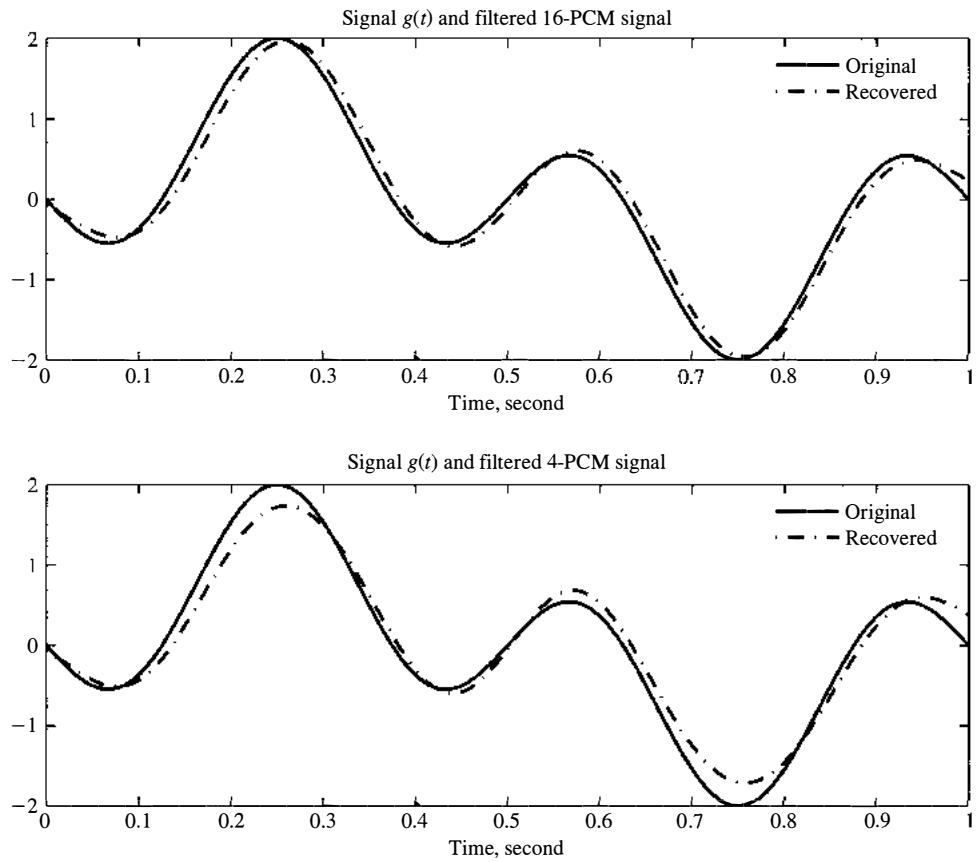


Figure 6.43 Comparison between the original signal and the PCM signals after low-pass filtering to recover the original message.

```
s_DMout(1)=-Delta/2;
for k=2:Num_it
    xvar=s_DMout(k-1);
    s_DMout(k)=xvar+Delta*sign(s_down(k-1)-xvar);
end
s_DMout=kron(s_DMout,p_zoh);
else
    warning('Error! ts/td is not an integer!');
    s_DMout=[];
end
end
```

To generate DM signals with different step sizes, we apply the same signal $g(t)$ as used in the PCM example. The MATLAB program `ExDM.m` applies three step sizes: $\Delta_1 = 0.2$, $\Delta_2 = 2\Delta_1$, and $\Delta_3 = 4\Delta_1$.

```
%  (ExDM.m)
% Example of sampling, quantization, and zero-order hold
clear;clf;
td=0.002;       %original sampling rate 500 Hz
```

```
t=[0:td:1.];    %time interval of 1 second
xsig=sin(2*pi*t)-sin(6*pi*t);   % 1Hz+3Hz sinusoids
Lsig=length(xsig);
ts=0.02;        %new sampling rate = 50Hz.
Nfact=ts/td;
% send the signal through a 16-level uniform quantizer
Delta1=0.2;     % First select a small Delta=0.2 in DM
s_DMout1=deltamod(xsig,Delta1,td,ts);
%   obtained the DM signal
%   plot the original signal and the DM signal in time domain
figure(1);
subplot(311);sfig1=plot(t,xsig,'k',t,s_DMout1(1:Lsig),'b');
set(sfig1,'Linewidth',2);
title('Signal {\it g}({\it t}) and DM signal')
xlabel('time (sec.)'); axis([0 1 -2.2 2.2]);
%
%  Apply DM again by doubling the Delta
Delta2=2*Delta1;          %
s_DMout2=deltamod(xsig,Delta2,td,ts);
%   obtained the DM signal
%   plot the original signal and the DM signal in time domain
subplot(312);sfig2=plot(t,xsig,'k',t,s_DMout2(1:Lsig),'b');
set(sfig2,'Linewidth',2);
title('Signal {\it g}({\it t}) and DM signal with doubled stepsize')
xlabel('time (sec.)'); axis([0 1 -2.2 2.2]);
%
Delta3=2*Delta2;          % Double the DM Delta again.
s_DMout3=deltamod(xsig,Delta3,td,ts);
%   plot the original signal and the DM signal in time domain
subplot(313);sfig3=plot(t,xsig,'k',t,s_DMout3(1:Lsig),'b');
set(sfig3,'Linewidth',2);
title('Signal {\it g}({\it t}) and DM signal with quadrupled
    stepsize')
xlabel('time (sec.)'); axis([0 1 -2.2 2.2]);
```

To illustrate the effect of DM, the resulting signals from the DM encoder are shown in Fig. 6.44. This example clearly shows that when the step size is too small (Δ_1), there is a severe overloading effect as the original signal varies so fast that the small step size is unable to catch up. Doubling the DM step size clearly solves the overloading problem in this example. However, quadrupling the step size (Δ_3) would lead to unnecessarily large quantization error. This example thus confirms our earlier analysis that a careful selection of the DM step size is critical.

REFERENCES

1. D. A. Linden, "A discussion of sampling theorem," *Proc. IRE*, vol. 47, no.7, pp. 1219–1226, July 1959.
2. H. P. Kramer, "A Generalized Sampling Theorem," *J. Math. Phys.*, vol. 38, pp. 68–72, 1959.

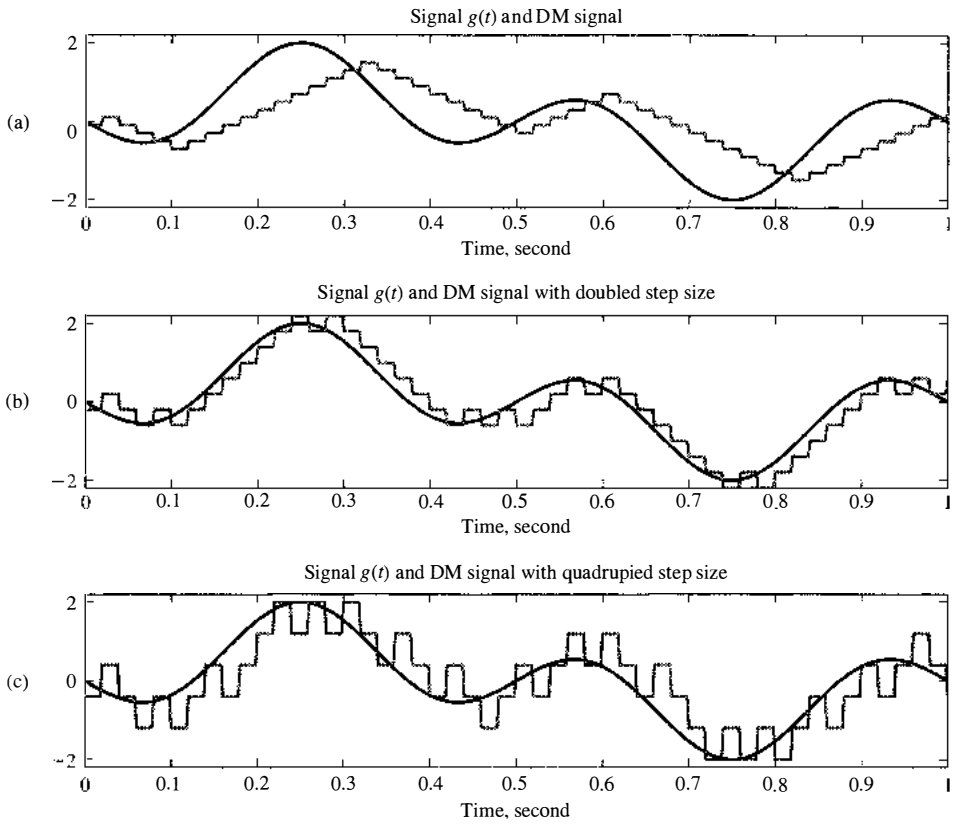


Figure 6.44 Examples of delta modulation output with three different step sizes: (a) small step size leads to overloading; (b) reasonable step size; (c) large step size causes large quantization errors.

3. W. R. Bennett, *Introduction to Signal Transmission*, McGraw-Hill, New York, 1970.
4. W. S. Anglin and J. Lambek, *The Heritage of Thales*, Springer, Berlin, 1995.
5. B. Smith, "Instantaneous Companding of Quantized Signals," *Bell Syst. Tech. J.*, vol. 36, pp. 653–709, May 1957.
6. ITU-T Standard Recommendation G.711, English, 1989.
7. ITU-T Standard Recommendation G.726, English, 1990.
8. C. L. Dammann, L. D. McDaniel, and C. L. Maddox, "D-2 Channel Bank Multiplexing and Coding," *Bell Syst. Tech. J.*, vol. 51, pp. 1675–1700, Oct. 1972.
9. K. W. Cattermole, *Principles of Pulse-Code Modulation*, Ilife, England, 1969.
10. Bell Telephone Laboratories, *Transmission Systems for Communication*, 4th ed., Bell, Murray Hill, NJ, 1970.
11. E. L. Gruenberg, *Handbook of Telemetry and Remote Control*, McGraw-Hill, New York, 1967.
12. J. B. O'Neal, Jr., "Delta Modulation Quantizing Noise: Analytical and Computer Simulation Results for Gaussian and Television Input Signals," *Bell Syst. Tech. J.*, pp. 117–141, Jan. 1966.
13. F. de Jager, "Delta Modulation, a Method of PCM Transmission Using the 1-Unit Code," *Philips Res. Rep.*, no. 7, pp. 442–466, 1952.
14. A. Tomozawa and H. Kaneko, "Companded Delta Modulation for Telephone Transmission," *IEEE Trans. Commun. Technol.*, vol. CT-16, pp. 149–157, Feb. 1968.
15. B. S. Atal, "Predictive Coding of Speech Signals at Low Bit Rates," *IEEE Trans. Commun.*, vol. COMM-30, pp. 600–614, 1982.
16. J. P. Campbell and T. E. Tremain, "Voiced/Unvoiced Classification of Speech with Applications to the U.S. Government LPC-10E Algorithm," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Tokyo, pp. 473–476, 1986.
17. A. Gersho, "Advances in Speech and Audio Compression," *Proc. IEEE*, vol. 82, pp. 900–918, 1994.

18. L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, NJ, 1978.
19. Lajos Hanzo, Jason Woodward, and Clare Sommerville, *Voice Compression and Communications*, Wiley, Hoboken; NJ, 2001.
20. N. Levinson, "The Wiener rms Error Criterion in Filter Design and Prediction," *J. Math. Phys.*, vol. 25, pp. 261–278, 1947.
21. A. H. Sayed, *Fundamentals of Adaptive Filtering*, Wiley-IEEE Press, Hoboken, NJ, 2003.
22. J. Y. Stein, *Digital Signal Processing: A Computer Science Perspective*, Wiley, Hoboken, NJ, 2000.
23. K. K. Paliwal and B. W. Kleijn, "Quantization of LPC Parameters," in *Speech Coding and Synthesis,* W. B. Kleijn and K. K. Paliwal, Eds. Elsevier Science, Amsterdam, 1995.
24. T. E. Tremain, "The Government Standard Linear Predictive Coding Algorithm LPC-10," *Speech Technol.*, 40–49, 1982.
25. M. R. Schroeder and B. S. Atal, "Code-Excited Linear Prediction (CELP): High-Quality Speech at Very Low Bit Rates," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*, vol. 10, pp. 937–940, 1985.
26. S. Mallat, "A Theory of Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. Pattern Anal. Machine Intel.*, vol. 11, pp. 674–693, 1989.
27. M. J. Smith and T. P. Barnwell, "Exact Reconstruction for Tree Structured Sub-Band Coders," *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. 34, no. 3, pp. 431–441, 1986.
28. B. G. Haskel, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG-2*, Chapman & Hall, New York, 1996.
29. J. L. Mitchell, W. B. Pennebaker, C.E Fogg, and D. J. LeGall, *MPEG Video Compression Standard*, Chapman & Hall, New York, 1996.
30. ITU-T Recommendation H.263, Video Coding for Low Bit Rate Communication.

PROBLEMS

6.1-1 Figure P6.1-1 shows Fourier spectra of signals $g_1(t)$ and $g_2(t)$. Determine the Nyquist interval and the sampling rate for signals $g_1(t)$, $g_2(t)$, $g_1^2(t)$, $g_2^m(t)$, and $g_1(t)g_2(t)$.
Hint: Use the frequency convolution and the width property of the convolution.

Figure P.6.1-1



6.1-2 Determine the Nyquist sampling rate and the Nyquist sampling interval for the signals:

(a) $\text{sinc}\,(100\pi t)$

(b) $\text{sinc}^2\,(100\pi t)$

(c) $\text{sinc}\,(100\pi t) + \text{sinc}\,(50\pi t)$

(d) $\text{sinc}\,(100\pi t) + 3\,\text{sinc}^2\,(60\pi t)$

(e) $\text{sinc}\,(50\pi t)\,\text{sinc}\,(100\pi t)$

6.1-3 A signal $g(t)$ band-limited to B Hz is sampled by a periodic pulse train $p_{T_s}(t)$ made up of a rectangular pulse of width $1/8B$ second (centered at the origin) repeating at the Nyquist rate ($2B$ pulses per second). Show that the sampled signal $\overline{g}(t)$ is given by

$$\overline{g}(t) = \frac{1}{4}g(t) + \sum_{n=1}^{\infty} \frac{2}{n\pi} \sin\left(\frac{n\pi}{4}\right) g(t) \cos 4n\pi Bt$$

Show that the signal $g(t)$ can be recovered by passing $\overline{g}(t)$ through an ideal low-pass filter of bandwidth B Hz and a gain of 4.

6.1-4 A signal $g(t) = \text{sinc}^2(5\pi t)$ is sampled (using uniformly spaced impulses) at a rate of **(i)** 5 Hz; **(ii)** 10 Hz; **(iii)** 20 Hz. For each of the three cases:

(a) Sketch the sampled signal.

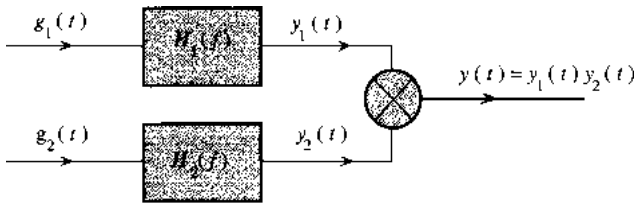
(b) Sketch the spectrum of the sampled signal.

(c) Explain whether you can recover the signal $g(t)$ from the sampled signal.

(d) If the sampled signal is passed through an ideal low-pass filter of bandwidth 5 Hz, sketch the spectrum of the output signal.

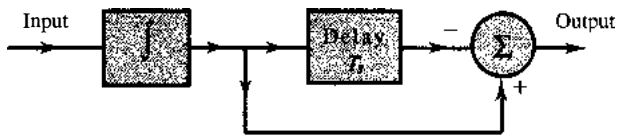
6.1-5 Signals $g_1(t) = 10^4\,\Pi(10^4 t)$ and $g_2(t) = \delta(t)$ are applied at the inputs of ideal low-pass filters $H_1(f) = \Pi(f/20{,}000)$ and $H_2(f) = \Pi(f/10{,}000)$ (Fig. P6.1-5). The outputs $y_1(t)$ and $y_2(t)$ of these filters are multiplied to obtain the signal $y(t) = y_1(t)y_2(t)$. Find the Nyquist rate of $y_1(t), y_2(t)$, and $y(t)$. Use the convolution property and the width property of convolution to determine the bandwidth of $y_1(t)y_2(t)$. See also Prob. 6.1-1.

Figure P.6.1-5



6.1-6 A zero-order hold circuit (Fig. P6.1-6) is often used to reconstruct a signal $g(t)$ from its samples.

Figure P.6.1-6



(a) Find the unit impulse response of this circuit.

(b) Find the transfer function $H(f)$ and sketch $|H(f)|$.

(c) Show that when a sampled signal $\overline{g}(t)$ is applied at the input of this circuit, the output is a staircase approximation of $g(t)$. The sampling interval is T_s.

6.1-7 **(a)** A first-order hold circuit can also be used to reconstruct a signal $g(t)$ from its samples. The impulse response of this circuit is

$$h(t) = \Delta\left(\frac{t}{2T_s}\right)$$

where T_s is the sampling interval. Consider a typical sampled signal $\overline{g}(t)$ and show that this circuit performs the linear interpolation. In other words, the filter output consists of sample tops connected by straight-line segments. Follow the procedure discussed in Sec. 6.1.1 (Fig. 6.2b).

(b) Determine the transfer function of this filter and its amplitude response, and compare it with the ideal filter required for signal reconstruction.

(c) This filter, being noncausal, is unrealizable. Suggest a modification that will make this filter realizable. How would such a modification affect the reconstruction of $g(t)$ from its samples? How would it affect the frequency response of the filter?

6.1-8 Prove that a signal cannot be simultaneously time-limited and band-limited.

Hint: Show that the contrary assumption leads to contradiction. Assume a signal simultaneously time-limited and band-limited so that $G(f) = 0$ for $|f| > B$. In this case, $G(f) = G(f)\,\Pi(f/2B')$ for $B' > B$. This means that $g(t)$ is equal to $g(t) * 2B'\text{sinc}\,(2\pi B't)$. Show that the latter cannot be time-limited.

6.2-1 The American Standard Code for Information Interchange (ASCII) has 128 characters, which are binary-coded. If a certain computer generates 100,000 characters per second, determine the following:

(a) The number of bits (binary digits) required per character.

(b) The number of bits per second required to transmit the computer output, and the minimum bandwidth required to transmit this signal.

(c) For single error detection capability, an additional bit (parity bit) is added to the code of each character. Modify your answers in parts **(a)** and **(b)** in view of this information.

6.2-2 A compact disc (CD) records audio signals digitally by using PCM. Assume that the audio signal bandwidth equals 15 kHz.

(a) If the Nyquist samples are uniformly quantized into $L = 65,536$ levels and then binary-coded, determine the number of binary digits required to encode a sample.

(b) If the audio signal has average power of 0.1 watt and peak voltage of 1 volt. Find the resulting signal-to-quantization-noise ratio (SQNR) of the uniform quantizer output in part **(a)**.

(c) Determine the number of binary digits per second (bit/s) required to encode the audio signal.

(d) For practical reasons discussed in the text, signals are sampled at a rate well above the Nyquist rate. Practical CDs use 44,100 samples per second. If $L = 65,536$, determine the number of bits per second required to encode the signal, and the minimum bandwidth required to transmit the encoded signal.

6.2-3 A television signal (video and audio) has a bandwidth of 4.5 MHz. This signal is sampled, quantized, and binary coded to obtain a PCM signal.

(a) Determine the sampling rate if the signal is to be sampled at a rate 20% above the Nyquist rate.

(b) If the samples are quantized into 1024 levels, determine the number of binary pulses required to encode each sample.

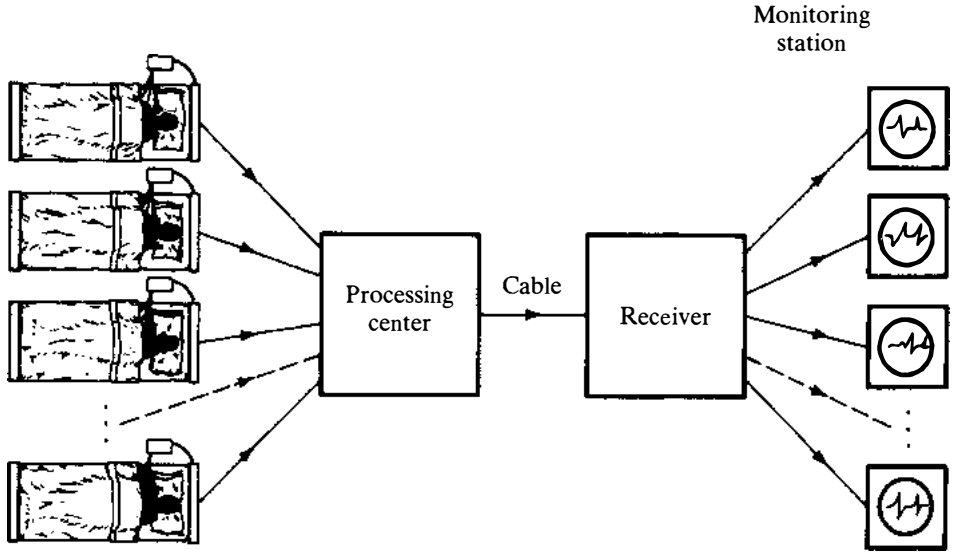
(c) Determine the binary pulse rate (bits per second) of the binary-coded signal, and the minimum bandwidth required to transmit this signal.

6.2-4 Five telemetry signals, each of bandwidth 240 Hz, are to be transmitted simultaneously by binary PCM. The signals must be sampled at least 20% above the Nyquist rate. Framing and synchronizing requires an additional 0.5% extra bits. A PCM encoder is used to convert these signals before they are time-multiplexed into a single data stream. Determine the minimum possible data rate (bits per second) that must be transmitted, and the minimum bandwidth required to transmit the multiplex signal.

6.2-5 It is desired to set up a central station for simultaneous monitoring of the electrocardiograms (ECGs) of 10 hospital patients. The data from the 10 patients are brought to a processing center over wires and are sampled, quantized, binary-coded, and time-division-multiplexed. The multiplexed

data are now transmitted to the monitoring station (Fig. P6.2-5). The ECG signal bandwidth is 100 Hz. The maximum acceptable error in sample amplitudes is 0.25% of the peak signal amplitude. The sampling rate must be at least twice the Nyquist rate. Determine the minimum cable bandwidth needed to transmit these data.

Figure P.6.2-5

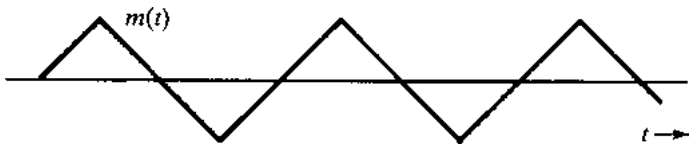


6.2-6 A message signal $m(t)$ is transmitted by binary PCM without compression. If the SQNR is required to be at least 47 dB, determine the minimum value of $L = 2^n$ required, assuming that $m(t)$ is sinusoidal. Determine the actual SQNR obtained with this minimum L.

6.2-7 Repeat Prob. 6.2-6 for $m(t)$ shown in Fig. P6.2-7.

Hint: The power of a periodic signal is its energy averaged over one cycle. In this case, however, because the signal amplitude takes on the same values every quarter cycle, the power can also be found by averaging the signal energy over a quarter cycle.

Figure P.6.2-7



6.2-8 For a PCM signal, determine L if the compression parameter $\mu = 100$ and the minimum SNR required is 45 dB. Determine the output SQNR with this value of L. Remember that L must be a power of 2, that is, $L = 2^n$ for a binary PCM.

6.2-9 A signal band-limited to 1 MHz is sampled at a rate 50% higher than the Nyquist rate and quantized into 256 levels by using a μ-law quantizer with $\mu = 255$.

(a) Determine the signal-to-quantization-noise ratio.

(b) The SQNR (the received signal quality) found in part **(a)** was unsatisfactory. It must be increased at least by 10 dB. Would you be able to obtain the desired SQNR without increasing

the transmission bandwidth if it was found that a sampling rate 20% above the Nyquist rate is adequate? If so, explain how. What is the maximum SQNR that can be realized in this way?

6.2-10 The output SQNR of a 10-bit PCM was found to be insufficient at 30 dB. To achieve the desired SNR of 42 dB, it was decided to increase the number of quantization levels L. Find the fractional increase in the transmission bandwidth required for this increase in L.

6.4-1 In a certain telemetry system, there are four analog signals $m_1(t)$, $m_2(t)$, $m_3(t)$, and $m_4(t)$. The bandwidth of $m_1(t)$ is 3.6 kHz, but for each of the remaining signals it is 1.4 kHz. These signals are to be sampled at rates no less than their respective Nyquist rates and are to be word-by-word multiplexed. This can be achieved by multiplexing the PAM samples of the four signals and then binary coding the multiplexed samples (as in the case of the PCM T1 carrier in Fig. 6.20a). Suggest a suitable multiplexing scheme for this purpose. What is the commutator frequency (in rotations per second)? *Note:* In this case you may have to sample some signal(s) at rates higher than their Nyquist rate(s).

6.4-2 Repeat Prob. 6.4-1 if there are four signals $m_1(t)$, $m_2(t)$, $m_3(t)$, and $m_4(t)$ with bandwidths 1200, 700, 300, and 200 Hz, respectively.

Hint: First multiplex m_2, m_3, and m_4 and then multiplex this composite signal with $m_1(t)$.

6.4-3 A signal $m_1(t)$ is band-limited to 3.6 kHz, and the three other signals $m_2(t)$, $m_3(t)$, and $m_4(t)$ are band-limited to 1.2 kHz each. These signals are sampled at the Nyquist rate and binary coded using 512 levels ($L = 512$). Suggest a suitable bit-by-bit multiplexing arrangement (as in Fig. 6.12). What is the commutator frequency (in rotations per second), and what is the output bit rate?

6.7-1 In a single-integration DM system, the voice signal is sampled at a rate of 64 kHz, similar to PCM. The maximum signal amplitude is normalized as $A_{\max} = 1$.

(a) Determine the minimum value of the step size σ to avoid slope overload.

(b) Determine the granular noise power N_o if the voice signal bandwidth is 3.4 kHz.

(c) Assuming that the voice signal is sinusoidal, determine S_o and the SNR.

(d) Assuming that the voice signal amplitude is uniformly distributed in the range $(-1, 1)$, determine S_o and the SNR.

(e) Determine the minimum transmission bandwidth.

7 PRINCIPLES OF DIGITAL DATA TRANSMISSION

Throughout most of the twentieth century, a significant percentage of communication systems was in analog form. However, by the end of the 1990s, the digital format began to dominate most applications. One does not need to look hard to witness the continuous migration from analog to digital communications: from audiocassette tape to MP3 and CD, from NTSC analog TV to digital HDTV, from traditional telephone to VoIP, and from VHS videotape to DVD. In fact, even the last analog refuge of broadcast radio is facing a strong digital competitor in the form of satellite radio. Given the dominating importance of digital communication systems in our lives today, it is never too early to study the basic principles and various aspects of digital data transmission, as we will do in this chapter.

This chapter deals with the problems of transmitting digital data over a channel. Hence, the starting messages are assumed to be digital. We shall begin by considering the binary case, where the data consist of only two symbols: **1** and **0**. We assign a distinct waveform (pulse) to each of these two symbols. The resulting sequence of these pulses is transmitted over a channel. At the receiver, these pulses are detected and are converted back to binary data (**1**s and **0**s).

7.1 DIGITAL COMMUNICATION SYSTEMS

A digital communication system consists of several components, as shown in Fig. 7.1. In this section, we conceptually outline their functionalities in the communication systems. The details of their analysis and design will be given in dedicated sections later in this chapter.

7.1.1 Source

The input to a digital system takes the form of a sequence of digits. The input could be the output from a data set, a computer, or a digitized audio signal (PCM, DM, or LPC), digital facsimile or HDTV, or telemetry data, and so on. Although most of the discussion in this chapter is confined to the binary case (communication schemes using only two symbols), the more general case of M-ary communication, which uses M symbols, will also be discussed in Secs. 7.7 and 7.9.

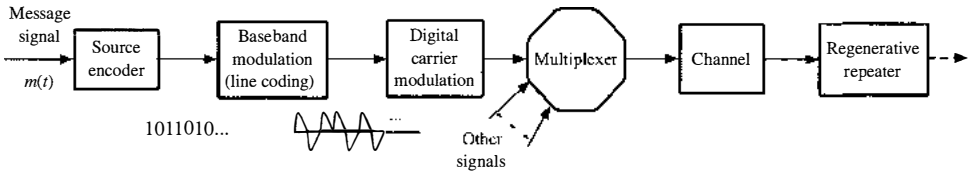


Figure 7.1 Fundamental building blocks of digital communication systems.

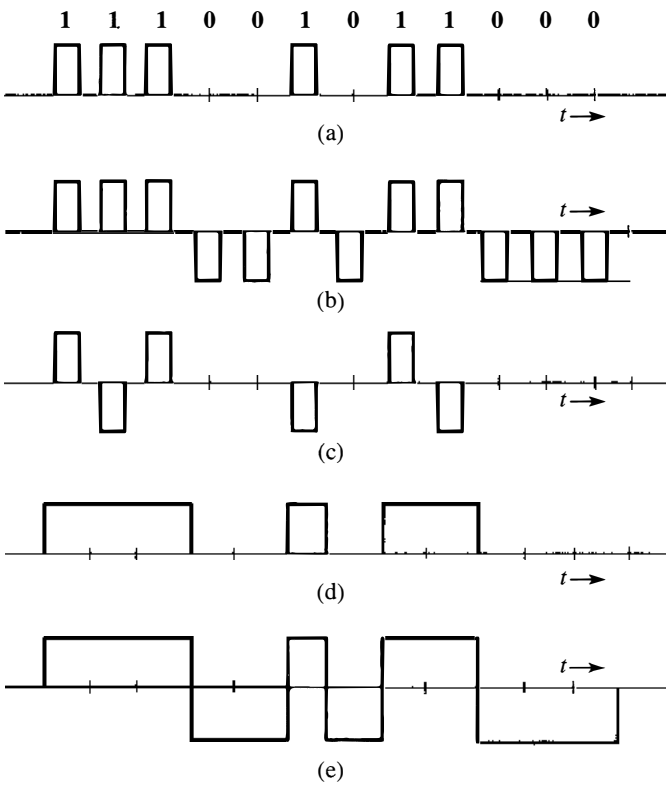


Figure 7.2 Line code examples: (a) on-off (RZ); (b) polar (RZ); (c) bipolar (RZ); (d) on-off (NRZ); (e) polar (NRZ).

7.1.2 Line Coder

The digital output of a source encoder is converted (or coded) into electrical pulses (waveforms) for the purpose of transmission over the channel. This process is called **line coding** or **transmission coding**. There are many possible ways of assigning waveforms (pulses) to the digital data. In the binary case (2 symbols), for example, conceptually the simplest line code is **on-off**, where a **1** is transmitted by a pulse $p(t)$ and a **0** is transmitted by no pulse (zero signal) as shown in Fig. 7.2a. Another commonly used code is **polar**, where **1** is transmitted by a pulse $p(t)$ and **0** is transmitted by a pulse $-p(t)$ (Fig. 7.2b). The polar scheme is the most power-efficient code because it requires the least power for a given noise immunity (error probability). Another popular code in PCM is **bipolar**, also known as **pseudoternary** or **alternate mark inversion (AMI)**, where **0** is encoded by no pulse and **1** is encoded by a pulse $p(t)$ or $-p(t)$ depending on whether the previous **1** is encoded by $-p(t)$ or $p(t)$. In short, pulses representing consecutive **1**s alternate in sign, as shown in Fig. 7.2c. This code has the advantage that if *one single* error is made in the detecting of pulses, the received pulse

sequence will violate the bipolar rule and the error can be detected (although not corrected) immediately.*

Another line code that appeared promising earlier is the duobinary (and modified duobinary) proposed by Lender.[1, 2] This code is better than the bipolar in terms of bandwidth efficiency. Its more prominent variant, the *modified duobinary* line code, has seen applications in hard disk drive read channels, in optical 10 Gbit/s transmission for metronetworks, and in the first-generation modems for integrated services digital networks (ISDN). Details of duobinary line codes will be discussed later in this chapter.

In our discussion so far, we have used half-width pulses just for the sake of illustration. We can select other widths also. Full-width pulses are often used in some applications. Whenever full-width pulses are used, the pulse amplitude is held to a constant value throughout the pulse interval (i.e., it does not have a chance to go to zero before the next pulse begins). For this reason, these schemes are called **non-return-to-zero** or **NRZ** schemes, in contrast to **return-to-zero** or **RZ** schemes (Fig. 7.2a–c). Figure 7.2d shows an on-off NRZ signal, whereas Fig. 7.2e shows a polar NRZ signal.

7.1.3 Multiplexer

Generally speaking, the capacity of a physical channel (e.g., coaxial cable, optic fiber) for transmitting data is much larger than the data rate of individual sources. To utilize this capacity effectively, we combine several sources by means of a digital multiplexer. The digital multiplexing can be achieved through frequency division or time division, as we have already discussed. Alternatively, code division is also a practical and effective approach (to be discussed in Chapter 11). Thus a physical channel is normally shared by several messages simultaneously.

7.1.4 Regenerative Repeater

Regenerative repeaters are used at regularly spaced intervals along a digital transmission line to detect the incoming digital signal and regenerate new "clean" pulses for further transmission along the line. This process periodically eliminates, and thereby combats, accumulation of noise and signal distortion along the transmission path. The ability of such regenerative repeaters to effectively eliminate noise and signal distortion effects is one of the biggest advantages of digital communication systems over their analog counterparts.

If the pulses are transmitted at a rate of R_b pulses per second, we require the periodic timing information—the clock signal at R_b Hz—to sample the incoming pulses at a repeater. This timing information can be extracted from the received signal itself if the line code is chosen properly. When the *RZ* polar signal in Fig. 7.2b is rectified, for example, it results in a periodic signal of clock frequency R_b Hz, which contains the desired periodic timing signal of frequency R_b Hz. When this signal is applied to a resonant circuit tuned to frequency R_b, the output, which is a sinusoid of frequency R_b Hz, can be used for timing. The on-off signal can be expressed as a sum of a periodic signal (of clock frequency) and a polar, or random, signal as shown in Fig. 7.3. Because of the presence of the periodic component, we can extract the timing information from this signal by using a resonant circuit tuned to the clock frequency. A bipolar signal, when rectified, becomes an on-off signal. Hence, its timing information can be extracted using the same way as that for an on-off signal.

* This assumes no more than one error in sequence. Multiple errors in sequence could cancel their respective effects and remain undetected. However, the probability of multiple errors is much smaller than that of single errors. Even

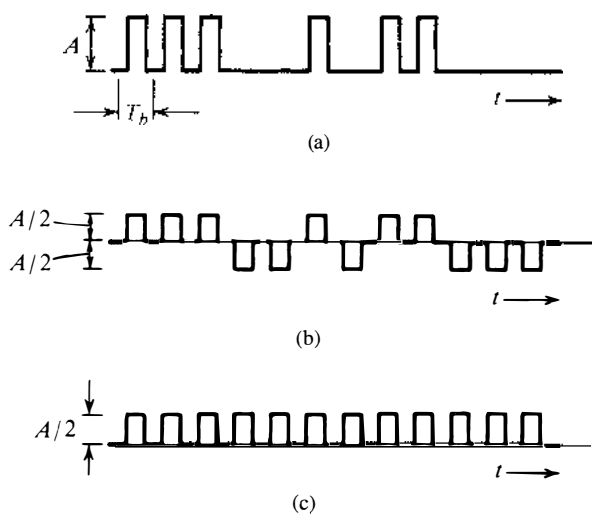


Figure 7.3 An on-off signal (a) is a sum of a random polar signal (b) and a clock frequency periodic signal (c).

The timing signal (the resonant circuit output) is sensitive to the incoming bit pattern. In the on-off or bipolar case, a **0** is transmitted by 'no pulse.' Hence, if there are too many **0**s in a sequence (no pulses), there is no signal at the input of the resonant circuit and the sinusoidal output of the resonant circuit starts decaying, thus causing error in the timing information. We shall discuss later ways of overcoming this problem. A line code in which the bit pattern does not affect the accuracy of the timing information is said to be a **transparent** line code. The *RZ* polar scheme (where each bit is transmitted by some pulse) is transparent, whereas the on-off and bipolar are nontransparent.

7.2 LINE CODING

Digital data can be transmitted by various **transmission** or **line codes**. We have given examples of on-off, polar, and bipolar. Each line code has its advantages and disadvantages. Among other desirable properties, a line code should have the following properties.

- *Transmission bandwidth* should be as small as possible.
- *Power efficiency*. For a given bandwidth and a specified detection error rate, the transmitted power should be as low as possible.
- *Error detection and correction capability*. It is desirable to detect, and preferably correct, detection errors. In a bipolar case, for example, a single error will cause bipolar violation and can easily be detected. Error correcting codes will be discussed in depth in Chapter 14.
- *Favorable power spectral density*. It is desirable to have zero power spectral density (PSD) at $f = 0$ (dc) because ac coupling and transformers are often used at the repeaters.* Significant power in low-frequency components should also be avoided because it causes dc wander in the pulse stream when ac coupling is used.

---

for single errors, we cannot tell exactly where the error is located. Therefore, this code can detect the presence of single errors, but it cannot correct them.

* The ac coupling is required because the dc paths provided by the cable pairs between the repeater sites are used to transmit the power needed to operate the repeaters.

- *Adequate timing content.* It should be possible to extract timing or clock information from the signal.
- *Transparency.* It should be possible to correctly transmit a digital signal regardless of the pattern of **1**s and **0**s. We saw earlier that a long string of **0**s could cause problems in timing extraction for the on-off and bipolar cases. A code is transparent if the data are so coded that for every possible sequence of data, the coded signal is received faithfully.

7.2.1 PSD of Various Line Codes

In Example 3.19 we discussed a procedure for finding the PSD of a polar pulse train. We shall use a similar procedure to find a general expression for PSD of the baseband modulation (line coding) output signals as shown in Fig. 7.1. In particular, we directly apply the relationship between the PSD and the autocorrelation function of the baseband modulation signal given in Section 3.8 [Eq. (3.85)].

In the following discussion, we consider a generic pulse $p(t)$ whose corresponding Fourier transform is $P(f)$. We can denote the line code symbol at time k as a_k. When the transmission rate is $R_b = 1/T_b$ pulses per second, the line code generates a pulse train constructed from the basic pulse $p(t)$ with amplitude a_k starting at time $t = kT_b$; in other words, the kth symbol is transmitted as $a_k p(t - kT_b)$. Figure 7.4a provides an illustration of a special pulse $p(t)$, whereas Fig. 7.4b shows the corresponding pulse train generated by the line coder at baseband. As shown

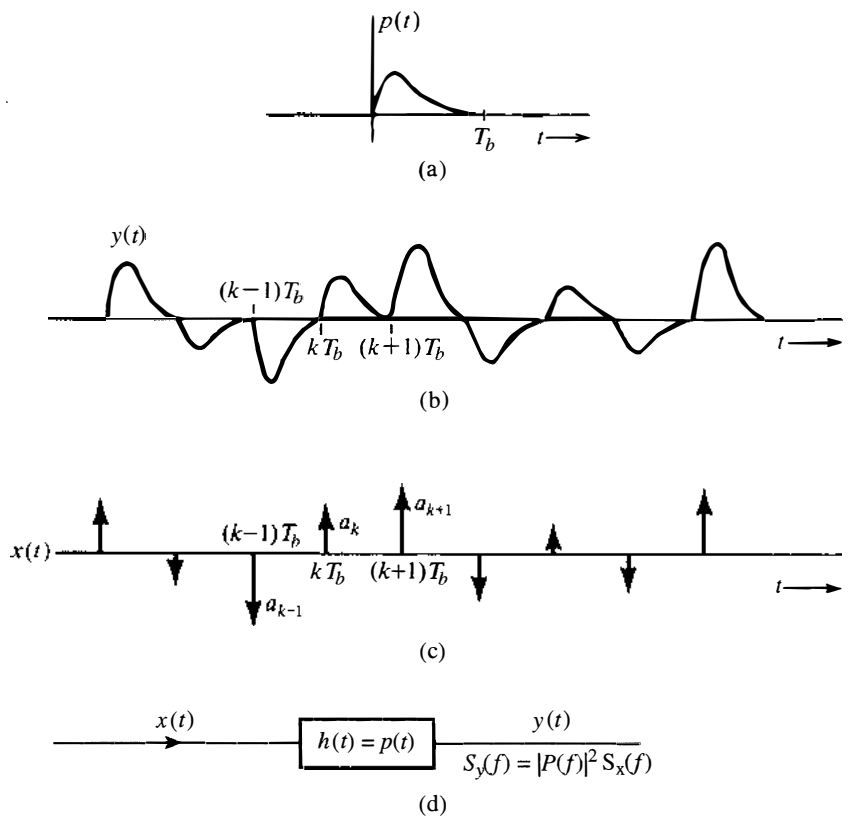


Figure 7.4 Random pulse-amplitude-modulated signal and its generation from a PAM impulse.

in Fig. 7.4b, counting a succession of symbol transmissions T_b second apart, the baseband signal is a pulse train of the form

$$y(t) = \sum a_k p(t - kT_b) \tag{7.1}$$

Note that the line coder determines the symbol $\{a_k\}$ as the amplitude of the pulse $p(t - kT_b)$.

The values a_k are random and depend on the line coder input and the line code itself; $y(t)$ is a pulse-amplitude-modulated (PAM) signal. The on-off, polar, and bipolar line codes are all special cases of this pulse train $y(t)$, where a_k takes on values 0, 1, or -1 randomly, subject to some constraints. We can, therefore, analyze many line codes according to the PSD of $y(t)$. Unfortunately, the PSD of $y(t)$ depends on both a_k and $p(t)$. If the pulse shape $p(t)$ changes, we may have to derive the PSD all over again. This difficulty can be overcome by the simple artifice of selecting a PAM signal $x(t)$ that uses a unit impulse for the basic pulse $p(t)$ (Fig. 7.4c). The impulses are at the intervals of T_b and the strength (area) of the kth impulse is a_k. If $x(t)$ is applied to the input of a filter that has a unit impulse response $h(t) = p(t)$ (Fig. 7.4d), the output will be the pulse train $y(t)$ in Fig. 7.4b. Also, applying Eq. (3.92), the PSD of $y(t)$ is

$$S_y(f) = |P(f)|^2 S_x(f)$$

This relationship allows us to determine $S_y(f)$, the PSD of a line code corresponding to any pulse shape $p(t)$, once we know $S_x(f)$. This approach is attractive because of its generality.

We now need to derive $\mathcal{R}_x(\tau)$, the time autocorrelation function of the impulse train $x(t)$. This can be conveniently done by considering the impulses as a limiting form of the rectangular pulses, as shown in Fig. 7.5a. Each pulse has a width $\epsilon \to 0$, and the kth pulse height

$$h_k = \frac{a_k}{\epsilon} \to \infty$$

This way, we guarantee that the strength of the kth impulse is a_k, or

$$\epsilon h_k = a_k$$

If we designate the corresponding rectangular pulse train by $\hat{x}(t)$, then by definition [Eq. (3.82) in Sec. 3.8]

$$\mathcal{R}_{\hat{x}}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} \hat{x}(t)\hat{x}(t - \tau)\, dt \tag{7.2}$$

Because $\mathcal{R}_{\hat{x}}(\tau)$ is an even function of τ [Eq. (3.83)], we need to consider only positive τ. To begin with, consider the case of $\tau < \epsilon$. In this case the integral in Eq. (7.2) is the area under the signal $\hat{x}(t)$ multiplied by $\hat{x}(t)$ delayed by $\tau\,(\tau < \epsilon)$. As seen from Fig. 7.5b, the area associated with the kth pulse is $h_k^2(\epsilon - \tau)$, and

$$\begin{aligned}
\mathcal{R}_{\hat{x}} &= \lim_{T\to\infty} \frac{1}{T} \sum_k h_k^2(\epsilon - \tau) \\
&= \lim_{T\to\infty} \frac{1}{T} \sum_k a_k^2 \left(\frac{\epsilon - \tau}{\epsilon^2}\right) \\
&= \frac{R_0}{\epsilon T_b}\left(1 - \frac{\tau}{\epsilon}\right)
\end{aligned} \tag{7.3a}$$

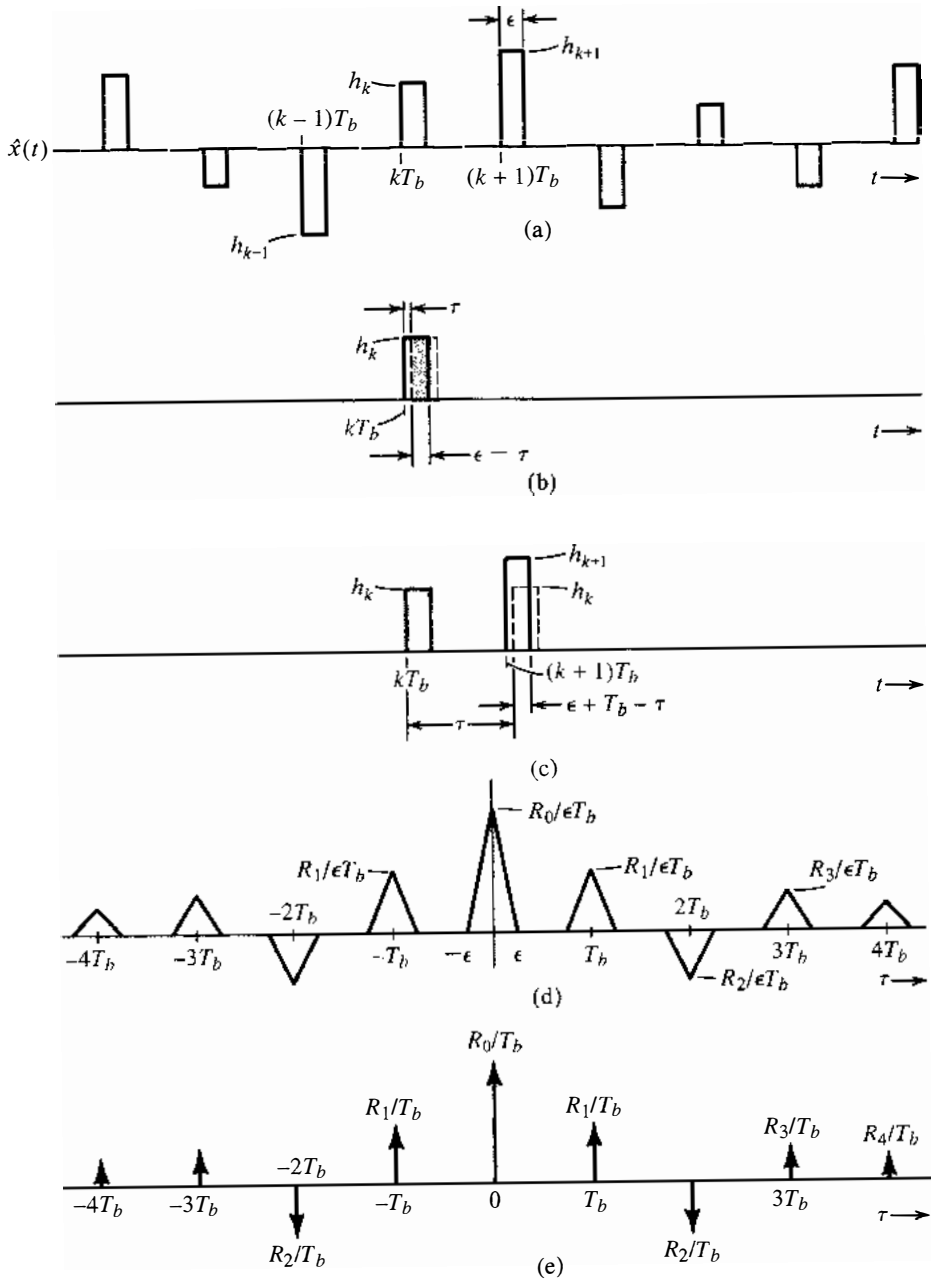


Figure 7.5 Derivation of PSD of a random PAM signal with a very narrow pulse of width ϵ and height $h_k = a_k/\epsilon$.

where

$$R_0 = \lim_{T\to\infty} \frac{T_b}{T} \sum_k a_k^2 \tag{7.3b}$$

During the averaging interval T ($T \to \infty$), there are N pulses ($N \to \infty$), where

$$N = \frac{T}{T_b} \tag{7.4}$$

and from Eq. (7.3b)

$$R_0 = \lim_{N\to\infty} \frac{1}{N} \sum_k a_k^2 \tag{7.5}$$

Observe that the summation is over N pulses. Hence, R_0 is the time average of the square of the pulse amplitudes a_k. Using our time average notation, we can express R_0 as

$$R_0 = \lim_{N\to\infty} \frac{1}{N} \sum_k a_k^2 = \overbrace{a_k^2} \tag{7.6}$$

We also know that $\mathcal{R}_{\hat{x}}(\tau)$ is an even function of τ [see Eq. (3.83)]. Hence, Eq. (7.3) can be expressed as

$$\mathcal{R}_{\hat{x}}(\tau) = \frac{R_0}{\epsilon T_b}\left(1 - \frac{|\tau|}{\epsilon}\right) \qquad |\tau| < \epsilon \tag{7.7}$$

This is a triangular pulse of height $R_0/\epsilon T_b$ and width 2ϵ centered at $\tau = 0$ (Fig. 7.5d). This is expected because as τ increases beyond ϵ, there is no overlap between the delayed signal $\hat{x}(t - \tau)$ and $\hat{x}(t)$; hence, $\mathcal{R}_{\hat{x}}(\tau) = 0$, as seen from Fig. 7.5d. But as we increase τ further, we find that the kth pulse of $\hat{x}(t - \tau)$ will start overlapping the $(k + 1)$th pulse of $\hat{x}(t)$ as τ approaches T_b (Fig. 7.5c). Repeating the earlier argument, we see that $\mathcal{R}_{\hat{x}}(\tau)$ will have another triangular pulse of width 2ϵ centered at $\tau = T_b$ and of height $R_1/\epsilon T_b$ where

$$\begin{aligned} R_1 &= \lim_{T\to\infty} \frac{T_b}{T} \sum_k a_k a_{k+1} \\ &= \lim_{N\to\infty} \frac{1}{N} \sum_k a_k a_{k+1} \\ &= \overbrace{a_k a_{k+1}} \end{aligned}$$

Observe that R_1 is obtained by multiplying every pulse strength (a_k) by the strength of its immediate neighbor (a_{k+1}), adding all these products, and then dividing by the total number of pulses. This is clearly the time average (mean) of the product $a_k a_{k+1}$ and is, in our notation, $\overbrace{a_k a_{k+1}}$. A similar thing happens around $\tau = 2T_b, 3T_b, \ldots$. Hence, $\mathcal{R}_{\hat{x}}(\tau)$ consists of a sequence of triangular pulses of width 2ϵ centered at $\tau = 0, \pm T_b, \pm 2T_b, \ldots$. The height of the pulses centered at $\pm nT_b$ is $R_n/\epsilon T_b$, where

$$\begin{aligned} R_n &= \lim_{T\to\infty} \frac{T_b}{T} \sum_k a_k a_{k+n} \\ &= \lim_{N\to\infty} \frac{1}{N} \sum_k a_k a_{k+n} \\ &= \overbrace{a_k a_{k+n}} \end{aligned}$$

R_n is essentially the discrete autocorrelation function of the line code symbols $\{a_k\}$.

To find $\mathcal{R}_x(\tau)$, we let $\epsilon \to 0$ in $\mathcal{R}_{\hat{x}}(\tau)$. As $\epsilon \to 0$, the width of each triangular pulse$\to 0$ and the height$\to \infty$ in such a way that the area is still finite. Thus, in the limit as $\epsilon \to 0$, the triangular pulses become impulses. For the nth pulse centered at nT_b, the height is $R_n/\epsilon T_b$ and the area is R_n/T_b. Hence, (Fig. 7.5e)

$$\mathcal{R}_x(\tau) = \frac{1}{T_b} \sum_{n=-\infty}^{\infty} R_n \, \delta(\tau - nT_b) \tag{7.8}$$

The PSD $S_x(f)$ is the Fourier transform of $\mathcal{R}_x(\tau)$. Therefore,

$$S_x(f) = \frac{1}{T_b} \sum_{n=-\infty}^{\infty} R_n e^{-jn2\pi fT_b} \tag{7.9}$$

Recognizing that $R_{-n} = R_n$ [because $\mathcal{R}(\tau)$ is an even function of τ], we have

$$S_x(f) = \frac{1}{T_b}\left[R_o + 2\sum_{n=1}^{\infty} R_n \cos n2\pi fT_b\right] \tag{7.10}$$

The input $x(t)$ to the filter with impulse response $h(t) = p(t)$ results in the output $y(t)$, as shown in Fig. 7.4d. If $p(t) \Longleftrightarrow P(f)$, the transfer function of the filter is $H(f) = P(f)$, and according to Eq. (3.91),

$$S_y(f) = |P(f)|^2 S_x(f) \tag{7.11a}$$

$$= \frac{|P(f)|^2}{T_b}\left[\sum_{n=-\infty}^{\infty} R_n e^{-jn2\pi fT_b}\right] \tag{7.11b}$$

$$= \frac{|P(f)|^2}{T_b}\left[R_o + 2\sum_{n=1}^{\infty} R_n \cos n2\pi fT_b\right] \tag{7.11c}$$

Thus, the PSD of a line code is fully characterized by its R_n and the pulse-shaping selection $P(f)$. We shall now use this general result to find the PSDs of various specific line codes by first determining the symbol autocorrelation R_n.

7.2.2 Polar Signaling

In polar signaling, **1** is transmitted by a pulse $p(t)$ and **0** is represented by $-p(t)$. In this case, a_k is equally likely to be 1 or -1, and a_k^2 is always 1. Hence,

$$R_0 = \lim_{N\to\infty} \frac{1}{N} \sum_k a_k^2$$

There are N pulses and $a_k^2 = 1$ for each one, and the summation on the right-hand side above is N. Hence,

$$R_0 = \lim_{N\to\infty} \frac{1}{N}(N) = 1 \tag{7.12a}$$

Moreover, both a_k and a_{k+1} are either 1 or -1. Hence, $a_k a_{k+1}$ is either 1 or -1. Because the pulse amplitude a_k is equally likely to be 1 and -1 on the average, out of N terms the product $a_k a_{k+1}$ is equal to 1 for $N/2$ terms and is equal to -1 for the remaining $N/2$ terms. Therefore,

Possible Values of $a_k a_{k+1}$

a_{k+1} \ a_k	-1	$+1$
-1	1	-1
$+1$	-1	1

$$R_1 = \lim_{N\to\infty} \frac{1}{N}\left[\frac{N}{2}(1) + \frac{N}{2}(-1)\right] = 0 \qquad (7.12b)$$

Arguing this way, we see that the product $a_k a_{k+n}$ is also equally likely to be 1 or -1. Hence,

$$R_n = 0 \qquad n \geq 1 \qquad (7.12c)$$

Therefore from Eq. (7.11c)

$$S_y(f) = \frac{|P(f)|^2}{T_b} R_0$$
$$= \frac{|P(f)|^2}{T_b} \qquad (7.13)$$

For the sake of comparison of various schemes, we shall consider a specific pulse shape. Let $p(t)$ be a rectangular pulse of width $T_b/2$ (half-width rectangular pulse), that is,

$$p(t) = \Pi\left(\frac{t}{T_b/2}\right) = \Pi\left(\frac{2t}{T_b}\right)$$

and

$$P(f) = \frac{T_b}{2}\,\text{sinc}\left(\frac{\pi f T_b}{2}\right) \qquad (7.14)$$

Therefore

$$S_y(f) = \frac{T_b}{4}\,\text{sinc}^2\left(\frac{\pi f T_b}{2}\right) \qquad (7.15)$$

Figure 7.6 shows the spectrum $S_y(f)$. It is clear that the polar signal has most of its power concentrated in lower frequencies. Theoretically, the spectrum becomes very small as frequency increases but never becomes totally zero above a certain frequency. To define a meaningful measure of bandwidth, we consider its *first non-dc null frequency* to be its **essential bandwidth**.*

From polar signal spectrum, the essential bandwidth of the signal is seen to be $2R_b$ Hz (where R_b is the clock frequency). This is 4 times the theoretical bandwidth (Nyquist bandwidth) required to transmit R_b pulses per second. Increasing the pulse width reduces the bandwidth (expansion in the time domain results in compression in the frequency domain).

* Strictly speaking, the location of the first null frequency above dc is not always a good measure of signal bandwidth. Whether the first non-dc null is a meaningful bandwidth depends on the amount of signal power contained in the main (or first) lobe of the PSD, as we will see later in the PSD comparison of several line codes (Fig. 7.9). In most practical cases, this approximation is acceptable for commonly used line codes and pulse shapes.

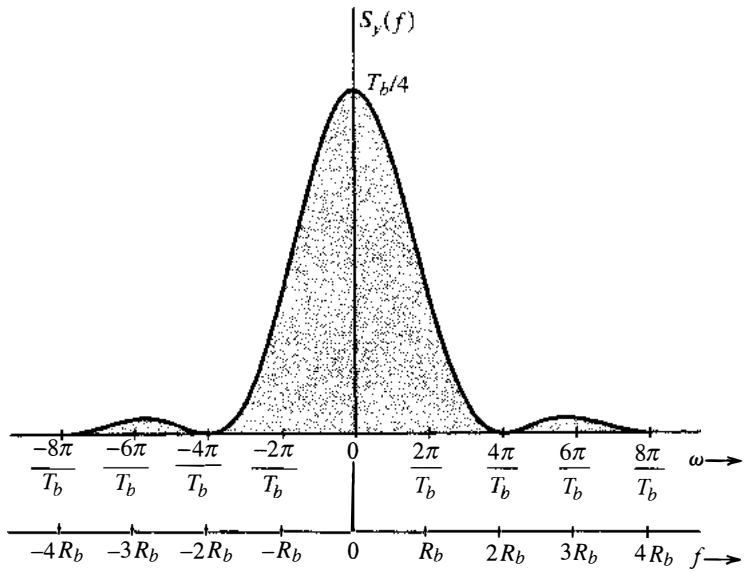


Figure 7.6 Power spectral density of a polar signal.

For a full-width pulse* (maximum possible pulse width), the essential bandwidth is half, that is R_b Hz. This is still twice the theoretical bandwidth. Thus, polar signaling is not the most bandwidth efficient.

Second, polar signaling has no capability for error detection or error correction. A third disadvantage of polar signaling is that it has nonzero PSD at dc ($f = 0$). This will rule out the use of ac coupling during transmission. The ac mode of coupling, which permits transformers and blocking capacitors to aid in impedance matching and bias removal, and allows dc powering of the line repeaters over the cable pairs, is very important in practice. Later, we shall show how a PSD of a line code may be forced to zero at dc by properly shaping $p(t)$.

On the positive side, polar signaling is the most efficient scheme from the power requirement viewpoint. For a given power, it can be shown that the error detection probability for a polar scheme is the lowest among all signaling techniques (see Chapter 10). Polar signaling is also transparent because there is always some pulse (positive or negative) regardless of the bit sequence. There is no discrete clock frequency component in the spectrum of the polar signal. Rectification of the *RZ* polar signal, however, yields a periodic signal of clock frequency and can readily be used to extract timing.

7.2.3 Constructing a DC Null in PSD by Pulse Shaping

Because $S_y(f)$, the PSD of a line code contains a factor $|P(f)|^2$, we can force the PSD to have a dc null by selecting a pulse $p(t)$ such that $P(f)$ is zero at dc ($f = 0$). Because

$$P(f) = \int_{-\infty}^{\infty} p(t)e^{-j2\pi ft}\,dt$$

* Scheme using the full-width pulse $p(t) = \Pi(t/T_b)$ is an example of a non-return-to-zero (NRZ) scheme. The half-width pulse scheme, on the other hand, is an example of a return-to-zero (RZ) scheme.

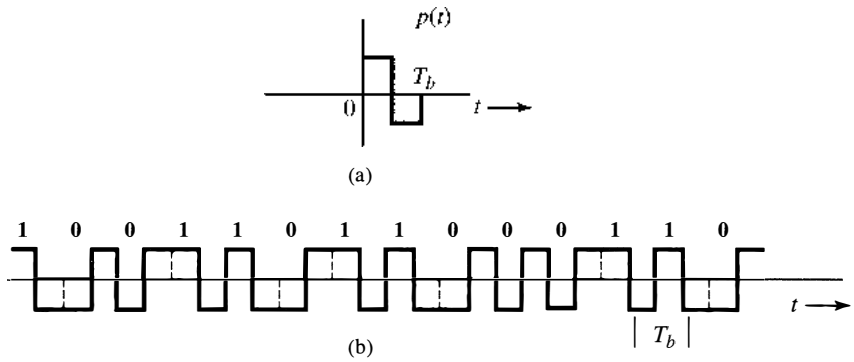


Figure 7.7 Split-phase (Manchester or twinned-binary) signal. (a) Basic pulse $p(t)$ for Manchester signaling. (b) Transmitted waveform for binary data sequence using Manchester signaling.

we have

$$P(0) = \int_{-\infty}^{\infty} p(t)\,dt$$

Hence, if the area under $p(t)$ is made zero, $P(0)$ is zero, and we have a dc null in the PSD. For a rectangular pulse, one possible shape of $p(t)$ to accomplish this is shown in Fig. 7.7a. When we use this pulse with polar line coding, the resulting signal is known as **Manchester code**, or **split-phase** (also called **twinned-binary**), signal. The reader can use Eq. (7.13), to show that for this pulse, the PSD of the Manchester line code has a dc null (see Prob. 7.2-2).

7.2.4 On-Off Signaling

In on-off signaling, a **1** is transmitted by a pulse $p(t)$ and a **0** is transmitted by no pulse. Hence, a pulse strength a_k is equally likely to be 1 or 0. Out of N pulses in the interval of T seconds, a_k is 1 for $N/2$ pulses and is 0 for the remaining $N/2$ pulses on the average. Hence,

$$R_0 = \lim_{N\to\infty} \frac{1}{N}\left[\frac{N}{2}(1)^2 + \frac{N}{2}(0)^2\right] = \frac{1}{2} \tag{7.16}$$

To compute R_n we need to consider the product $a_k a_{k+n}$. Since a_k and a_{k+n} are equally likely to be 1 or 0, the product $a_k a_{k+n}$ is equally likely to be 1×1, 1×0, 0×1 or 0×0, that is, 1, 0, 0, 0. Therefore on the average, the product $a_k a_{k+n}$ is equal to 1 for $N/4$ terms and 0 for $3N/4$ terms and

$$R_n = \lim_{N\to\infty} \frac{1}{N}\left[\frac{N}{4}(1) + \frac{3N}{4}(0)\right] = \frac{1}{4} \qquad n \geq 1 \tag{7.17}$$

Therefore, [Eq. (7.9)]

$$S_x(f) = \frac{1}{2T_b} + \frac{1}{4T_b}\sum_{\substack{n=-\infty \\ n\neq 0}}^{\infty} e^{-jn2\pi fT_b} \tag{7.18a}$$

$$= \frac{1}{4T_b} + \frac{1}{4T_b}\sum_{n=-\infty}^{\infty} e^{-jn2\pi fT_b} \tag{7.18b}$$

Equation (7.18b) is obtained from Eq. (7.18a) by splitting the term $1/2T_b$ corresponding to R_0 into two: $1/4T_b$ outside the summation and $1/4T_b$ inside the summation (corresponding to $n = 0$). We now use the formula (see the footnote for a proof*)

$$\sum_{n=-\infty}^{\infty} e^{-jn2\pi fT_b} = \frac{1}{T_b} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_b}\right)$$

Substitution of this result in Eq. (7.18b) yields

$$S_x(f) = \frac{1}{4T_b} + \frac{1}{4T_b^2} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_b}\right) \tag{7.19a}$$

and the desired PSD of the on-off waveform $y(t)$ is [from Eq. (7.11a)]

$$S_y(f) = \frac{|P(f)|^2}{4T_b}\left[1 + \frac{1}{T_b} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_b}\right)\right] \tag{7.19b}$$

Note that unlike the continuous PSD spectrum of polar signaling, the on-off PSD of Eq. (7.19b) also has an additional discrete part. This discrete part may be nullified if the pulse shape is chosen such that

$$P\left(\frac{n}{T_b}\right) = 0 \qquad n = 0, \pm 1, \ldots$$

For the example case of a half-width rectangular pulse [see Eq. (7.14)],

$$S_y(f) = \frac{T_b}{16} \operatorname{sinc}^2\left(\frac{\pi fT_b}{2}\right)\left[1 + \frac{1}{T_b} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_b}\right)\right] \tag{7.20}$$

The resulting PSD is shown in Fig. 7.8. The continuous component of the spectrum is $(T_b/16)\ \operatorname{sinc}^2(\pi fT_b/2)$. This is identical (except for a scaling factor) to the spectrum of the polar signal [Eq. (7.15)]. The discrete component is represented by the product of an impulse train with the continuous component $(T_b/16)\ \operatorname{sinc}^2(\pi fT_b/2)$. Hence this component appears as periodic impulses with the continuous component as the envelope. Moreover, the impulses repeat at the clock frequency $R_b = 1/T_b$ because its fundamental frequency is $2\pi/T_b$ rad/s, or $1/T_b$ Hz. This is a logical result because as Fig. 7.3 shows, an on-off signal can be expressed as a sum of a polar and a periodic component. The polar component $y_1(t)$ is exactly half

---

* The impulse train in Fig. 3.23a of Example 3.11 is $\delta_{T_b}(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_b)$. Moreover, the Fourier series for this impulse train as found in Eq. (2.67) is

$$\sum_{n=-\infty}^{\infty} \delta(t - nT_b) = \frac{1}{T_b} \sum_{n=-\infty}^{\infty} e^{jn2\pi R_b t} \qquad R_b = \frac{1}{T_b}$$

We take the Fourier transform of both sides of this equation, and use the fact that $\delta(t - nT_b) \Longleftrightarrow e^{-jn2\pi fT_b}$ and $e^{jn2\pi R_b t} \Longleftrightarrow \delta(f - nR_b)$. This yields

$$\sum_{n=-\infty}^{\infty} e^{-jn2\pi fT_b} = \frac{1}{T_b} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_b}\right)$$

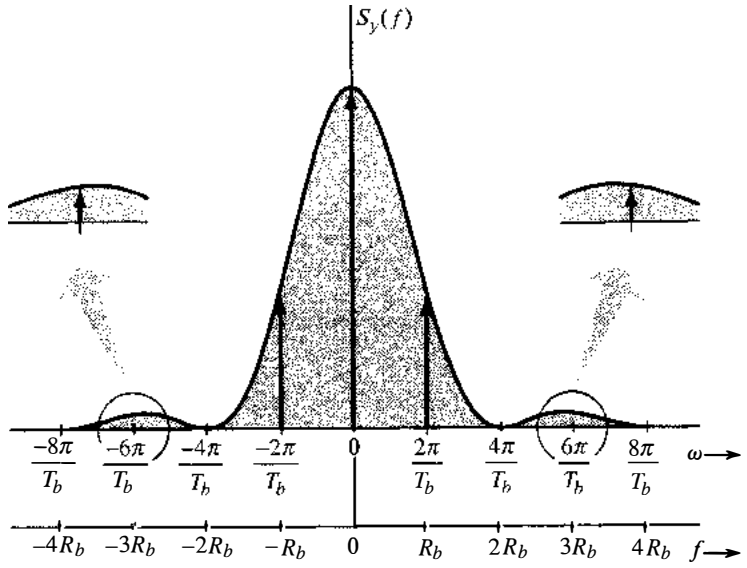


Figure 7.8 Power spectral density (PSD) of an on-off signal.

the polar signal discussed earlier. Hence, the PSD of this component is one-fourth the PSD in Eq. (7.15). The periodic component is of clock frequency R_b; it consists of discrete components of frequency R_b and its harmonics.

On-off signaling has very little to brag about. For a given transmitted power, it is less immune to noise interference than the polar scheme, which uses a positive pulse for **1** and a negative pulse for **0**. This is because the noise immunity depends on the difference of amplitudes representing **1** and **0**. Hence, for the same immunity, if on-off signaling uses pulses of amplitudes 2 and 0, polar signaling need use only pulses of amplitudes 1 and -1. It is simple to show that on-off signaling requires twice as much power as polar signaling. If a pulse of amplitude 1 or -1 has energy E, then the pulse of amplitude 2 has energy $(2)^2E = 4E$. Because $1/T_b$ digits are transmitted per second, polar signal power is $(E)(1/T_b) = E/T_b$. For the on-off case, on the other hand, each pulse energy is $4E$, though on average such a pulse is transmitted over half of the time while nothing is transmitted over the other half. Hence, the average signal power of on-off is

$$\frac{1}{T_b}\left(4E\frac{1}{2} + 0 \cdot \frac{1}{2}\right) = \frac{2E}{T_b}$$

which is twice that required for the polar signal. Moreover, unlike the polar case, on-off signaling is not transparent. A long string of **0**s (or offs) causes the absence of a signal and can lead to errors in timing extraction. In addition, all the disadvantages of polar signaling, (e.g., excessive transmission bandwidth, nonzero power spectrum at dc, no error detection (or correction) capability are also present in on-off signaling.

7.2.5 Bipolar Signaling

The signaling scheme used in PCM for telephone networks is called bipolar (pseudoternary or alternate mark inverted). A **0** is transmitted by no pulse, and a **1** is transmitted by a pulse

$p(t)$ or $-p(t)$, depending on whether the previous **1** was transmitted by $-p(t)$ or $p(t)$. With consecutive pulses alternating, we can avoid dc wander and thus cause a dc null in the PSD. Bipolar signaling actually uses three symbols $[p(t), 0,$ and $-p(t)]$, and, hence, it is in reality ternary rather than binary signaling.

To calculate the PSD, we have

$$R_o = \lim_{N\to\infty} \frac{1}{N} \sum_k a_k^2$$

On the average, half of the a_ks are 0, and the remaining half are either 1 or -1, with $a_k^2 = 1$. Therefore,

$$R_o = \lim_{N\to\infty} \frac{1}{N} \left[\frac{N}{2}(\pm 1)^2 + \frac{N}{2}(0)^2 \right] = \frac{1}{2}$$

To compute R_1, we consider the pulse strength product $a_k a_{k+1}$. There are four equally likely sequences of two bits: **11, 10, 01, 00**. Since bit **0** is encoded by no pulse ($a_k = 0$), the product $a_k a_{k+1}$ is zero for the last three of these sequences. This means, on the average, that $3N/4$ combinations have $a_k a_{k+1} = 0$ and only $N/4$ combinations have nonzero $a_k a_{k+1}$. Because of the bipolar rule, the bit sequence **11** can be encoded only by two consecutive pulses of opposite polarities. This means the product $a_k a_{k+1} = -1$ for the $N/4$ combinations. Therefore

$$R_1 = \lim_{N\to\infty} \frac{1}{N} \left[\frac{N}{4}(-1) + \frac{3N}{4}(0) \right] = -\frac{1}{4}$$

To compute R_2 in a similar way, we need to observe the product $a_k a_{k+2}$. For this, we need to consider all possible combinations of three bits in sequence. There are eight equally likely combinations: **111, 101, 110, 100, 011, 010, 001, 000**. The last six combinations have either the first and/or the last bit **0**. Hence $a_k a_{k+2} = 0$ for all these six combinations. The first two combinations are the only ones that yield nonzero $a_k a_{k+2}$. From the bipolar rule, the first and the third pulses in the combination **111** are of the same polarity, yielding $a_k a_{k+2} = 1$. But for **101**, the first and the third pulse are of opposite polarity, yielding $a_k a_{k+2} = -1$. Thus, on the average, $a_k a_{k+2} = 1$ for $N/8$ terms, -1 for $N/8$ terms and 0 for $3N/4$ terms. Hence,

$$R_2 = \lim_{N\to\infty} \frac{1}{N} \left[\frac{N}{8}(1) + \frac{N}{8}(-1) + \frac{3N}{8}(0) \right] = 0$$

In general

$$R_n = \lim_{N\to\infty} \frac{1}{N} \sum_k a_k a_{k+n}$$

For $n > 2$, the product $a_k a_{k+n}$ can be 1, -1, or 0. Moreover, an equal number of combinations have values 1 and -1. This causes $R_n = 0$. Thus

$$R_n = 0 \qquad n > 1$$

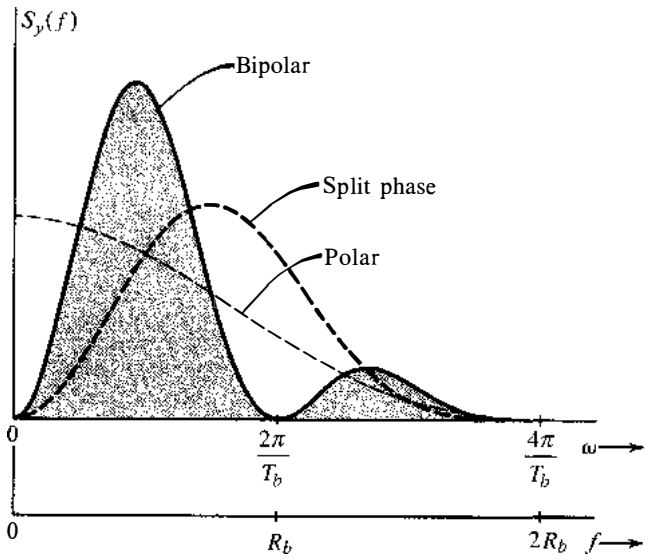


Figure 7.9 PSD of bipolar, polar, and split-phase signals normalized for equal powers. Half-width rectangular pulses are used.

and [see Eq. (7.11c)]

$$S_y(f) = \frac{|P(f)|^2}{2T_b}\,[1 - \cos 2\pi f T_b] \tag{7.21a}$$

$$= \frac{|P(f)|^2}{T_b}\sin^2(\pi f T_b) \tag{7.21b}$$

Note that $S_y(f) = 0$ for $f = 0$ (dc), regardless of $P(f)$. Hence, the PSD has a dc null, which is desirable for ac coupling. Moreover, $\sin^2(\pi f T_b) = 0$ at $f = 1/T_b$, that is, at $f = 1/T_b = R_b$ Hz. Thus, regardless of $P(f)$, we are assured of the first non-dc null bandwidth R_b Hz. For the half-width pulse

$$S_y(f) = \frac{T_b}{4}\,\text{sinc}^2\left(\frac{\pi f T_b}{2}\right)\sin^2(\pi f T_b) \tag{7.22}$$

This is shown in Fig. 7.9. The essential bandwidth of the signal is R_b ($R_b = 1/T_b$), which is half that of polar using the same half-width pulse or on-off signaling and twice the theoretical minimum bandwidth. Observe that we were able to obtain the bandwidth R_b for polar (or on-off) case for full-width pulse. For the bipolar case, the bandwidth is R_b Hz whether the pulse is half-width or full-width.

Bipolar signaling has several advantages: (1) its spectrum has a dc null; (2) its bandwidth is not excessive; (3) it has single-error-detection capability. This is because even single detection error will cause a violation of the alternating pulse rule, and this will be immediately detected. If a bipolar signal is rectified, we get an on-off signal that has a discrete component at the clock frequency. Among the disadvantages of a bipolar signal is the requirement for twice as much power (3 dB) as a polar signal needs. This is because bipolar detection is essentially equivalent to on-off signaling from the detection point of view. One distinguishes between $+p(t)$ or $-p(t)$ from 0 rather than between $\pm p(t)$.

Another disadvantage of bipolar signaling is that it is not transparent. In practice, various substitution schemes are used to prevent long strings of logic zeros from allowing the extracted clock signals to drift away. We shall now discuss two such schemes.

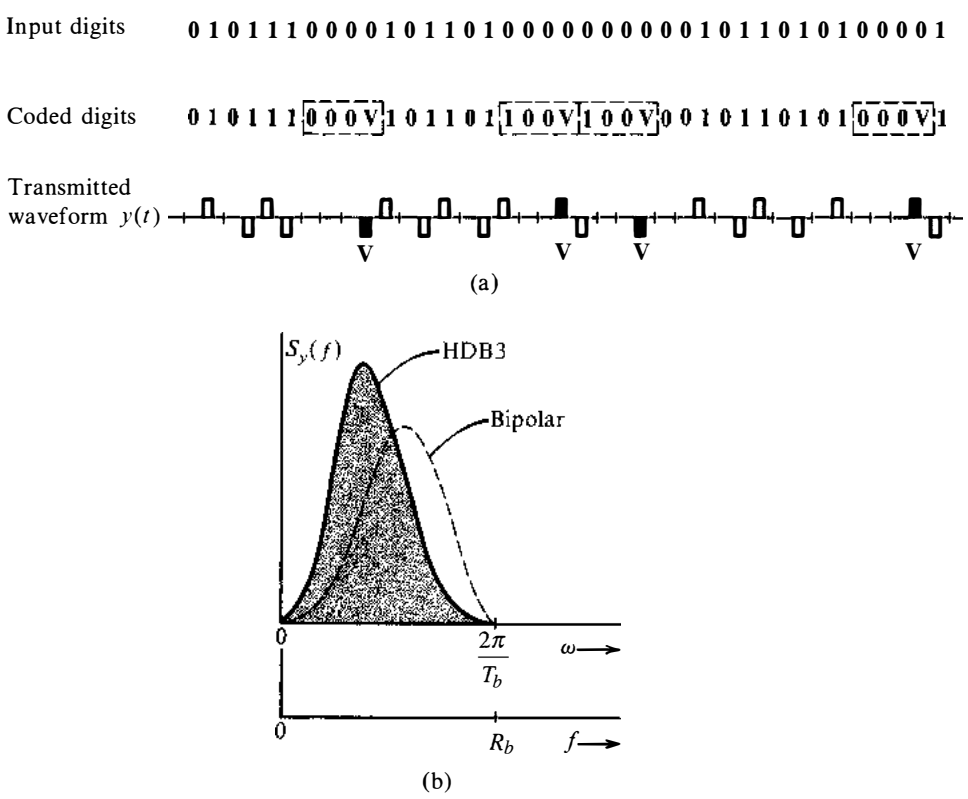


Figure 7.10 (a) HDB3 signal and (b) its PSD.

High-Density Bipolar (HDB) Signaling

The HDB scheme is an ITU (formerly CCITT) standard. In this scheme the problem of nontransparency in bipolar signaling is eliminated by adding pulses when the number of consecutive **0**s exceeds N. Such a modified coding is designated as **high-density bipolar coding** (HDBN), where N can take on any value 1, 2, 3, ... The most important of the HDB codes is HDB3 format, which has been adopted as an international standard.

The basic idea of the HDBN code is that when a run of $N + 1$ zeros occurs, this group of zeros is replaced by one of the special $N + 1$ binary digit sequences. To increase the timing content of the signal, the sequences are chosen to include some binary **1**s. The **1**s included deliberately violate the bipolar rule for easy identification of the substituted sequence. In HDB3 coding, for example, the special sequences used are **000V** and **B00V** where **B=1** that conforms to the bipolar rule and **V=1** that violates the bipolar rule. The choice of sequence **000V** or **B00V** is made in such a way that consecutive **V** pulses alternate signs to avoid dc wander and to maintain the dc null in the PSD. This requires that the sequence **B00V** be used when there are an even number of **1**s following the last special sequence and the sequence **000V** be used when there are an odd number of **1**s following the last sequence. Figure 7.10a shows an example of this coding. Note that in the sequence **B00V**, both **B** and **V** are encoded by the same pulse. The decoder has to check two things—the bipolar violations and the number of **0**s preceding each violation to determine if the previous **1** is also a substitution.

Despite deliberate bipolar violations, HDB signaling retains error detecting capability. Any single error will insert a spurious bipolar violation (or will delete one of the deliberate violations). This will become apparent when, at the next violation, the alternation of violations does not appear. This also shows that deliberate violations can be detected despite single errors. Figure 7.10b shows the PSD of HDB3 as well as that of a bipolar signal to facilitate comparison.[3]

Binary with N Zero Substitution (BNZS) Signaling

A class of line codes similar to HDBN is the **binary with N zero substitution, or BNZS** code, where if N zeros occur in succession, they are replaced, by one of the two special sequences containing some **1**s to increase timing content. There are deliberate bipolar violations just as in HDBN. Binary with eight-zero substitution (B8ZS) is used in DS1 signals of the digital telephone hierarchy in Chapter 6. It replaces any string of eight zeros in length with a sequence of ones and zeros containing two bipolar violations. Such a sequence is unlikely to be counterfeited by errors, and any such sequence received by a digital channel bank is replaced by a string of eight logic zeros prior to decoding. The sequence used as a replacement consists of the pattern **000VB0VB**. Similarly, in **B6ZS** code used in DS2 signals, a string of six zeros is replaced with **0VB0VB**, and DS3 signal features a three-zero B3ZS code. The B3ZS code is slightly more complex than the others in that either **B0V** or **00V** is used, the choice being made so that the number of **B** pulses between consecutive **V** pulses is odd. These BNZS codes with $N = 3, 6$, or 8 involve bipolar violations and must therefore be carefully replaced by their equivalent zero strings at the receiver.

There are many other transmission (line) codes, too numerous to list here. A list of codes and appropriate references can be found in Bylanski and Ingram.[3]

7.3 PULSE SHAPING

The PSD $S_y(f)$ of a digital signal $y(t)$ can be controlled by a choice of line code or by $P(f)$, the pulse shape. In the last section we discussed how the PSD is controlled by a line code. In this section we examine how $S_y(f)$ is influenced by the pulse shape $p(t)$, and we learn how to shape a pulse $p(t)$ to achieve a desired $S_y(f)$. The PSD $S_y(f)$ is strongly and directly influenced by the pulse shape $p(t)$ because $S_y(f)$ contains the term $|P(f)|^2$. Thus, in comparison to the nature of the line code, the pulse shape is a more direct and potent factor in terms of shaping the PSD $S_y(f)$.

7.3.1 Intersymbol Interferences (ISI) and Effect

In the last section, we used a simple half-width rectangular pulse $p(t)$ for the sake of illustration. Strictly speaking, in this case the bandwidth of $S_y(f)$ is infinite, since $P(f)$ has infinite bandwidth. But we found that the essential bandwidth of $S_y(f)$ was finite. For example, most of the power of a bipolar signal is contained within the essential band 0 to R_b Hz. Note, however, that the PSD is small but is still nonzero in the range $f > R_b$ Hz. Therefore, when such a signal is transmitted over a channel of bandwidth R_b Hz, a significant portion of its spectrum is transmitted, but a small portion of the spectrum is suppressed. In Sec. 3.5 and Sec. 3.6, we saw how such a spectral distortion tends to spread the pulse (dispersion). Spreading of a pulse beyond its allotted time interval T_b will cause it to interfere with neighboring pulses. This is known as **intersymbol interference** or **ISI**.

ISI is *not* noise. ISI is caused by nonideal channels that are not distortionless over the entire signal bandwidth. In the case of half-width rectangular pulse, the signal bandwidth is, strictly speaking, infinity. ISI, as a manifestation of channel distortion, can cause errors in pulse detection if it is large enough.

To resolve the difficulty of ISI, let us review briefly our problem. We need to transmit a pulse every T_b interval, the kth pulse being $a_k p(t-kT_b)$. The channel has a finite bandwidth, and

we are required to detect the pulse amplitude a_k correctly (i.e., without ISI). In our discussion so far, we have considered time-limited pulses. Since such pulses cannot be band-limited, part of their spectra is suppressed by a band-limited channel. This causes pulse distortion (spreading out) and, consequently, ISI. We can try to resolve this difficulty by using pulses that are band-limited to begin with so that they can be transmitted intact over a band-limited channel. But band-limited pulses cannot be time-limited. Obviously, various pulses will overlap and cause ISI. Thus, whether we begin with time-limited pulses or band-limited pulses, it appears that ISI cannot be avoided. It is inherent in the finite transmission bandwidth. Fortunately, there is an escape from this blind alley. Pulse amplitudes can be detected correctly despite pulse spreading (or overlapping), if there is no ISI at the decision-making instants. This can be accomplished by a properly shaped band-limited pulse. To eliminate ISI, Nyquist proposed three different criteria for pulse shaping,[4] where the pulses are allowed to overlap. Yet, they are shaped to cause zero (or controlled) interference with all the other pulses at the decision-making instants. Thus, by limiting the noninterference requirement only at the decision-making instants, we eliminate the need for the pulse to be totally nonoverlapping. We shall consider only the first two criteria. The third is much less useful than the first two criteria,[5] and hence, will not be considered here.

7.3.2 Nyquist's First Criterion for Zero ISI

In the first method, Nyquist achieves zero ISI by choosing a pulse shape that has a nonzero amplitude at its center (say $t = 0$) and zero amplitudes at $t = \pm nT_b$ ($n = 1, 2, 3, \ldots$), where T_b is the separation between successive transmitted pulses (Fig. 7.11a). Thus,

$$p(t) = \begin{cases} 1 & t = 0 \\ 0 & t = \pm nT_b \end{cases} \qquad \left(T_b = \frac{1}{R_b}\right) \tag{7.23}$$

A pulse satisfying this criterion causes zero ISI at all the remaining pulse centers, or signaling instants as shown in Fig. 7.11a, where we show several successive pulses (dashed) centered at $t = 0, T_b, 2T_b, 3T_b, \ldots$ ($T_b = 1/R_b$). For the sake of convenience, we have shown all pulses

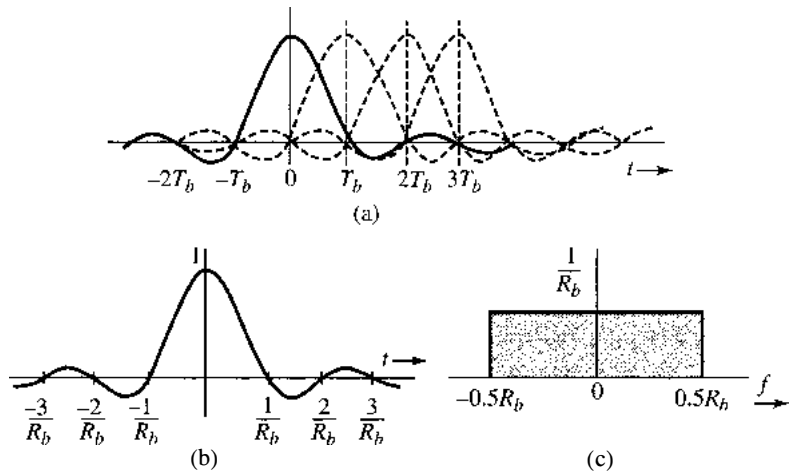


Figure 7.11 The minimum bandwidth pulse that satisfies Nyquist's first criterion and its spectrum.

to be positive.* It is clear from this figure that the samples at $t = 0, T_b, 2T_b, 3T_b, \ldots$ consist of the amplitude of only one pulse (centered at the sampling instant) with no interference from the remaining pulses.

Now transmission of R_b bit/s requires a theoretical minimum bandwidth $R_b/2$ Hz. It would be nice if a pulse satisfying Nyquist's criterion had this minimum bandwidth $R_b/2$ Hz. Can we find such a pulse $p(t)$? We have already solved this problem (Example 6.1 with $B = R_b/2$), where we showed that there exists one (and only one) pulse which meets Nyquist's criterion (7.23) and has a bandwidth $R_b/2$ Hz. This pulse, $p(t) = \text{sinc}\,(\pi R_b t)$, (Fig. 7.11b) has the property

$$\text{sinc}\,(\pi R_b t) = \begin{cases} 1 & t = 0 \\ 0 & t = \pm nT_b \end{cases} \qquad \left(T_b = \frac{1}{R_b}\right) \tag{7.24a}$$

Moreover, the Fourier transform of this pulse is

$$P(f) = \frac{1}{R_b}\Pi\left(\frac{f}{R_b}\right) \tag{7.24b}$$

which has a bandwidth $R_b/2$ Hz as seen from Fig. 7.11c. We can use this pulse to transmit at a rate of R_b pulses per second without ISI, over a bandwidth only $R_b/2$.

This scheme shows that we can attain the theoretical limit of performance by using a sinc pulse. Unfortunately, this pulse is impractical because it starts at $-\infty$. We will have to wait an infinite time to generate it. Any attempt to truncate it would increase its bandwidth beyond $R_b/2$ Hz. But even if this pulse were realizable, it would have an undesirable feature: namely, it decays too slowly at a rate $1/t$. This causes some serious practical problems. For instance, if the nominal data rate of R_b bit/s required for this scheme deviates a little, the pulse amplitudes will not vanish at the other pulse centers. Because the pulses decay only as $1/t$, the cumulative interference at any pulse center from all the remaining pulses is of the form $\sum(1/n)$. It is well known that the infinite series of this form does not converge and can add up to a very large value. A similar result occurs if everything is perfect at the transmitter but the sampling rate at the receiver deviates from the rate of R_b Hz. Again, the same thing happens if the sampling instants deviate a little because of pulse time jitter, which is inevitable even in the most sophisticated systems. This scheme therefore fails unless everything is perfect, which is a practical impossibility. And all this is because sinc $(\pi R_b t)$ decays too slowly (as $1/t$). The solution is to find a pulse $p(t)$ that satisfies Eq. (7.23) but decays faster than $1/t$. Nyquist has shown that such a pulse requires a bandwidth $kR_b/2$, with $1 \leq k \leq 2$.

This can be proved as follows. Let $p(t) \Longleftrightarrow P(f)$, where the bandwidth of $P(f)$ is in the range $(R_b/2,\ R_b)$ (Fig. 7.12a). The desired pulse $p(t)$ satisfies Eq. (7.23). If we sample $p(t)$ every T_b seconds by multiplying $p(t)$ by $\delta_{T_b}(t)$, (an impulse train), then because of the property (7.23), all the samples, except the one at the origin, are zero. Thus, the sampled signal $\overline{p}(t)$ is

$$\overline{p}(t) = p(t)\delta_{T_b}(t) = \delta(t) \tag{7.25}$$

Following the analysis of Eq. (6.4) in Chapter 6, we know that the spectrum of a sampled signal $\overline{p}(t)$ is $(1/T_b$ times) the spectrum of $p(t)$ repeating periodically at intervals of the sampling

* Actually, a pulse corresponding to **0** would be negative. But considering all positive pulses does not affect our reasoning. Showing negative pulses would make the figure needlessly confusing.

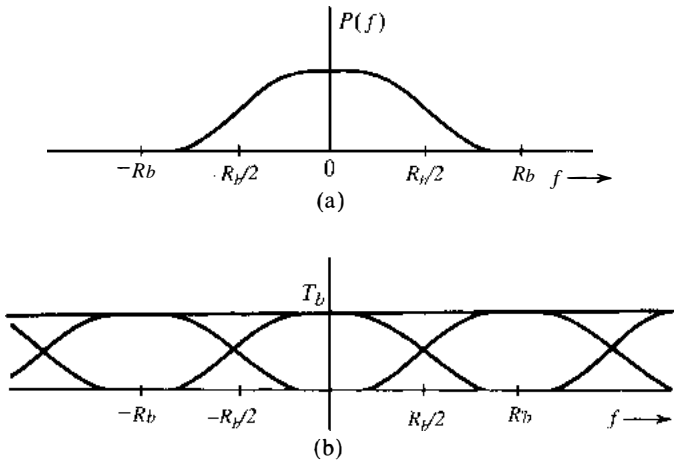


Figure 7.12
Derivation of the zero ISI Nyquist criterion pulse.

frequency R_b. Therefore, the Fourier transform of both sides of Eq. (7.25) yields

$$\frac{1}{T_b}\sum_{n=-\infty}^{\infty} P(f - nR_b) = 1 \qquad \text{where} \qquad R_b = \frac{1}{T_b} \tag{7.26}$$

or

$$\sum_{n=-\infty}^{\infty} P(f - nR_b) = T_b \tag{7.27}$$

Thus, the sum of the spectra formed by repeating $P(f)$ spaced R_b apart is a constant T_b, as shown in Fig. 7.12b.*

Consider the spectrum in Fig. 7.12b over the range $0 < f < R_b$. Over this range only two terms $P(f)$ and $P(f - R_b)$ in the summation in Eq. (7.27) are involved. Hence

$$P(f) + P(f - R_b) = T_b \qquad 0 < f < R_b$$

Letting $x = f - R_b/2$, we have

$$P(x + 0.5R_b) + P(x - 0.5R_b) = T_b \qquad |x| < 0.5R_b \tag{7.28a}$$

or, alternatively,

$$P\left(x + \frac{R_b}{2}\right) + P\left(x - \frac{R_b}{2}\right) = T_b \qquad |x| < 0.5R_b \tag{7.28b}$$

Use of the conjugate symmetry property [Eq. (3.11)] on Eq. (7.28) yields

$$P\left(\frac{R_b}{2} + x\right) + P^*\left(\frac{R_b}{2} - x\right) = T_b \qquad |x| < 0.5R_b \tag{7.29}$$

* Observe that if $R_b > 2B$, where B is the bandwidth (in hertz) of $P(f)$, the repetitions of $P(f)$ are nonoverlapping, and condition (7.27) cannot be satisfied. For $R_b = 2B$, the condition is satisfied only for the ideal low-pass $P(f)[p(t) = \text{sinc}(\pi R_b t)]$, which is not realizable. Hence, we must have $B > R_b/2$.

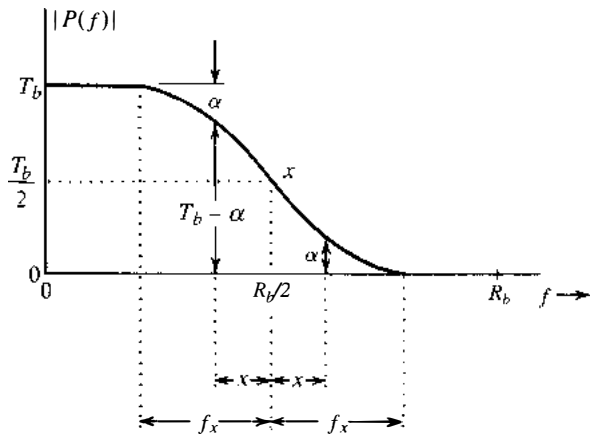


Figure 7.13 Vestigial (raised-cosine) spectrum.

If we choose $P(f)$ to be real-valued and positive then only $|P(f)|$ needs to satisfy Eq. (7.29). Because $|P(f)|$ is real, Eq. (7.29) implies

$$\left|P\left(\frac{R_b}{2}+x\right)\right| + \left|P\left(\frac{R_b}{2}-x\right)\right| = T_b \qquad |x| < 0.5R_b \tag{7.30}$$

Hence, $|P(f)|$ should be of the form shown in Fig. 7.13. This curve has an odd symmetry about the set of axes intersecting at point α [the point on $|P(f)|$ curve at $f = R_b/2$]. Note that this requires that

$$|P(0.5R_b)| = 0.5|P(0)|$$

The bandwidth, in hertz, of $P(f)$ is $0.5R_b + f_x$, where f_x is the bandwidth in excess of the minimum bandwidth $R_b/2$. Let r be the ratio of the excess bandwidth f_x to the theoretical minimum bandwidth $R_b/2$:

$$\begin{aligned} r &= \frac{\text{excess bandwidth}}{\text{theoretical minimum bandwidth}} \\ &= \frac{f_x}{0.5R_b} \\ &= 2f_xT_b \end{aligned} \tag{7.31}$$

Observe that because f_x cannot be larger than $R_b/2$,

$$0 \le r \le 1 \tag{7.32}$$

In terms of frequency f, the theoretical minimum bandwidth is $R_b/2$ Hz, and the excess bandwidth is $f_x = rR_b/2$ Hz. Therefore, the bandwidth of $P(f)$ is

$$B_T = \frac{R_b}{2} + \frac{rR_b}{2} = \frac{(1+r)R_b}{2} \tag{7.33}$$

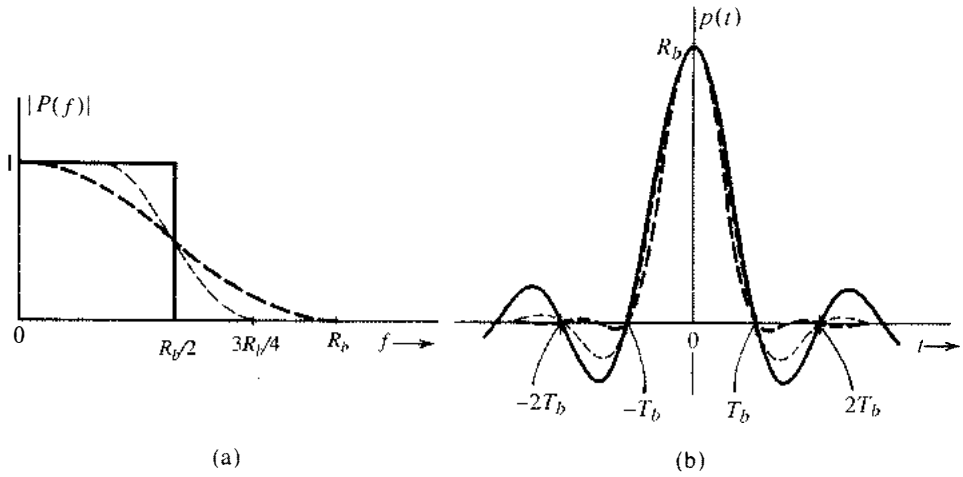


Figure 7.14 Pulses satisfying Nyquist's first criterion: solid curve, ideal $f_x = 0$ $(r = 0)$; light dashed curve, $f_x = R_b/4$ $(r = 0.5)$; heavy dashed curve, $f_x = R_b/2$ $(r = 1)$.

The constant r is called the **roll-off factor** and is also expressed in terms of percent. For example, if $P(f)$ is a Nyquist first criterion spectrum with a bandwidth that is 50% higher than the theoretical minimum, its roll-off factor $r = 0.5$ or 50%.

A filter having an amplitude response with the same characteristics is required in the vestigial sideband modulation discussed in Sec. 4.5 [Eq. (4.26)]. For this reason, we shall refer to the spectrum $P(f)$ in Eqs. (7.29) and (7.30) as a **vestigial spectrum**. The pulse $p(t)$ in Eq. (7.23) has zero ISI at the centers of all other pulses transmitted at a rate of R_b pulses per second. A pulse $p(t)$ that causes zero ISI at the centers of all the remaining pulses (or signaling instants) is the Nyquist first criterion pulse. We have shown that a pulse with a vestigial spectrum [Eq. (7.29) or Eq. (7.30)] satisfies the Nyquist's first criterion for zero ISI.

Because $0 \leq r < 1$, the bandwidth of $P(f)$ is restricted to the range $R_b/2$ to R_b Hz. The pulse $p(t)$ can be generated as a unit impulse response of a filter with transfer function $P(f)$. But because $P(f) = 0$ over a frequency band, it violates the Paley-Wiener criterion and is therefore unrealizable. However, the vestigial roll-off characteristic is gradual, and it can be more closely approximated by a practical filter. One family of spectra that satisfies Nyquist's first criterion is

$$P(f) = \begin{cases} 1, & |f| < \dfrac{R_b}{2} - f_x \\ \dfrac{1}{2}\left[1 - \sin \pi \left(\dfrac{f - R_b/2}{2f_x}\right)\right] & \left|f - \dfrac{R_b}{2}\right| < f_x \\ 0, & |f| > \dfrac{R_b}{2} + f_x \end{cases} \tag{7.34}$$

Figure 7.14a shows three curves from this family, corresponding to $f_x = 0$ $(r = 0)$, $f_x = R_b/4$ $(r = 0.5)$ and $f_x = R_b/2$ $(r = 1)$. The respective impulse responses are shown in Fig. 7.14b. It can be seen that increasing f_x (or r) improves $p(t)$; that is, more gradual cutoff reduces the oscillatory nature of $p(t)$ and causes it to decay more rapidly in time domain. For

the case of the maximum value of $f_x = R_b/2$ ($r = 1$), Eq. (7.34) reduces to

$$P(f) = \frac{1}{2}(1 + \cos \pi f T_b)\,\Pi\left(\frac{f}{2R_b}\right) \tag{7.35a}$$

$$= \cos^2\left(\frac{\pi f T_b}{2}\right)\Pi\left(\frac{f T_b}{2}\right) \tag{7.35b}$$

This characteristic of Eq. (7.34) is known in the literature as the **raised-cosine** characteristic, because it represents a cosine raised by its peak amplitude. Eq. (7.35) is also known as the **full-cosine roll-off** characteristic. The inverse Fourier transform of this spectrum is readily found as (see Prob 7.3-8)

$$p(t) = R_b \frac{\cos \pi R_b t}{1 - 4R_b^2 t^2}\,\text{sinc}\,(\pi R_b t) \tag{7.36}$$

This pulse is shown in Fig. 7.14b ($r = 1$). We can make several important observations about the raised-cosine pulse. First, the bandwidth of this pulse is R_b Hz and has a value R_b at $t = 0$ and is zero not only at all the remaining signaling instants but also at points midway between all the signaling instants. Second, it decays rapidly, as $1/t^3$. As a result, the raised-cosine pulse is relatively insensitive to deviations of R_b, sampling rate, timing jitter, and so on. Furthermore, the pulse-generating filter with transfer function $P(f)$ [Eq. (7.35b)] is closely realizable. The phase characteristic that goes along with this filter is very nearly linear, so that no additional phase equalization is needed.

It should be remembered that it is the pulses received at the detector input that should have the form for zero ISI. In practice, because the channel is not ideal (distortionless), the transmitted pulses should be shaped so that after passing through the channel with transfer function $H_c(f)$, they will be received with the proper shape (such as raised-cosine pulses) at the receiver. Hence, the transmitted pulse $p_i(t)$ should satisfy

$$P_i(f)H_c(f) = P(f)$$

where $P(f)$ has the vestigial spectrum in Eq. (7.30). For convenience, the transfer function $H_c(f)$ as a channel may also include a receiver filter designed to reject interference and other out-of-band noises.

Example 7.1 Determine the pulse transmission rate in terms of the transmission bandwidth B_T and the roll-off factor r. Assume a scheme using Nyquist's first criterion.

From Eq. (7.33)

$$R_b = \frac{2}{1 + r} B_T$$

Because $0 \le r \le 1$, the pulse transmission rate varies from $2B_T$ to B_T, depending on the choice of r. A smaller r gives a higher signaling rate. But the pulse $p(t)$ decays slowly, creating the same problems as those discussed for the sinc pulse. For the raised-cosine pulse $r = 1$ and $R_b = B_T$, we achieve half the theoretical maximum rate. But the pulse decays faster as $1/t^3$ and is less vulnerable to ISI.

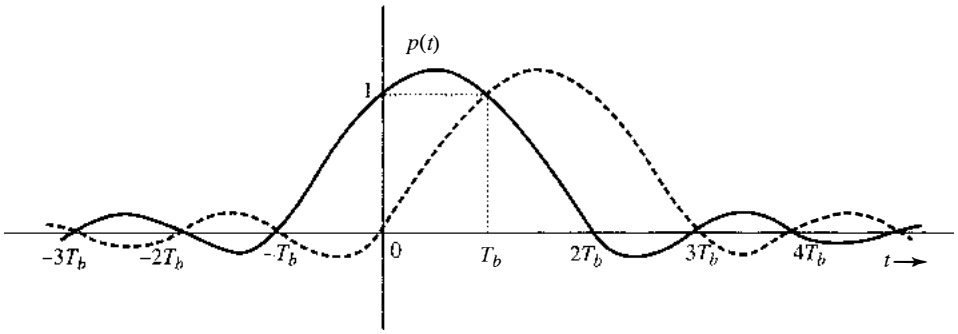


Figure 7.15 Communication using controlled ISI or Nyquist second criterion pulses.

7.3.3 Controlled ISI or Partial Response Signaling

The Nyquist criterion pulse requires in a bandwidth somewhat larger than the theoretical minimum. If we wish to further reduce the pulse bandwidth, we must find a way to widen the pulse $p(t)$ (the wider the pulse, the narrower the bandwidth). Widening the pulse may result in interference (ISI) with the neighboring pulses. However, in the binary transmission with just two possible symbols, a known and controlled amount of ISI may be possible to remove or compensate because there are only a few possible interference patterns.

Consider a pulse specified by (see Fig. 7.15):

$$p(nT_b) = \begin{cases} 1 & n = 0, 1 \\ 0 & \text{for all other } n \end{cases} \tag{7.37}$$

This leads to a known and controlled ISI from the kth pulse to the very next transmitted pulse. We use polar signaling by means of this pulse. Thus, **1** is transmitted by $p(t)$ and **0** is transmitted by using the pulse $-p(t)$. The received signal is sampled at $t = nT_b$, and the pulse $p(t)$ has zero value at all n except for $n = 0$ and 1, where its value is 1 (Fig. 7.15). Clearly, such a pulse causes zero ISI with all the pulses except the succeeding pulse. Therefore, we need to worry about the ISI with the succeeding pulse only. Consider two such successive pulses located at 0 and T_b, respectively. If both pulses were positive, the sample value of the resulting signal at $t = T_b$ would be 2. If the both pulses were negative, the sample value would be -2. But if the two pulses were of opposite polarity, the sample value would be 0. With only these three possible values, the signal sample clearly allows us to make correct decision at the sampling instants. The decision rule is as follows. If the sample value is positive, the present bit is **1** and the previous bit is also **1**. If the sample value is negative, the present bit is **0** and the previous bit is also **0**. If the sample value is zero, the present bit is the opposite of the previous bit. Knowledge of the previous bit then allows the determination of the present bit.

Table 7.1 shows a transmitted bit sequence, the sample values of the received signal $x(t)$ (assuming no errors causes by channel noise), and the detector decision. This example also indicates the error detecting property of this scheme. Examination of samples of the waveform $y(t)$ in Table 7.1 shows that there are always an even number of zero-valued samples between two full-valued samples of the same polarity and an odd number of zero-valued samples between two full-valued samples of opposite polarity. Thus, the first sample value of $x(t)$ is 2, and the next full-valued sample (the fourth sample) is 2. Between these full-valued samples

TABLE 7.1
Transmitted Bits and the Received Samples in Controlled ISI Signaling

Information sequence	1	1	0	1	1	0	0	0	1	0	1	1	1
Samples $y(kT_b)$	1	2	0	0	2	0	−2	−2	0	0	0	2	2
Detected sequence	1	1	0	1	1	0	0	0	1	0	1	1	1

of the same polarity, there are an even number (i.e., 2) of zero-valued samples. If one of the sample values is detected wrong, this rule is violated, and the error is detected.

The pulse $p(t)$ goes to zero at $t = -T_b$ and $2T_b$, resulting in the pulse width (of the primary lobe) 50% higher than that of the first criterion pulse. This pulse broadening in the time domain leads to reduction of its bandwidth. This is the second criterion proposed by Nyquist. This scheme of controlled ISI is also known as **correlative** or **partial-response** scheme. A pulse satisfying the second criterion in Eq. (7.37) is also known as the **duobinary pulse**.

7.3.4 Example of a Duobinary Pulse

If we restrict the pulse bandwidth to $R_b/2$, then following the procedure of Example 7.1, we can show that (see Prob 7.3-9) only the following pulse $p(t)$ meets the requirement in Eq. (7.37) for the duobinary pulse:

$$p(t) = \frac{\sin(\pi R_b t)}{\pi R_b t(1 - R_b t)} \tag{7.38}$$

The Fourier transform $P(f)$ of the pulse $p(t)$ is given by (see Prob 7.3-9)

$$P(f) = \frac{2}{R_b} \cos\left(\frac{\pi f}{R_b}\right) \Pi\left(\frac{f}{R_b}\right) e^{-j\pi f/R_b} \tag{7.39}$$

The pulse $p(t)$ and its amplitude spectrum $|P(f)|$ are shown in Fig. 7.16.* This pulse transmits binary data at a rate of R_b bit/s and has the theoretical minimum bandwidth $R_b/2$ Hz. Equation (7.38) shows that this pulse decays rapidly with time as $1/t^2$. This pulse is not ideally realizable because $p(t)$ is noncausal and has infinite duration [because $P(f)$ is band-limited]. However, it decays rapidly (as $1/t^2$), and therefore can be closely approximated.

It may come as a surprise that we are able to achieve the theoretical rate using the duobinary pulse. In fact, it is an illusion. The theoretical rate of transmission is 2 pieces of independent information per second per hertz bandwidth. We have achieved this rate for binary information. Here is the catch! A piece of binary information does not qualify as an independent piece of information because it cannot take on an arbitrary value. It must be selected from a finite set. The duobinary pulse would fail if the pulses were truly independent pieces of information, that is, if the pulses were to have arbitrary amplitudes. The scheme works only because the binary pulses take on finite known values, and hence, there are only a finite (known) number of interference patterns between pulses, which permits correct determination of pulse amplitudes despite interference.

* The phase spectrum is linear with $\theta_p(f) = -\pi f T_b$.

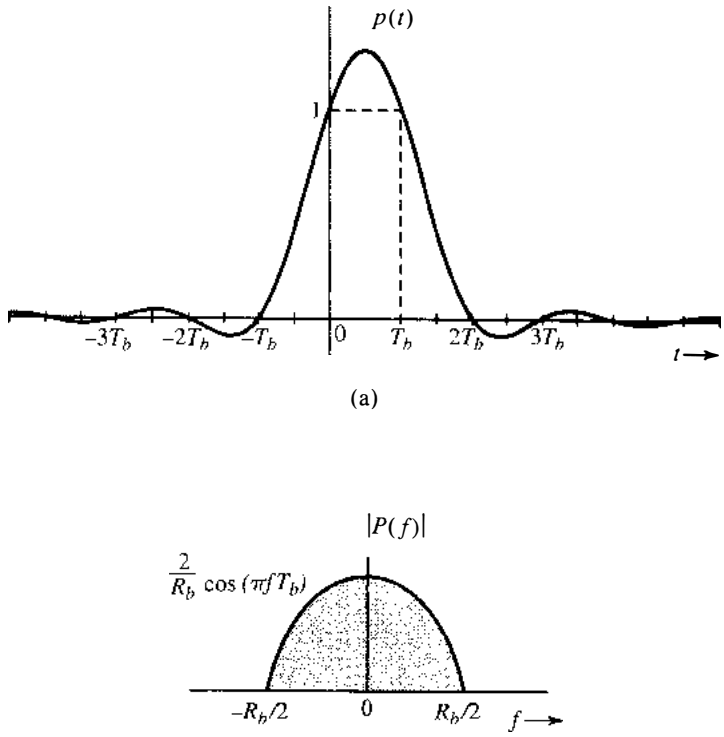


Figure 7.16
(a) The minimum bandwidth pulse that satisfies the duobinary pulse criterion and (b) its spectrum.

7.3.5 Pulse Relationship between Zero-ISI, Duobinary, and Modified Duobinary

Now we can establish the simple relationship between a pulse $p_a(t)$ satisfying the first Nyquist criterion (zero ISI) and a duobinary pulse $p_b(t)$ (with controlled ISI). From Eqs. (7.23) and (7.37), it is clear that $p_a(kT_b)$ and $p_b(kT_b)$ only differ for $k = 1$. They have identical sample values for all other integer k. Therefore, one can easily construct a pulse $p_b(t)$ from $p_a(t)$ by

$$p_b(t) = p_a(t) + p_a(t - T_b)$$

This addition is the "controlled" ISI or partial-response signaling that we deliberately introduced to reduce the bandwidth requirement. To see what effect "duobinary" signaling has on the spectral bandwidth, consider the relationship of the two pulses in the frequency domain:

$$P_b(f) = P_a(f)[1 + e^{-j2\pi fT_b}] \tag{7.40a}$$

$$|P_b(f)| = |P_a(f)|\sqrt{2(1 + \cos(2\pi fT_b)}2\,|\cos(\pi fT_b)| \tag{7.40b}$$

We can see that partial-response signaling is actually forcing a frequency null at $2\pi fT_b = \pi$ or, equivalently $f = 0.5/T_b$. Therefore, conceptually we can see how partial-response signaling provides an additional opportunity to reshape the PSD or the transmission bandwidth. Indeed, duobinary signaling, by forcing a frequency null at $0.5/T_b$, forces its essential bandwidth to be at the minimum transmission bandwidth needed for a data rate of $1/T_b$ (as discussed in Sec. 6.1.3).

In fact, many physical channels such as magnetic recording have a zero gain at dc. Therefore, it makes no sense for the baseband signal to have any dc component in its PSD. Modified partial-response signaling is often adopted to force a null at dc. One notable example is the so-called **modified duobinary** signaling that requires

$$p_c(nT_b) = \begin{cases} 1 & n = -1 \\ -1 & n = 1 \\ 0 & \text{for all other integers } n \end{cases} \tag{7.41}$$

A similar argument indicates that $p_c(t)$ can be generated from any pulse $p_a(t)$ satisfying the first Nyquist criterion via

$$p_c(t) = p_a(t + T_b) - p_a(t - T_b)$$

Equivalently, in the frequency domain, the duobinary pulse is

$$P_c(f) = 2jP_a(f) \sin(2\pi f T_b)$$

which uses $\sin(2\pi f T_b)$ to force a null at dc to comply with the physical channel constraint.

7.3.6 Detection of Duobinary Signaling and Differential Encoding

For the controlled ISI method of duobinary signaling, Fig. 7.17 shows the basic transmitter diagram. We now take a closer look at the relationship of all the data symbols at the baseband and the detection procedure. For binary message bit $I_k = 0$, or 1, the polar symbols are simply

$$a_k = 2I_k - 1$$

Under the controlled ISI, the samples of the transmission signal $y(t)$ are

$$y(kT_b) = b_k = a_k + a_{k-1} \tag{7.42}$$

The question for the receiver is how to *detect* I_k from $y(kT_b)$ or b_k. This question can be answered by first considering all the possible values of b_k or $y(kT_b)$. Because $a_k = \pm 1$, then $b_k = 0, \pm 2$. From Eq. (7.42), it is evident that

$$\begin{array}{lll} b_k = 2 & \Rightarrow \quad a_k = 1 & \text{or } I_k = 1 \\ b_k = -2 & \Rightarrow \quad a_k = -1 & \text{or } I_k = 0 \\ b_k = 0 & \Rightarrow \quad a_k = -a_{k-1} & \text{or } I_k = 1 - I_{k-1} \end{array} \tag{7.43}$$

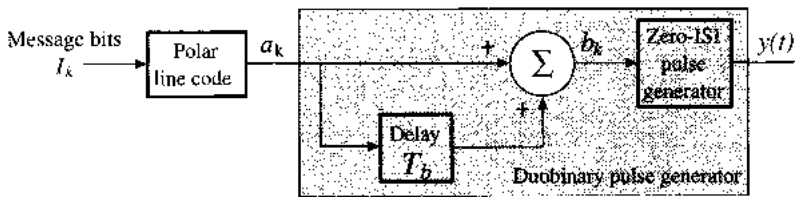


Figure 7.17 Equivalent duobinary signaling.

Therefore, a simple detector of duobinary signaling is to first detect all the bits I_k corresponding to $b_k = \pm 2$. The remaining $\{b_k\}$ are zero-valued samples that imply transition: that is, the current digit is **1** and the previous digit is **0**, or vice versa. This means the digit detection must be is based on the previous digit. An example of this digit-by-digit detection was shown in Table 7.1. The disadvantage of the detection method in Eq. (7.43) is that when $y(kT_b) = 0$, the current bit decision depends on the previous bit decision. If the previous digit were detected incorrectly, then the error would tend to propagate, until a sample value of ± 2 appears. To mitigate this error propagation problem, we apply a effective mechanism known as **differential coding**.

Figure 7.18 illustrates a duobinary signal generator by introducing an additional differential encoder prior to partial-response pulse generation. As shown in Fig. 7.18, differential encoding is a very simple step that changes the relationship between line code and the message bits. Differential encoding generates a new binary sequence

$$p_k = I_k \oplus p_{k-1} \qquad \text{modulo 2}$$

with the assumption that the precoder initial state is either $p_0 = 0$ or $p_0 = 1$. Now, the precoder output enters a polar line coder and generates

$$a_k = 2p_k - 1$$

Because of the duobinary signaling $b_k = a_k + a_{k-1}$ and the zero-ISI pulse generator, the samples of the received signal $y(t)$ without noise become

$$\begin{aligned} y(kT_b) = b_k &= a_k + a_{k-1} \\ &= 2(p_k + p_{k-1}) - 2 \\ &= 2\,(p_{k-1} \oplus I_k + p_{k-1} - 1) \\ &= \begin{cases} 2(1 - I_k) & p_{k-1} = 1 \\ 2(I_k - 1) & p_{k-1} = 0 \end{cases} \end{aligned} \tag{7.44}$$

Based on Eq. (7.44), we can summarize the direct relationship between the message bits and the sample values as

$$y(kT_b) = \begin{cases} 0 & I_k = 1 \\ \pm 2 & I_k = 0 \end{cases} \tag{7.45}$$

This relationship serves as our basis for a symbol-by-symbol detection algorithm. In short, the decision algorithm is based on the current sample $y(kT_b)$. When there is no noise, $y(kT_b) = b_k$ and the receiver decision is

$$I_k = \frac{2 - |y(kT_b)|}{2} \tag{7.46}$$

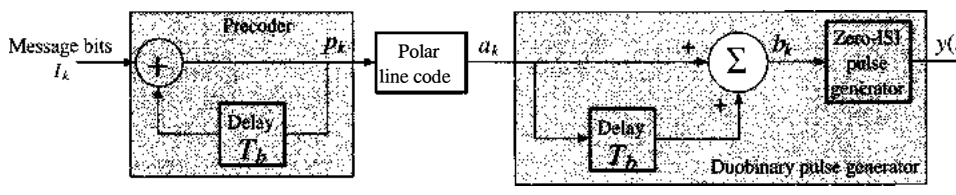


Figure 7.18 Differential encoded duobinary signaling.

TABLE 7.2
Binary Duobinary Signaling with Differential Encoding

Time k	0	1	2	3	4	5	6	7	8	9	10	11	12	13
I_k		1	1	0	1	1	0	0	0	1	0	1	1	1
p_k	0	1	0	0	1	0	0	0	0	1	1	0	1	0
a_k	−1	1	−1	−1	1	−1	−1	−1	−1	1	1	−1	1	−1
b_k		0	0	−2	0	0	−2	−2	−2	0	2	0	0	0
Detected bits		1	1	0	1	1	0	0	0	1	0	1	1	1

Therefore, the incorporation of differential encoding with duobinary signaling not only simplifies the decision rule but also makes the decision independent of the previous digit and eliminates error propagation. In Table 7.2, the example of Table 7.1 is recalculated with differential encoding. The decoding relationship of Eq. (7.45) is clearly shown in this example.

The differential encoding defined for binary information symbols can be conveniently generalized to nonbinary symbols. When the information symbols I_k are M-ary, the only change to the differential encoding block is to replace "modulo 2" with "modulo M." Similarly, other generalized partial-response signaling such as the modified duobinary must also face the error propagation problem at its detection. A suitable type of differential encoding can be similarly adopted to prevent error propagation.

7.3.7 Pulse Generation

A pulse $p(t)$ satisfying a Nyquist criterion can be generated as the unit impulse response of a filter with transfer function $P(f)$. This will not always be easy. A better method is to generate the waveform directly, using a transversal filter (tapped delay line) discussed here. The pulse $p(t)$ to be generated is sampled with a sufficiently small sampling interval T_s (Fig. 7.19a), and the filter tap gains are set in proportion to these sample values in sequence, as shown in Fig. 7.19b. When a narrow rectangular pulse with the width T_s, the sampling interval, is applied at the input of the transversal filter, the output will be a staircase approximation of $p(t)$. This output, when passed through a low-pass filter, is smoothed out. The approximation can be improved by reducing the pulse sampling interval T_s.

It should be stressed once again that the pulses arriving at the detector input of the receiver need to meet the desired Nyquist criterion. Hence, the transmitted pulses should be so shaped that after passing through the channel, they are received in the desired (Nyquist) form. In practice, however, pulses need not be shaped rigidly at the transmitter. The final shaping can be carried out by an equalizer at the receiver, as discussed later (Sec. 7.5).

7.4 SCRAMBLING

In general, a scrambler tends to make the data more random by removing long strings of **1**s or **0**s. Scrambling can be helpful in timing extraction by removing long strings of **0**s in binary data. Scramblers, however, are primarily used for preventing unauthorized access to the data, and they are optimized for that purpose. Such optimization may actually result in generation of a long string of zeros in the data. The digital network must be able to cope with these long zero strings by using the zero replacement techniques discussed in Sec. 7.2.

Figure 7.19
Pulse generation by transversal filter.

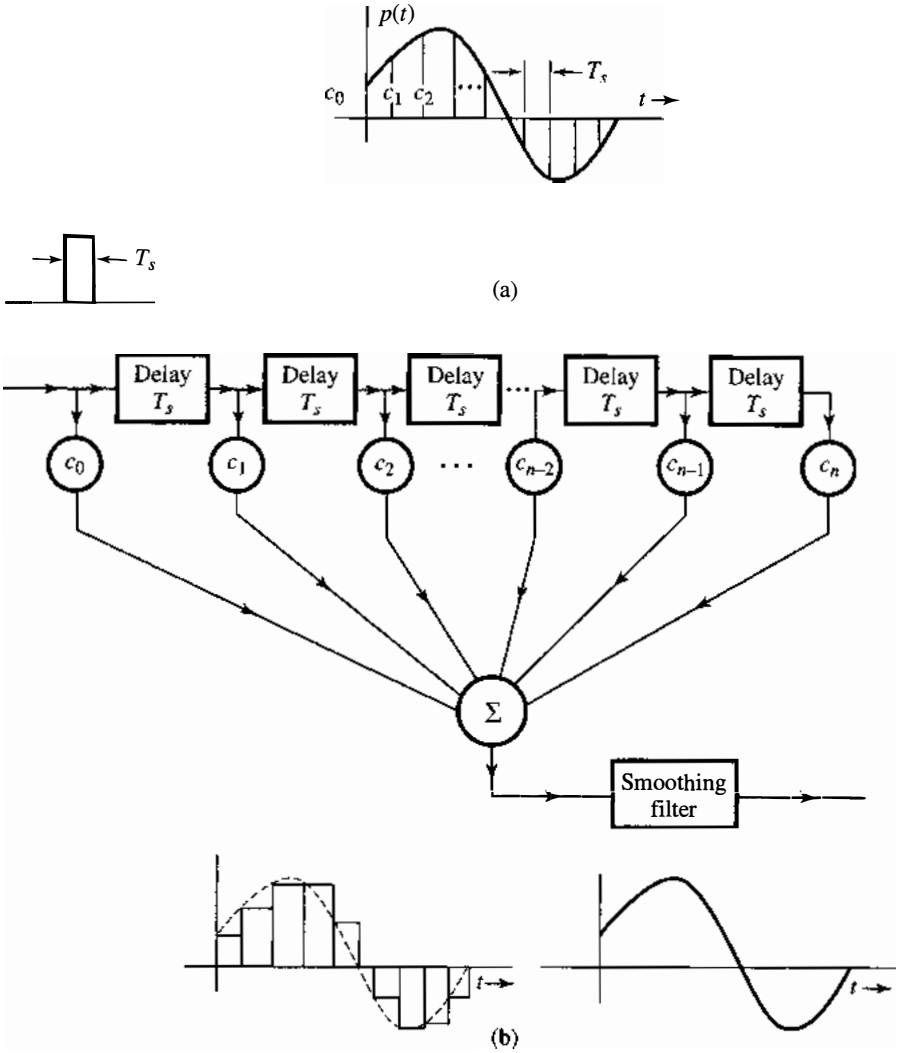


Figure 7.20
(a) Scrambler.
(b) Descrambler.

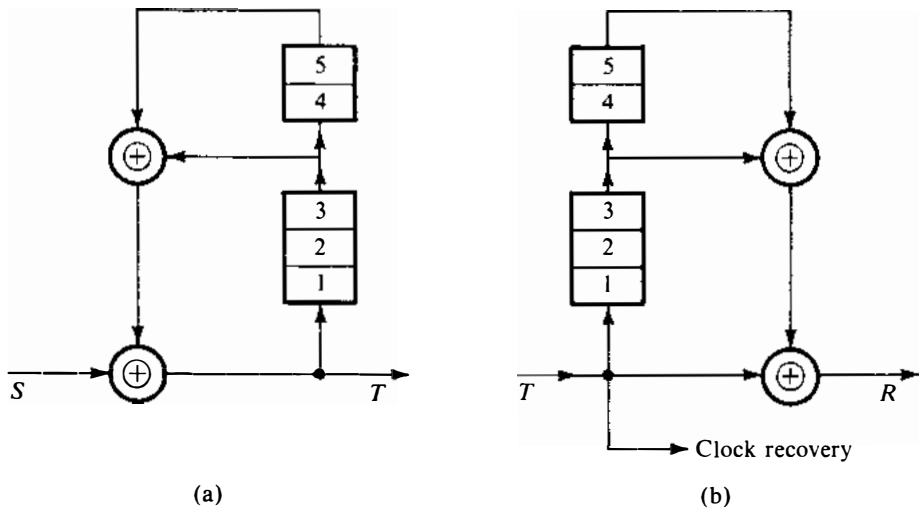


Figure 7.20 shows a typical scrambler and descrambler. The scrambler consists of a feedback shift register, and the matching descrambler has a feedforward shift register, as shown in Fig. 7.20. Each stage in the shift register delays a bit by one unit. To analyze the scrambler and the matched descrambler, consider the output sequence T of the scrambler (Fig. 7.20a). If S is the input sequence to the scrambler, then

$$S \oplus D^3T \oplus D^5T = T \tag{7.47}$$

where D represents the delay operator; that is, D^nT is the sequence T delayed by n units. Now, recall that the modulo 2 sum of any sequence with itself gives a sequence of all **0**s. Adding $(D^3 \oplus D^5)T$ to both sides of Eq. (7.47), we get

$$\begin{aligned} S &= T \oplus (D^3 \oplus D^5)T \\ &= [1 \oplus (D^3 \oplus D^5)]T \\ &= (1 \oplus F)T \end{aligned} \tag{7.48}$$

where $F = D^3 \oplus D^5$.

To design the descrambler at the receiver, we start with T, the sequence received at the descrambler. From Eq. (7.48), it follows that

$$T \oplus FT = T \oplus (D^3 \oplus D^5)T = S$$

This equation, in which we regenerate the input sequence S from the received sequence T, is readily implemented by the descrambler shown in Fig. 7.20b.

Note that a single detection error in the received sequence T will affect three output bits in R. Hence, scrambling has the disadvantage of causing multiple errors for a single received bit error.

Example 7.2 The data stream **101010100000111** is fed to the scrambler in Fig. 7.20a. Find the scrambler output T, assuming the initial content of the registers to be zero.

From Fig. 7.20a we observe that initially $T = S$, and the sequence S enters the register and is returned as $(D^3 \oplus D^5)S = FS$ through the feedback path. This new sequence FS again enters the register and is returned as F^2S, and so on. Hence

$$\begin{aligned} T &= S \oplus FS \oplus F^2S \oplus F^3S \oplus \cdots \\ &= (1 \oplus F \oplus F^2 \oplus F^3 \oplus \cdots)S \end{aligned} \tag{7.49}$$

Recognizing that

$$F = D^3 \oplus D^5$$

we have

$$F^2 = (D^3 \oplus D^5)(D^3 \oplus D^5) = D^6 \oplus D^{10} \oplus D^8 \oplus D^8$$

Because modulo-2 addition of any sequence with itself is zero, $D^8 \oplus D^8 = 0$, and

$$F^2 = D^6 \oplus D^{10}$$

Similarly

$$F^3 = (D^6 \oplus D^{10})(D^3 \oplus D^5) = D^9 \oplus D^{11} \oplus D^{13} \oplus D^{15}$$

and so on. Hence [see Eq. (7.49)],

$$T = (1 \oplus D^3 \oplus D^5 \oplus D^6 \oplus D^9 \oplus D^{10} \oplus D^{11} \oplus D^{12} \oplus D^{13} \oplus D^{15} \cdots)S$$

Because D^nS is simply the sequence S delayed by n bits, various terms in the above equation correspond to the following sequences:

$$\begin{aligned}
S &= \mathbf{101010100000111} \\
D^3S &= \mathbf{000101010100000111} \\
D^5S &= \mathbf{0000010101010000111} \\
D^6S &= \mathbf{0000001010101000001111} \\
D^9S &= \mathbf{000000000101010100000111} \\
D^{10}S &= \mathbf{0000000000101010100000111} \\
D^{11}S &= \mathbf{00000000000101010100000111} \\
D^{12}S &= \mathbf{000000000000101010100000111} \\
D^{13}S &= \mathbf{0000000000000101010100000111} \\
\underline{D^{15}S} &= \underline{\mathbf{00000000000000010101010000111}} \\
T &= \mathbf{101110001101001}
\end{aligned}$$

Note that the input sequence contains the periodic sequence **10101010** $\cdots$, as well as a long string of **0**s. The scrambler output effectively removes the periodic component, as well as the long string of **0**s. The input sequence has 15 digits. The scrambler output up to the 15th digit only is shown, because all the output digits beyond 15 depend on input digits beyond 15, which are not given.

Readers can verify that the descrambler output is indeed S when the foregoing sequence T is applied at its input.

7.5 DIGITAL RECEIVERS AND REGENERATIVE REPEATERS

Basically, a receiver or a regenerative repeater performs three functions: (1) reshaping incoming pulses by means of an equalizer, (2) extracting the timing information required to sample incoming pulses at optimum instants, and (3) making symbol detection decisions based on the pulse samples. The repeater shown in Fig. 7.21 consists of a receiver plus a "regenerator." A complete repeater may also include provision for separation of dc power from ac signals. This

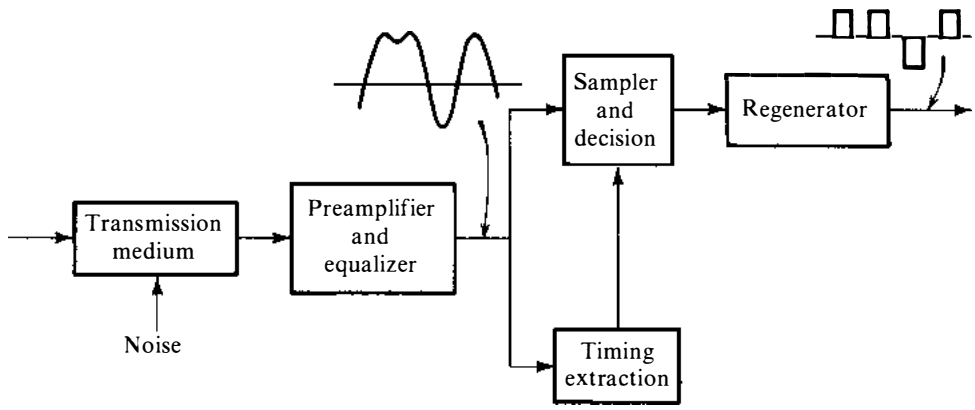


Figure 7.21 Regenerative repeater.

is normally accomplished by transformer-coupling the signals and bypassing the dc around the transformers to the power supply circuitry.*

7.5.1 Equalizers

A pulse train is attenuated and distorted by the transmission medium. The attenuation can be compensated by the preamplifier, whereas the distortion is compensated by the equalizer. Channel distortion is in the form of dispersion, which is caused by an attenuation of certain *critical frequency components* of the data pulse train. Theoretically, an equalizer should have a frequency characteristic that is the inverse of that of the transmission medium. This will restore the critical frequency components and eliminate pulse dispersion. Unfortunately, this also enhances the received channel noise by boosting its components at these critical frequencies. This undesirable phenomenon is known as *noise amplification.*

For digital signals, however, complete equalization is really not necessary, because a detector only needs to make relatively simple decisions—such as whether the pulse is positive or negative (or whether the pulse is present or absent). Therefore, considerable pulse dispersion can be tolerated. Pulse dispersion results in ISI and the consequent increase in error detection. Noise increase resulting from the equalizer (which boosts the high frequencies) also increases the detection error probability. For this reason, design of an optimum equalizer involves an inevitable compromise between reducing ISI and reducing the channel noise. A judicious choice of the equalization characteristics is a central feature in all well-designed digital communication systems.[6]

Zero-Forcing Equalizer

It is really not necessary to eliminate or minimize ISI (interference) with neighboring pulses for all t. All that is needed is to eliminate or minimize interference with neighboring pulses at their respective *sampling instants* only. This is because the receiver decision is based on sample values only. This kind of (relaxed) equalization can be accomplished by equalizers using the transversal filter structure encountered earlier. Unlike traditional filters, transversal

* The repeater usually includes circuitry to protect the electronics of the regenerator from high-voltage transients induced by power surges and lightning. Special transformer windings may be provided to couple fault-locate signals into a cable pair dedicated to the purpose.

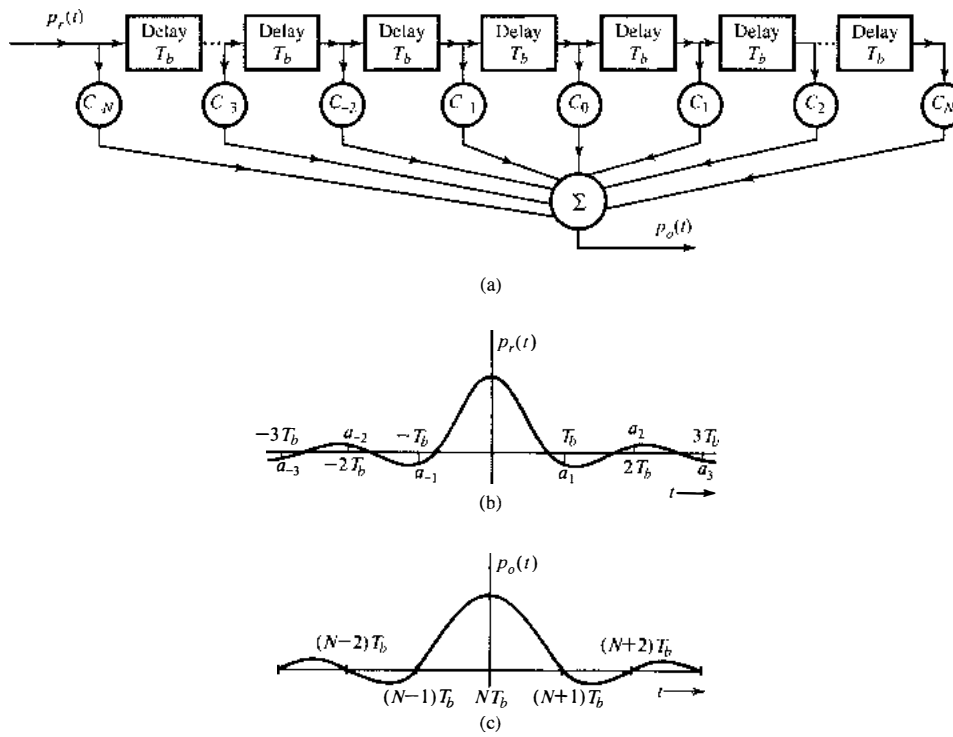


Figure 7.22 Zero-forcing equalizer analysis.

filter equalizers are easily adjustable to compensate against different channels or even slowly time-varying channels. The design goal is to force the equalizer output pulse to have zero ISI values at the sampling (decision-making) instants. In other words, the equalizer output pulses satisfy the Nyquist or the controlled ISI criterion. The time delay T between successive taps is chosen to be T_b, the interval between pulses.

To begin, set the tap gains $c_0 = 1$ and $c_k = 0$ for all other values of k in the transversal filter in Fig. 7.22a. Thus the output of the filter will be the same as the input delayed by NT_b. For a single pulse $p_r(t)$ (Fig. 7.22b) at the input of the transversal filter with the tap setting just given, the filter output $p_o(t)$ will be exactly $p_r(t - NT_b)$, that is, $p_r(t)$ delayed by NT_b. This delay has no practical effect on our communication system and is not relevant to our discussion. Hence, for convenience, we shall ignore this delay. This means that $p_r(t)$ in Fig. 7.22b also represents the filter output $p_o(t)$ for this tap setting ($c_0 = 1$ and $c_k = 0$, $k \neq 0$). We require that the output pulse $p_o(t)$ satisfy the Nyquist's criterion or the controlled ISI criterion, as the case may be. For the Nyquist criterion, the output pulse $p_o(t)$ must have zero values at all the multiples of T_b. From Fig. 7.22b, we see that the pulse amplitudes a_1, a_{-1}, and a_2 at T_b, $-T_b$, and $2T_b$, respectively, are not negligible. By adjusting the tap gains (c_k), we generate additional shifted pulses of proper amplitudes that will force the resulting output pulse to have desired values at $t = 0, \pm T_b, \pm 2T_b, \ldots$.

The output $p_o(t)$ (Fig 7.22c) is the sum of pulses of the form $c_k p_r(t - kT_b)$ (ignoring the delay of NT_b). Thus

$$p_o(t) = \sum_{n=-N}^{N} c_n p_r(t - nT_b) \tag{7.50}$$

The samples of $p_o(t)$ at $t = kT_b$ are

$$p_o(kT_b) = \sum_{n=-N}^{N} c_n p_r(kT_b - nT_b) \qquad k = 0, \pm 1, \pm 2, \pm 3, \ldots \tag{7.51a}$$

By using a more convenient notation $p_r[k]$ to denote $p_r(kT_b)$ and $p_o[k]$ to denote $p_o(kT_b)$, Eq. (7.51a) can be expressed as

$$p_o[k] = \sum_{n=-N}^{N} c_n p_r[k - n] \qquad k = 0, \pm 1, \pm 2, \pm 3, \ldots \tag{7.51b}$$

Nyquist's first criterion requires the samples $p_o[k] = 0$ for $k \neq 0$, and $p_o[k] = 1$ for $k = 0$. Upon substituting these values in Eq. (7.51b), we obtain a set of infinite simultaneous equations in terms of $2N + 1$ variables. Clearly, it is not possible to solve all the equations. However, if we specify the values of $p_o[k]$ only at $2N + 1$ points as

$$p_o[k] = \begin{cases} 1 & k = 0 \\ 0 & k = \pm 1, \ \pm 2, \ldots, \pm N \end{cases} \tag{7.52}$$

then a unique solution exists. This assures that a pulse will have zero interference at sampling instants of N preceding and N succeeding pulses. Because the pulse amplitude decays rapidly, interference beyond the Nth pulse is not significant for $N > 2$, in general. Substitution of the condition (7.52) into Eq. (7.51b) yields a set of $2N + 1$ simultaneous equations for $2N + 1$ variables. These $2N + 1$ equations can be rewritten in the matrix form of

$$\underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\mathbf{p}_o} = \underbrace{\begin{bmatrix} p_r[0] & p_r[-1] & \cdots & p_r[-2N+1] & p_r[-2N] \\ p_r[1] & p_r[0] & \cdots & p_r[-2N+2] & p_r[-2N+1] \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ p_r[2N-1] & p_r[2N-2] & \ddots & p_r[0] & p_r[-1] \\ p_r[2N] & p_r[2N-1] & \cdots & p_r[1] & p_r[0] \end{bmatrix}}_{\mathbf{P}_r} \underbrace{\begin{bmatrix} c_{-N} \\ c_{-N+1} \\ \vdots \\ c_{-1} \\ c_0 \\ c_1 \\ \vdots \\ c_{N-1} \\ c_N \end{bmatrix}}_{\mathbf{c}} \tag{7.53}$$

In this compact expression, the $(2N + 1) \times (2N + 1)$ matrix $\mathbf{P}_r$ has identical entries along all the diagonal lines. Such a matrix is known as the Toeplitz matrix and is commonly encountered in describing convolutive relationships. A Toeplitz matrix is fully determined by its first row and first column. It has some nice properties and admits simpler algorithms for computing its inverse (see, e.g., the method by Trench[7]). The tap gain c_k can be obtained by solving this set

of equations by taking the inverse of the matrix $\mathbf{P}_r$

$$\mathbf{c} = \mathbf{P}_r^{-1}\mathbf{p}_o$$

Example 7.3 For the received pulse $p_r(t)$ in Fig. 7.22b, let

$$\begin{aligned} p_r[0] &= 1 \\ p_r[1] = -0.3 \quad & p_r[2] = 0.1 \\ p_r[-1] = -0.2 \quad & p_r[-2] = 0.05 \end{aligned}$$

Design a three-tap ($N = 1$) equalizer.

Substituting the foregoing values in Eq. (7.53), we obtain

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & -0.2 & 0.05 \\ -0.3 & 1 & -0.2 \\ 0.1 & -0.3 & 1 \end{bmatrix} \begin{bmatrix} c_{-1} \\ c_0 \\ c_1 \end{bmatrix} \tag{7.54}$$

Solution of this set yields $c_{-1} = 0.210$, $c_0 = 1.13$, and $c_1 = 0.318$. This tap setting assures us that $p_0[0] = 1$ and $p_0[-1] = p_0[1] = 0$. The ideal output $p_o(t)$ is sketched in Fig. 7.22c.

Note that the equalizer determined from Eq. (7.53) can guarantee only the zero ISI condition of Eq. (7.52). In other words, ISI is zero only for $k = 0, \pm 1, \ldots, \pm N$. In fact, for k outside this range, it is quite common that the samples $p_o(kT_b) \neq 0$, indicating some residual ISI. For instance, consider the equalizer problem in Example 7.3. The samples of the equalized pulse has zero ISI for $k = -1, 0, 1$. However, from

$$p_o[k] = \sum_{n=-N}^{N} c_n p_r[k-n]$$

we can see that the three-tap zero-forcing equalizer parameters will lead to

$$\begin{aligned} p_o[-3] = 0.010 \quad & p_o[-2] = 0.0145 \quad p_o[2] = 0.0176 \\ p_o[3] = 0.0318 \quad & p_o[k] = 0 \quad k = 0, \pm 1, \pm 4, \ldots \end{aligned}$$

It is therefore clear that not all the ISI has been removed because of these four nonzero samples of the equalizer output pulse. In fact, because we only have $2N + 1$ ($N = 1$ in Example 7.3) parameters in the equalizer, it is impossible to force $p_o[k] = 0$ for all k unless $N = \infty$. This means that we will not be able to design a practical finite tap equalizer that achieves perfect zero ISI. Still, when N is sufficiently large, then typically the residual nonzero sample values will be small, indicating that most of the ISI has been suppressed.

Minimum Mean Square Error (MMSE) Method

In practice, an alternative approach is to minimize the mean square difference between the equalizer output response $p_o[k]$ and the desired zero ISI response. This is known as the minimum mean square error (MMSE) method for designing transversal filter equalizers. The MMSE method does not try to force the pulse samples to zero at $2N$ points. Instead, we minimize the squared errors averaged over a set of output samples. This method involves more simultaneous equations. Thus we must find the equalizer tap values to minimize the average (mean) square error over a larger window $[-K, K]$:

$$\text{MSE} \triangleq \frac{1}{2K+1} \sum_{k=-K}^{K} (p_o[k] - \delta[k])^2$$

where we use a function known as the Kronecker delta

$$\delta[k] = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}$$

The solution to this minimization problem can be better represented in matrix form as

$$\mathbf{c} = \mathbf{P}_r^{\dagger} \mathbf{p}_o$$

where $\mathbf{P}_r^{\dagger}$ represents the Moore-Penrose pseudo-inverse of the nonsquare matrix $\mathbf{P}_r$ of size $(2K + 1) \times (2N + 1)$. The MMSE design often leads to a more robust equalizer for the reduction of ISI.

Adaptive Equalization and Other More General Equalizers

The equalizer filter structure that is described here has the simplest form. Practical digital communication systems often apply much more sophisticated equalizer structures and more advanced equalization algorithms.[6] Because of the probabilistic tools needed, we will defer detailed coverage on the specialized topic of equalization to Chapter 12.

7.5.2 Timing Extraction

The received digital signal needs to be sampled at precise instants. This requires a clock signal at the receiver in synchronism with the clock signal at the transmitter (**symbol** or **bit synchronization**), delayed by the channel response. Three general methods of synchronization exist:

1. Derivation from a primary or a secondary standard (e.g., transmitter and receiver slaved to a master timing source).
2. Transmitting a separate synchronizing signal (pilot clock).
3. Self-synchronization, where the timing information is extracted from the received signal itself.

Because of its high cost, the first method is suitable for large volumes of data and high-speed communication systems. The second method, in which part of the channel capacity is used to transmit timing information, is suitable when the available capacity is large in comparison to the data rate and when additional transmission power can be spared. The third method is

a very efficient method of timing extraction or clock recovery because the timing is derived from the received message signal itself. An example of the self-synchronization method will be discussed here.

We have already shown that a digital signal, such as an on-off signal (Fig. 7.3a), contains a discrete component of the clock frequency itself (Fig. 7.3c). Hence, when the on-off binary signal is applied to a resonant circuit tuned to the clock frequency, the output signal is the desired clock signal.

Not all the binary signals contain a discrete component of the clock frequency. For example, a bipolar signal has no discrete component of any frequency [see Eq. (7.21) or Fig. 7.9]. In such cases, it may be possible to extract timing by using a *nonlinear device* to generate a frequency tone that is related to the timing clock. In the bipolar case, for instance, a simple rectification converts a bipolar signal to an on-off signal, which can readily be used to extract timing.

Small random deviations of the incoming pulses from their ideal location (known as **timing jitter**) are always present, even in the most sophisticated systems. Although the source emits pulses at the right instants, subsequent operations during transmission (e.g., Doppler shift) tend to cause pulses to deviate from these original positions. The Q of the tuned circuit used for timing extraction must be large enough to provide an adequate suppression of timing jitter, yet small enough to meet the stability requirements. During the intervals in which there are no pulses in the input, the oscillation continues because of the flywheel effect of the high-Q circuit. But still the oscillator output is sensitive to the pulse pattern; for example, during a long string of **1**s the output amplitude will increase, whereas during a long string of **0**s it will decrease. This introduces additional jitter in the timing signal extracted.

The complete timing extractor and time pulse generator for a polar case in shown in Fig. 7.23. The sinusoidal output of the oscillator (timing extractor) is passed through a phase shifter that adjusts the phase of the timing signal so that the timing pulses occur at the maximum points. This method is used to recover the clock at each of the regenerators in a PCM system. The jitter introduced by successive regenerators adds up, and after a certain number of regenerators it is necessary to use a regenerator with a more sophisticated clock recovery system such as a phase-locked loop.

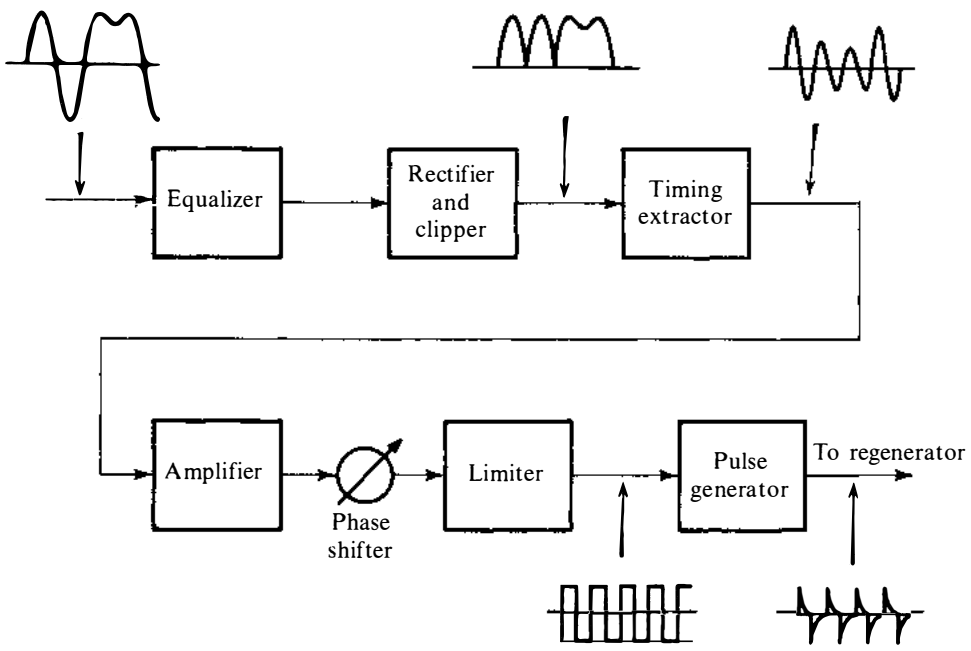


Figure 7.23 Timing extraction.

Timing Jitter

Variations of the pulse positions or sampling instants cause timing jitter. This results from several causes, some of which are dependent on the pulse pattern being transmitted, whereas others are not. The former are cumulative along the chain of regenerative repeaters, since all the repeaters are affected in the same way, whereas the other forms of jitter are random from regenerator to regenerator and therefore tend to partially cancel out their mutual effects over a long-haul link. Random forms of jitter are caused by noise, interference, and mistuning of the clock circuits. Pattern-dependent jitter results from clock mistuning, amplitude-to-phase conversion in the clock circuit, and ISI, which alters the position of the peaks of the input signal according to the pattern. The rms value of the jitter over a long chain of N repeaters can be shown to increase as $\sqrt{N}$.

Jitter accumulation over a digital link may be reduced by buffering the link with an elastic store and clocking out the digit stream under the control of a highly stable phase-locked loop. Jitter reduction is necessary about every 200 miles in a long digital link to keep the maximum jitter within reasonable limits.

7.5.3 Detection Error

Once the transmission has passed through the equalizer, detection can take place at the detector that samples the received signal based on the clock provided by the timing extractor. The signal received at the detector consists of the equalized pulse train plus a random channel noise. The noise can cause error in pulse detection. Consider, for example, the case of polar transmission using a basic pulse $p(t)$ (Fig. 7.24a). This pulse has a peak amplitude A_p. A typical received pulse train is shown in Fig. 7.24b. Pulses are sampled at their peak values. If noise were absent, the sample of the positive pulse (corresponding to **1**) would be A_p and that of the negative pulse (corresponding to **0**) would be $-A_p$.* Because of noise, these samples would be $\pm A_p + n$ where n is the random noise amplitude (see Fig. 7.24b). From the symmetry of the situation, the detection threshold is zero; that is, if the pulse sample value is positive, the digit is detected as **1**; if the sample value is negative, the digit is detected as **0**.

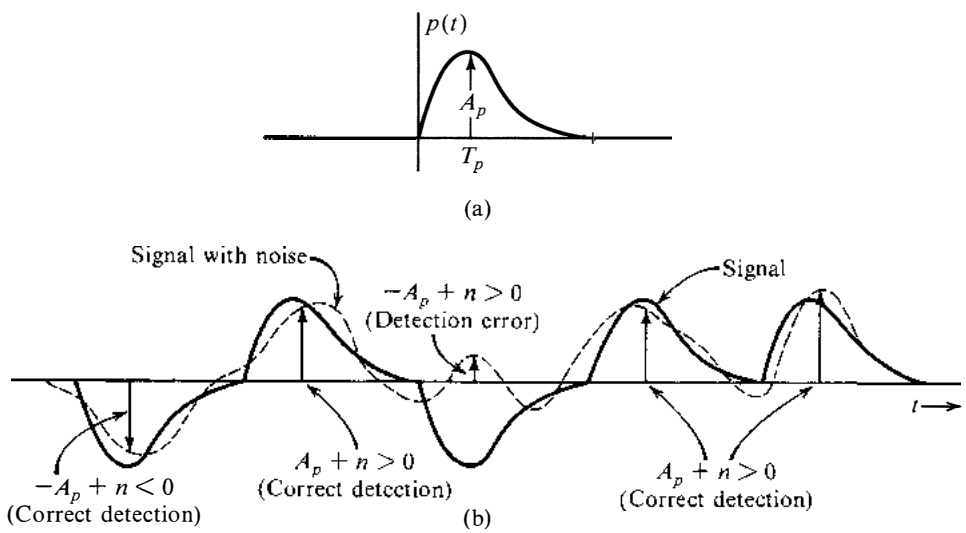


Figure 7.24 Error probability in threshold detection.

* This assumes zero ISI.

The detector's decision of whether to declare **1** or **0** could be made readily from the pulse sample, except that the noise value n is random, meaning that its exact value is unpredictable. It may have a large or a small value, and it can be negative as well as positive. It is possible that **1** is transmitted but n at the sampling instant has a large negative value. This will make the sample value $A_p + n$ small or even negative. On the other hand, if **0** is transmitted and n has a large positive value at the sampling instant, the sample value $-A_p + n$ can be positive and the digit will be detected wrongly as **1**. This is clear from Fig. 7.24b.

The performance of digital communication systems is typically specified by the average number of detection errors. For example, if two cellphones (receivers) in the same spot are attempting to detect the same transmission from a cellular tower, the cellphone with the lower number of detection errors is the better receiver. It is likely to have fewer dropped calls and less trouble receiving clear speech. However, because noise is random, sometimes one cellphone may be better while other times the other cellphone may have fewer errors. The real measure of receiver performance is therefore the average ratio of the number of errors to the total number of transmitted data. Thus, the meaningful performance comparison is the likelihood of detection error, or the **detection error probability**.

Because the precise analysis and evaluation of this error likelihood require the knowledge and tools from probability theory, we will postpone error analysis until after the introduction of probability in Chapter 8. Later, in Chapter 10, we will discuss fully the error probability analysis of different digital communication systems for different noise models as well as system designs against different noises. For example, Gaussian noise can generally characterize the random channel noise from thermal effects and intersystem cross talk. Optimum detectors can be designed to minimize the error likelihood against Gaussian noise. However, switching transients, lightning strikes, power line load switching, and other singular events cause very high level noise pulses of short duration to contaminate the cable pairs that carry digital signals. These pulses, collectively called **impulse noise**, cannot conveniently be engineered away, and they constitute the most prevalent source of errors from the environment outside the digital systems. Errors are virtually never, therefore, found in isolation, but occur in bursts of up to several hundred at a time. To correct error burst, we use special **burst error correcting codes** described in Chapter 14.

7.6 EYE DIAGRAMS: AN IMPORTANT TOOL

In the last section, we studied the effect of noise and channel ISI on the detection of digital transmissions. We also described the design of equalizers to compensate the channel distortion and explained the timing-extraction process. We now present a practical engineering tool known as the **eye diagram**. The eye diagram is easy to generate and is often applied by engineers on received signals because it makes possible the visual examination of severity of the ISI, the accuracy of timing extraction, the noise immunity, and other important factors.

We need only a basic oscilloscope to generate the eye diagram. Given a baseband signal at the channel output

$$y(t) = \sum a_k p(t - kT_b)$$

it can be applied to the vertical input of the oscilloscope. The time base of the scope is triggered at the same rate $1/T_b$ as that of the incoming pulses, and it yields a sweep lasting exactly T_b, the interval of one transmitted data symbol a_k. The oscilloscope shows the superposition of

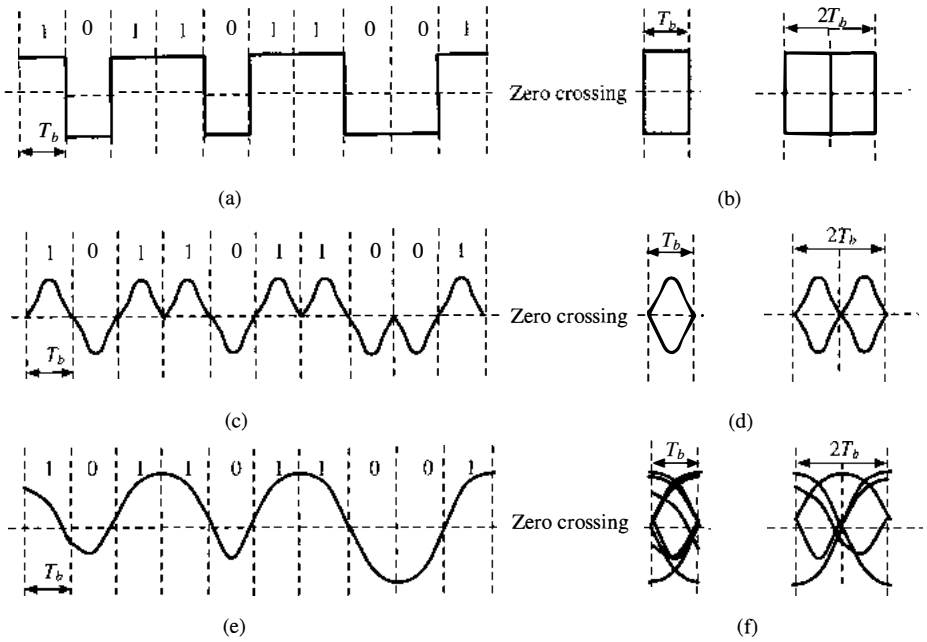


Figure 7.25 The eye diagram.

many traces of length T_b from the channel output $y(t)$. What appears on the oscilloscope is simply the input signal (vertical input) cut up every T_b and then superimposed on top of one another. The resulting pattern on the oscilloscope looks like a human eye, hence the name eye diagram. More generally, we can also apply a time sweep that lasts m symbol intervals, or mT_b. The oscilloscope pattern is simply the input signal (vertical input) cut up every mT_b and then superimposed on top of one another. The oscilloscope will then display an eye diagram that is mT_b wide and has the shape of m eyes in a horizontal row.

We now present an example. Consider the transmission of a binary signal by polar NRZ pulses (Fig. 7.25a). Its eye diagrams are shown in Fig. 7.25b for the time base of T_b and $2T_b$, respectively. In this example, the channel has infinite bandwidth to pass the NRZ pulse and there is no channel distortion. Hence, we obtain eye diagrams with totally *open* eye(s). We can also consider a channel output using the same polar line code and a different (RZ) pulse shape, as shown in Fig. 7.25c. The resulting eye diagrams are shown in Fig. 7.25d. In this case, the eye is wide open only at the midpoint of the pulse duration. With proper timing extraction, the receiver should sample the received signal right at the midpoint where the eye is totally open, to achieve the best noise immunity at the decision point (Sec. 7.5.3). This is because the midpoint of the eye represents the best sampling instant of each pulse, where the pulse amplitude is maximum without interference from any other neighboring pulse (zero ISI).

We now consider a channel that is distortive or has finite bandwidth, or both. After passing through this nonideal channel, the NRZ polar signal of Fig. 7.25a becomes the waveform of Fig. 7.25e. The received signal pulses are no longer rectangular but are be rounded, distorted, and spread out. The eye diagrams are not fully open anymore, as shown in Fig. 7.25f. In this case, the ISI is not zero. Hence, pulse values at their respective sampling instants will deviate from the full-scale values by a varying amount in each trace, causing blurs, resulting in a partially closed eye pattern.

In the presence of channel noise, the eye will tend to close in all cases. Weaker noise will cause proportionately less closing. The decision threshold with respect to which symbol

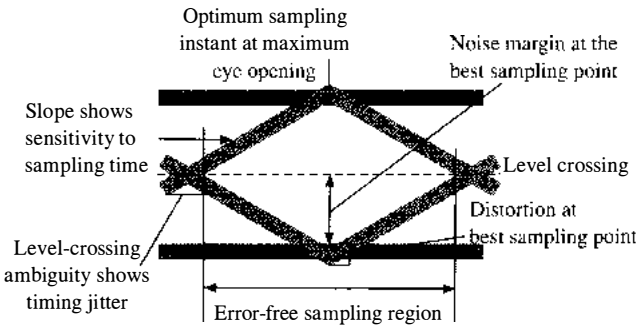


Figure 7.26 Reading an eye diagram.

(**1** or **0**) was transmitted is the midpoint of the eye.* Observe that for zero ISI, the system can tolerate noise of up to half the vertical opening of the eye. Any noise value larger than this amount can cause a decision error if its sign is opposite to the sign of the data symbol. Because ISI reduces the eye opening, it clearly reduces noise tolerance. The eye diagram is also used to determine optimum tap settings of the equalizer. Taps are adjusted to obtain the maximum vertical and horizontal eye opening.

The eye diagram is a very effective tool for signal analysis during real-time experiments. It not only is simple to generate, it also provides very rich and important information about the quality and susceptibility of the received digital signal. From the typical eye diagram given in Fig. 7.26, we can extract several key measures regarding the signal quality.

- *Maximum opening point.* The eye opening amount at the sampling and decision instant indicates that amount of noise the detector can tolerate without making an error. The quantity is known as the *noise margin.* The instant of maximum eye opening indicates the optimum sampling or decision-making instant.
- *Sensitivity to timing jitter.* The width of the eye indicates the time interval over which correct decision can still be made, and it is desirable to have an eye with the maximum horizontal opening. If the decision-making instant deviates from the instant when the eye has a maximum vertical opening, the margin of noise tolerance is reduced. This causes higher error probability in pulse detection. The slope of the eye shows how fast the noise tolerance is reduced and, hence, the sensitivity of the decision noise tolerance to variation of the sampling instant. It demonstrates the effects of timing jitter.
- *Level-crossing (timing) jitter.* Typically, practical receivers extract timing information about the pulse rate and the sampling clock from the (zero) level crossing of the received signal waveform. The variation of level crossing can be seen from the width of the eye corners. This measure provides information about the timing jitter such a receiver is expected to experience.

Finally, we provide a practical eye diagram example for a polar signaling waveform. In this case, we select a cosine roll-off pulse that satisfy Nyquist's first criterion of zero ISI. The roll-off factor is chosen to be $r = 0.5$. The eye diagram is shown in Fig. 7.27 for a time base of $2T_b$. In fact, even for the same signal, the eye diagrams may be somewhat different for different time offset (or initial point) values. Figure 7.27a illustrates the eye diagram of this polar signaling waveform for a display time offset of $T_b/2$, whereas Fig. 7.27b shows the

* This is true for a two-level decision [e.g., when $p(t)$ and $-p(t)$ are used for **1** and **0**, respectively]. For a three-level decision (e.g., bipolar signaling), there will be two thresholds.

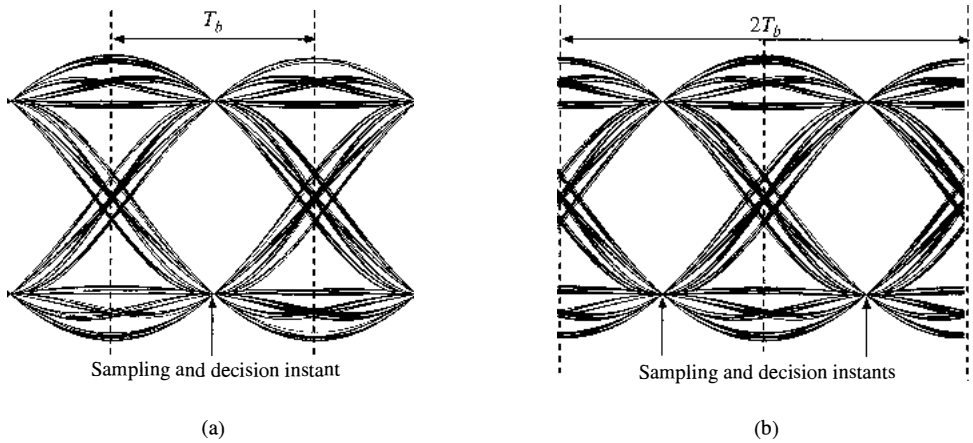


Figure 7.27 Eye diagrams of a polar signaling system using a raised cosine pulse with roll-off factor 0.5: (a) over 2 symbol periods $2T_b$ with a time shift $T_b/2$; (b) without time shift.

normal eye diagram when the display time offset value is zero. It is clear from comparison that these two diagrams have a simple horizontal circular shift relationship. By observing the maximum eye opening, we can see that this baseband signal has zero ISI, confirming the basic feature of the raised-cosine pulse. On the other hand, because Nyquist's first criterion places no requirement on the zero crossing of the pulse, the eye diagram indicates that timing jitter would be likely.

7.7 PAM: M-ARY BASEBAND SIGNALING FOR HIGHER DATA RATE

Regardless of which line code is used, binary baseband modulations have one thing in common: they all transmit one bit of information over the interval of T_b second, or at the bit rate of $1/T_b$ bit per second. If the transmitter would like to send bits at a much higher rate, T_b may be shortened. For example, to increase the bit rate by M, T_b must be reduced by the same factor of M; however, there is a heavy price to be paid in bandwidth. As we demonstrated in Fig. 7.9, the bandwidth of baseband modulation is proportional to the pulse rate $1/T_b$. Shortening T_b by a factor of M will certainly increase the required channel bandwidth by M. Fortunately, reducing T_b is not the only way to increase data rate. A very effective practical solution is to allow each pulse to carry multiple bits. We explain this concept here.

For each symbol transmission within the time interval of T_b to carry more bits, there must be more than two symbols to choose from. By increasing the number of symbols to M, we ensure that the information transmitted by each symbol will also increase with M. For example, when $M = 4$ (4-ary, or quaternary), we have four basic symbols, or pulses, available for communication (Fig. 7.28a). A sequence of two binary digits can be transmitted by just one 4-ary symbol. This is because a sequence of two bits can form only four possible sequences (viz., **11, 10, 01,** and **00**). Because we have four distinct symbols available, we can assign one of the four symbols to each of these combinations (Fig. 7.28a). Each symbol now occupies a time duration of T_s. A signaling example for a short sequence is given in Fig. 7.28b and the 4-ary eye-diagram is shown in Fig. 7.28c.

This signaling allows us to transmit each pair of bits by one 4-ary pulse (Fig. 7.28b). Hence, to transmit n bits, we need only $(n/2)$ 4-ary pulses. This means one 4-ary symbol can transmit the information of two binary digits. Also, because three bits can form $2 \times 2 \times 2 = 8$ combinations, a group of three bits can be transmitted by one 8-ary symbol. Similarly, a group

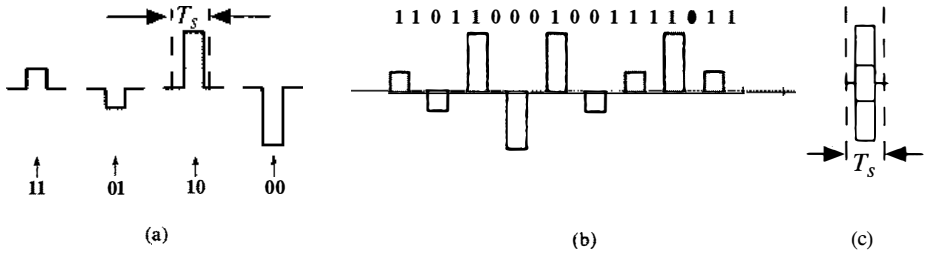


Figure 7.28 4-Ary PAM signaling: (a) four RZ symbols; (b) baseband transmission; (c) the 4-ary RZ eye diagram.

of four bits can be transmitted by one 16-ary symbol. In general, the information I_M transmitted by an M-ary symbol is

$$I_M = \log_2 M \text{ bits} \tag{7.55}$$

This means we can increase the rate of information transmission by increasing M.

This special M-ary signaling is known as the **pulse amplitude modulation (PAM)** because the data information is conveyed by the varying pulse amplitude. We should note here that pulse amplitude modulation is only one of many possible choices of M-ary signaling. There are an infinite number of such choices. Still, only a limited few are truly effective in combating noise and efficient in saving bandwidth and power. A more detailed discussion of other M-ary signaling schemes will be presented a little later, in Sec. 7.9.

As in most system designs, there are always prices to pay for every possible gain. The price paid by PAM to increase data rate is power. As M increases, the transmitted power also increases as M. This is because to have the same noise immunity, the minimum separation between pulse amplitudes should be comparable to that of binary pulses. Therefore, pulse amplitudes increase with M (Fig. 7.28). It can be shown that the transmitted power increases as M^2 (Prob. 7.7-5). Thus, to increase the rate of communication by a factor of $\log_2 M$, the power required increases as M^2. Because the transmission bandwidth depends only on the pulse rate and not on pulse amplitudes, the bandwidth is independent of M. We will use the following example of PSD analysis to illustrate this point.

Example 7.4 Determine the PSD of the quaternary (4-ary) baseband signaling in Fig. 7.28 when the message bits **1** and **0** are equally likely.

The 4-ary line code has four distinct symbols corresponding to the four different combinations of two message bits. One such mapping is

$$a_k = \begin{cases} -3 & \text{message bits } 00 \\ -1 & \text{message bits } 01 \\ +1 & \text{message bits } 10 \\ +3 & \text{message bits } 11 \end{cases} \tag{7.56}$$

Therefore, all four values of a_k are equally likely, each with a chance of 1 in 4. Recall that

$$R_0 = \lim_{N\to\infty} \frac{1}{N} \sum_k a_k^2$$

Within the summation, 1/4 of the a_k will be ± 1, and ± 3. Thus,

$$R_0 = \lim_{N\to\infty} \frac{1}{N}\left[\frac{N}{4}(-3)^2 + \frac{N}{4}(-1)^2 + +\frac{N}{4}(1)^2 + +\frac{N}{4}(3)^2\right] = 5$$

On the other hand, for $n > 0$, we need to determine

$$R_n = \lim_{N\to\infty} \frac{1}{N} \sum_k a_k a_{k+n}$$

To find this average value, we build a table with all the possible values of the product $a_k a_{k+n}$:

Possible Values of $a_k a_{k+n}$

a_{k+n} \ a_k	-3	-1	$+1$	$+3$
-3	9	3	-3	-9
-1	3	1	-1	-3
$+1$	-3	-1	1	3
$+3$	-9	-3	3	9

From the foregoing table listing all the possible products of $a_k a_{k+n}$, we see that each product in the summation $a_k a_{k+n}$ can take on any of the following six values ± 1, ± 3, ± 9. First, (± 1, ± 9) are equally likely (1 in 8). On the other hand, ± 3 are equally likely (1 in 4). Thus, we can show that

$$R_n = \lim_{N\to\infty} \frac{1}{N}\left[\frac{N}{8}(-9) + \frac{N}{8}(+9) + \frac{N}{8}(-1) + \frac{N}{8}(+1) + \frac{N}{4}(-3) + \frac{N}{4}(+3)\right] = 0$$

As a result,

$$S_x(f) = \frac{5}{T_s} \quad \Longrightarrow \quad S_y(f) = \frac{5}{T_s}|P(f)|^2$$

Thus, the M-ary line code generates the same PSD shape as binary polar signaling. The only difference is that it utilizes 5 times the original signal power.

Although most terrestrial digital telephone network uses binary encoding, the subscriber loop portion of the integrated services digital network (ISDN) uses the quaternary code, 2B1Q, similar to Fig. 7.28a. It uses NRZ pulses to transmits 160 kbit/s of data at a **baud** rate (pulse rate) of 80 kbit/s. Of the various line codes examined by the ANSI standards committee, 2B1Q provided the greatest baud rate reduction in the noisy and cross-talk-prone local cable plant environment.

Pulse Shaping and Eye Diagrams in PAM: In this case, we can use the Nyquist criterion pulses because these pulses have zero ISI at the sample points, and, therefore, their amplitudes can be correctly detected by sampling at the pulse centers. We can also use the controlled ISI (partial-response signaling) for M-ary signaling.[8]

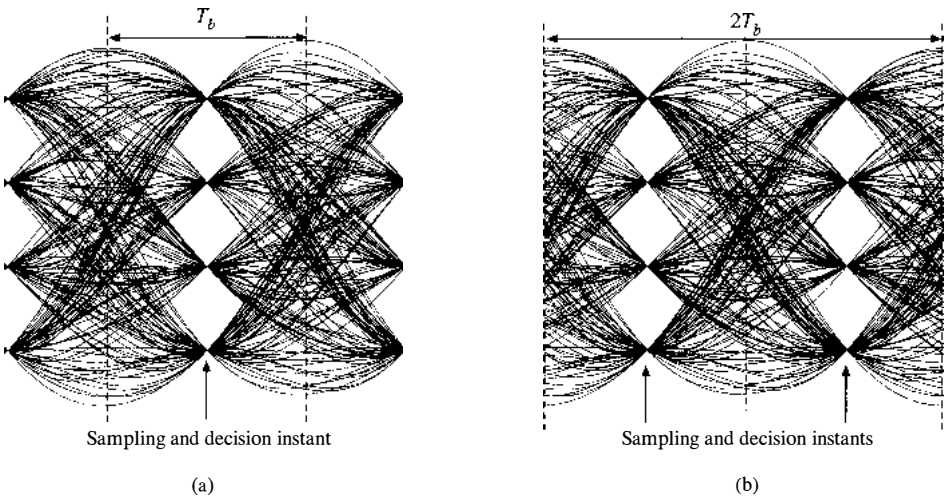


Figure 7.29 Eye diagrams of a 4-ary PAM signaling system using a raised-cosine pulse with roll-off factor 0.5: (a) over two symbol periods $2T_b$ with time offset $T_b/2$; (b) without time offset.

Eye diagrams can also be generated for M-ary PAM by using the same method used for binary modulations. Because of multilevel signaling, the eye diagram should have M levels at the optimum sampling instants even when ISI is zero. Here we generate the practical eye diagram example for a four-level PAM signal that uses the same cosine roll-off pulse with roll-off factor $r = 0.5$ that was used in the eye diagram of Fig. 7.27. The corresponding eye diagrams with time offsets of $T_b/2$ and 0 are given in Fig. 7.29a and b, respectively. Once again, no ISI is observed at the sampling instants. The eye diagrams clearly show four equally separated signal values without ISI at the optimum sampling points.

7.8 DIGITAL CARRIER SYSTEMS

Thus far, we have discussed baseband digital systems, where signals are transmitted directly without any shift in frequency. Because baseband signals have sizable power at low frequencies, they are suitable for transmission over a pair of wires and coaxial cables. Much of the modern communication is conducted this way. However, baseband signals cannot be transmitted over a radio link or satellites because this would necessitate impractically large antennas to efficiently radiate the low-frequency spectrum of the signal. Hence, for these applications, the signal spectrum must be shifted to a high-frequency range. A spectrum shift to higher frequencies is also required to transmit several messages simultaneously by sharing the large bandwidth of the transmission medium. As seen in Chapter 4, the spectrum of a signal can be shifted to a higher frequency by applying the baseband digital signal to modulate a high-frequency sinusoid (carrier).

In transmitting and receiving digital carrier signals, we need a modulator and demodulator to transmit and receive data. The two devices, **mo**dulator and **de**modulator are usually packaged in one unit called a **modem** for two-way (duplex) communications.

7.8.1 Basic Binary Carrier Modulations

There are two basic forms of carrier modulation: amplitude modulation and angle modulation. In amplitude modulation, the carrier amplitude is varied in proportion to the modulating signal (i.e., the baseband signal). This is shown in Fig. 7.30. An unmodulated carrier cos $\omega_c t$ is shown

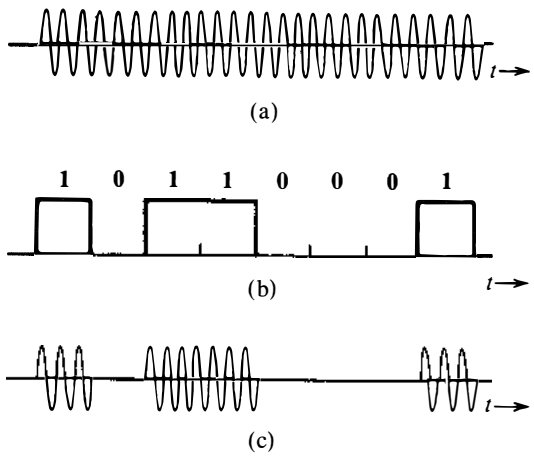


Figure 7.30
(a) The carrier $\cos \omega_c t$. (b) The modulating signal $m(t)$. (c) ASK: the modulated signal $m(t) \cos \omega_c t$.

in Fig. 7.30a. The on-off baseband signal $m(t)$ (the modulating signal) is shown in Fig. 7.30b. It can be written according to Eq. (7.1) as

$$m(t) = \sum a_k p(t - kT_b), \qquad \text{where} \qquad p(t) = \Pi\left(\frac{t - T_b/2}{T_b}\right)$$

The line code $a_k = 0, 1$ is on-off. When the carrier amplitude is varied in proportion to $m(t)$, we can write the carrier modulated signal as

$$\varphi_{\text{ASK}}(t) = m(t) \cos \omega_c t \tag{7.57}$$

shown in Fig. 7.30c. Note that the modulated signal is still an on-off signal. This modulation scheme of transmitting binary data is known as **on-off keying (OOK)** or **amplitude shift keying (ASK)**.

Of course, the baseband signal $m(t)$ may utilize a pulse $p(t)$ different from the rectangular one shown in the example of Fig. 7.30. This will generate an ASK signal that does not have a constant amplitude during the transmission of **1** ($a_k = 1$).

If the baseband signal $m(t)$ were polar (Fig. 7.31a), the corresponding modulated signal $m(t) \cos \omega_c t$ would appear as shown in Fig. 7.31b. In this case, if $p(t)$ is the basic pulse, we are transmitting **1** by a pulse $p(t) \cos \omega_c t$ and **0** by $-p(t) \cos \omega_c t = p(t) \cos (\omega_c t + \pi)$. Hence, the two pulses are π radians apart in phase. The information resides in the phase or the sign of the pulse. For this reason this scheme is known as **phase shift keying (PSK)**. Note that the transmission is still polar. In fact, just like ASK, the PSK modulated carrier signal has the same form

$$\varphi_{\text{PSK}}(t) = m(t) \cos \omega_c t \qquad m(t) = \sum a_k p(t - kT_b) \tag{7.58}$$

with the difference that the line code is polar $a_k = \pm 1$.

When data are transmitted by varying the frequency, we have the case of **frequency shift keying (FSK)**, as shown in Fig. 7.31c. A **0** is transmitted by a pulse of frequency ω_{c_0}, and **1** is transmitted by a pulse of frequency ω_{c_1}. The information about the transmitted data resides in the carrier frequency. The FSK signal may be viewed as a sum of two interleaved ASK signals, one with a modulating frequency ω_{c_0}, and the other with a modulating frequency ω_{c_1}. We can

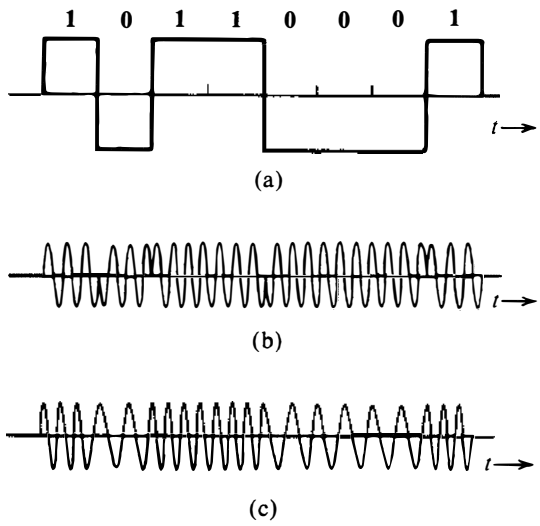


Figure 7.31 (a) The modulating signal $m(t)$. (b) PSK: the modulated signal $m(t)\cos \omega_c t$. (c) FSK: the modulated signal.

use the binary ASK expression of Eq. (7.57) to write the FSK signal as,

$$\varphi_{\text{FSK}}(t) = \sum a_k p(t - kT_b)\cos \omega_{c_1} t + \sum (1 - a_k)p(t - kT_b)\cos \omega_{c_0} t$$

where $a_k = 0, 1$ is on-off. Thus the FSK signal is a superposition of two AM signals with different carrier frequencies and different but complementary amplitudes.

In practice, ASK as an on-off scheme is commonly used today in optical fiber communications in the form of laser-intensity modulation. PSK is commonly applied in digital satellite communications and was also used in earlier telephone modems (2400 and 4800 bit/s). As for FSK, AT&T in 1962 developed one of the earliest telephone-line modems called 103A; it uses FSK to transmit 300 bit/s at two frequencies, 1070 and 1270 Hz, and receives FSK at 2025 and 2225 Hz.

7.8.2 PSD of Digital Carrier Modulation

We have just shown that the binary carrier modulations of ASK, PSK, and FSK can all be written into some forms of $m(t)\cos \omega_c t$. To determine the PSD of the ASK, PSK, and FSK signals, it would be helpful for us to first find the relationship between the PSD of $m(t)$ and the PSD of the modulated signal

$$\varphi(t) = m(t)\cos \omega_c t$$

Recall from Eq. (3.80) that the PSD of $\varphi(t)$ is

$$S_\varphi(f) = \lim_{T \to \infty} \frac{|\Psi_T(f)|^2}{T}$$

where $\Psi_T(f)$ is the Fourier transform of the truncated signal

$$\begin{aligned}\varphi_T(t) &= \varphi(t)[u(t+T/2) - u(t-T/2)] \\ &= m(t)[u(t+T/2) - u(t-T/2)] \cos \omega_c t \\ &= m_T(t) \cos \omega_c t \end{aligned} \tag{7.59}$$

Here $m_T(t)$ is the truncated baseband signal with Fourier transform $M_T(f)$. Applying the frequency shift property [see Eq. (3.36)], we have

$$\Psi_T(f) = \frac{1}{2}[M_T(f - f_c) + M_T(f + f_c)]$$

As a result, the PSD of the modulated carrier signal $\varphi(t)$ is

$$S_\varphi(f) = \lim_{T\to\infty} \frac{1}{4} \frac{|M_T(f + f_c) + M_T(f - f_c)|^2}{T}$$

Because $M(f)$ is a baseband signal, $M_T(f + f_c)$ and $M_T(f - f_c)$ have zero overlap as $T \to \infty$ as long as f_c is larger than the bandwidth of $M(f)$. Therefore, we conclude that

$$\begin{aligned} S_\varphi(f) &= \lim_{T\to\infty} \frac{1}{4} \left[\frac{|M_T(f + f_c)|^2}{T} + \frac{|M_T(f - f_c)|^2}{T} \right] \\ &= \frac{1}{4} S_M(f + f_c) + \frac{1}{4} S_M(f - f_c) \end{aligned} \tag{7.60}$$

In other words, for an appropriately chosen carrier frequency, modulation causes a shift in the baseband signal PSD.

Now, the ASK signal in Fig. 7.30c, fits this model, with $m(t)$ being an on-off signal (using a full-width or NRZ pulse). Hence, the PSD of the ASK signal is the same as that of an on-off signal (Fig. 7.4b) shifted to $\pm f_c$ as shown in Fig. 7.32a. Remember that by using a full-width rectangular pulse $p(t)$,

$$P\left(\frac{n}{T_b}\right) = 0 \qquad n = \pm 1, \pm 2, \ldots$$

In this case, the baseband on-off PSD has no discrete components except at dc in Fig. 7.30b. Therefore, the ASK spectrum has discrete component only at ω_c.

The PSK signal also fits this modulation description where $m(t)$ is a polar signal using a full-width NRZ pulse. Therefore, the PSD of a PSK signal is the same as that of the polar baseband signal shifted to $\pm\omega_c$, as shown in Fig. 7.32b. Note that this PSD has the same shape (with a different scaling factor) as the PSD of the ASK minus its discrete components.

Finally, we have shown that the FSK signal may be viewed as a sum of two interleaved ASK signals using the full-width pulse. Hence, the spectrum of FSK is the sum of two ASK spectra at frequencies ω_{c_0} and ω_{c_1}, as shown in Fig. 7.32c. It can be shown that by properly choosing ω_{c_0} and ω_{c_1} and by maintaining phase continuity during frequency switching, discrete components can be eliminated at ω_{c_0} and ω_{c_1}. Thus, no discrete components appear in this spectrum. It is important to note that the bandwidth of FSK is higher than that of ASK or PSK.

As observed earlier, polar signaling is the most power-efficient scheme. The PSK, being polar, requires 3 dB less power than ASK (or FSK) for the same noise immunity, that is, for the same error probability in pulse detection.

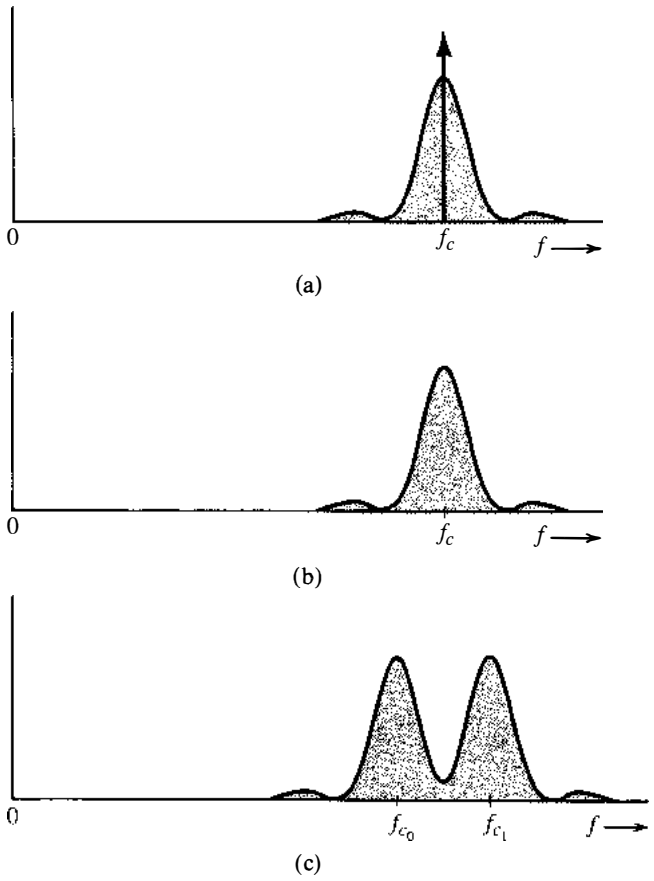


Figure 7.32 PSD of (a) ASK, (b) PSK, and (c) FSK.

Of course, we can also modulate bipolar, or any other scheme discussed earlier. Also, note that the use of the NRZ rectangular pulse in Fig. 7.30 or 7.31 is for the sake of illustration only. In practice, baseband pulses may be spectrally shaped to eliminate ISI.

7.8.3 Connections between Analog and Digital Carrier Modulations

There is a natural and clear connection between ASK and AM because the message information is directly reflected in the varying amplitude of the modulated signals. Because of its nonnegative amplitude, ASK is essentially an AM signal with modulation index $\mu = 1$. There is a similar connection between FSK and FM. FSK is simply an FM signal with only limited number of instantaneous frequencies.

The connection between PSK and analog modulation is a bit more subtle. For PSK, the modulated signal can be written as

$$\varphi_{\text{PSK}}(t) = A\cos(\omega_c t + \theta_k) \qquad kT_b \le t < kT_b + T_b$$

It can therefore be connected with PM. However, a closer look at the PSK signal reveals that because of the constant phase θ_k, its instantaneous frequency, in fact, does not change. In fact,

we can rewrite the PSK signal

$$\begin{aligned}\varphi_{\text{PSK}}(t) &= A\cos\theta_k\cos\omega_c t - A\sin\theta_k\sin\omega_c t \\ &= a_k\cos\omega_c t + b_k\sin\omega_c t \qquad kT_b \le t < kT_b + T_b \end{aligned} \tag{7.61}$$

by letting $a_k = A\cos\theta_k$ and $b_k = -A\sin\theta_k$. From Eq. (7.61), we recognize its strong resemblance to the QAM signal representation in Sec. 4.4. Therefore, the digital PSK modulation is closely connected with the analog QAM signal. In particular, $\theta = 0,\ \pi$ for binary PSK. Thus, binary PSK can be written as

$$\pm A\cos\omega_c t$$

This is effectively a digital manifestation of the DSB-SC amplitude modulation. In fact, as will be discussed later, by letting a_k take on multilevel values while setting $b_k = 0$ we can generate another digital carrier modulation known as the pulse amplitude modulation (or PAM), which can carry multiple bits during each modulation time-interval T_b.

As we have studied in Chapter 4, DSB-SC amplitude modulation is more power efficient than AM. Binary PSK is therefore more power efficient than ASK. In terms of bandwidth utilization, we can see from their connection to analog modulations that ASK and PSK have identical bandwidth occupation while FSK requires larger bandwidth. These observations intuitively corroborate our PSD results of Fig. 7.32.

7.8.4 Demodulation

Demodulation of digital-modulated signals is similar to that of analog-modulated signals. Because of the connections between ASK and AM, between FSK and FM, and between PSK and QAM (or DSB-SC AM), different demodulation techniques used for the analog modulations can be directly applied to their digital counterparts.

ASK Detection

Just like AM, ASK (Fig. 7.30c), can be demodulated both coherently (for synchronous detection) or noncoherently (for envelope detection). The coherent detector requires more elaborate equipment and has superior performance, especially when the signal power (hence SNR) is low. For higher SNR, the envelope detector performs almost as well as the coherent detector. Hence, coherent detection is not often used for ASK because it will defeat its very purpose (the simplicity of detection). If we can avail ourselves of a synchronous detector, we might as well use PSK, which has better power efficiency than ASK.

FSK Detection

Once again, the binary FSK can be viewed as two interleaved ASK signals with carrier frequencies ω_{c_0} and ω_{c_1}, respectively (Fig. 7.32c). Therefore, FSK can be detected coherently or noncoherently. In noncoherent detection, the incoming signal is applied to a pair of filters tuned to ω_{c_0} and ω_{c_1}, respectively. Each filter is followed by an envelope detector (see Fig. 7.33a). The outputs of the two envelope detectors are sampled and compared. If a **0** is transmitted by a pulse of frequency ω_{c_0}, then this pulse will appear at the output of the filter tuned to ω_{c_0}. Practically no signal appears at the output of the filter tuned to ω_{c_1}. Hence, the sample of the envelope detector output following the ω_{c_0} filter will be greater than the sample of the envelope

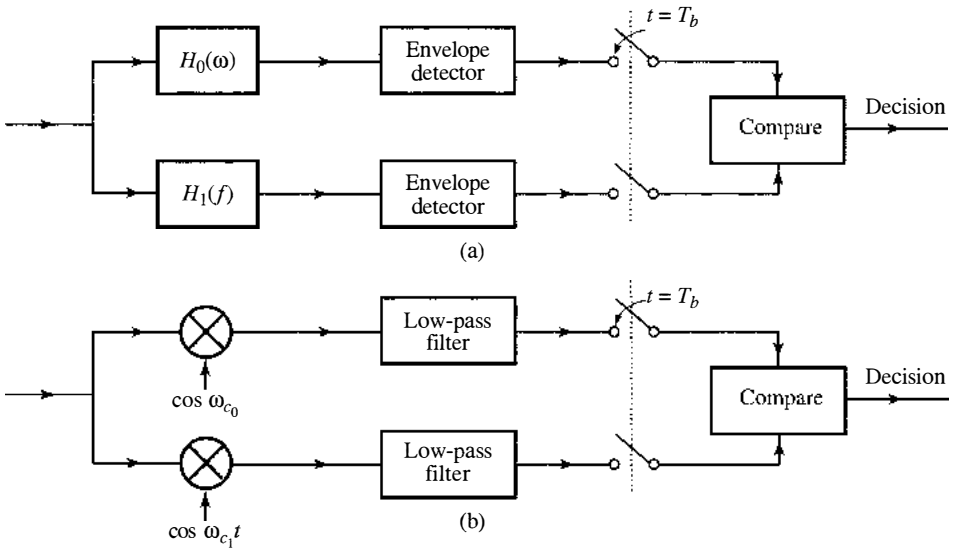


Figure 7.33
(a) Noncoherent detection of FSK. (b) Coherent detection of FSK.

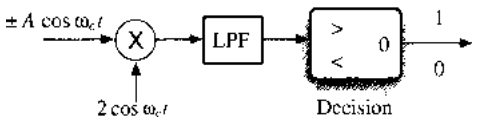


Figure 7.34
Coherent binary PSK detector (similar to a DSB-SC demodulator).

detector output following the ω_{c_1} filter, and the receiver decides that a **0** was transmitted. In the case of a **1**, the opposite happens.

Of course, FSK can also be detected coherently by generating two references of frequencies ω_{c_0} and ω_{c_1}, for the two demodulators, to demodulate the signal received and then comparing the outputs of the two demodulators as shown in Fig. 7.33b. Thus, coherent FSK detector must generate two carriers in synchronization with the modulation carriers. Once again, this complex demodulator defeats the purpose of FSK, which is designed primarily for simpler, noncoherent detection. In practice, coherent FSK detection is not in use.

PSK Detection

In binary PSK, a **1** is transmitted by a pulse $A \cos \omega_c t$ and a **0** is transmitted by a pulse $-A \cos \omega_c t$ (Fig. 7.31b). The information in PSK signals therefore resides in the carrier phase. Just as in DSB-SC, these signals cannot be demodulated via envelope detection because the envelope stays constant for both **1** and **0** (Fig. 7.31b). The coherent detector of the binary PSK modulation is shown in Fig. 7.34. The coherent detection is similar to that used for analog signals. Methods of carrier acquisition have been discussed in Sec. 4.8.

Differential PSK

Although envelope detection cannot be used for PSK detection, it is still possible to exploit the finite number of modulation phase values for noncoherent detection. Indeed, PSK signals may be demodulated noncoherently by means of an ingenious method known as **differential PSK**,

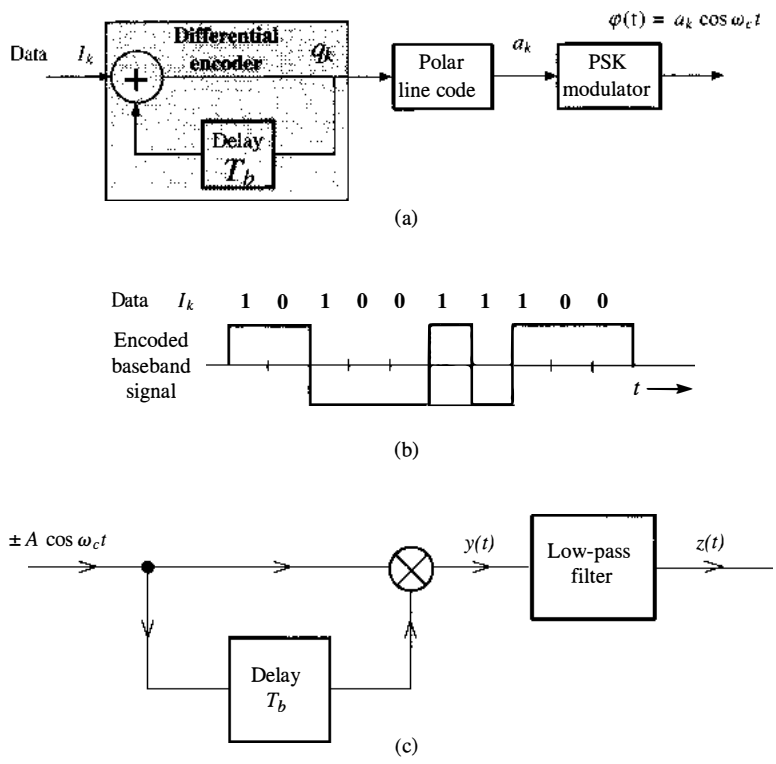


Figure 7.35
(a) Differential encoding; (b) encoded signal; (c) differential PSK receiver.

or DPSK. The principle of differential detection is for the receiver to detect the relative phase change between successive modulated phases θ_k and θ_{k-1}. Since the phase value in PSK is finite (equaling to 0 and π in binary PSK), the transmitter can encode the information data into the phase difference $\theta_k - \theta_{k-1}$. For example, a phase difference of zero represents **0** whereas a phase difference of π signifies **1**.

This technique is known as **differential encoding** (before modulation). In one differential code, a **0** is encoded by the same pulse used to encode the previous data bit (no transition), and a **1** is encoded by the negative of the pulse used to encode the previous data bit (transition). Differential encoding is simple to implement, as shown in Fig. 7.35a. Notice that the addition is modulo-2. The encoded signal is shown in Fig. 7.35b. Thus a transition in the line code pulse sequence indicates **1** and no transition indicates **0**. The modulated signal consists of pulses

$$A \cos (\omega_c t + \theta_k) = \pm A \cos \omega_c t$$

If the data bit is **0**, the present pulse and the previous pulse have the same polarity or phase; both pulses are either $A \cos \omega_c t$ or $-A \cos \omega_c t$. If the data bit is **1**, the present pulse and the previous pulse are of opposite polarities or phases; if the present pulse is $A \cos \omega_c t$, the previous pulse is $-A \cos \omega_c t$, and vice versa.

In demodulation of DPSK (Fig. 7.35c), we avoid generation of a local carrier by observing that the received modulated signal itself is a carrier ($\pm A \cos \omega_c t$) with a possible sign ambiguity. For demodulation, in place of the carrier, we use the received signal delayed by T_b (one bit interval). If the received pulse is identical to the previous pulse, the product $y(t) = A^2 \cos^2 \omega_c t = (A^2/2)(1 + \cos 2\omega_c t)$, and the low-pass filter output $z(t) = A^2/2$. We immediately detect the present bit as **0**. If the received pulse and the previous pulse are of

TABLE 7.3
Differential Encoding and Detection of Binary DPSK

Time k	0	1	2	3	4	5	6	7	8	9	10
I_k		1	0	1	0	0	1	1	1	0	0
q_k	0	1	1	0	0	0	1	0	1	1	1
Line code a_k	−1	1	1	−1	−1	−1	1	−1	1	1	1
θ_k	π	0	0	π	π	π	0	π	0	0	0
$\theta_k - \theta_{k-1}$		π	0	π	0	0	π	π	π	0	0
Detected bits		1	0	1	0	0	1	1	1	0	0

opposite polarity, $y(t) = -A^2 \cos^2 \omega_c t$ and $z(t) = -A^2/2$, and the present bit is detected as **0**. Table 7.3 illustrates a specific example of the encoding and decoding.

Thus, in terms of demodulation complexity, ASK, FSK, and DPSK can all be noncoherently detected without a synchronous carrier at the receiver. On the other hand, PSK must be coherently detected. Noncoherent detection, however, comes with a price in terms of noise immunity. From the point of view of noise immunity, coherent PSK is superior to all other schemes. PSK also requires smaller bandwidth than FSK (see Fig. 7.32). Quantitative discussion of this topic can be found in Chapter 10.

7.9 M-ARY DIGITAL CARRIER MODULATION

The binary digital carrier modulations of ASK, FSK, and PSK all transmit one bit of information over the interval of T_b second, or at the bit rate of $1/T_b$ bit/s. Similar to digital baseband transmission, higher bit rate transmission can be achieved either by reducing T_b or by applying M-ary signaling; the first option requires more bandwidth; the second requires more power. In most communication systems, bandwidth is strictly limited. Thus, to conserve bandwidth, an effective way to increase transmission data rate is to generalize binary modulation by employing M-ary signaling. Specifically, we can apply M-level ASK, M-frequency FSK, and M-phase PSK modulations.

M-ary ASK and Noncoherent Detection

M-ary ASK is a very simple generalization of binary ASK. Instead of sending only

$$\varphi(t) = 0 \text{ for } \mathbf{0} \qquad \text{and} \qquad \varphi(t) = A \cos \omega_c t \text{ for } \mathbf{1}$$

M-ary ASK can send $\log_2 M$ bits each time by transmitting, for example,

$$\varphi(t) = 0, A \cos \omega_c t, 2A \cos \omega_c t, \ldots, (M-1)A \cos \omega_c t$$

This is still an AM signal that uses M different amplitudes and a modulation index of $\mu = 1$. Its bandwidth remains the same as that of the binary ASK, while its power is increased proportionally with M^2. Its demodulation would again be achieved via envelope detection or coherent detection.

M-ary FSK and Orthogonal Signaling

M-FSK is similarly generated by selecting one sinusoid from the set $\{A \cos 2\pi f_i t, i = 1, \ldots, M\}$ to transmit a particular pattern of $\log_2 M$ bits. Generally for FSK, we can

design a frequency increment δf and let

$$f_m = f_1 + (m-1)\delta f \qquad m = 1, 2, \ldots, M$$

For this FSK with equal frequency separation, the frequency deviation (in analyzing the FM signal) is

$$\Delta f = \frac{f_M - f_1}{2} = \frac{1}{2}(M-1)\delta f$$

It is therefore clear that the selection of the frequency set $\{f_i\}$ determines the performance and the bandwidth of the FSK modulation. If δf is chosen too large, then the M-ary FSK will use too much bandwidth. On the other hand, if δf is chosen too small, then over the time interval of T_b second, different FSK symbols will show virtually no difference and the receiver will be unable to distinguish the different symbols reliably. Thus large δf leads to bandwidth waste, whereas small δf is prone to detection error due to transmission noise and interference.

The task of M-ary FSK design is to determine a small enough δf that each FSK symbol $A\cos\omega_i t$ is highly distinct from all other FSK symbols. One solution to this problem of FSK signal design actually can be found in the discussion of orthogonal signal space in Sec. 2.6.2. If we can design FSK symbols to be orthogonal in T_b by selecting a small δf (or Δf), then the FSK signals will be truly distinct over T_b, and the bandwidth consumption will be small.

To find the minimum δf that leads to an orthogonal set of FSK signals, the orthogonality condition according to Sec. 2.6.2 requires that

$$\int_0^{T_b} A\cos(2\pi f_m t)\, A\cos(2\pi f_n t)\, dt = 0 \qquad m \neq n \tag{7.62}$$

We can use this requirement to find the minimum δf. First of all,

$$\begin{aligned}\int_0^{T_b} A\cos(2\pi f_m t)\, A\cos(2\pi f_n t)\, dt &= \frac{A^2}{2}\int_0^{T_b} [\cos 2\pi(f_m + f_n)t + \cos 2\pi(f_m - f_n)t]\, dt \\ &= \frac{A^2}{2}T_b \frac{\sin 2\pi(f_m + f_n)T_b}{2\pi(f_m + f_n)T_b} + \frac{A^2}{2}T_b \frac{\sin 2\pi(f_m - f_n)T_b}{2\pi(f_m - f_n)T_b}\end{aligned} \tag{7.63}$$

Since in practical modulations, $(f_m + f_n)T_b$ is very large (often no smaller than 10^3), the first term in Eq. (7.63) is effectively zero and negligible. Thus, the orthogonality condition reduces to the requirement that for any integer $m \neq n$,

$$\frac{A^2}{2}\frac{\sin 2\pi(f_m - f_n)T_b}{2\pi(f_m - f_n)} = 0$$

Because $f_m = f_1 + (m-1)\delta f$, for mutual orthogonality we have

$$\sin[2\pi(m-n)\delta f T_b] = 0 \qquad m \neq n$$

From this requirement, it is therefore clear that the smallest δf to satisfy the mutual orthogonality condition is

$$\delta f = \frac{1}{2T_b}\ \text{Hz}$$

This choice of minimum frequency separation is known as the *minimum shift* FSK. Since it forms an orthogonal set of symbols, it is often known as orthogonal signaling.

We can in fact describe the *minimum shift* FSK geometrically by applying the concept of orthonormal basis functions in Sec. 2.6. Let

$$\psi_i(t) = \sqrt{\frac{2}{T_b}} \cos 2\pi \left(f_1 + \frac{i-1}{2T_b}\right) t \qquad i = 1, 2, \ldots, M$$

It can be simply verified that

$$\int_0^{T_b} \psi_m(t)\psi_n(t)\, dt = \begin{cases} 1 & m = n \\ 0 & m \neq n \end{cases}$$

Thus, each of the FSK symbol can be written as

$$A \cos 2\pi f_m t = A\sqrt{\frac{T_b}{2}}\psi_m(t) \qquad m = 1, 2, \ldots, M$$

The geometrical relationship of the two FSK symbols for $M = 2$ is easily captured by Fig. 7.36.

The demodulation of M-ary FSK signals follows the same approach as the binary FSK demodulation. Generalizing the binary FSK demodulators of Fig. 7.33 we can apply a bank of M coherent or noncoherent detectors to the M-ary FSK signal before making a decision based on the strongest detector branch.

Earlier in the PSD analysis of baseband modulations, we showed that the baseband digital signal bandwidth at the symbol interval of T_b can be approximated by $1/T_b$. Therefore, for the minimum shift FSK, $\Delta f = (M - 1)/(4T_b)$, and its bandwidth according to Carson's rule is approximately

$$2(\Delta f + B) = \frac{M-3}{2T_b}$$

In fact, it can be in general shown that the bandwidth of an orthogonal M-ary scheme is M times that of the binary scheme [see Sec. 10.7, Eq. (10.123)]. Therefore, in an M-ary orthogonal scheme, the rate of communication increases by a factor of $\log_2 M$ at the cost of M-fold transmission bandwidth increase. For a comparable noise immunity, the transmitted power is practically independent of M in the orthogonal scheme. Therefore, unlike M-ary ASK, M-ary FSK does not require more transmission power. However, its bandwidth requirement increases almost linearly with M (compared with binary FSK or M-ary ASK).

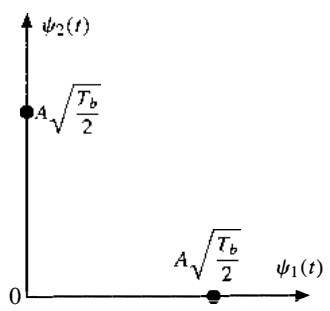


Figure 7.36 Binary FSK symbols in the two-dimensional orthogonal signal space.

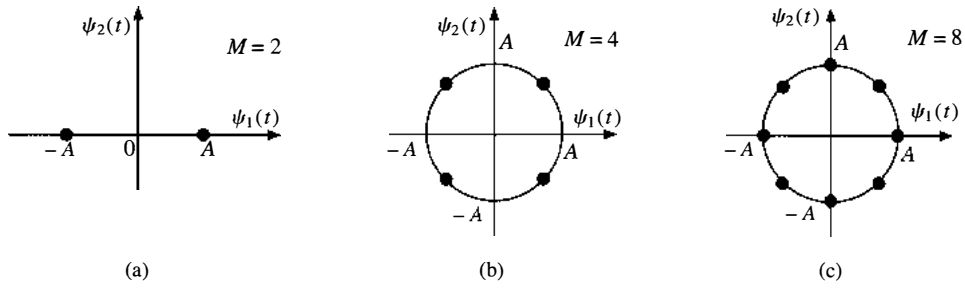


Figure 7.37 *M*-ary PSK symbols in the orthogonal signal space: (a) $M = 2$; (b) $M = 4$; (c) $M = 8$.

M-ary PSK, PAM, and QAM

By making a small modification to Eq. (7.61), PSK signals in general can be written into the format of

$$\varphi_{\text{PSK}}(t) = a_m\sqrt{\frac{2}{T_b}}\cos\omega_c t + b_m\sqrt{\frac{2}{T_b}}\sin\omega_c t \qquad 0 \le t < T_b \tag{7.64a}$$

in which $a_m = A\cos\theta_m$ and $b_m = -A\sin\theta_m$. In fact, based on the analysis in Sec. 2.6, $\sqrt{2/T_b}\cos\omega_c t$ and $\sqrt{2/T_b}\sin\omega_c t$ are orthogonal to each other. Furthermore, they are normalized over $[0, T_b]$. As a result, we can represent all PSK symbols in a two-dimensional signal space with basis functions

$$\psi_1(t) = \sqrt{\frac{2}{T_b}}\cos\omega_c t \qquad \psi_2(t) = \sqrt{\frac{2}{T_b}}\sin\omega_c t$$

such that

$$\varphi_{\text{PSK}}(t) = a_m\psi_1(t) + b_m\psi_2(t) \tag{7.64b}$$

We can geometrically illustrate the relationship of the PSK symbols in the signal space (Fig. 7.37). Equation (7.64) means that PSK modulations can be represented as QAM signal. In fact, because the signal is PSK, the signal points must meet a special requirement that

$$\begin{aligned} a_m^2 + b_m^2 &= A^2\cos^2\theta_m + (-A)^2\sin^2\theta_m \\ &= A^2 = \text{constant} \end{aligned} \tag{7.64c}$$

In other words, all the signal points must stay on a circle of radius A. In practice, all the signal points are chosen to be equally spaced in the interest of obtaining the best immunity against noise. Therefore, for M-ary PSK signaling, the angles are typically chosen uniformly as

$$\theta_m = \theta_0 + \frac{2\pi}{M}(m-1) \qquad m = 1, 2, \ldots, M$$

The special PSK signaling with $M = 4$ is an extremely popular and powerful digital modulation format.* It in fact is a summation of two binary PSK signals, one using the

* QPSK has several effective variations including the offset QPSK.

(in-phase) carrier of cos $\omega_c t$ while the other uses the (quadrature) carrier of sin $\omega_c t$ of the same frequency. For this reason, it is also known as **quadrature PSK (QPSK)**. We can transmit and receive both of these signals on the same channel, thus doubling the transmission rate.

To further generalize the PSK to achieve higher data rate, we can see that the PSK representation of Eq. (7.64) is a special case of the quadrature amplitude modulation (QAM) signal discussed in Chapter 4 (Fig. 4.19). The only difference lies in the requirement by PSK that the modulated signal have a constant magnitude (modulus) A. In fact, the much more flexible and general QAM signaling format can be conveniently used for digital modulation as well. The signal transmitted by an M-ary QAM system can be written as

$$\begin{aligned} p_i(t) &= a_i p(t) \cos \omega_c t + b_i p(t) \sin \omega_c t \\ &= r_i p(t) \cos (\omega_c t - \theta_i) \qquad i = 1, 2, \ldots, M \end{aligned}$$

where

$$r_i = \sqrt{a_i^2 + b_i^2} \qquad \text{and} \qquad \theta_i = \tan^{-1} \frac{b_i}{a_i} \tag{7.65}$$

and $p(t)$ is a properly shaped baseband pulse. The simplest choice of $p(t)$ would be a rectangular pulse

$$p(t) = \sqrt{\frac{2}{T_b}} [u(t) - u(t - T_b)]$$

Certainly, better pulses can also be applied conserve bandwidth.

Figure 7.38a shows the QAM modulator and demodulator. Each of the two signals $m_1(t)$ and $m_2(t)$ is a baseband $\sqrt{M}$-ary pulse sequence. The two signals are modulated by two carriers of the same frequency but in phase quadrature. The digital QAM signal $p_i(t)$ can be generated by means of QAM by letting $m_1(t) = a_i p(t)$ and $m_2(t) = b_i p(t)$. Both $m_1(t)$ and $m_2(t)$ are baseband PAM signals. The eye diagram of the QAM signal consists of the in-phase component $m_1(t)$ and the quadrature component $m_2(t)$. Both exhibit the M-ary baseband PAM eye diagram, as discussed earlier in Sec. 7.6.

The geometrical representation of M-ary QAM can be extended from the PSK signal space by simply removing the constant modulus constraint Eq. (7.64c). One very popular and practical choice of r_i and θ_i for $M = 16$ is shown graphically in Fig. 7.38b. The transmitted pulse $p_i(t)$ can take on 16 distinct forms, and is, therefore, a 16-ary pulse. Since $M = 16$, each pulse can transmit the information of $\log_2 16 = 4$ binary digits. This can be done as follows: there are 16 possible sequences of four binary digits and there are 16 combinations (a_i, b_i) in Fig. 7.38b. Thus, every possible four-bit sequence is transmitted by a particular (a_i, b_i) or (r_i, θ_i). Therefore, one signal pulse $r_i p(t) \cos (\omega_c t - \theta_i)$ transmits four bits. Compared with binary PSK (or BPSK), the 16-ary QAM bit rate is quadrupled without increasing the bandwidth. The transmission rate can be increased further by increasing the value of M.

Modulation as well as demodulation can be performed by using the system in Fig. 7.38a. The inputs are $m_1(t) = a_i p(t)$ and $m_2(t) = b_i p(t)$. The two outputs at the demodulator are $a_i p(t)$ and $b_i p(t)$. From knowledge of (a_i, b_i), we can determine the four transmitted bits. Further analysis of 16-ary QAM on a noisy channel is carried out in Sec. 10.6 [Eq. (10.104)]. The practical value of this 16-ary QAM signaling becomes fully evident when we consider its broad range of applications. In fact, 16-QAM is used in the V.32 telephone data/fax modems (9600 bit/s), in high-speed cable modems, and in modern satellite digital television broadcasting.

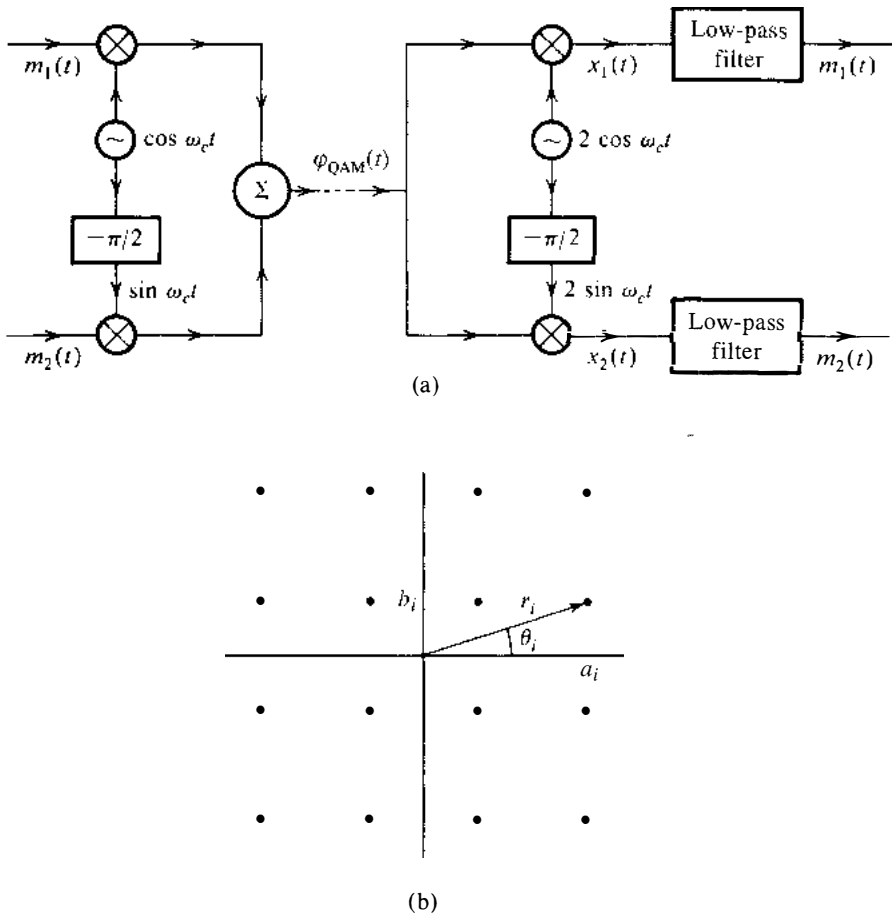


Figure 7.38 (a) QAM or quadrature multiplexing and (b) 16-point QAM ($M = 16$).

Note that if we disable the data stream that modulates sin $\omega_c t$ in QAM, then all the signaling points can be reduced to a single dimension. Upon setting $m_2(t) = 0$, QAM becomes

$$p_i(t) = a_i p(t) \cos \omega_c t, \qquad t \in [0, T_b]$$

This degenerates into the pulse amplitude modulation or PAM. Comparison of the signal expression of $p_i(t)$ with the analog DSB-SC signal makes it clear that PAM is the digital version of the DSB-SC signal. Just as analog QAM is formed by the superposition of two DSB-SC amplitude modulations in phase quadrature, digital QAM consists of two PAM signals, each having $\sqrt{M}$ signaling levels. Similarly, like the relationship between analog DSB-SC and QAM, PAM requires the same amount of bandwidth as QAM does. However, PAM is much less efficient because it would need M modulation signaling levels in one dimension, whereas QAM requires only $\sqrt{M}$ signaling levels in each of the two orthogonal QAM dimensions.

Trading Power and Bandwidth

In Chapter 10 we shall discuss several other types of M-ary signaling. The nature of the exchange between the transmission bandwidth and the transmitted power (or SNR) depends on the choice of M-ary scheme. For example, in orthogonal signaling, the transmitted power is practically independent of M but the transmission bandwidth increases with M. Contrast this

to the PAM case, where the transmitted power increases roughly with M^2 while the bandwidth remains constant. Thus, M-ary signaling allows us great flexibility in trading signal power (or SNR) for transmission bandwidth. The choice of the appropriate system will depend upon the particular circumstances. For instance, it will be appropriate to use QAM signaling if the bandwidth is at a premium (as in telephone lines) and to use orthogonal signaling when power is at a premium (as in space communication).

7.10 MATLAB EXERCISES

In this section, we provide MATLAB programs to generate the eye diagrams. The first step is to specify the basic pulse shapes in PAM. The next four short programs are used to generate NRZ, RZ, half-sinusoid, and raised-cosine pulses.

```
% (pnrz.m)
% generating a rectangular pulse of width T
% Usage function pout=pnrz(T);
function pout=prect(T);
pout=ones(1,T);
end
```

```
% (prz.m)
% generating a rectangular pulse of width T/2
% Usage function pout=prz(T);
function pout=prz(T);
pout=[zeros(1,T/4) ones(1,T/2) zeros(1,T/4)];
end
```

```
% (psine.m)
% generating a sinusoid pulse of width T
%
function pout=psine(T);
pout=sin(pi*[0:T-1]/T);
end
```

```
% (prcos.m)
%  Usage   y=prcos(rollfac,length, T)
function y=prcos(rollfac,length, T)
% rollfac = 0 to 1 is the rolloff factor
% length is the onesided pulse length in the number of T
% length = 2T+1;
% T is the oversampling rate
y=rcosfir(rollfac, length, T,1, 'normal');
end
```

The first program (binary_eye.m) uses the four different pulses to generate eye diagrams of binary polar signaling.

```
% (binary_eye.m)
% generate and plot eyediagrams
%
clear;clf;
data = sign(randn(1,400));  % Generate 400 random bits
Tau=64;                     % Define the symbol period
dataup=upsample(data, Tau); % Generate impulse train
yrz=conv(dataup,prz(Tau));  % Return to zero polar signal
yrz=yrz(1:end-Tau+1);
ynrz=conv(dataup,pnrz(Tau)); % Non-return to zero polar
ynrz=ynrz(1:end-Tau+1);
ysine=conv(dataup,psine(Tau));  % half sinusoid polar
ysine=ysine(1:end-Tau+1);
Td=4;                       % truncating raised cosine to 4 periods
yrcos=conv(dataup,prcos(0.5,Td,Tau)); % rolloff factor = 0.5
yrcos=yrcos(2*Td*Tau:end-2*Td*Tau+1); % generating RC pulse train
eye1=eyediagram(yrz,2*Tau,Tau,Tau/2);title('RZ eye-diagram');
eye2=eyediagram(ynrz,2*Tau,Tau,Tau/2);title('NRZ eye-diagram');
eye3=eyediagram(ysine,2*Tau,Tau,Tau/2);title('Half-sine eye-diagram');
eye4=eyediagram(yrcos,2*Tau,Tau); title('Raised-cosine eye-diagram');
```

The second program (Mary_eye.m) uses the four different pulses to generate eye diagrams of four-level PAM signaling.

```
% (Mary_eye.m)
% generate and plot eyediagrams
%
%
clear;clf;
data = sign(randn(1,400))+2* sign(randn(1,400)); % 400 PAM symbols
Tau=64;                     % Define the symbol period
dataup=upsample(data, Tau); % Generate impulse train
yrz=conv(dataup,prz(Tau));  % Return to zero polar signal
yrz=yrz(1:end-Tau+1);
ynrz=conv(dataup,pnrz(Tau)); % Non-return to zero polar
ynrz=ynrz(1:end-Tau+1);
ysine=conv(dataup,psine(Tau));  % half sinusoid polar
ysine=ysine(1:end-Tau+1);
Td=4;                       % truncating raised cosine to 4 periods
yrcos=conv(dataup,prcos(0.5,Td,Tau)); % rolloff factor = 0.5
yrcos=yrcos(2*Td*Tau:end-2*Td*Tau+1); % generating RC pulse train
eye1=eyediagram(yrz,2*Tau,Tau,Tau/2);title('RZ eye-diagram');
eye2=eyediagram(ynrz,2*Tau,Tau,Tau/2);title('NRZ eye-diagram');
eye3=eyediagram(ysine,2*Tau,Tau,Tau/2);title('Half-sine eye-diagram');
eye4=eyediagram(yrcos,2*Tau,Tau); title('Raised-cosine eye-diagram');
```

REFERENCES

1. A. Lender, "Duobinary Technique for High Speed Data Transmission," *IEEE Trans. Commun. Electron.*, vol. CE-82, pp. 214–218, May 1963.
2. A. Lender, "Correlative Level Coding for Binary-Data Transmission," *IEEE Spectrum*, vol. 3, no. 2, pp. 104–115, Feb. 1966.
3. P. Bylanski and D. G. W. Ingram, *Digital Transmission Systems*, Peter Peregrinus Ltd., Hertshire, England, 1976.
4. H. Nyquist, "Certain Topics in Telegraph Transmission Theory," *AIEE Trans.*, vol. 47, p. 817, April 1928.
5. E. D. Sunde, *Communication Systems Engineering Technology*, Wiley, New York, 1969.
6. R. W. Lucky and H. R. Rudin, "Generalized Automatic Equalization for Communication Channels," *IEEE Int. Commun. Conf.*, vol. 22, 1966.
7. W. F. Trench, "An Algorithm for the Inversion of Finite Toeplitz Matrices," *J. SIAM*, vol. 12, pp. 515–522, Sept. 1964.
8. A. Lender, Chapter 7, in *Digital Communications: Microwave Applications*, K. Feher, Ed., Prentice-Hall, Englewood Cliffs, NJ, 1981.

PROBLEMS

7.2-1 Consider a full-width rectangular pulse shape

$$p(t) = \Pi(t/T_b)$$

(a) Find PSDs for the polar, on-off, and bipolar signaling.

(b) Sketch roughly the PSDs and find their bandwidths. For each case, compare the bandwidth to the case where $p(t)$ is a half-width rectangular pulse.

7.2-2 **(a)** A random binary data sequence **110100101**$\cdots$ is transmitted by using a Manchester (split-phase) line code with the pulse $p(t)$ shown in Fig. 7.7a. Sketch the waveform $y(t)$.

(b) Derive $S_y(f)$, the PSD of a Manchester (split-phase) signal in part **(a)** assuming **1** and **0** equally likely. Roughly sketch this PSD and find its bandwidth.

7.2-3 If the pulse shape is

$$p(t) = \Pi\left(\frac{t}{0.5T_b}\right)$$

use differential code (see Fig. 7.18) to derive the PSD for a binary signal. Determine the PSD $S_y(f)$.

7.2-4 The **duobinary** line coding proposed by Lender is also ternary like bipolar, but it requires only half the bandwidth of bipolar. In practice, duobinary coding is indirectly realized by using a special pulse shape as discussed in Sec. 7.3 (see Fig. 7.18). In this code, a **0** is transmitted by no pulse, and a **1** is transmitted by a pulse $p(t)$ or $-p(t)$ using the following rule. A **1** is encoded by the same pulse as that used for the previous **1** if there are an even number of **0**s between them. It is encoded by a pulse of opposite polarity if there are an odd number of **0**s between them. A number 0 is considered to be an even number. Like bipolar, this code also has a single error detection capability, because correct reception implies that between successive pulses of the same polarity, an even number of **0**s must occur, and between successive pulses of opposite polarity, an odd number of **0**s must occur.

(a) Assuming half-width rectangular pulse, sketch the duobinary signal $y(t)$ for the random binary sequence

1110001101001010 $\cdots$

(b) Determine R_0, R_1, and R_2 for this code. Assume (or you may show if you like) that $R_n = 0$ for all $n > 2$. Find and sketch the PSD for this line code (assuming half-width pulse). Show that its bandwidth is $R_b/2$ Hz, half that of bipolar.

7.3-1 Data at a rate of 6 kbit/s is to be transmitted over a leased line of bandwidth 4 kHz by using Nyquist criterion pulses. Determine the maximum value of the roll-off factor r that can be used.

7.3-2 In a certain telemetry system, there are eight analog measurements, each of bandwidth 2 kHz. Samples of these signals are time-division-multiplexed, quantized, and binary-coded. The error in sample amplitudes cannot be greater than 1% of the peak amplitude.

(a) Determine L, the number of quantization levels.

(b) Find the transmission bandwidth B_T if Nyquist criterion pulses with roll-off factor $r = 0.2$ are used. The sampling rate must be at least 25% above the Nyquist rate.

7.3-3 A leased telephone line of bandwidth 3 kHz is used to transmit binary data. Calculate the data rate (in bits per second) that can be transmitted if we use:

(a) Polar signal with rectangular half-width pulses.

(b) Polar signal with rectangular full-width pulses.

(c) Polar signal using Nyquist criterion pulses of $r = 0.25$.

(d) Bipolar signal with rectangular half-width pulses.

(e) Bipolar signal with rectangular full-width pulses.

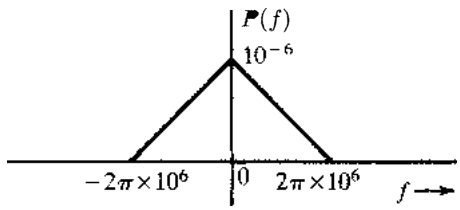
7.3-4 The Fourier transform $P(f)$ of the basic pulse $p(t)$ used in a certain binary communication system is shown in Fig. P7.3-4.

(a) From the shape of $P(f)$, explain at what pulse rate this pulse would satisfy Nyquist's criterion.

(b) Find $p(t)$ and verify that this pulse does (or does not) satisfy the Nyquist's criterion.

(c) If the pulse does satisfy the Nyquist criterion, what is the transmission rate (in bits per second) and what is the roll-off factor?

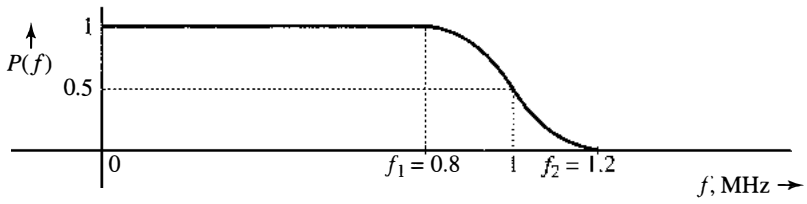
Figure P.7.3-4



7.3-5 A pulse $p(t)$ whose spectrum $P(f)$ is shown in Fig. P7.3-5 satisfies Nyquist's criterion. If $f_1 = 0.8$ MHz and $f_2 = 1.2$ MHz, determine the maximum rate at which binary data can be transmitted by this pulse using Nyquist's criterion. What is the roll-off factor?

7.3-6 Binary data at a rate of 1 Mbit/s is to be transmitted by using Nyquist criterion pulses with $P(f)$ shown in Fig. P7.3-5, The frequencies f_1 and f_2 of the spectrum are adjustable. The channel available for the transmission of this data has a bandwidth of 700 kHz. Determine f_1 and f_2 and the roll-off factor.

Figure P.7.3-5



7.3-7 Show that the inverse Fourier transform of $P(f)$ in Eq. (7.39) is indeed second criterion pulse $p(t)$ given in Eq. (7.38).

Hint: Use Eq. (3.32) to find the inverse transform of $P(f)$ in Eq. (7.39) and express $\text{sinc}(x)$ in the form $\sin x/x$.

7.3-8 Show that the inverse Fourier transform of $P(f)$ (the raised cosine pulse spectrum in Eq. (7.35) is the pulse $p(t)$ given in Eq (7.36).

Hint: Use Eq. (3.32) to find the inverse transform of $P(f)$ in Eq. (7.39) and express $\text{sinc}(x)$ in the form $\sin x/x$.

7.3-9 Show that there exists one (and only one) pulse $p(t)$ of bandwidth $R_b/2$ Hz that satisfies the criterion of second criterion pulse [Eq. (7.37)]. Show that this pulse is given by

$$p(t) = \{\text{sinc}(\pi R_b t) + \text{sinc}[\pi R_b(t - T_b)]\} = \frac{\sin(\pi R_b t)}{\pi R_b t(1 - R_b t)}$$

and its Fourier transform is $P(f)$ given in Eq. (7.39).

Hint: For a pulse of bandwidth $R_b/2$, the Nyquist interval is $1/R_b = T_b$, and the conditions (7.37) give the Nyquist sample values at $t = \pm nT_b$. Use the interpolation formula [Eq. (6.10)] with $B = R_b/2$, $T_s = T_b$ to construct $p(t)$. In determining $P(f)$, recognize that $(1 + e^{-j2\pi fT_b}) = e^{-j\pi fT_b}(e^{j\pi fT_b} + e^{-j\pi fT_b}2)$.

7.3-10 In a binary data transmission using duobinary pulses, sample values were read as follows:

$$1\,2\,0\;-2\;-2\,0\,0\;-2\,0\,2\,0\,0\,2\,0\,0\,0\;-2$$

(a) Explain if there is any error in detection.

(b) If there is no detection error, determine the received bit sequence.

7.3-11 In a binary data transmission using duobinary pulses, sample values of the received pulses were read as follows:

$$1\,2\,0\,0\,0\;-2\,0\,0\;-2\,0\,2\,0\,0\;-2\,0\,2\,2\,0\;-2$$

(a) Explain if there is any error.

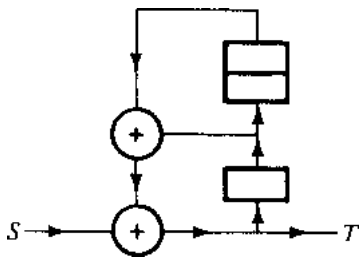
(b) Can you guess the correct transmitted digit sequence? There is more than one possible correct sequence. Give as many correct sequences as possible, assuming that more than one detection error is extremely unlikely.

7.4-1 In Example 7.2, when the sequence $S =$ **101010100000111** was applied to the input of the scrambler in Fig. 7.20a, the output T was found to be **101110001101001**. Verify that when this

sequence T is applied to the input of the descrambler in Fig. 7.20b, the output is the original input sequence, $S = \mathbf{101010100000111}$.

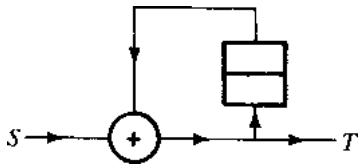
7.4-2 Design a descrambler for the scrambler of Fig. P7.4-2. If a sequence $S = \mathbf{101010100000111}$. is applied to the input of this scrambler, determine the output sequence T. Verify that if this T is applied to the input of the descrambler, the output is the sequence S.

Figure P.7.4-2



7.4-3 Repeat Prob. 7.4-2 if the scrambler shown in Fig. P7.4-3 is concatenated with the scrambler in Fig. P7.4-2 to form a composite scrambler.

Figure P.7.4-3



7.5-1 In a certain binary communication system that uses Nyquist's criterion pulses, a received pulse $p_r(t)$ (see Fig. 7.22a) has the following nonzero sample values:

$$p_r(0) = 1$$
$$p_r(T_b) = 0.1 \qquad p_r(-T_b) = 0.3$$
$$p_r(2T_b) = -0.02 \qquad p_r(-2T_b) = -0.07$$

(a) Determine the tap settings of a three-tap, zero-forcing equalizer.

(b) Using the equalizer in part **(a)**, find the residual nonzero ISI.

7.7-1 In a PAM scheme with $M = 16$:

(a) Determine the minimum transmission bandwidth required to transmit data at a rate of 12,000 bits/sec with zero ISI.

(b) Determine the transmission bandwidth if Nyquist criterion pulses with a roll-off factor $r = 0.2$ are used to transmit data.

7.7-2 An audio signal of bandwidth 4 kHz is sampled at a rate 25% above the Nyquist rate and quantized. The quantization error is not to exceed 0.1% of the signal peak amplitude. The resulting quantized samples are now coded and transmitted by 4-ary pulses.

(a) Determine the minimum number of 4-ary pulses required to encode each sample.

(b) Determine the minimum transmission bandwidth required to transmit this data with zero ISI.

(c) If 4-ary pulses satisfying Nyquist's criterion with 25% roll-off are used to transmit this data, determine the transmission bandwidth.

7.7-3 Binary data is transmitted over a certain channel at a rate R_b bit/s. To reduce the transmission bandwidth, it is decided to use 16-ary PAM signaling to transmit this data.

(a) By what factor is the bandwidth reduced?

(b) By what factor is the transmitted power increased, assuming minimum separation between pulse amplitudes to be the same in both cases?

Hint: Take the pulse amplitudes to be $\pm A/2$, $\pm 3A/2$, $\pm 5A/2$, $\pm 7A/2$, $\cdots$, $\pm 15A/2$ so that the minimum separation between various amplitude levels is A (same as that in the binary case pulses $\pm A/2$). Assume that all 16 levels are equally likely. Recall also that multiplying a pulse by a constant k increases its energy k^2-fold.

7.7-4 An audio signal of bandwidth 10 kHz is sampled at a rate of 24 kHz, quantized into 256 levels and coded by means of M-ary PAM pulses satisfying Nyquist's criterion with a roll-off factor $r = 0.2$. A 30 kHz bandwidth is available to transmit the data. Determine the best value of M.

7.7-5 Consider a case of binary transmission via polar signaling that uses half-width rectangular pulses of amplitudes $A/2$ and $-A/2$. The data rate is R_b bit/s.

(a) What is the minimum transmission bandwidth and the transmitted power.

(b) This data is to be transmitted by M-ary rectangular half-width pulses of amplitudes

$$\pm A/2,\ \pm 3A/2,\ \pm 5A/2, \ldots, \pm[(M-1)/2]A$$

Note that to maintain about the same noise immunity, the minimum pulse amplitude separation is A. If each of the M-ary pulses is equally likely to occur, show that the transmitted power is

$$P = \frac{(M^2-1)A^2}{24 \log_2 M}$$

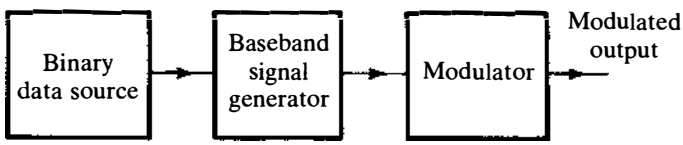
Also determine the transmission bandwidth.

7.8-1 Figure P7.8-1 shows a binary data transmission scheme. The baseband signal generator uses full-width pulses and polar signaling. The data rate is 1 Mbit/s.

(a) If the modulator generates a PSK signal, what is the bandwidth of the modulated output?

(b) If the modulator generates FSK with the difference $f_{c1} - f_{c0} = 100$ kHz (see Fig. 7.32c), determine the modulated signal bandwidth.

Figure P.7.8-1



7.8-2 Repeat Prob. 7.8-1 if, instead of full-width pulses, Nyquist's criterion pulses with $r = 0.2$ are used.

7.8-3 Repeat Prob. 7.8-1 if a multiamplitude scheme with $M = 4$ (PAM signaling with full-width pulse) is used. In FSK [Prob. 7.8-1, part **(b)**], assume that successive amplitude levels are transmitted by frequencies separated by 100 kHz.

8 FUNDAMENTALS OF PROBABILITY THEORY

Thus far, we have been studying signals whose values at any instant t are determined by their analytical or graphical description. These are called **deterministic** signals, implying complete certainty about their values at any moment t. Such signals, which can be specified with certainty, cannot convey information. It will be seen in Chapter 13 that information is inherently related to uncertainty. The higher the uncertainty about a signal (or message) to be received, the higher its information content. If a message to be received is specified (i.e., if it is known beforehand), then it contains no uncertainty and conveys no new information to the receiver. Hence, signals that convey information must be unpredictable. In addition to information-bearing signals, noise signals that perturb information signals in a system are also unpredictable (otherwise they can simply be subtracted). These unpredictable message signals and noise waveforms are examples of **random processes** that play key roles in communication systems and their analysis.

Random phenomena arise either because of our partial ignorance of the generating mechanism (as in message or noise signals) or because the laws governing the phenomena may be fundamentally random (as in quantum mechanics). Yet in another situation, such as the outcome of rolling a die, it is possible to predict the outcome provided we know exactly all the conditions: the angle of the throw, the nature of the surface on which it is thrown, the force imparted by the player, and so on. The exact analysis, however, is so complex and so sensitive to all the conditions that it is impractical to carry it out, and we are content to accept the outcome prediction on an average basis. Here the random phenomenon arises from our unwillingness to carry out the exact and full analysis because it is impractical to amass all the conditions precisely or not worth the effort.

We shall begin with a review of the basic concepts of the theory of probability, which forms the basis for describing random processes.

8.1 CONCEPT OF PROBABILITY

To begin the discussion of probability, we must define some basic elements and important terms. The term **experiment** is used in probability theory to describe a process whose outcome cannot be fully predicted because the conditions under which it is performed cannot be predetermined with sufficient accuracy and completeness. Tossing a coin, rolling a die, and drawing a card

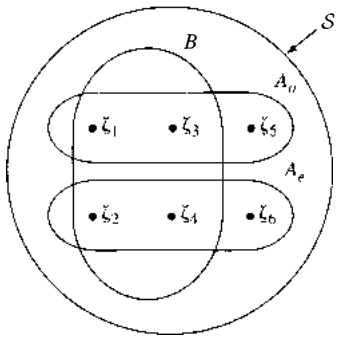


Figure 8.1 Sample space for a throw of a die.

from a deck are some examples of such experiments. An experiment may have several separately identifiable **outcomes**. For example, rolling a die has six possible identifiable outcomes (1, 2, 3, 4, 5, and 6). **An event** is a subset of outcomes that share some common characteristics. An event occurs if the outcome of the experiment belongs to the specific subset of outcomes defining the event. In the experiment of rolling a die, for example, the event "odd number on a throw" can result from any one of three outcomes (viz., 1, 3, and 5). Hence, this event is a set consisting of three outcomes (1, 3, and 5). Thus, events are groupings of outcomes into classes among which we choose to distinguish. The ideas of **experiment**, **outcomes**, and **events** form the basic foundation of probability theory. These ideas can be better understood by using the concepts of set theory.

We define the **sample space** $\mathcal{S}$ as a collection of all possible and separately identifiable outcomes of an experiment. In other words, the **sample space** $\mathcal{S}$ specifies the **experiment**. Each outcome is an **element**, or **sample point**, of this space $\mathcal{S}$ and can be conveniently represented by a point in the sample space. In the experiment of rolling a die, for example, the sample space consists of six elements represented by six sample points ζ_1, ζ_2, ζ_3, ζ_4, ζ_5, and ζ_6, where ζ_i represents the outcome "a number i is thrown" (Fig. 8.1). The event, on the other hand, is a subset of $\mathcal{S}$. The event "an odd number is thrown," denoted by A_o, is a subset of $\mathcal{S}$ (or a set of sample points ζ_1, ζ_3, and ζ_5). Similarly, the event A_e, "an even number is thrown," is another subset of $\mathcal{S}$ (or a set of sample points ζ_2, ζ_4, and ζ_6):

$$A_o = (\zeta_1, \zeta_3, \zeta_5) \qquad A_e = (\zeta_2, \zeta_4, \zeta_6)$$

Let us denote the event "a number equal to or less than 4 is thrown" as B. Thus,

$$B = (\zeta_1, \zeta_2, \zeta_3, \zeta_4)$$

These events are clearly marked in Fig. 8.1. Note that an outcome can also be an event, because an outcome is a subset of $\mathcal{S}$ with only one element.

The **complement** of any event A, denoted by A^c, is the event containing all points not in A. Thus, for the event B in Fig. 8.1, $B^c = (\zeta_5, \zeta_6)$, $A_o^c = A_e$, and $A_e^c = A_o$. An event that has no sample points is a **null event**, which is denoted by $\emptyset$ and is equal to $\mathcal{S}^c$.

The **union** of events A and B, denoted by $A \cup B$, is the event that contains all points in A and B. This is the event stated as having "an outcome of either A or B." For the events in Fig. 8.1,

$$A_o \cup B = (\zeta_1, \zeta_3, \zeta_5, \zeta_2, \zeta_4)$$
$$A_e \cup B = (\zeta_2, \zeta_4, \zeta_6, \zeta_1, \zeta_3)$$

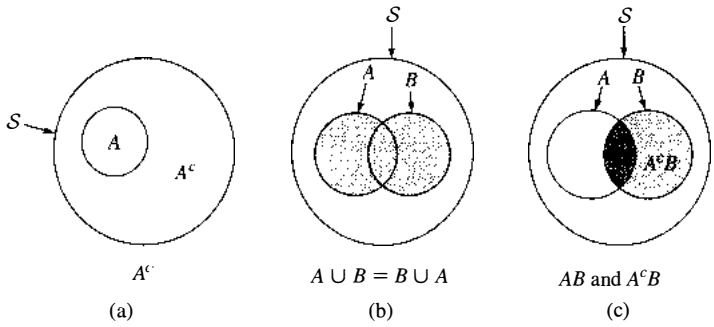


Figure 8.2 Representation of (a) complement, (b) union, and (c) intersection of events.

Observe that the union operation commutes:

$$A \cup B = B \cup A \tag{8.1}$$

The **intersection** of events A and B, denoted by $A \cap B$ or simply by AB, is the event that contains points common to A and B. This is the event that "outcome is both A and B," also known as the **joint event** $A \cap B$. Thus, the event A_eB, "a number that is even and equal to or less than 4 is thrown," is a set (ζ_2, ζ_4), and similarly for A_oB,

$$A_eB = (\zeta_2, \zeta_4) \qquad A_oB = (\zeta_1, \zeta_3)$$

Observe that the intersection also commutes

$$A \cap B = B \cap A \tag{8.2}$$

All these concepts can be demonstrated on a Venn diagram (Fig. 8.2). If the events A and B are such that

$$A \cap B = \emptyset \tag{8.3}$$

then A and B are said to be **disjoint**, or **mutually exclusive**, events. This means events A and B cannot occur simultaneously. In Fig. 8.1 events A_e and A_o are mutually exclusive, meaning that in any trial of the experiment if A_e occurs, A_o cannot occur at the same time, and vice versa.

Relative Frequency and Probability

Although the outcome of an experiment is unpredictable, there is a *statistical regularity* about the outcomes. For example, if a coin is tossed a large number of times, about half the times the outcome will be "heads," and the remaining half of the times it will be "tails." We may say that the relative frequency of the two outcomes "heads" or "tails" is one-half. This relative frequency represents the likelihood of a particular event.

Let A be one of the events of interest in an experiment. If we conduct a sequence of N independent trials* of this experiment, and if the event A occurs in $N(A)$ out of these N trials, then the fraction

$$f(A) = \lim_{N \to \infty} \frac{N(A)}{N} \tag{8.4}$$

* Trials conducted under similar discernible conditions.

is called the **relative frequency** of the event A. Observe that for small N, the fraction $N(A)/N$ may vary widely with N. As N increases, the fraction will approach a limit because of statistical regularity.

The probability of an event has the same connotations as the relative frequency of that event. Hence, we estimate the probability of each event, as the relative frequency of that event.* Therefore, to an event A, we assign the probability $P(A)$ as

$$P(A) = \lim_{N\to\infty} \frac{N(A)}{N} \tag{8.5}$$

From Eq. (8.5), it follows that

$$0 \le P(A) \le 1 \tag{8.6}$$

Example 8.1 Assign probabilities to each of the six outcomes in Fig. 8.1.

Because each of the six outcomes is equally likely in a large number of independent trials, each outcome will appear in one-sixth of the trials. Hence,

$$P(\zeta_i) = \frac{1}{6} \qquad i = 1,\ 2,\ 3,\ 4,\ 5,\ 6 \tag{8.7}$$

Consider now the two events A and B of an experiment. Suppose we conduct N independent trials of this experiment and events A and B occur in $N(A)$ and $N(B)$ trials, respectively. If A and B are mutually exclusive (or disjoint), then if A occurs, B cannot occur, and vice versa. Hence, the event $A \cup B$ occurs in $N(A) + N(B)$ trials and

$$\begin{aligned} P(A \cup B) &= \lim_{N\to\infty} \frac{N(A) + N(B)}{N} \\ &= P(A) + P(B) \qquad \text{if } A \cap B = \emptyset \end{aligned} \tag{8.8}$$

This result can be extended to more than two mutually exclusive events. In other words, if events $\{A_i\}$ are mutually exclusive such that

$$A_i \cap A_j = \emptyset \qquad i \neq j$$

then

$$P\left(\bigcup_i A_i\right) = \sum_i P(A_i)$$

---

* Observe that we are not defining the probability by the relative frequency. To a given event, a probability is closely estimated by the relative frequency of the event when this experiment is repeated many times. Modern theory of probability, being a branch of mathematics, starts with certain axioms about probability [Eqs. (8.6), (8.8), and (8.11)]. It assumes that somehow these probabilities are assigned by nature. We use relative frequency to estimate probability because it is reasonable in the sense that it closely approximates our experience and expectation of "probability."

Example 8.2 Assign probabilities to the events A_e, A_o, B, A_eB, and A_oB in Fig. 8.1.

Because $A_e = (\zeta_2 \cup \zeta_4 \cup \zeta_6)$ where ζ_2, ζ_4, and ζ_6 are mutually exclusive,

$$P(A_e) = P(\zeta_2) + P(\zeta_4) + P\zeta_6)$$

From Eq. (8.7) it follows that

$$P(A_e) = \frac{1}{2} \tag{8.9a}$$

Similarly,

$$P(A_o) = \frac{1}{2} \tag{8.9b}$$

$$P(B) = \frac{2}{3} \tag{8.9c}$$

From Fig. 8.1 we also observe that

$$A_eB = \zeta_2 \cup \zeta_4$$

and

$$P(A_eB) = P(\zeta_2) + P(\zeta_4) = \frac{1}{3} \tag{8.10a}$$

Similarly,

$$P(A_oB) = \frac{1}{3} \tag{8.10b}$$

We can also show that

$$P(\mathcal{S}) = 1 \tag{8.11}$$

This result can be proved by using the relative frequency. Let an experiment be repeated N times (N large). Because $\mathcal{S}$ is the union of all possible outcomes, $\mathcal{S}$ occurs in every trial. Hence, N out of N trials lead to event $\mathcal{S}$, and the result follows.

Example 8.3 Two dice are thrown. Determine the probability that the sum on the dice is seven.

For this experiment, the sample space contains 36 sample points because 36 possible outcomes exist. All the outcomes are equally likely. Hence, the probability of each outcome is 1/36.

A sum of seven can be obtained by the six combinations: (1, 6), (2, 5), (3, 4), (4, 3) (5, 2), and (6, 1). Hence, the event "a seven is thrown" is the union of six outcomes, each with probability 1/36. Therefore,

$$P(\text{"a seven is thrown"}) = \frac{1}{36} + \frac{1}{36} + \frac{1}{36} + \frac{1}{36} + \frac{1}{36} + \frac{1}{36} = \frac{1}{6}$$

Example 8.4 A coin is tossed four times in succession. Determine the probability of obtaining exactly two heads.

A total of $2^4 = 16$ distinct outcomes are possible, all of which are equally likely because of the symmetry of the situation. Hence, the sample space consists of 16 points, each with probability 1/16. The 16 outcomes are as follows:

HHHH	*TTTT*
HHHT	*TTTH*
HHTH	*TTHT*
⟶ *HHTT*	⟶ *TTHH*
HTHH	*THTT*
⟶ *HTHT*	⟶ *THTH*
⟶ *HTTH*	⟶ *THHT*
HTTT	*THHH*

Six out of these 16 outcomes lead to the event "obtaining exactly two heads" (arrows). Because all of the six outcomes are disjoint (mutually exclusive),

$$P(\text{obtaining exactly two heads}) = \frac{6}{16} = \frac{3}{8}$$

In Example 8.4, the method of listing all possible outcomes quickly becomes unwieldy as the number of tosses increases. For example, if a coin is tossed just 10 times, the total number of outcomes is 1024. A more convenient approach would be to apply the results of combinatorial analysis used in Bernoulli trials, to be discussed shortly.

Conditional Probability and Independent Events

Conditional Probability: It often happens that the probability of one event is influenced by the outcome of another event. As an example, consider drawing two cards in succession from a deck. Let A denote the event that the first card drawn is an ace. We do not replace the card drawn in the first trial. Let B denote the event that the second card drawn is an ace. It is evident that the probability of drawing an ace in the second trial will be influenced by the outcome of the first draw. If the first draw does not result in an ace, then the probability of obtaining an ace in the second trial is 4/51. The probability of event B thus depends on whether event A occurs. We now introduce the **conditional probability** $P(B|A)$ to denote the probability of event B when it is known that event A has occurred. $P(B|A)$ is read as "probability of B given A."

Let there be N trials of an experiment, in which the event A occurs n_1 times. Of *these* n_1 trials, event B occurs n_2 times. It is clear that n_2 is the number of times that the joint event $A \cap B$ (Fig. 8.2c) occurs. That is,

$$P(A \cap B) = \lim_{N\to\infty}\left(\frac{n_2}{N}\right) = \lim_{N\to\infty}\left(\frac{n_1}{N}\right)\left(\frac{n_2}{n_1}\right)$$

Note that $\lim_{N\to\infty}(n_1/N) = P(A)$. Also, $\lim_{N\to\infty}(n_2/n_1) = P(B|A)$,* because B occurs n_2 of the n_1 times that A occurred. This represents the conditional probability of B given A. Therefore,

$$P(A \cap B) = P(A)P(B|A) \tag{8.12}$$

and

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \qquad \text{provided } P(A) > 0 \tag{8.13a}$$

Using a similar argument, we obtain

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \qquad \text{provided } P(B) > 0 \tag{8.13b}$$

It follows from Eqs. (8.13) that

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \tag{8.14a}$$

$$P(B|A) = \frac{P(B)P(A|B)}{P(A)} \tag{8.14b}$$

Equations (8.14) are called **Bayes' rule**. In Bayes' rule, one conditional probability is expressed in terms of the reversed conditional probability.

Example 8.5 An experiment consists of drawing two cards from a deck in succession (without replacing the first card drawn). Assign a value to the probability of obtaining two red aces in two draws.

Let A and B be the events "red ace in the first draw" and "red ace in the second draw," respectively. We wish to determine $P(A \cap B)$,

$$P(A \cap B) = P(A)P(B|A)$$

and the relative frequency of A is 2/52 = 1/26. Hence,

$$P(A) = \frac{1}{26}$$

---

* Here we are implicitly using the fact that $n_1 \to \infty$ as $N \to \infty$. This is true provided the ratio $\lim_{N\to\infty}(n_1/N) \neq 0$, that is, if $P(A) \neq 0$.

Also, $P(B|A)$ is the probability of drawing a red ace in the second draw given that the first draw was a red ace. The relative frequency of this event is 1/51, so

$$P(B|A) = \frac{1}{51}$$

Hence,

$$P(A \cap B) = \left(\frac{1}{26}\right)\left(\frac{1}{51}\right) = \frac{1}{1326}$$

Independent Events: Under conditional probability, we presented an example where the occurrence of one event was influenced by the occurrence of another. There are, of course, many examples in which two or more events are entirely independent; that is, the occurrence of one event in no way influences the occurrence of the other event. As an example, we again consider the drawing of two cards in succession, but in this case we replace the card obtained in the first draw and shuffle the deck before the second draw. In this case, the outcome of the second draw is in no way influenced by the outcome of the first draw. Thus $P(B)$, the probability of drawing an ace in the second draw, is independent of whether the event A (drawing an ace in the first trial) occurs. Thus, the events A and B are independent. The conditional probability $P(B|A)$ is given by $P(B)$.

The event B is said to be **independent** of the event A if and only if

$$P(A \cap B) = P(A)P(B) \tag{8.15a}$$

Note that if the events A and B are independent, it follows from Eqs. (8.13a) and (8.15b) that

$$P(B|A) = P(B) \tag{8.15b}$$

This relationship states that if B is independent of A, then its probability is not affected by the event A. Naturally, if event B is independent of event A, then event A is also independent of B. It can been seen from Eqs. (8.14) that

$$P(A|B) = P(A) \tag{8.15c}$$

Note that there is a huge difference between **independent events** and **mutually exclusive events**. If A and B are mutually exclusive, then $A \cap B$ is empty and $P(A \cap B) = 0$. If A and B are mutually exclusive, then A and B cannot occur at the same time. This clearly means that they are NOT independent events.

Bernoulli Trials

In Bernoulli trials, if a certain event A occurs, we call it a "success." If $P(A) = p$, then the probability of success is p. If q is the probability of failure, then $q = 1 - p$. We shall find the probability of k successes in n (Bernoulli) trials. The outcome of each trial is independent of the outcomes of the other trials. It is clear that in n trials, if success occurs in k trials, failure occurs in $n - k$ trials. Since the outcomes of the trials are independent, the probability of this event is clearly $p^n(1-p)^{n-k}$, that is,

$$P(k \text{ successes in a specific order in } n \text{ trials}) = p^k(1-p)^{n-k}$$

But the event of "k successes in n trials" can occur in many different ways (different orders). It is well known from combinatorial analysis that there are

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{8.16}$$

ways in which k positions can be taken from n positions (which is the same as the number of ways of achieving k successes in n trials).

This can be proved as follows. Consider an urn containing n distinguishable balls marked $1, 2, \ldots, n$. Suppose we draw k balls from this urn without replacing them. The first ball could be any one of the n balls, the second ball could be any one of the remaining $(n-1)$ balls, and so on. Hence, the total number of ways in which k balls can be drawn is

$$n(n-1)(n-2)\ldots(n-k+1) = \frac{n!}{(n-k)!}$$

Next, consider any one set of the k balls drawn. These balls can be ordered in different ways. We could label any one of the k balls as number 1, and any one of the remaining $(k-1)$ balls as number 2, and so on. This will give a total of $k(k-1)(k-2)\cdots 1 = k!$ distinguishable patterns formed from the k balls. The total number of ways in which k things can be taken from n things is $n!/(n-k)!$ But many of these ways will use the same k things, arranged in different order. The ways in which k things can be taken from n things without regard to order (unordered subset k taken from n things) is $n!/(n-k)!$ divided by $k!$ This is precisely defined by Eq. (8.16).

This means the probability of k successes in n trials is

$$\begin{aligned} P(k \text{ successes in } n \text{ trials}) &= \binom{n}{k} p^k (1-p)^{n-k} \\ &= \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k} \end{aligned} \tag{8.17}$$

Tossing a coin and observing the number of heads is a Bernoulli trial with $p = 0.5$. Hence, the probability of observing k heads in n tosses is

$$P(k \text{ heads in } n \text{ tosses}) = \binom{n}{k} (0.5)^k (0.5)^{n-k} = \frac{n!}{k!(n-k)!} (0.5)^n$$

Example 8.6 A binary symmetric channel (BSC) has an error probability P_e (i.e., the probability of receiving **0** when **1** is transmitted, or vice versa, is P_e). Note that the channel behavior is symmetrical with respect to **0** and **1**. Thus,

$$P(\mathbf{0}|\mathbf{1}) = P(\mathbf{1}|\mathbf{0}) = P_e$$

and

$$P(\mathbf{0}|\mathbf{0}) = P(\mathbf{1}|\mathbf{1}) = 1 - P_e$$

where $P(y|x)$ denotes the probability of receiving y when x is transmitted. A sequence of n binary digits is transmitted over this channel. Determine the probability of receiving exactly k digits in error.

The reception of each digit is independent of the other digits. This is an example of a Bernoulli trial with the probability of success $p = P_e$ ("success" here is receiving a digit in error). Clearly, the probability of k successes in n trials (k errors in n digits) is

$$P(\text{receiving } k \text{ out of } n \text{ digits in error}) = \binom{n}{k} P_e^k (1 - P_e)^{n-k}$$

For example, if $P_e = 10^{-5}$, the probability of receiving two digits wrong in a sequence of eight digits is

$$\binom{8}{2} (10^{-5})^2 (1 - 10^{-5})^6 \simeq \frac{8!}{2!\,6!} 10^{-10} = (2.8)10^{-9}$$

Example 8.7 PCM Repeater Error Probability

In pulse code modulation, regenerative repeaters are used to detect pulses (before they are lost in noise) and retransmit new, clean pulses. This combats the accumulation of noise and pulse distortion.

A certain PCM channel consists of n identical links in tandem (Fig. 8.3). The pulses are detected at the end of each link and clean new pulses are transmitted over the next link. If P_e is the probability of error in detecting a pulse over any one link, show that P_E, the probability of error in detecting a pulse over the entire channel (over the n links in tandem), is

$$P_E \simeq nP_e \qquad nP_e \ll 1$$

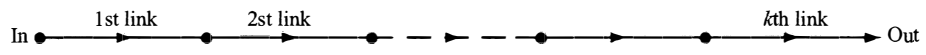


Figure 8.3 A PCM repeater.

The probabilities of detecting a pulse correctly over one link and over the entire channel (n links in tandem) are $1 - P_e$ and $1 - P_E$, respectively. A pulse can be detected correctly over the entire channel if either the pulse is detected correctly over every link or errors are made over an even number of links only.

$$\begin{aligned} 1 - P_E = {} & P(\text{correct detection over all links}) \\ & + P(\text{error over two links only}) \\ & + P(\text{error over four links only}) + \cdots \\ & + P\left(\text{error over } 2\left\lfloor \frac{n}{2} \right\rfloor \text{ links only}\right) \end{aligned}$$

where $\lfloor a \rfloor$ denotes the largest integer less than or equal to a.
Because pulse detection over each link is independent of the other links (see Example 8.6),

$$P(\text{correct detection over all } n \text{ links}) = (1 - P_e)^n$$

and

$$P(\text{error over } k \text{ links only}) = \frac{n!}{k!(n-k)!} P_e^k (1 - P_e)^{n-k}$$

Hence,

$$1 - P_E = (1 - P_e)^n + \sum_{k=2,4,6,\ldots}^{\alpha} \frac{n!}{k!(n-k)!} P_e^k (1 - P_e)^{n-k}$$

In practice, $P_e \ll 1$, so only the first two terms on the right-hand side of this equation are of significance. Also, $(1 - P_e)^{n-k} \simeq 1$, and

$$\begin{aligned} 1 - P_E &\simeq (1 - P_e)^n + \frac{n!}{2!(n-2)!} P_e^2 \\ &= (1 - P_e)^n + \frac{n(n-1)}{2} P_e^2 \end{aligned}$$

If $nP_e \ll 1$, then the second term can also be neglected, and

$$\begin{aligned} 1 - P_E &\simeq (1 - P_e)^n \\ &\simeq 1 - nP_e \qquad nP_e \ll 1 \end{aligned}$$

and

$$P_E \simeq nP_e$$

We can explain this result heuristically by considering the transmission of N ($N \to \infty$) pulses. Each link makes NP_e errors, and the total number of errors is approximately nNP_e (approximately, because some of the erroneous pulses over a link will be erroneous over other links). Thus the overall error probability is nP_e.

Example 8.8

In binary communication, one of the techniques used to increase the reliability of a channel is to repeat a message several times. For example, we can send each message (**0** or **1**) three times. Hence, the transmitted digits are **000** (for message **0**) or **111** (for message **1**). Because of channel noise, we may receive any one of the eight possible combinations of three binary digits. The decision as to which message is transmitted is made by the majority rule; that is, if at least two of the three detected digits are **0**, the decision is **0**, and so on. This scheme permits correct reception of data even if one out of three digits is in error. Detection error occurs only if at least two out of three digits are received in error. If P_e is the error probability of one digit, and $P(\epsilon)$ is the probability of making a wrong decision in this scheme, then

$$\begin{aligned} P(\epsilon) &= \sum_{k=2}^{3} \binom{3}{k} P_e^k (1 - P_e)^{3-k} \\ &= 3P_e^2(1 - P_e) + P_e^3 \end{aligned}$$

In practice, $P_e \ll 1$, and

$$P(\epsilon) \simeq 3P_e^2$$

For instance, if $P_e = 10^{-4}$, $P(\epsilon) \simeq 3 \times 10^{-8}$. Thus, the error probability is reduced from 10^{-4} to 3×10^{-8}. We can use any odd number of repetitions for this scheme to function.

In this example, higher reliability is achieved at the cost of a reduction in the rate of information transmission by a factor of 3. We shall see in Chapter 14 that more efficient ways exist to effect a trade-off between reliability and the rate of transmission through the use of error correction codes.

Multiplication Rule for Conditional Probabilities

As shown in Eq. (8.12), we can write the joint event

$$P(A \cap B) = P(A)P(B/A)$$

This rule on joint events can be generalized for multiple events $A_1, A_2, \ldots, A_n$ via iterations. If $A_1A_2 \cdots A_n \neq \emptyset$, then we have

$$P(A_1A_2 \cdots A_n) = \frac{P(A_1A_2 \cdots A_n)}{P(A_1A_2 \cdots A_{n-1})} \cdot \frac{P(A_1A_2 \cdots A_{n-1})}{P(A_1A_2 \cdots A_{n-2})} \cdots \frac{P(A_1A_2)}{P(A_1)} \cdot P(A_1) \tag{8.18a}$$

$$= P(A_n|A_1A_2 \cdots A_{n-1}) \cdot P(A_{n-1}|A_1A_2 \cdots A_{n-2}) \cdots P(A_2|A_1) \cdot P(A_1) \tag{8.18b}$$

Note that since $A_1A_2 \cdots A_n \neq \emptyset$, every denominator in Eq. (8.18a) is positive and well defined.

Example 8.9 Suppose a box of diodes consist of N_g good diodes and N_b bad diodes. If five diodes are randomly selected, one at a time, without replacement, determine the probability of obtaining the sequence of diodes in the order of *good*, *bad*, *good*, *good*, *bad*.

We can denote G_i as the event that the ith draw is a good diode. We are interested in the event of $G_1G_2^cG_3G_4G_5^c$.

$$P(G_1G_2^cG_3G_4G_5^c) = P(G_1)P(G_2^c|G_1)P(G_3|G_1G_2^c)P(G_4|G_1G_2^cG_3)P(G_5^c|G_1G_2^cG_3G_4)$$

$$= \frac{N_g}{N_g + N_b} \cdot \frac{N_b}{N_g + N_b - 1} \cdot \frac{N_g - 1}{N_b + N_g - 2} \cdot \frac{N_g - 2}{N_g + N_b - 3} \cdot \frac{N_b - 1}{N_g + N_b - 4}$$

To Divide and Conquer: The Total Probability Theorem

In analyzing a particular event of interest, sometimes a direct approach to evaluating its probability can be difficult because there can be so many different outcomes to enumerate. When dealing with such problems, it is often advantageous to adopt the *divide-and-conquer* approach by separating all the possible causes leading to the particular event of interest B. The total probability theorem provides a perfect tool for analyzing the probability of such problems.

We define $\mathcal{S}$ as the sample space of the experiment of interest. As shown in Fig. 8.4, the entire sample space can be partitioned into n disjoint events $A_1, \ldots, A_n$. We can now state the theorem:

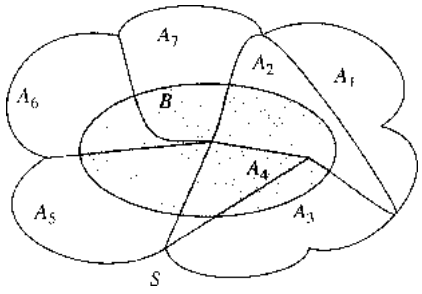


Figure 8.4
The event of interest B and the partition of S by $\{A_i\}$.

Total Probability Theorem: Let n disjoint events $A_1, \ldots, A_n$ form a partition of the sample space S such that

$$\bigcup_{i=1}^{n} A_i = S \quad \text{and} \quad A_i \cap A_j = \emptyset, \quad \text{if} \quad i \neq j$$

Then the probability of an event B can be written as

$$P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$$

Proof: The proof of this theorem is quite simple based on Fig. 8.4. Since $\{A_i\}$ form a partition of S, then

$$\begin{aligned} B &= B \cap S = B \cap (A_1 \cup A_2 \cup \cdots \cup A_n) \\ &= (A_1 B) \cup (A_2 B) \cup \cdots \cup (A_n B) \end{aligned}$$

Because $\{A_i\}$ are disjoint, so are $\{A_i B\}$. Thus,

$$P(B) = \sum_{i=1}^{n} P(A_i B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$$

This theorem can simplify the analysis of the more complex event of interest B by identifying all different causes A_i for B. By quantifying the effect of A_i on B through $P(B|A_i)$, the theorem allows us to "divide-and-conquer" a complex problem (of event B).

Example 8.10 The decoding of a data packet may be in error because of N distinct error patterns $E_1, E_2, \ldots, E_N$ it encounters. These error patterns are mutually exclusive, each with probability $P(E_i) = p_i$. When the error pattern E_i occurs, the data packet would be incorrectly decoded with probability q_i. Find the probability that the data packet is incorrectly decoded.

Solution

We apply total probability theorem to tackle this problem. First, define B as the event that the data packet is incorrectly decoded. Based on the problem, we know that

$$P(B|E_i) = q_i \quad \text{and} \quad P(E_i) = p_i$$

Furthermore, the data packet has been incorrectly decoded. Therefore

$$\sum_{i=1}^{n} p_i = 1$$

Applying the total probability theorem, we find that

$$P(B) = \sum_{i=1}^{n} P(B|E_i)P(E_i) = \sum_{i=1}^{n} q_i p_i$$

Isolating a Particular Cause: Bayes' Theorem

The total probability theorem facilitates the probabilistic analysis of a complex event by using a *divide-and-conquer* approach. In practice, it may also be of interest to determine the likelihood of a particular cause of an event among many disjoint possible causes. Bayes' theorem provides the solution to this problem.

Bayes' Theorem: Let n disjoint events $A_1, \ldots, A_n$ form a partition of the sample space $\mathcal{S}$. Let B be an event with $P(B) > 0$. Then for $j = 1, \ldots, n$,

$$P(A_j|B) = \frac{P(B|A_j)P(A_j)}{P(B)} = \frac{P(B|A_j)P(A_j)}{\sum_{i=1}^{n} P(B|A_i)P(A_i)}$$

The proof is already given by the theorem itself.

Bayes' theorem provides a simple method for computing the conditional probability of A_j given that B has occurred. The probability $P(A_j|B)$ is often known as the *posterior probability* of event A_j. It describes, among n possible causes of B, the probability that B may be caused by A_j. In other words, Bayes' theorem isolates and finds the relative likelihood of each possible cause to an event of interest.

Example 8.11 A communication system always encounters one of three possible interference waveforms: F_1, F_2, or F_3. The probability of each interference is 0.8, 0.16, and 0.04, respectively. The communication system fails with probabilities 0.01, 0.1, and 0.4 when it encounters F_1, F_2, and F_3, respectively. Given that the system has failed, find the probability that the failure is a result of F_1, F_2, or F_3, respectively.

Denote B as the event of system failure. We know from the description that

$$P(F_1) = 0.8 \qquad P(F_2) = 0.16 \qquad P(F_3) = 0.04$$

Furthermore, the effect of each interference on the system is given by

$$P(B|F_1) = 0.01 \qquad P(B|F_2) = 0.1 \qquad P(B|F_3) = 0.4$$

Now following Bayes' theorem, we find that

$$P(F_1|B) = \frac{P(B|F_1)P(F_1)}{\sum_{i=1}^{3} P(B|F_i)P(F_i)} = \frac{(0.01)(0.8)}{(0.01)(0.8) + (0.1)(0.16) + (0.4)(0.04)} = 0.2$$

$$P(F_2|B) = \frac{P(B|F_2)P(F_2)}{\sum_{i=1}^{3} P(B|F_i)P(F_i)} = 0.4$$

$$P(F_3|B) = \frac{P(B|F_3)P(F_3)}{\sum_{i=1}^{3} P(B|F_i)P(F_i)} = 0.4$$

Example 8.11 illustrates the major difference between the *posterior probability* $P(F_i|B)$ and the *prior probability* $P(F_i)$. Although the prior probability $P(F_3) = 0.04$ is the lowest among the three possible interferences, once the failure event B has occurred, $P(F_3|B) = 0.4$ is actually one of the most likely events. Bayes' theorem is an important tool in communications for determining the relative likelihood of a particular cause to an event.

Axiomatic Theory of Probability

The relative frequency definition of probability is intuitively appealing. Unfortunately, it has some serious mathematical objections. Logically there is no reason why we should get the same estimate of the relative frequency whether we base it on 10,000 trials or on 20. Moreover, in the relative frequency definition, it is not clear when and in what mathematical sense the limit in Eq. (8.5) exists. If we consider a set of an infinite number of trials, we can partition such a set into several subsets, such as odd and even numbered trials. Each of these subsets (of infinite trials each) would have its own relative frequency. So far, all the attempts to prove that the relative frequencies of all the subsets are equal have been futile.[1] There are some other difficulties also. For instance, in some cases, such as Julius Caesar having visited Great Britain, it is an experiment for which we cannot repeat the event an infinite number of trials. Thus, we can never know the probability of such an event. We, therefore, need to develop a theory of probability that is not tied down to any particular definition of probability. In other words, we must separate the empirical and the formal problems of probability. Assigning probabilities to events is an empirical aspect, and setting up purely formal calculus to deal with probabilities (assigned by whatever empirical method) is the formal aspect.

It is instructive to consider here the basic difference between physical sciences and mathematics. Physical sciences are based on **inductive logic**, while mathematics is strictly a **deductive logic**. Inductive logic consists of making a large number of observations and then generalizing, from these observations, laws that will explain these observations. For instance, history and experience tell us that every human being must die someday. This leads to a law that *humans are mortals*. This is inductive logic. Based on a law (or laws) obtained by inductive logic, we can make further deductions. The statement "John is a human being, so he must die some day" is an example of deductive logic. Deriving the laws of the physical sciences is basically an exercise in inductive logic, whereas mathematics is pure deductive logic. In a physical science we make observations in a certain field and generalize these observations into laws such as Ohm's law, Maxwell's equations, and quantum mechanics. There are no other proofs for these inductively obtained laws; they are found to be true by observation. But once we have such inductively formulated laws (axioms or hypotheses), by using thought process, we can deduce additional results based on these *basic laws or axioms* alone. This is the proper domain of mathematics. All these deduced results have to be proved rigorously based on a set of axioms. Thus, based on Maxwell's equations alone, we can derive the laws of the propagation of electromagnetic waves.

This discussion shows that the discipline of mathematics can be summed up in one aphorism: "This implies that." In other words, if we are given a certain set of axioms (hypotheses),

then, based upon these axioms alone, what else is true? As Bertrand Russell puts it: "Pure mathematics consists entirely of such asseverations as that, if such and such proposition is true of anything, then such and such another proposition is true of that thing." Seen in this light, it may appear that assigning probability to an event may not necessarily be the responsibility of the mathematical discipline of probability. Under mathematical discipline, we need to start with a set of axioms about probability and then investigate what else can be said about probability based on this set of axioms alone. We start with a concept (as yet undefined) of probability and postulate axioms. The axioms must be internally consistent and should conform to the observed relationships and behavior of probability in the practical and the intuitive sense. It is beyond the scope of this book to discuss how these axioms are formulated. The modern theory of probability starts with Eqs. (8.6), (8.8), and (8.11) as its axioms. Based on these three axioms alone, what else is true is the essence of modern theory of probability. The relative frequency approach uses Eq. (8.5) to define probability, and Eqs. (8.5), (8.8), and (8.11) follow as a consequence of this definition. In the axiomatic approach, on the other hand, we do not say anything about how we assign probability $P(A)$ to an event A; rather, we postulate that the probability function must obey the three postulates or axioms in Eqs. (8.6), (8.8), and (8.11). The modern theory of probability does not concern itself with the problem of assigning probabilities to events. It assumes that somehow the probabilities were assigned to these events a priori.

If a mathematical model is to conform to the real phenomenon, we must assign these probabilities in away that is consistent with an empirical and an intuitive understanding of probability. The concept of relative frequency is admirably suited for this. Thus, although we use relative frequency to assign (not define) probabilities, it is all under the table, not a part of the mathematical discipline of probability.

8.2 RANDOM VARIABLES

The outcome of an experiment may be a real number (as in the case of rolling a die), or it may be nonnumerical and describable by a phrase (such as "heads" or "tail" in tossing a coin). From a mathematical point of view, it is simpler to have numerical values for all outcomes. For this reason, we assign a real number to each sample point according to some rule. If there are m sample points $\zeta_1, \zeta_2, \ldots, \zeta_m$, then using some convenient rule, we assign a real number $\mathrm{x}(\zeta_i)$ to sample point ζ_i ($i = 1, 2, \ldots, m$). In the case of tossing a coin, for example, we may assign the number 1 for the outcome heads and the number -1 for the outcome tails (Fig. 8.5).

Thus, x(.) is a function that maps sample points $\zeta_1, \zeta_2, \ldots, \zeta_m$ into real numbers $x_1, x_2, \ldots, x_n$.* We now have a **random variable** x that takes on values $x_1, x_2, \ldots, x_n$. We shall use roman type (x) to denote a random variable (RV) and italic type (e.g., $x_1, x_2, \ldots, x_n$) to denote the value it takes. The probability of an RV x taking a value x_i is $P_{\mathrm{x}}(x_i)$ = Probability of "$\mathrm{x} = x_i$."

Discrete Random Variables

A random variable is discrete if there exists a denumerable sequence of distinct numbers x_i such that

$$\sum_i P_{\mathrm{x}}(x_i) = 1 \tag{8.19}$$

* The number m is not necessarily equal to n. More than one sample point can map into one value of x.

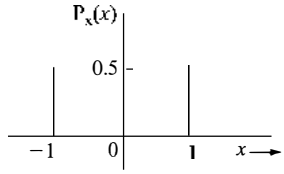


Figure 8.5 Probabilities in a coin-tossing experiment.

Thus, a discrete RV can assume only certain discrete values. An RV that can assume any value over a continuous set is called a **continuous** random variable.

Example 8.12 Two dice are thrown. The sum of the points appearing on the two dice is an RV x. Find the values taken by x, and the corresponding probabilities.

We see that x can take on all integral values from 2 through 12. Various probabilities can be determined by the method outlined in Example 8.3.

There are 36 sample points in all, each with probability 1/36. Dice outcomes for various values of x are shown in Table 8.1. Note that although there are 36 sample points, they all map into 11 values of x. This is because more than one sample point maps into the same value of x. For example, six sample points map into $x = 7$.

The reader can verify that $\sum_{i=2}^{12} P_x(x_i) = 1$.

TABLE 8.1

Value of x_i	Dice Outcomes	$P_x(x_i)$
2	(1, 1)	1/36
3	(1, 2), (2, 1)	2/36 = 1/18
4	(1, 3), (2, 2), (3, 1)	3/36 = 1/12
5	(1, 4), (2, 3), (3, 2), (4, 1)	4/36 = 1/9
6	(1, 5), (2, 4), (3, 3), (4, 2), (5, 1)	5/36
7	(1, 6), (2, 5), (3, 4), (4, 3), (5, 2), (6, 1)	6/36 =1/6
8	(2, 6), (3, 5), (4, 4), (5, 3), (6, 2)	5/36
9	(3, 6), (4, 5), (5, 4), (6, 3)	4/36 = 1/9
10	(4, 6), (5, 5), (6, 4)	3/36 = 1/12
11	(5, 6), (6, 5)	2/36 = 1/18
12	(6, 6)	1/36

The preceding discussion can be extended to two RVs, x and y. The joint probability $P_{xy}(x_i, y_j)$ is the probability that "$x = x_i$ and $y = y_j$." Consider, for example, the case of a coin tossed twice in succession. If the outcomes of the first and second tosses are mapped into RVs x and y, then x and y each takes values 1 and −1. Because the outcomes of the two tosses are independent, x and y are independent, and

$$P_{xy}(x_i, y_j) = P_x(x_i)\, P_y(y_j)$$

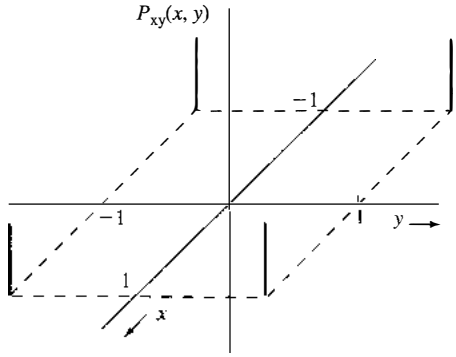


Figure 8.6 Representation of joint probabilities of two random variables.

and

$$P_{xy}(1, 1) = P_{xy}(1, -1) = P_{xy}(-1, 1) = P_{xy}(-1, -1) = \tfrac{1}{4}$$

These probabilities are plotted in Fig. 8.6.

For a general case where the variable x can take values $x_1, x_2, \ldots, x_n$ and the variable y can take values $y_1, y_2, \ldots, y_m$, we have

$$\sum_i \sum_j P_{xy}(x_i, y_j) = 1 \tag{8.20}$$

This follows from the fact that the summation on the left is the probability of the union of all possible outcomes and must be unity (a certain event).

Conditional Probabilities

If x and y are two RVs, then the conditional probability of $x = x_i$ given $y = y_j$ is denoted by $P_{x|y}(x_i|y_j)$. Moreover,

$$\sum_i P_{x|y}(x_i|y_j) = \sum_j P_{y|x}(y_j|x_i) = 1 \tag{8.21}$$

This can be proved by observing that probabilities $P_{x|y}(x_i|y_j)$ are specified over the sample space corresponding to the condition $y = y_j$. Hence, $\sum_i P_{x|y}(x_i|y_j)$ is the probability of the union of all possible outcomes of x (under the condition $y = y_j$) and must be unity (a certain event). A similar argument applies to $\sum_j P_{y|x}(y_j|x_i)$. Also from Eq. (8.12), we have

$$P_{xy}(x_i, y_j) = P_{x|y}(x_i|y_j)P_y(y_j) = P_{y|x}(y_j|x_i)P_x(x_i) \tag{8.22}$$

Bayes' rule follows from Eq. (8.22). Also from Eq. (8.22), we have

$$\begin{aligned} \sum_i P_{xy}(x_i, y_j) &= \sum_i P_{x|y}(x_1|y_j)P_y(y_j) \\ &= P_y(y_j) \sum_i P_{x|y}(x_i|y_j) \\ &= P_y(y_j) \end{aligned} \tag{8.23a}$$

Similarly,

$$P_{\mathrm{x}}(x_i) = \sum_j P_{\mathrm{xy}}(x_i, y_j) \tag{8.23b}$$

The probabilities $P_{\mathrm{x}}(x_i)$ and $P_{\mathrm{y}}(y_j)$ are called **marginal probabilities**. Equations (8.23) show how to determine marginal probabilities from joint probabilities. Results of Eqs. (8.20) through (8.23) can be extended to more than two RVs.

Example 8.13 A binary symmetric channel (BSC) error probability is P_e. The probability of transmitting **1** is Q, and that of transmitting **0** is $1 - Q$ (Fig. 8.7). Determine the probabilities of receiving **1** and **0** at the receiver.

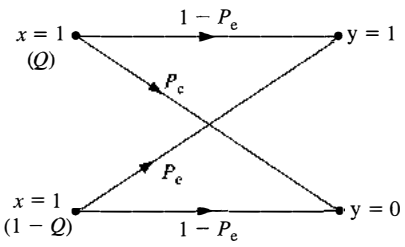


Figure 8.7 Binary symmetric channel (BSC).

If x and y are the transmitted digit and the received digit, respectively, then for a BSC,

$$P_{\mathrm{y|x}}(\mathbf{0}|\mathbf{1}) = P_{\mathrm{y|x}}(\mathbf{1}|\mathbf{0}) = P_e$$
$$P_{\mathrm{y|x}}(\mathbf{0}|\mathbf{0}) = P_{\mathrm{y|x}}(\mathbf{1}|\mathbf{1}) = 1 - P_e$$

Also,

$$P_{\mathrm{x}}(\mathbf{1}) = Q \qquad \text{and} \qquad P_{\mathrm{x}}(\mathbf{0}) = 1 - Q$$

We need to find $P_{\mathrm{y}}(\mathbf{1})$ and $P_{\mathrm{y}}(\mathbf{0})$. From the total probability theorem,

$$P_{\mathrm{y}}(y_j) = \sum_i P_{\mathrm{x}}(x_i) P_{\mathrm{y|x}}(y_j|x_i)$$

we find

$$\begin{aligned} P_{\mathrm{y}}(\mathbf{1}) &= P_{\mathrm{x}}(\mathbf{0}) P_{\mathrm{y|x}}(\mathbf{1}|\mathbf{0}) + P_{\mathrm{x}}(\mathbf{1}) P_{\mathrm{y|x}}(\mathbf{1}|\mathbf{1}) \\ &= (1 - Q)P_e + Q(1 - P_e) \end{aligned}$$

Similarly,

$$P_{\mathrm{y}}(\mathbf{0}) = (1 - Q)(1 - P_e) + QP_e$$

These answers seem almost obvious from Fig. 8.7.

Note that because of channel errors, the probability of receiving a digit **1** is not the same as that of transmitting **1**. The same is true of **0**.

Example 8.14 Over a certain binary communication channel, the symbol **0** is transmitted with probability 0.4 and **1** is transmitted with probability 0.6. It is given that $P(\epsilon|\mathbf{0}) = 10^{-6}$ and $P(\epsilon|\mathbf{1}) = 10^{-4}$, where $P(\epsilon|x_i)$ is the probability of detecting the error given that x_i is transmitted. Determine $P(\epsilon)$, the error probability of the channel.

If $P(\epsilon, x_i)$ is the joint probability that x_i is transmitted and it is detected wrongly, then the total probability theorem yields

$$\begin{aligned} P(\epsilon) &= \sum_i P(\epsilon|x_i)P(x_i) \\ &= P_{\mathrm{x}}(\mathbf{0})P(\epsilon|\mathbf{0}) + P_{\mathrm{x}}(\mathbf{1})P(\epsilon|\mathbf{1}) \\ &= 0.4(10^{-6}) + 0.6(10^{-4}) \\ &= 0.604(10^{-4}) \end{aligned}$$

Note that $P(\epsilon|\mathbf{0}) = 10^{-6}$ means that on the average, one out of 1 million received **0**s will be detected erroneously. Similarly, $P(\epsilon|\mathbf{1}) = 10^{-4}$ means that on the average, one out of 10,000 received **1**s will be in error. But $P(\epsilon) = 0.604(10^{-4})$ indicates that on the average, one out of $1/0.604(10^{-4}) \simeq 16{,}556$ digits (regardless of whether they are **1**s or **0**s) will be received in error.

Cumulative Distribution Function

The **cumulative distribution function (CDF)** $F_{\mathrm{x}}(x)$ of an RV x is the probability that x takes a value less than or equal to x; that is,

$$F_{\mathrm{x}}(x) = P(\mathrm{x} \leq x) \tag{8.24}$$

We can show that a CDF $F_{\mathrm{x}}(x)$ has the following four properties:

$$1.\ F_{\mathrm{x}}(x) \geq 0 \tag{8.25a}$$

$$2.\ F_{\mathrm{x}}(\infty) = 1 \tag{8.25b}$$

$$3.\ F_{\mathrm{x}}(-\infty) = 0 \tag{8.25c}$$

$$4.\ F_{\mathrm{x}}(x) \text{ is a nondecreasing function, that is,} \tag{8.25d}$$

$$F_{\mathrm{x}}(x_1) \leq F_{\mathrm{x}}(x_2) \text{ for } x_1 \leq x_2 \tag{8.25e}$$

The first property is obvious. The second and third properties are proved by observing that $F_{\mathrm{x}}(\infty) = P(\mathrm{x} \leq \infty)$ and $F_{\mathrm{x}}(-\infty) = P(\mathrm{x} \leq -\infty)$. To prove the fourth property, we have, from Eq. (8.24),

$$\begin{aligned} F_{\mathrm{x}}(x_2) &= P(\mathrm{x} \leq x_2) \\ &= P[(\mathrm{x} \leq x_1) \cup (x_1 < \mathrm{x} \leq x_2)] \end{aligned}$$

Because $x \leq x_1$ and $x_1 < x \leq x_2$ are disjoint, we have

$$\begin{aligned} F_x(x_2) &= P(x \leq x_1) + P(x_1 < x \leq x_2) \\ &= F_x(x_1) + P(x_1 < x \leq x_2) \end{aligned} \tag{8.26}$$

Because $P(x_1 < x \leq x_2)$ is nonnegative, the result follows.

Example 8.15 In an experiment, a trial consists of four successive tosses of a coin. If we define an RV x as the number of heads appearing in a trial, determine $P_x(x)$ and $F_x(x)$.

A total of 16 distinct equiprobable outcomes are listed in Example 8.4. Various probabilities can be readily determined by counting the outcomes pertaining to a given value of x. For example, only one outcome maps into x=0, whereas six outcomes map into x=2. Hence, $P_x(0) = 1/16$ and $P_x(2) = 6/16$. In the same way, we find

$$\begin{aligned} P_x(0) &= P_x(4) = 1/16 \\ P_x(1) &= P_x(3) = 4/16 = 1/4 \\ P_x(2) &= 6/16 = 3/8 \end{aligned}$$

The probabilities $P_x(x_i)$ and the corresponding CDF $F_x(x_i)$ are shown in Fig. 8.8.

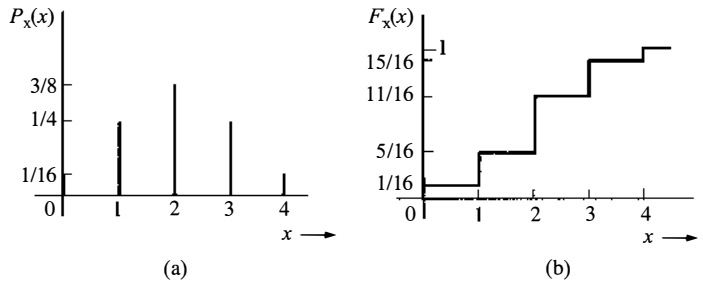


Figure 8.8 (a) Probabilities $P_x(x_i)$ and (b) the cumulative distribution function (CDF).

Continuous Random Variables

A continuous RV x can assume any value in a certain interval. In a continuum of any range, an uncountably infinite number of possible values exist, and $P_x(x_i)$, the probability that $x = x_i$, as one of the uncountably infinite values, is generally zero. Consider the case of a temperature **T** at a certain location. We may suppose that this temperature can assume any of a range of values. Thus, an infinite number of possible temperature values may prevail, and the probability that the random variable **T** will assume a certain value T_i is zero. The situation is somewhat similar to that described in Sec. 3.1 in connection with a continuously loaded beam (Fig. 3.5b). There is a loading along the beam at every point, but at any one point the load is zero. The meaningful measure in that case was the loading (or weight) not at a point, but over a finite interval. Similarly, for a continuous RV, the meaningful quantity is not the probability that $x = x_i$ but the probability that $x < x \leq x + \Delta x$. For such a measure, the CDF is eminently suited because the latter probability is simply $F_x(x + \Delta x) - F_x(x)$ [see Eq. (8.26)]. Hence, we begin our study of continuous RVs with the CDF.

Properties of the CDF [Eqs. (8.25) and (8.26)] derived earlier are general and are valid for continuous as well as discrete RVs.

Probability Density Function: From Eq. (8.26), we have

$$F_{\mathrm{x}}(x + \Delta x) = F_{\mathrm{x}}(x) + P(x < \mathrm{x} \leq x + \Delta x) \tag{8.27a}$$

If $\Delta x \to 0$, then we can also express $F_{\mathrm{x}}(x + \Delta x)$ via Taylor expansion as

$$F_{\mathrm{x}}(x + \Delta x) \simeq F_{\mathrm{x}}(x) + \frac{dF_{\mathrm{x}}(x)}{dx}\Delta x \tag{8.27b}$$

From Eqs. (8.27), it follows that as $\Delta x \to 0$,

$$\frac{dF_{\mathrm{x}}(x)}{dx}\Delta x = P(x < \mathrm{x} \leq x + \Delta x) \tag{8.28}$$

We designated the derivative of $F_{\mathrm{x}}(x)$ with respect to x by $p_{\mathrm{x}}(x)$ (Fig. 8.9),

$$\frac{dF_{\mathrm{x}}(x)}{dx} = p_{\mathrm{x}}(x) \tag{8.29}$$

The function $p_{\mathrm{x}}(x)$ is called the **probability density function (PDF)** of the RV x. It follows from Eq. (8.28) that the probability of observing the RV x in the interval $(x,\ x + \Delta x)$ is $p_{\mathrm{x}}(x)\Delta x$ $(\Delta x \to 0)$. This is the area under the PDF $p_{\mathrm{x}}(x)$ over the interval Δx, as shown in Fig. 8.9b.

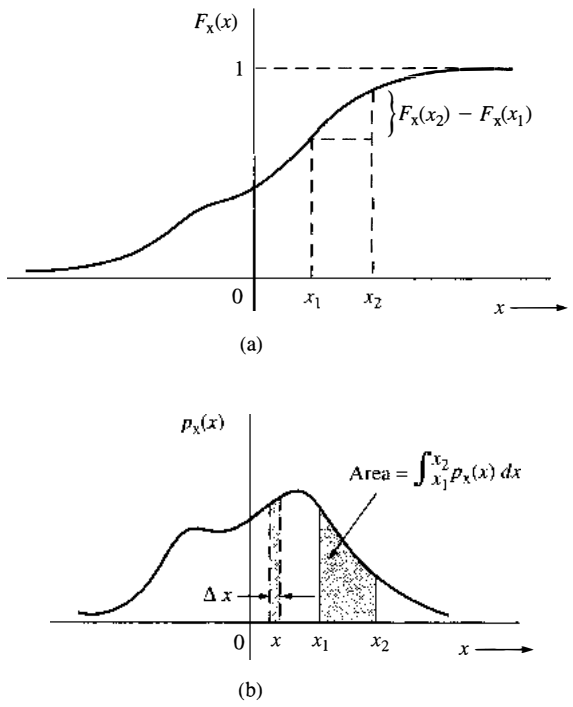


Figure 8.9 (a) Cumulative distribution function (CDF). (b) Probability density function (PDF).

From Eq. (8.29), we can see that

$$F_{\mathrm{x}}(x) = \int_{-\infty}^{x} p_{\mathrm{x}}(u)\, du \tag{8.30}$$

Here we use the fact that $F_{\mathrm{x}}(-\infty) = 0$. We also have from Eq. (8.26)

$$\begin{aligned} P(x_1 < \mathrm{x} \leq x_2) &= F_{\mathrm{x}}(x_2) - F_{\mathrm{x}}(x_1) \\ &= \int_{-\infty}^{x_2} p_{\mathrm{x}}(x)\, dx - \int_{-\infty}^{x_1} p_{\mathrm{x}}(x)\, dx \\ &= \int_{x_1}^{x_2} p_{\mathrm{x}}(x)\, dx \end{aligned} \tag{8.31}$$

Thus, the probability of observing x in any interval (x_1, x_2) is given by the area under the PDF $p_{\mathrm{x}}(x)$ over the interval (x_1, x_2), as shown in Fig. 8.9b. Compare this with a continuously loaded beam (Fig. 3.5b), where the weight over any interval was given by an integral of the loading density over the interval.

Because $F_{\mathrm{x}}(\infty) = 1$, we have

$$\int_{-\infty}^{\infty} p_{\mathrm{x}}(x)\, dx = 1 \tag{8.32}$$

This also follows from the fact that the integral in Eq. (8.32) represents the probability of observing x in the interval $(-\infty, \infty)$. Every PDF must satisfy the condition in Eq. (8.32). It is also evident that the PDF must not be negative, that is,

$$p_{\mathrm{x}}(x) \geq 0$$

Although it is true that the probability of an impossible event is **0** and that of a certain event is **1**, the converse is not true. An event whose probability is **0** is not necessarily an impossible event, and an event with a probability of **1** is not necessarily a certain event. This may be illustrated by the following example. The temperature T of a certain city on a summer day is an RV taking on any value in the range of 5 to 50°C. Because the PDF $p_{\mathrm{T}}(T)$ is continuous, the probability that T = 34.56, for example, is zero. But this is not an impossible event. Similarly, the probability that T takes on any value but 34.56 is **1**, although this is not a certain event. In fact, a continuous RV x takes every value in a certain range. Yet $p_{\mathrm{x}}(x)$, the probability that $\mathrm{x} = x$, is zero for every x in that range.

We can also determine the PDF $p_{\mathrm{x}}(x)$ for a discrete random variable. Because the CDF $F_{\mathrm{x}}(x)$ for the discrete case is always a sequence of step functions (Fig. 8.8), the PDF (the derivative of the CDF) will consist of a train of positive impulses. If an RV x takes values $x_1, x_2, \ldots, x_n$ with probabilities $a_1, a_2, \ldots, a_n$, respectively, then

$$F_{\mathrm{x}}(x) = a_1 u(x - x_1) + a_2 u(x - x_2) + \cdots + a_n u(x - x_n) \tag{8.33a}$$

This can be easily verified from Example 8.15 (Fig. 8.8). Hence,

$$\begin{aligned} p_{\mathrm{x}}(x) &= a_1\, \delta(x - x_1) + a_2\, \delta(x - x_2) + \cdots + a_n\, \delta(x - x_n) \\ &= \sum_{r=1}^{n} a_r\, \delta(x - x_r) \end{aligned} \tag{8.33b}$$

It is, of course, possible to have a mixed case, where a PDF may have a continuous part and an impulsive part (see Prob. 8.2-4).

The Gaussian Random Variable

Consider a PDF (Fig. 8.10)

$$p_X(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \tag{8.34}$$

This is a case of the well-known standard **Gaussian**, or **normal**, probability density. It has zero mean and unit variance. This function was named after the famous mathematician Carl Friedrich Gauss.

The CDF $F_X(x)$ in this case is

$$F_X(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-x^2/2}\, dx$$

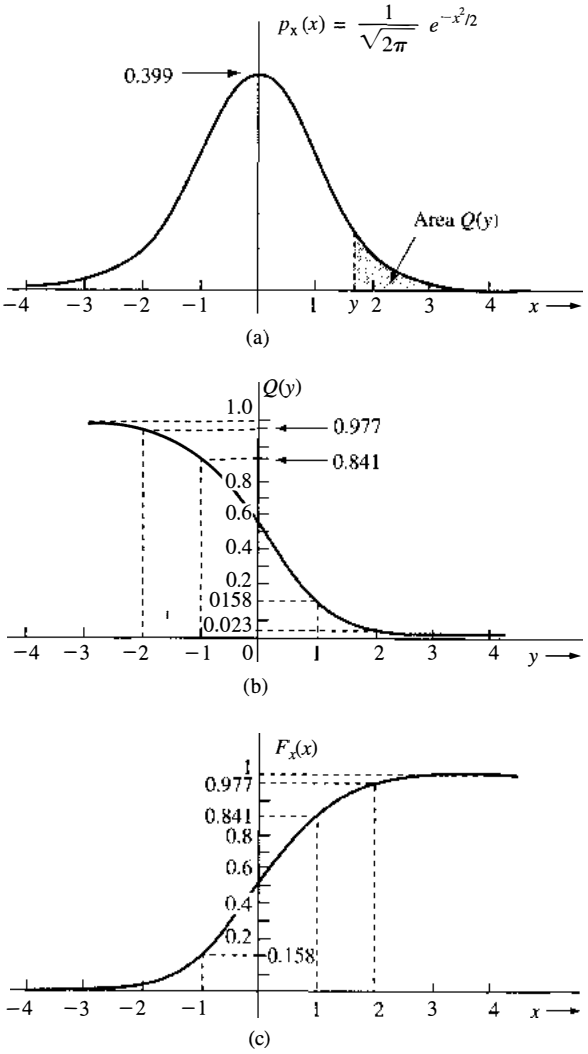


Figure 8.10 (a) Gaussian PDF. (b) Function $Q(y)$. (c) CDF of the Gaussian PDF.

This integral cannot be evaluated in a closed form and must be computed numerically. It is convenient to use the function $Q(.)$, defined as[2]

$$Q(y) \triangleq \frac{1}{\sqrt{2\pi}} \int_y^\infty e^{-x^2/2}\, dx \tag{8.35}$$

The area under $p_X(x)$ from y to ∞ (shaded in Fig. 8.10a) is* $Q(y)$. From the symmetry of $p_X(x)$ about the origin, and the fact that the total area under $p_X(x) = 1$, it follows that

$$Q(-y) = 1 - Q(y) \tag{8.36}$$

Observe that for the PDF in Fig. 8.10a, the CDF is given by (Fig. 8.10c)

$$F_X(x) = 1 - Q(x) \tag{8.37}$$

The function $Q(x)$ is tabulated in Table 8.2 (see also later: Fig. 8.12d). This function is widely tabulated and can be found in most of the standard mathematical tables.[2, 3] It can be shown that,[4]

$$Q(x) \simeq \frac{1}{x\sqrt{2\pi}} e^{-x^2/2} \qquad \text{for } x \gg 1 \tag{8.38a}$$

For example, when $x = 2$, the error in this approximation is 18.7%. But for $x = 4$ it is 10.4% and for $x = 6$ it is 2.3%.

A much better approximation to $Q(x)$ is

$$Q(x) \simeq \frac{1}{x\sqrt{2\pi}} \left(1 - \frac{0.7}{x^2}\right) e^{-x^2/2} \qquad x > 2 \tag{8.38b}$$

The error in this approximation is just within 1% for $x > 2.15$. For larger values of x the error approaches 0.

A more general Gaussian density function has two parameters (m, σ) and is (Fig. 8.11)

$$p_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-m)^2/2\sigma^2} \tag{8.39}$$

For this case,

$$F_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-(x-m)^2/2\sigma^2}\, dx$$

* The function $Q(x)$ is closely related to functions erf (x) and erfc (x),

$$\text{erfc}\,(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-y^2}\, dy = 2Q(x\sqrt{2})$$

Therefore,

$$Q(x) = \frac{1}{2}\,\text{erfc}\left(\frac{x}{\sqrt{2}}\right) = \frac{1}{2}\left[1 - \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$$

TABLE 8.2[3]
$Q(x)$

x	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0000	.5000	.4960	.4920	.4880	.4840	.4801	.4761	.4721	.4681	.4641
.1000	.4602	.4562	.4522	.4483	.4443	.4404	.4364	.4325	.4286	.4247
.2000	.4207	.4168	.4129	.4090	.4052	.4013	.3974	.3936	.3897	.3859
.3000	.3821	.3783	.3745	.3707	.3669	.3632	.3594	.3557	.3520	.3483
.4000	.3446	.3409	.3372	.3336	.3300	.3264	.3228	.3192	.3156	.3121
.5000	.3085	.3050	.3015	.2981	.2946	.2912	.2877	.2843	.2810	.2776
.6000	.2743	.2709	.2676	.2643	.2611	.2578	.2546	.2514	.2483	.2451
.7000	.2420	.2389	.2358	.2327	.2296	.2266	.2236	.2206	.2177	.2148
.8000	.2119	.2090	.2061	.2033	.2005	.1977	.1949	.1922	.1894	.1867
.9000	.1841	.1814	.1788	.1762	.1736	.1711	.1685	.1660	.1635	.1611
1.000	.1587	.1562	.1539	.1515	.1492	.1469	.1446	.1423	.1401	.1379
1.100	.1357	.1335	.1314	.1292	.1271	.1251	.1230	.1210	.1190	.1170
1.200	.1151	.1131	.1112	.1093	.1075	.1056	.1038	.1020	.1003	.9853E-01
1.300	.9680E-01	.9510E-01	.9342E-01	.9176E-01	.9012E-01	.8851E-01	.8691E-01	.8534E-01	.8379E-01	.8226E-01
1.400	.8076E-01	.7927E-01	.7780E-01	.7636E-01	.7493E-01	.7353E-01	.7215E-01	.7078E-01	.6944E-01	.6811E-01
1.500	.6681E-01	.6552E-01	.6426E-01	.6301E-01	.6178E-01	.6057E-01	.5938E-01	.5821E-01	.5705E-01	.5592E-01
1.600	.5480E-01	.5370E-01	.5262E-01	.5155E-01	.5050E-01	.4947E-01	.4846E-01	.4746E-01	.4648E-01	.4551E-01
1.700	.4457E-01	.4363E-01	.4272E-01	.4182E-01	.4093E-01	.4006E-01	.3920E-01	.3836E-01	.3754E-01	.3673E-01
1.800	.3593E-01	.3515E-01	.3438E-01	.3362E-01	.3288E-01	.3216E-01	.3144E-01	.3074E-01	.3005E-01	.2938E-01
1.900	.2872E-01	.2807E-01	.2743E-01	.2680E-01	.2619E-01	.2559E-01	.2500E-01	.2442E-01	.2385E-01	.2330E-01
2.000	.2275E-01	.2222E-01	.2169E-01	.2118E-01	.2068E-01	.2018E-01	.1970E-01	.1923E-01	.1876E-01	.1831E-01
2.100	.1786E-01	.1743E-01	.1700E-01	.1659E-01	.1618E-01	.1578E-01	.1539E-01	.1500E-01	.1463E-01	.1426E-01
2.200	.1390E-01	.1355E-01	.1321E-01	.1287E-01	.1255E-01	.1222E-01	.1191E-01	.1160E-01	.1130E-01	.1101E-01
2.300	.1072E-01	.1044E-01	.1017E-01	.9903E-02	.9642E-02	.9387E-02	.9137E-02	.8894E-02	.8656E-02	.8424E-02
2.400	.8198E-02	.7976E-02	.7760E-02	.7549E-02	.7344E-02	.7143E-02	.6947E-02	.6756E-02	.6569E-02	.6387E-02
2.500	.6210E-02	.6037E-02	.5868E-02	.5703E-02	.5543E-02	.5386E-02	.5234E-02	.5085E-02	.4940E-02	.4799E-02
2.600	.4661E-02	.4527E-02	.4396E-02	.4269E-02	.4145E-02	.4025E-02	.3907E-02	.3793E-02	.3681E-02	.3573E-02
2.700	.3467E-02	.3364E-02	.3264E-02	.3167E-02	.3072E-02	.2980E-02	.2890E-02	.2803E-02	.2718E-02	.2635E-02
2.800	.2555E-02	.2477E-02	.2401E-02	.2327E-02	.2256E-02	.2186E-02	.2118E-02	.2052E-02	.1988E-02	.1926E-02
2.900	.1866E-02	.1807E-02	.1750E-02	.1695E-02	.1641E-02	.1589E-02	.1538E-02	.1489E-02	.1441E-02	.1395E-02
3.000	.1350E-02	.1306E-02	.1264E-02	.1223E-02	.1183E-02	.1144E-02	.1107E-02	.1070E-02	.1035E-02	.1001E-02
3.100	.9676E-03	.9354E-03	.9043E-03	.8740E-03	.8447E-03	.8164E-03	.7888E-03	.7622E-03	.7364E-03	.7114E-03
3.200	.6871E-03	.6637E-03	.6410E-03	.6190E-03	.5976E-03	.5770E-03	.5571E-03	.5377E-03	.5190E-03	.5009E-03
3.300	.4834E-03	.4665E-03	.4501E-03	.4342E-03	.4189E-03	.4041E-03	.3897E-03	.3758E-03	.3624E-03	.3495E-03
3.400	.3369E-03	.3248E-03	.3131E-03	.3018E-03	.2909E-03	.2802E-03	.2701E-03	.2602E-03	.2507E-03	.2415E-03
3.500	.2326E-03	.2241E-03	.2158E-03	.2078E-03	.2001E-03	.1926E-03	.1854E-03	.1785E-03	.1718E-03	.1653E-03
3.600	.1591E-03	.1531E-03	.1473E-03	.1417E-03	.1363E-03	.1311E-03	.1261E-03	.1213E-03	.1166E-03	.1121E-03
3.700	.1078E-03	.1036E-03	.9961E-04	.9574E-04	.9201E-04	.8842E-04	.8496E-04	.8162E-04	.7841E-04	.7532E-04
3.800	.7235E-04	.6948E-04	.6673E-04	.6407E-04	.6152E-04	.5906E-04	.5669E-04	.5442E-04	.5223E-04	.5012E-04
3.900	.4810E-04	.4615E-04	.4427E-04	.4247E-04	.4074E-04	.3908E-04	.3747E-04	.3594E-04	.3446E-04	.3304E-04
4.000	.3167E-04	.3036E-04	.2910E-04	.2789E-04	.2673E-04	.2561E-04	.2454E-04	.2351E-04	.2252E-04	.2157E-04
4.100	.2066E-04	.1978E-04	.1894E-04	.1814E-04	.1737E-04	.1662E-04	.1591E-04	.1523E-04	.1458E-04	.1395E-04
4.200	.1335E-04	.1277E-04	.1222E-04	.1168E-04	.1118E-04	.1069E-04	.1022E-04	.9774E-05	.9345E-05	.8934E-05
4.300	.8540E-05	.8163E-05	.7801E-05	.7455E-05	.7124E-05	.8807E-05	.6503E-05	.6212E-05	.5934E-05	.5668E-05
4.400	.5413E-05	.5169E-05	.4935E-05	.4712E-05	.4498E-05	.4294E-05	.4098E-05	.3911E-05	.3732E-05	.3561E-05
4.500	.3398E-05	.3241E-05	.3092E-05	.2949E-05	.2813E-05	.2682E-05	.2558E-05	.2439E-05	.2325E-05	.2216E-05
4.600	.2112E-05	.2013E-05	.1919E-05	.1828E-05	.1742E-05	.1660E-05	.1581E-05	.1506E-05	.1434E-05	.1366E-05
4.700	.1301E-05	.1239E-05	.1179E-05	.1123E-05	.1069E-05	.1017E-05	.9680E-06	.9211E-06	.8765E-06	.8339E-06
4.800	.7933E-06	.7547E-06	.7178E-06	.6827E-06	.6492E-06	.6173E-06	.5869E-06	.5580E-06	.5304E-06	.5042E-06
4.900	.4792E-06	.4554E-06	.4327E-06	.4111E-06	.3906E-06	.3711E-06	.3525E-06	.3448E-06	.3179E-06	.3019E-06
5.000	.2867E-06	.2722E-06	.2584E-06	.2452E-06	.2328E-06	.2209E-06	.2096E-06	.1989E-06	.1887E-06	.1790E-06
5.100	.1698E-06	.1611E-06	.1528E-06	.1449E-06	.1374E-06	.1302E-06	.1235E-06	.1170E-06	.1109E-06	.1051E-06

(*continued*)

TABLE 8.2
Continued

x	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
5.200	.9964E-07	.9442E-07	.8946E-07	.8476E-07	.8029E-07	.7605E-07	.7203E-07	.6821E-07	.6459E-07	.6116E-07
5.300	.5790E-07	.5481E-07	.5188E-07	.4911E-07	.4647E-07	.4398E-07	.4161E-07	.3937E-07	.3724E-07	.3523E-07
5.400	.3332E-07	.3151E-07	.2980E-07	.2818E-07	.2664E-07	.2518E-07	.2381E-07	.2250E-07	.2127E-07	.2010E-07
5.500	.1899E-07	.1794E-07	.1695E-07	.1601E-07	.1512E-07	.1428E-07	.1349E-07	.1274E-07	.1203E-07	.1135E-07
5.600	.1072E-07	.1012E-07	.9548E-08	.9010E-08	.8503E-08	.8022E-08	.7569E-08	.7140E-08	.6735E-08	.6352E-08
5.700	.5990E-08	.5649E-08	.5326E-08	.5022E-08	.4734E-08	.4462E-08	.4206E-08	.3964E-08	.3735E-08	.3519E-08
5.800	.3316E-08	.3124E-08	.2942E-08	.2771E-08	.2610E-08	.2458E-08	.2314E-08	.2179E-08	.2051E-08	.1931E-08
5.900	.1818E-08	.1711E-08	.1610E-08	.1515E-08	.1425E-08	.1341E-08	.1261E-08	.1186E-08	.1116E-08	.1049E-08
6.000	.9866E-09	.9276E-09	.8721E-09	.8198E-09	.7706E-09	.7242E-09	.6806E-09	.6396E-09	.6009E-09	.5646E-09
6.100	.5303E-09	.4982E-09	.4679E-09	.4394E-09	.4126E-09	.3874E-09	.3637E-09	.3414E-09	.3205E-09	.3008E-09
6.200	.2823E-09	.2649E-09	.2486E-09	.2332E-09	.2188E-09	.2052E-09	.1925E-09	.1805E-09	.1692E-09	.1587E-09
6.300	.1488E-09	.1395E-09	.1308E-09	.1226E-09	.1149E-09	.1077E-09	.1009E-09	.9451E-10	.8854E-10	.8294E-10
6.400	.7769E-10	.7276E-10	.6814E-10	.6380E-10	.5974E-10	.5593E-10	.5235E-10	.4900E-10	.4586E-10	.4292E-10
6.500	.4016E-10	.3758E-10	.3515E-10	.3288E-10	.3077E-10	.2877E-10	.2690E-10	.2516E-10	.2352E-10	.2199E-10
6.600	.2056E-10	.1922E-10	.1796E-10	.1678E-10	.1568E-10	.1465E-10	.1369E-10	.1279E-10	.1195E-10	.1116E-10
6.700	.1042E-10	.9731E-11	.9086E-11	.8483E-11	.7919E-11	.7392E-11	.6900E-11	.6439E-11	.6009E-11	.5607E-11
6.800	.5231E-11	.4880E-11	.4552E-11	.4246E-11	.3960E-11	.3692E-11	.3443E-11	.3210E-11	.2993E-11	.2790E-11
6.900	.2600E-11	.2423E-11	.2258E-11	.2104E-11	.1960E-11	.1826E-11	.1701E-11	.1585E-11	.1476E-11	.1374E-11
7.000	.1280E-11	.1192E-11	.1109E-11	.1033E-11	.9612E-12	.8946E-12	.8325E-12	.7747E-12	.7208E-12	.6706E-12
7.100	.6238E-12	.5802E-12	.5396E-12	.5018E-12	.4667E-12	.4339E-12	.4034E-12	.3750E-12	.3486E-12	.3240E-12
7.200	.3011E-12	.2798E-12	.2599E-12	.2415E-12	.2243E-12	.2084E-12	.1935E-12	.1797E-12	.1669E-12	.1550E-12
7.300	.1439E-12	.1336E-12	.1240E-12	.1151E-12	.1068E-12	.9910E-13	.9196E-13	.8531E-13	.7914E-13	.7341E-13
7.400	.6809E-13	.6315E-13	.5856E-13	.5430E-13	.5034E-13	.4667E-13	.4326E-13	.4010E-13	.3716E-13	.3444E-13
7.500	.3191E-13	.2956E-13	.2739E-13	.2537E-13	.2350E-13	.2176E-13	.2015E-13	.1866E-13	.1728E-13	.1600E-13
7.600	.1481E-13	.1370E-13	.1268E-13	.1174E-13	.1086E-13	.1005E-13	.9297E-14	.8600E-14	.7954E-14	.7357E-14
7.700	.6803E-14	.6291E-14	.5816E-14	.5377E-14	.4971E-14	.4595E-14	.4246E-14	.3924E-14	.3626E-14	.3350E-14
7.800	.3095E-14	.2859E-14	.2641E-14	.2439E-14	.2253E-14	.2080E-14	.1921E-14	.1773E-14	.1637E-14	.1511E-14
7.900	.1395E-14	.1287E-14	.1188E-14	.1096E-14	.1011E-14	.9326E-15	.8602E-15	.7934E-15	.7317E-15	.6747E-15
8.000	.6221E-15	.5735E-15	.5287E-15	.4874E-15	.4492E-15	.4140E-15	.3815E-15	.3515E-15	.3238E-15	.2983E-15
8.100	.2748E-15	.2531E-15	.2331E-15	.2146E-15	.1976E-15	.1820E-15	.1675E-15	.1542E-15	.1419E-15	.1306E-15
8.200	.1202E-15	.1106E-15	.1018E-15	.9361E-16	.8611E-16	.7920E-16	.7284E-16	.6698E-16	.6159E-16	.5662E-16
8.300	.5206E-16	.4785E-16	.4398E-16	.4042E-16	.3715E-16	.3413E-16	.3136E-16	.2881E-16	.2646E-16	.2431E-16
8.400	.2232E-16	.2050E-16	.1882E-16	.1728E-16	.1587E-16	.1457E-16	.1337E-16	.1227E-16	.1126E-16	.1033E-16
8.500	.9480E-17	.8697E-17	.7978E-17	.7317E-17	.6711E-17	.6154E-17	.5643E-17	.5174E-17	.4744E-17	.4348E-17
8.600	.3986E-17	.3653E-17	.3348E-17	.3068E-17	.2811E-17	.2575E-17	.2359E-17	.2161E-17	.1979E-17	.1812E-17
8.700	.1659E-17	.1519E-17	.1391E-17	.1273E-17	.1166E-17	.1067E-17	.9763E-18	.8933E-18	.8174E-18	.7478E-18
8.800	.6841E-18	.6257E-18	.5723E-18	.5234E-18	.4786E-18	.4376E-18	.4001E-18	.3657E-18	.3343E-18	.3055E-18
8.900	.2792E-18	.2552E-18	.2331E-18	.2130E-18	.1946E-18	.1777E-18	.1623E-18	.1483E-18	.1354E-18	.1236E-18
9.000	.1129E-18	.1030E-18	.9404E-19	.8584E-19	.7834E-19	.7148E-19	.6523E-19	.5951E-19	.5429E-19	.4952E-19
9.100	.4517E-19	.4119E-19	.3756E-19	.3425E-19	.3123E-19	.2847E-19	.2595E-19	.2365E-19	.2155E-19	.1964E-19
9.200	.1790E-19	.1631E-19	.1486E-19	.1353E-19	.1232E-19	.1122E-19	.1022E-19	.9307E-20	.8474E-20	.7714E-20
9.300	.7022E-20	.6392E-20	.5817E-20	.5294E-20	.4817E-20	.4382E-20	.3987E-20	.3627E-20	.3299E-20	.3000E-20
9.400	.2728E-20	.2481E-20	.2255E-20	.2050E-20	.1864E-20	.1694E-20	.1540E-20	.1399E-20	.1271E-20	.1155E-20
9.500	.1049E-20	.9533E-21	.8659E-21	.7864E-21	.7142E-21	.6485E-21	.5888E-21	.5345E-21	.4852E-21	.4404E-21
9.600	.3997E-21	.3627E-21	.3292E-21	.2986E-21	.2709E-21	.2458E-21	.2229E-21	.2022E-21	.1834E-21	.1663E-21
9.700	.1507E-21	.1367E-21	.1239E-21	.1123E-21	.1018E-21	.9223E-22	.8358E-22	.7573E-22	.6861E-22	.6215E-22
9.800	.5629E-22	.5098E-22	.4617E-22	.4181E-22	.3786E-22	.3427E-22	.3102E-22	.2808E-22	.2542E-22	.2300E-22
9.900	.2081E-22	.1883E-22	.1704E-22	.1541E-22	.1394E-22	.1261E-22	.1140E-22	.1031E-22	.9323E-23	.8429E-23
10.00	.7620E-23	.6888E-23	.6225E-23	.5626E-23	.5084E-23	.4593E-23	.4150E-23	.3749E-23	.3386E-23	.3058E-23

Notes: (1) E-01 should be read as $\times 10^{-1}$; E-02 should be read as $\times 10^{-2}$, and so on.
(2) This table lists $Q(x)$ for x in the range of 0 to 10 in the increments of 0.01. To find $Q(5.36)$, for example, look up the row starting with $x = 5.3$. The sixth entry in this row (under 0.06) is the desired value 0.4161×10^{-7}.

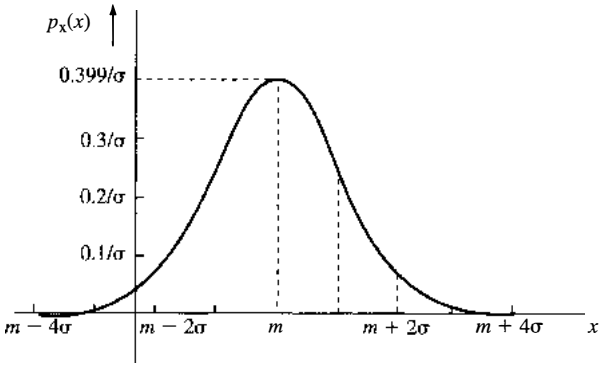


Figure 8.11 Gaussian PDF with mean m and variance σ^2.

Letting $(x-m)/\sigma = z$,

$$F_x(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{(x-m)/\sigma} e^{-z^2/2}\, dz$$

$$= 1 - Q\left(\frac{x-m}{\sigma}\right) \tag{8.40a}$$

Therefore,

$$P(x \le x) = 1 - Q\left(\frac{x-m}{\sigma}\right) \tag{8.40b}$$

and

$$P(x > x) = Q\left(\frac{x-m}{\sigma}\right) \tag{8.40c}$$

The Gaussian PDF is perhaps the most important PDF in the field of communications. The majority of the noise processes observed in practice are Gaussian. The amplitude n of a Gaussian noise signal is an RV with a Gaussian PDF. This means the probability of observing n in an interval $(n,\ n+\Delta n)$ is $p_n(n)\Delta n$, where $p_n(n)$ is of the form in Eq. (8.39) [with $m = 0$].

Example 8.16 Threshold Detection

Over a certain binary channel, messages m= **0** and **1** are transmitted with equal probability by using a positive and a negative pulse, respectively. The received pulse corresponding to **1** is $p(t)$, shown in Fig. 8.12a, and the received pulse corresponding to **0** is $-p(t)$. Let the peak amplitude of $p(t)$ be A_p at $t = T_p$. Because of the channel noise $n(t)$, the received pulses will be (Fig. 8.12c)

$$\pm p(t) + n(t)$$

To detect the pulses at the receiver, each pulse is sampled at its peak amplitude. In the absence of noise, the sampler output is either A_p (for m=**1**) or $-A_p$ (for m=**0**). Because of the channel noise, the sampler output is $\pm A_p + n$, where n, the noise amplitude at the sampling instant (Fig. 8.12b), is an RV. For Gaussian noise, the PDF of n is (Fig. 8.12b)

$$p_n(n) = \frac{1}{\sigma_n\sqrt{2\pi}} e^{-n^2/2\sigma_n^2} \tag{8.41}$$

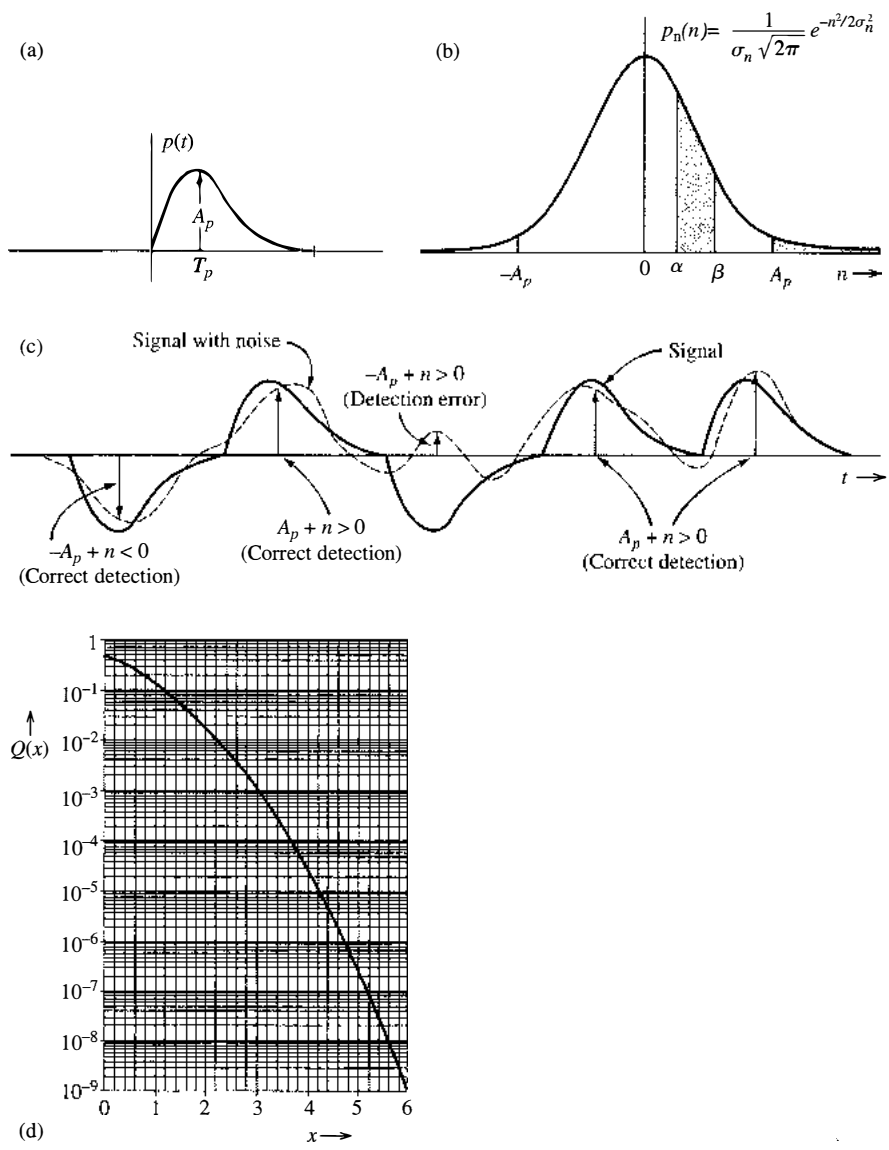


Figure 8.12
Error probability in threshold detection: (a) transmitted pulse; (b) noise PDF; (c) received pulses with noise; (d) detection error probability.

Because of the symmetry of the situation, the optimum detection threshold is zero; that is, the received pulse is detected as a **1** or a **0**, depending on whether the sample value is positive or negative.

Because noise amplitudes range from $-\infty$ to ∞, the sample value $-A_p + \text{n}$ can occasionally be positive, causing the received **0** to be read as **1** (see Fig. 8.12b). Similarly, $A_p + \text{n}$ can occasionally be negative, causing the received **1** to be read as **0**. If **0** is transmitted, it will be detected as **1** if $-A_p + \text{n} > 0$, that is, if $\text{n} > A_p$.

If $P(\epsilon|\mathbf{0})$ is the error probability given that **0** is transmitted, then

$$P(\epsilon|\mathbf{0}) = P(\text{n} > A_p)$$

Because $P(\mathrm{n} > A_p)$ is the shaded area in Fig. 8.12b to the right of A_p, from Eq. (8.40c) [with m = **0**] it follows that

$$P(\epsilon|\mathbf{0}) = Q\left(\frac{A_p}{\sigma_\mathrm{n}}\right) \tag{8.42a}$$

Similarly,

$$\begin{aligned} P(\epsilon|\mathbf{1}) &= P(\mathrm{n} < -A_p) \\ &= Q\left(\frac{A_p}{\sigma_\mathrm{n}}\right) = P(\epsilon|\mathbf{0}) \end{aligned} \tag{8.42b}$$

and

$$\begin{aligned} P_e &= \sum_i P(\epsilon,\, m_i) \\ &= \sum_i P(m_i)P(\epsilon|m_i) \\ &= Q\left(\frac{A_p}{\sigma_n}\right)\sum_i P(m_i) \\ &= Q\left(\frac{A_p}{\sigma_\mathrm{n}}\right) \end{aligned} \tag{8.42c}$$

The error probability P_e can be found from Fig. 8.12d.

Joint Distribution

For two RVs x and y, we define a CDF $F_{\mathrm{xy}}(x,\, y)$ as follows:

$$F_{\mathrm{xy}}(x,\, y) \triangleq P(\mathrm{x} \le x \text{ and } \mathrm{y} \le y) \tag{8.43}$$

and the joint PDF $p_{\mathrm{xy}}(x,\, y)$ as

$$p_{\mathrm{xy}}(x,\, y) = \frac{\partial^2}{\partial x\, \partial y} F_{\mathrm{xy}}(x,\, y) \tag{8.44}$$

Arguing along lines similar to those used for a single variable, we can show that as $\Delta x \to 0$ and $\Delta y \to 0$

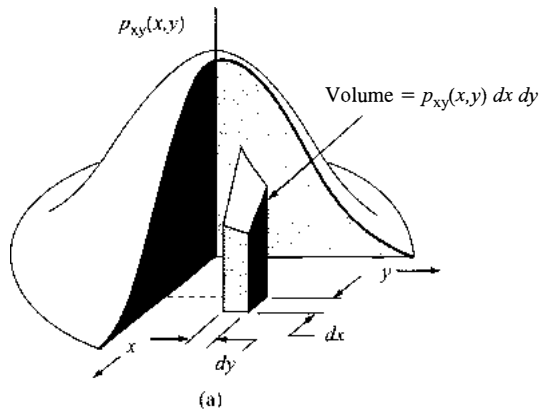
$$p_{\mathrm{xy}}(x,\, y)\Delta x\, \Delta y = P(x < \mathrm{x} \le x + \Delta x,\, y < \mathrm{y} \le y + \Delta y) \tag{8.45}$$

Hence, the probability of observing the variables x in the interval $(x,\, x + \Delta x)$ and y in the interval $(y,\, y + \Delta y)$ jointly is given by the volume under the joint PDF $p_{\mathrm{xy}}(x,\, y)$ over the region bounded by $(x,\, x + \Delta x)$ and $(y,\, y + \Delta y)$, as shown in Fig. 8.13a.

From Eq. (8.45), it follows that

$$P(x_1 < x \le x_2,\, y_1 < y \le y_2) = \int_{x_1}^{x_2}\int_{y_1}^{y_2} p_{xy}(x,\, y)\, dx\, dy \tag{8.46}$$

Thus, the probability of jointly observing x in the interval (x_1, x_2) and y in the interval $(y_1,\, y_2)$ is the volume under the PDF over the region bounded by $(x_1,\, x_2)$ and $(y_1,\, y_2)$.



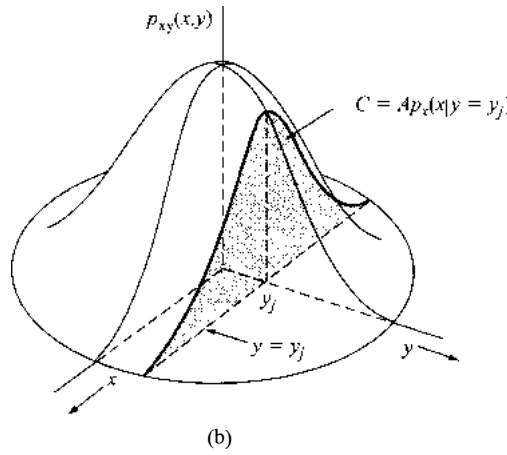


Figure 8.13
(a) Joint PDF.
(b) Conditional PDF.

The event of observing x in the interval $(-\infty, \infty)$ and observing y in the interval $(-\infty, \infty)$ is a certainty. Hence,

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{xy}(x, y)\,dx\,dy = 1 \tag{8.47}$$

Thus, the total volume under the joint PDF must be unity.

When we are dealing with two RVs x and y, the individual probability densities $p_x(x)$ and $p_y(y)$ can be obtained from the joint density $p_{xy}(x, y)$. These individual densities are also called **marginal densities**. To obtain these densities, we note that $p_x(x)\,\Delta x$ is the probability of observing x in the interval $(x, x + \Delta x)$. The value of y may lie anywhere in the interval $(-\infty, \infty)$. Hence,

$$\begin{aligned}
\lim_{\Delta x\to 0} p_x(x)\Delta x &= \lim_{\Delta x\to 0} \text{Probability } (x < x \le x + \Delta x,\ -\infty < y \le \infty) \\
&= \lim_{\Delta x\to 0}\int_{x}^{x+\Delta x}\int_{-\infty}^{\infty} p_{xy}(x, y)\,dx\,dy \\
&= \lim_{\Delta x\to 0}\int_{-\infty}^{\infty} p_{xy}(x, y)\,dy\int_{x}^{x+\Delta x} dx \\
&= \lim_{\Delta x\to 0}\Delta x\int_{-\infty}^{\infty} p_{xy}(x, y)\,dy
\end{aligned}$$

The last two steps follow from the fact that $p_{xy}(x, y)$ is constant over $(x, x + \Delta x)$ because $\Delta x \to 0$. Therefore,

$$p_x(x) = \int_{-\infty}^{\infty} p_{xy}(x, y)\, dy \tag{8.48a}$$

Similarly,

$$p_y(y) = \int_{-\infty}^{\infty} p_{xy}(x, y)\, dx \tag{8.48b}$$

In terms of the CDF, we have

$$F_y(y) = F_{xy}(\infty, y) \tag{8.49a}$$

$$F_x(x) = F_{xy}(x, \infty) \tag{8.49b}$$

These results may be generalized for multiple RVs $x_1, x_2, \ldots, x_n$.

Conditional Densities

The concept of conditional probabilities can be extended to the case of continuous RVs. We define the conditional PDF $p_{x|y}(x|y_j)$ as the PDF of x given that y has the value y_j. This is equivalent to saying that $p_{x|y}(x|y_j)\Delta x$ is the probability of observing x in the range $(x, x+\Delta x)$, given that $y = y_j$. The probability density $p_{x|y}(x|y_j)$ is the intersection of the plane $y = y_j$ with the joint PDF $p_{xy}(x, y)$ (Fig. 8.13b). Because every PDF must have unit area, however, we must normalize the area under the intersection curve C to unity to get the desired PDF. Hence, C is $Ap_{x|y}(x|y)$, where A is the area under C. An extension of the results derived for the discrete case yields

$$p_{x|y}(x|y)p_y(y) = p_{xy}(x, y) \tag{8.50a}$$

$$p_{y|x}(y|x)p_x(x) = p_{xy}(x, y) \tag{8.50b}$$

and

$$p_{x|y}(x|y) = \frac{p_{y|x}(y|x)p_x(x)}{p_y(y)} \tag{8.51a}$$

Equation (8.51a) is Bayes' rule for continuous RVs. When we have mixed variables (i.e., discrete and continuous), the mixed form of Bayes' rule is

$$P_{x|y}(x|y)p_y(y) = P_x(x)p_{y|x}(y|x) \tag{8.51b}$$

where x is a discrete RV and y is a continuous RV.*

Note that $p_{x|y}(x|y)$ is still, first and foremost, a probability density function. Thus,

$$\int_{-\infty}^{\infty} p_{x|y}(x|y)\, dx = \frac{\int_{-\infty}^{\infty} p_{xy}(x, y)\, dx}{p_y(y)} = \frac{p_y(y)}{p_y(y)} = 1 \tag{8.52}$$

* It may be worth noting that $P_{x|y}(x|y)$ is conditioned on an event $y = y$ that has probability zero.

Independent Random Variables

The continuous RVs x and y are said to be independent if

$$p_{x|y}(x|y) = p_x(x) \tag{8.53a}$$

In this case from Eqs. (8.53a) and (8.51) it follows that

$$p_{y|x}(y|x) = p_y(y) \tag{8.53b}$$

This implies that for independent RVs x and y,

$$p_{xy}(x, y) = p_x(x)p_y(y) \tag{8.53c}$$

Based on Eq. (8.53c), the joint CDF is also separable:

$$\begin{aligned} F_{xy}(x, y) &= \int_{-\infty}^{x}\int_{-\infty}^{y} p_{xy}(v, w)\, dw\, dv \\ &= \int_{-\infty}^{x} p_x(v)\, dv \cdot \int_{-\infty}^{y} p_y(w)\, dw \\ &= F_x(x) \cdot F_y(y) \end{aligned} \tag{8.54}$$

Example 8.17 Rayleigh Density

The Rayleigh density is characterized by the PDF (Fig. 8.14b)

$$p_r(r) = \begin{cases} \dfrac{r}{\sigma^2} e^{-r^2/2\sigma^2} & r \geq 0 \\ 0 & r < 0 \end{cases} \tag{8.55}$$

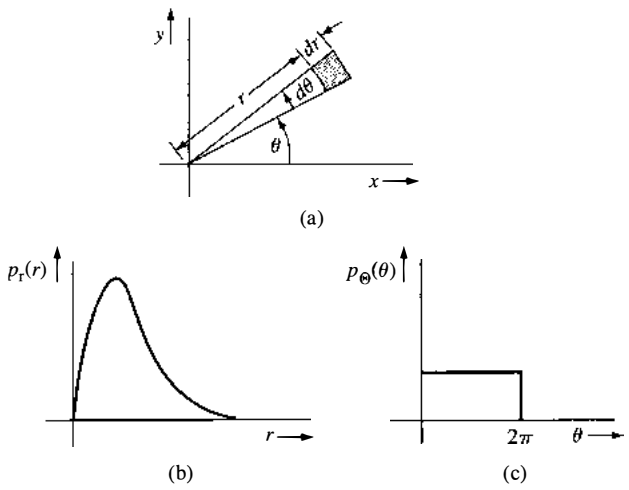


Figure 8.14 Derivation of the Rayleigh density.

A Rayleigh RV can be derived from two independent Gaussian RVs as follows. Let x and y be independent Gaussian variables with identical PDFs:

$$p_{\mathrm{x}}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2}$$

$$p_{\mathrm{y}}(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-y^2/2\sigma^2}$$

Then

$$p_{\mathrm{xy}}(x,\ y) = p_{\mathrm{x}}(x)p_{\mathrm{y}}(y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{8.56}$$

The joint density appears somewhat like the bell-shaped surface shown in Fig. 8.13. The points in the (x, y) plane can also be described in polar coordinates as $(r,\ \theta)$, where (Fig. 8.14a)

$$\mathrm{r} = \sqrt{\mathrm{x}^2 + \mathrm{y}^2} \qquad \Theta = \tan^{-1}\frac{\mathrm{y}}{\mathrm{x}}$$

In Fig. 8.14a, the shaded region represents $r < \mathrm{r} \leq r + dr$ and $\theta < \Theta \leq \theta + d\theta$ (where dr and $d\theta$ both $\to$ 0). Hence, if $p_{\mathrm{r}\Theta}(r,\ \theta)$ is the joint PDF of r and Θ, then by definition [Eq. (8.45)], the probability of observing r and Θ in this region is $p_{\mathrm{r}\Theta}(r,\ \theta)\, dr\, d\theta$. But we also know that this probability is $p_{\mathrm{xy}}(x,\ y)$ times the area $r\, dr\, d\theta$ of the shaded region.

Hence, [Eq. (8.56)]

$$\frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)2\sigma^2}\, r\, dr\, d\theta = p_{\mathrm{r}\Theta}(r,\ \theta)\, dr\, d\theta$$

and

$$\begin{aligned} p_{\mathrm{r}\Theta}(r,\ \theta) &= \frac{r}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \\ &= \frac{r}{2\pi\sigma^2} e^{-r^2/2\sigma^2} \end{aligned} \tag{8.57}$$

and [Eq. (8.48a)]

$$p_{\mathrm{r}}(r) = \int_{-\infty}^{\infty} p_{\mathrm{r}\Theta}\,(r,\ \theta)\, d\theta$$

Because Θ exists only in the region $(0,\ 2\pi)$,

$$\begin{aligned} p_{\mathrm{r}}(r) &= \int_0^{2\pi} \frac{r}{2\pi\sigma^2} e^{-r^2/2\sigma^2}\, d\theta \\ &= \frac{r}{\sigma^2} e^{-r^2/2\sigma^2} u(r) \end{aligned} \tag{8.58a}$$

Note that r is always greater than 0. In a similar way, we find

$$p_{\Theta}(\theta) = \begin{cases} \dfrac{1}{2\pi} & 0 \leq \Theta < 2\pi \\ 0 & \text{otherwise} \end{cases} \tag{8.58b}$$

RVs r and Θ are independent because $p_{r\Theta}(r,\ \theta) = p_r(r)p_\Theta(\theta)$. The PDF $p_r(r)$ is the **Rayleigh density function**. We shall later show that the envelope of narrowband Gaussian noise has a Rayleigh density. Both $p_r(r)$ and $p_\Theta(\theta)$ are shown in Fig. 8.14b and c.

8.3 STATISTICAL AVERAGES (MEANS)

Averages are extremely important in the study of RVs. To find a proper definition for the average of a random variable x, consider the problem of determining the average height of the entire population of a country. Let us assume that we have enough resources to gather data about the height of every person. If the data is recorded within the accuracy of an inch, then the height x of every person will be approximated to one of the n numbers $x_1, x_2, \ldots, x_n$. If there are N_i persons of height x_i, then the average height $\bar{x}$ is given by

$$\bar{x} = \frac{N_1x_1 + N_2x_2 + \cdots + N_nx_n}{N}$$

where the total number of persons is $N = \sum_i N_i$. Hence,

$$\bar{x} = \frac{N_1}{N}x_1 + \frac{N_2}{N}x_2 + \cdots + \frac{N_n}{N}x_n$$

In the limit as $N \to \infty$, the ratio N_i/N approaches $P_x(x_i)$ according to the relative frequency definition of the probability. Hence,

$$\bar{x} = \sum_{i=1}^{n} x_iP_x(x_i)$$

The mean value is also called the **average value**, or **expected value**, of the RV x and is denoted by $E[x]$. Thus,

$$\bar{x} = E[x] = \sum_i x_iP_x(x_i) \tag{8.59a}$$

We shall use both these notations, our choice depending on the circumstances and convenience.

If the RV x is continuous, an argument similar to that used in arriving at Eq. (8.59a) yields

$$\bar{x} = E[x] = \int_{-\infty}^{\infty} xp_x(x)\,dx \tag{8.59b}$$

This result can be derived by approximating the continuous variable x with a discrete variable by quantizing it in steps of Δx and then letting $\Delta x \to 0$.

Equation (8.59b) is more general and includes Eq. (8.59a), because the discrete RV can be considered as a continuous RV with an impulsive density. In such a case, Eq. (8.59b) reduces to Eq. (8.59a).

As an example, consider the general Gaussian PDF given by (Fig. 8.11)

$$p_x(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-(x-m)^2/2\sigma^2} \tag{8.60a}$$

From Eq. (8.59b) we have

$$\bar{x} = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} x e^{-(x-m)^2/2\sigma^2}\, dx$$

Changing the variable to $x = y + m$ yields

$$\begin{aligned}\bar{x} &= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} (y+m) e^{-y^2/2\sigma^2}\, dy \\ &= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} y e^{-y^2/2\sigma^2}\, dy + m\left[\frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-y^2/2\sigma^2}\, dy\right]\end{aligned}$$

The first integral inside the bracket is zero, because the integrand is an odd function of y. The term inside the square brackets is the integration of the Gaussian PDF, and is equal to 1. Hence,

$$\bar{x} = m \tag{8.60b}$$

Mean of a Function of a Random Variable

It is often necessary to find the mean value of a function of a RV. For instance, in practice we are often interested in the mean square amplitude of a signal. The mean square amplitude is the mean of the square of the amplitude x, that is, $\overline{x^2}$.

In general, we may seek the mean value of an RV y that is a function of the RV x; that is, we wish to find $\bar{y}$ where $y = g(x)$. Let x be a discrete RV that takes values $x_1, x_2, \ldots, x_n$ with probabilities $P_x(x_1), P_x(x_2), \ldots, P_x(x_n)$, respectively. But because $y = g(x)$, y takes values $g(x_1), g(x_2), \ldots, g(x_n)$ with probabilities $P_x(x_1), P_x(x_2), \ldots, P_x(x_n)$, respectively. Hence, from Eq. (8.59a) we have

$$\bar{y} = \overline{g(x)} = \sum_{i=1}^{n} g(x_i) P_x(x_i) \tag{8.61a}$$

If x is a continuous RV, a similar line of reasoning leads to

$$\overline{g(x)} = \int_{-\infty}^{\infty} g(x) p_x(x)\, dx \tag{8.61b}$$

Example 8.18 The output voltage of sinusoid generator is $A \cos \omega t$. This output is sampled randomly (Fig. 8.15a). The sampled output is an RV x, which can take on any value in the range $(-A, A)$. Determine the mean value $(\bar{x})$ and the mean square value $(\overline{x^2})$ of the sampled output x.

If the output is sampled at a random instant t, the output x is a function of the RV t:

$$x(t) = A \cos \omega t$$

If we let $\omega t = \Theta$, Θ is also an RV, and if we consider only modulo-2π values of Θ, then the RV Θ lies in the range $(0, 2\pi)$. Because t is randomly chosen, Θ can take any value in the range $(0, 2\pi)$ with uniform probability. Because the area under the PDF must be unity, $p_\Theta(\theta)$ is as shown in Fig. 8.15b.

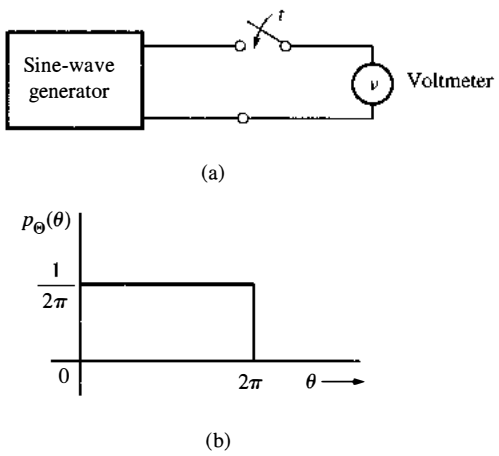


Figure 8.15 Random sampling of a sine-wave generator.

The RV x is thus a function of another RV, Θ,

$$\mathrm{x} = A\cos\Theta$$

Hence, from Eq. (8.61b),

$$\overline{\mathrm{x}} = \int_0^{2\pi} x p_\Theta(\theta)\, d\theta = \frac{1}{2\pi}\int_0^{2\pi} A\cos\theta\, d\theta = 0$$

and

$$\overline{\mathrm{x}^2} = \int_0^{2\pi} x^2 p_\Theta(\theta)\, d\theta = \frac{A^2}{2\pi}\int_0^{2\pi} \cos^2\theta\, d\theta$$
$$= \frac{A^2}{2}$$

Similarly, for the case of two variables x and y, we have

$$\overline{g(\mathrm{x},\ \mathrm{y})} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,\ y) p_{\mathrm{xy}}(x,\ y)\, dx\, dy \tag{8.62}$$

Mean of the Sum

If $g_1(\mathrm{x},\ \mathrm{y})$, $g_2(\mathrm{x},\ \mathrm{y})$, $\ldots$, $g_n(\mathrm{x},\ \mathrm{y})$ are functions of the RVs x and y, then

$$\overline{g_1(\mathrm{x},\ \mathrm{y}) + g_2(\mathrm{x},\ \mathrm{y}) + \cdots + g_n(\mathrm{x},\ \mathrm{y})} = \overline{g_1(\mathrm{x},\ \mathrm{y})} + \overline{g_2(\mathrm{x},\ \mathrm{y})} + \cdots + \overline{g_n(\mathrm{x},\ \mathrm{y})} \tag{8.63a}$$

The proof is trivial and follows directly from Eq. (8.62).

Thus, the mean (expected value) of the sum is equal to the sum of the means. An important special case is

$$\overline{\mathrm{x} + \mathrm{y}} = \overline{\mathrm{x}} + \overline{\mathrm{y}} \tag{8.63b}$$

Equation (8.63a) can be extended to functions of any number of RVs.

Mean of the Product of Two Functions

Unfortunately, there is no simple result [as in Eq. (8.63)] for the product of two functions. For the special case where

$$g(\mathrm{x},\ \mathrm{y}) = g_1(\mathrm{x})g_2(\mathrm{y}) \tag{8.64a}$$

$$\overline{g_1(\mathrm{x})g_2(\mathrm{y})} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g_1(\mathrm{x})g_2(\mathrm{y})p_{\mathrm{xy}}(x,\ y)\,dx\,dy$$

If x and y are independent, then [Eq. (8.53c)]

$$p_{\mathrm{xy}}(x,\ y) = p_{\mathrm{x}}(x)p_{\mathrm{y}}(y)$$

and

$$\begin{aligned}\overline{g_1(\mathrm{x})g_2(\mathrm{y})} &= \int_{-\infty}^{\infty} g_1(x)p_{\mathrm{x}}(x)\,dx \int_{-\infty}^{\infty} g_2(y)p_{\mathrm{y}}(y)\,dy \\ &= \overline{g_1(\mathrm{x})}\ \overline{g_2(\mathrm{y})} \qquad \text{if x and y independent}\end{aligned} \tag{8.64b}$$

A special case of this is

$$\overline{\mathrm{xy}} = \overline{\mathrm{x}}\,\overline{\mathrm{y}} \qquad \text{if x and y independent} \tag{8.64c}$$

Moments

The nth **moment** of an RV x is defined as the mean value of x^n. Thus, the nth moment of x is

$$\overline{\mathrm{x}^n} \triangleq \int_{-\infty}^{\infty} x^n p_{\mathrm{x}}(x)\,dx \tag{8.65a}$$

The nth **central moment** of an RV x is defined as

$$\overline{(\mathrm{x} - \overline{\mathrm{x}})^n} \triangleq \int_{-\infty}^{\infty} (x - \overline{\mathrm{x}})^n p_{\mathrm{x}}(x)\,dx \tag{8.65b}$$

The second central moment of an RV x is of special importance. It is called the **variance** of x and is denoted by σ_{x}^2, where σ_{x} is known as the **standard deviation (SD)** of the RV x. By definition,

$$\begin{aligned}\sigma_{\mathrm{x}}^2 &= \overline{(\mathrm{x} - \overline{\mathrm{x}})^2} \\ &= \overline{\mathrm{x}^2} - 2\overline{\mathrm{x}}\,\overline{\mathrm{x}} + \overline{\mathrm{x}}^2 = \overline{\mathrm{x}^2} - 2\overline{\mathrm{x}}^2 + \overline{\mathrm{x}}^2 \\ &= \overline{\mathrm{x}^2} - \overline{\mathrm{x}}^2\end{aligned} \tag{8.66}$$

Thus, the variance of x is equal to the mean square value minus the square of the mean. When the mean is zero, the variance is the mean square; that is, $\overline{\mathrm{x}^2} = \sigma_{\mathrm{x}}^2$.

Example 8.19 Find the mean square and the variance of the Gaussian RV with the PDF in Eq. (8.39) [see Fig. 8.11].

We have

$$\overline{x^2} = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} x^2 e^{-(x-m)^2/2\sigma^2}\, dx$$

Changing the variable to $y = (x - m)/\sigma$ and integrating, we get

$$\overline{x^2} = \sigma^2 + m^2 \tag{8.67a}$$

Also, from Eqs. (8.66) and (8.60b),

$$\begin{aligned} \sigma_x^2 &= \overline{x^2} - \overline{x}^2 \\ &= (\sigma^2 + m^2) - (m)^2 \\ &= \sigma^2 \end{aligned} \tag{8.67b}$$

Hence, a Gaussian RV described by the density in Eq. (8.60a) has mean m and variance σ^2. In other words, the Gaussian density function is completely specified by the first moment ($\overline{x}$) and the second moment ($\overline{x^2}$).

Example 8.20 Mean Square of the Uniform Quantization Error in PCM

In the PCM scheme discussed in Chapter 6, a signal band-limited to B Hz is sampled at a rate of $2B$ samples per second. The entire range $(-m_p, m_p)$ of the signal amplitudes is partitioned into L uniform intervals, each of magnitude $2m_p/L$ (Fig. 8.16a). Each sample is approximated to the midpoint of the interval in which it falls. Thus, sample m in Fig. 8.16a is approximated by a value $\hat{m}$, the midpoint of the interval in which m falls. Each sample is thus approximated (quantized) to one of the L numbers.

The difference $q = m - \hat{m}$ is the quantization error and is an RV. We shall determine $\overline{q^2}$, the mean square value of the quantization error. From Fig. 8.16a it can be seen that q is a continuous RV existing over the range $(-m_p/L, m_p/L)$ and is zero outside this range. If we assume that it is equally likely for the sample to lie anywhere in the quantizing interval,* then the PDF of q is uniform

$$p_q(q) = L/2m_p \qquad q \in (-m_p/L, m_p/L)$$

* Because the quantizing interval is generally very small, variations in the PDF of signal amplitudes over the interval are small and this assumption is reasonable.

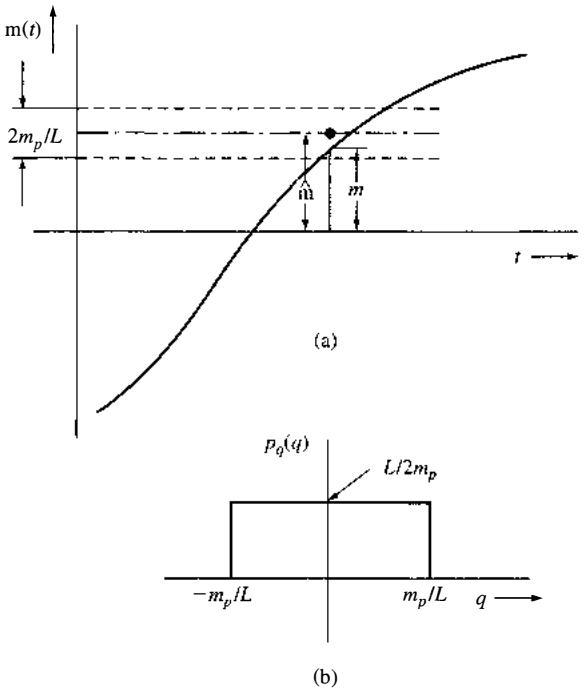


Figure 8.16 (a) Quantization error in PCM and (b) its PDF.

as shown in Fig. 8.16b, and

$$\begin{aligned}\overline{q^2} &= \int_{-m_p/L}^{m_p/L} q^2 p_q(q)\, dq \\ &= \frac{L}{2m_p}\frac{q^3}{3}\Big|_{-m_p/L}^{m_p/L} \\ &= \frac{1}{3}\left(\frac{m_p}{L}\right)^2 \end{aligned} \tag{8.68a}$$

From Fig. 8.16b it can be seen that $\overline{q} = 0$. Hence,

$$\sigma_q^2 = \overline{q^2} = \frac{1}{3}\left(\frac{m_p}{L}\right)^2 \tag{8.68b}$$

Example 8.21 Mean Square Error Caused by Channel Noise in PCM

Quantization noise is one of the sources of error in PCM. The other source of error is channel noise. Each quantized sample is coded by a group of n binary pulses. Because of channel noise, some of these pulses are incorrectly detected at the receiver. Hence, the decoded sample value $\tilde{m}$ at the receiver will differ from the quantized sample value $\hat{m}$ that is transmitted. The error $\epsilon = \hat{m} - \tilde{m}$ is a random variable. Let us calculate $\overline{\epsilon^2}$, the mean square error in the sample value caused by the channel noise.

To begin with, let us determine the values that ϵ can take and the corresponding probabilities. Each sample is transmitted by n binary pulses. The value of ϵ depends on the position of the incorrectly detected pulse. Consider, for example, the case of $L = 16$ transmitted by four binary pulses ($n = 4$), as shown in Fig. 1.5. Here the transmitted code **1101** represents a value of 13. A detection error in the first digit changes the received code to **0101**, which is a value of 5. This causes an error $\epsilon = 8$. Similarly, an error in the second digit gives $\epsilon = 4$. Errors in the third and the fourth digits will give $\epsilon = 2$ and $\epsilon = 1$, respectively. In general, the error in the ith digit causes an error $\epsilon_i = (2^{-i})16$. For a general case, the error $\epsilon_i = (2^{-i})F$, where F is the full scale, that is, $2m_p$, in PCM. Thus,

$$\epsilon_i = (2^{-i})(2m_p) \qquad i = 1, 2, \ldots, n$$

Note that the error ϵ is a discrete RV. Hence,*

$$\overline{\epsilon^2} = \sum_{i=1}^{n} \epsilon_i^2 P_\epsilon(\epsilon_i) \tag{8.69}$$

Because $P_\epsilon(\epsilon_i)$ is the probability that $\epsilon = \epsilon_i$, $P_\epsilon(\epsilon_i)$ is the probability of error in the detection of the ith digit. Because the error probability of detecting any one digit is the same as that of any other, that is, P_e,

$$\begin{aligned}\overline{\epsilon^2} &= P_e \sum_{i=1}^{n} \epsilon_i^2 \\ &= P_e \sum_{i=1}^{n} 4m_p^2(2^{-2i}) \\ &= 4m_p^2 P_e \sum_{i=1}^{n} 2^{-2i}\end{aligned}$$

This summation is a geometric progression with a common ratio $r = 2^{-2}$, with the first term $a_1 = 2^{-2}$ and the last term $a_n = 2^{-2n}$. Hence (see Appendix E.4),

$$\begin{aligned}\overline{\epsilon^2} &= 4m_p^2 P_e \left[\frac{(2^{-2})2^{-2n} - 2^{-2}}{2^{-2} - 1}\right] \\ &= \frac{4m_p^2 P_e(2^{2n} - 1)}{3(2^{2n})}\end{aligned} \tag{8.70a}$$

Note that the magnitude of the error ϵ varies from $2^{-1}(2m_p)$ to $2^{-n}(2m_p)$. The error ϵ can be positive as well as negative. For example, $\epsilon = 8$ because of a first-digit error in **1101**. But the corresponding error ϵ will be -8 if the transmitted code is **0101**. Of course the sign of ϵ does not matter in Eq. (8.69). It must be remembered, however, that ϵ varies from $-2^{-n}(2m_p)$ to $2^{-n}(2m_p)$ and its probabilities are symmetrical about

* Here we are assuming that the error can occur only in one of the n digits. But more than one digit may be in error. Because the digit error probability $P_e \ll 1$ (on the order 10^{-5} or less), however, the probability of more than one wrong digit is extremely small (see Example 8.6), and its contribution $\epsilon_i^2 P_\epsilon(\epsilon_i)$ is negligible.

$\epsilon = 0$. Hence, $\bar{\epsilon} = 0$ and

$$\sigma_\epsilon^2 = \overline{\epsilon^2} = \frac{4m_p^2 P_e(2^{2n} - 1)}{3(2^{2n})} \tag{8.70b}$$

Variance of a Sum of Independent Random Variables

The variance of a sum of independent RVs is equal to the sum of their variances. Thus, if x and y are independent RVs and

$$z = x + y$$

then

$$\sigma_z^2 = \sigma_x^2 + \sigma_y^2 \tag{8.71}$$

This can be shown as follows:

$$\begin{aligned}
\sigma_z^2 = \overline{(z - \bar{z})^2} &= \overline{[x + y - (\bar{x} + \bar{y})]^2} \\
&= \overline{[(x - \bar{x}) + (y - \bar{y})]^2} \\
&= \overline{(x - \bar{x})^2} + \overline{(y - \bar{y})^2} + \overline{2(x - \bar{x})(y - \bar{y})} \\
&= \sigma_x^2 + \sigma_y^2 + \overline{2(x - \bar{x})(y - \bar{y})}
\end{aligned}$$

Because x and y are independent RVs, $(x - \bar{x})$ and $(y - \bar{y})$ are also independent RVs. Hence, from Eq. (8.64b) we have

$$\overline{(x - \bar{x})(y - \bar{y})} = \overline{(x - \bar{x})} \cdot \overline{(y - \bar{y})}$$

But

$$\overline{(x - \bar{x})} = \bar{x} - \bar{x} = \bar{x} - \bar{x} = 0$$

Similarly,

$$\overline{(y - \bar{y})} = 0$$

and

$$\sigma_z^2 = \sigma_x^2 + \sigma_y^2$$

This result can be extended to any number of variables. If RVs x and y both have zero means (i.e., $\bar{x} = \bar{y} = 0$), then $\bar{z} = \bar{x} + \bar{y} = 0$. Also, because the variance equals the mean square value when the mean is zero, it follows that

$$\overline{z^2} = \overline{(x + y)^2} = \overline{x^2} + \overline{y^2} \tag{8.72}$$

provided $\bar{x} = \bar{y} = 0$, and provided x and y are independent RVs.

Example 8.22 Total Mean Square Error in PCM

In PCM, as seen in Examples 8.20 and 8.21, a signal sample m is transmitted as a quantized sample $\hat{m}$, causing a quantization error $q = m - \hat{m}$. Because of channel noise, the transmitted sample $\hat{m}$ is read as $\tilde{m}$, causing a detection error $\epsilon = \hat{m} - \tilde{m}$. Hence, the actual signal sample m is received as $\tilde{m}$ with a total error

$$m - \tilde{m} = (m - \hat{m}) + (\hat{m} - \tilde{m}) = q + \epsilon$$

where both q and ϵ are zero mean RVs. Because the quantization error q and the channel-noise error ϵ are independent, the mean square of the sum is [see Eq. (8.72)]

$$\overline{(m - \tilde{m})^2} = \overline{(q + \epsilon)^2} = \overline{q^2} + \overline{\epsilon^2}$$
$$= \frac{1}{3}\left(\frac{m_p}{L}\right)^2 + \frac{4m_p^2 P_\epsilon (2^{2n} - 1)}{3(2^{2n})}$$

Also, because $L = 2^n$,

$$\overline{(m - \tilde{m})^2} = \overline{q^2} + \overline{\epsilon^2} = \frac{m_p{}^2}{3(2^{2n})}[1 + 4P_\epsilon(2^{2n} - 1)] \tag{8.73}$$

Chebyshev's Inequality

The standard deviation σ_x of an RV x is a measure of the width of its PDF. The larger the σ_x, the wider the PDF. Figure 8.17 illustrates this effect for a Gaussian PDF. Chebyshev's inequality is a statement of this fact. It states that for a zero mean RV x

$$P(|x| \leq k\sigma_x) \geq 1 - \frac{1}{k^2} \tag{8.74}$$

This means the probability of observing x within a few standard deviations is very high. For example, the probability of finding |x| within $3\sigma_x$ is equal to or greater than 0.88. Thus, for a PDF with $\sigma_x = 1$, $P(|x| \leq 3) \geq 0.88$, whereas for a PDF with $\sigma_x = 3$, $P(|x| \leq 9) \geq 0.88$. It is clear that the PDF with $\sigma_x = 3$ is spread out much more than the PDF with $\sigma_x = 1$. Hence,

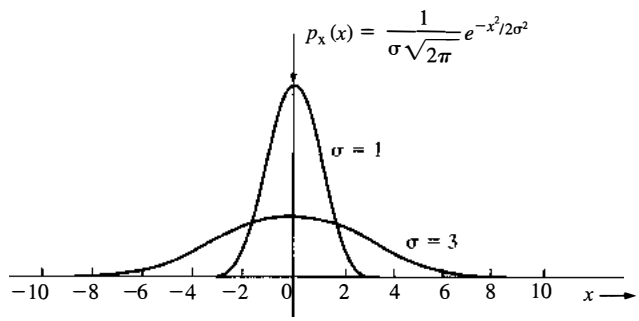


Figure 8.17 Gaussian PDF with standard deviations $\sigma = 1$ and $\sigma = 3$.

σ_x or σ_x^2 is often used as a measure of the width of a PDF. In Chapter 10, we shall use this measure to estimate the bandwidth of a signal spectrum. The proof of Eq. (8.74) is as follows:

$$\sigma_x^2 = \int_{-\infty}^{\infty} x^2 p_x(x)\, dx$$

Because the integrand is positive,

$$\sigma_x^2 \geq \int_{|x|\geq k\sigma_x} x^2 p_x(x)\, dx$$

If we replace x by its smallest value $k\sigma_x$, the inequality still holds,

$$\sigma_x^2 \geq k^2\sigma_x^2 \int_{|x|\geq k\sigma_x} p_x(x)\, dx = k^2\sigma_x^2 P(|x| \geq k\sigma_x)$$

or

$$P(|x| \geq k\sigma_x) \leq \frac{1}{k^2}$$

Hence,

$$P(|x| < k\sigma_x) \geq 1 - \frac{1}{k^2}$$

This inequality can be generalized for a nonzero mean RV as:

$$P(|x - \bar{x}| < k\sigma_x) \geq 1 - \frac{1}{k^2} \tag{8.75}$$

Example 8.23 Estimate the width, or spread, of a Gaussian PDF [Eq. (8.60a)]

For a Gaussian RV [see Eqs. (8.35) and (8.40b)]

$$P(|x - \bar{x}| < \sigma) = 1 - 2Q(1) = 0.6826$$
$$P(|x - \bar{x}| < 2\sigma) = 1 - 2Q(2) = 0.9546$$
$$P(|x - \bar{x}| < 3\sigma) = 1 - 2Q(3) = 0.9974$$

This means that the area under the PDF over the interval $(\bar{x} - 3\sigma,\ \bar{x} + 3\sigma)$ is 99.74% of the total area. A negligible fraction (0.26%) of the area lies outside this interval. Hence, the width, or spread, of the Gaussian PDF may be considered roughly $\pm 3\sigma$ about its mean, giving a total width of roughly 6σ.

8.4 CORRELATION

Often we are interested in determining the nature of dependence between two entities, such as smoking and lung cancer. Consider a random experiment with two outcomes described by

RVs x and y. We conduct several trials of this experiment and record values of x and y for each trial. From this data, it may be possible to determine the nature of a dependence between x and y. The covariance of RVs x and y is one measure that is simple to compute and can yield useful information about the dependence between x and y.

The covariance σ_{xy} of two RVs is defined as

$$\sigma_{xy} \triangleq \overline{(x - \bar{x})(y - \bar{y})} \tag{8.76}$$

Note that the concept of covariance is a natural extension of the concept of variance, which is defined as

$$\sigma_x^2 = \overline{(x - \bar{x})(x - \bar{x})}$$

Let us consider a case of two variables x and y that are dependent such that they tend to vary in harmony; that is, if x increases y increases, and if x decreases y also decreases. For instance, x may be the average daily temperature of a city and y the volume of soft drink sales that day in the city. It is reasonable to expect the two quantities to vary in harmony for a majority of the cases. Suppose we consider the following experiment: pick a random day and record the average temperature of that day as the value of x and the soft drink sales volume that day as the value of y. We perform this measurement over several days (several trials of the experiment) and record the data x and y for each trial. We now plot points (x, y) for all the trials. This plot, known as the **scatter diagram**, may appear as shown in Fig. 8.18a. The plot shows that when x is large, y is likely to be large. Note the use of the word *likely*. It is not *always* true that y will be large if x is large, but it is true most of the time. In other words, in a few cases, a low average temperature will be paired with higher soft drink sales owing to some atypical situation, such as a major soccer match. This is quite obvious from the scatter diagram in Fig. 8.18a.

To continue this example, the variable $x - \bar{x}$ represents the difference between actual and average values of x, and $y - \bar{y}$ represents the difference between actual and average values of y. It is more instructive to plot $(y - \bar{y})$ vs. $(x - \bar{x})$. This is the same as the scatter diagram in Fig. 8.18a with the origin shifted to $(\bar{x}, \bar{y})$, as in Fig. 8.18b, which shows that a day with an above-average temperature is likely to produce above-average soft drink sales, and a day with a below-average temperature is likely to produce below-average soft drink sales.

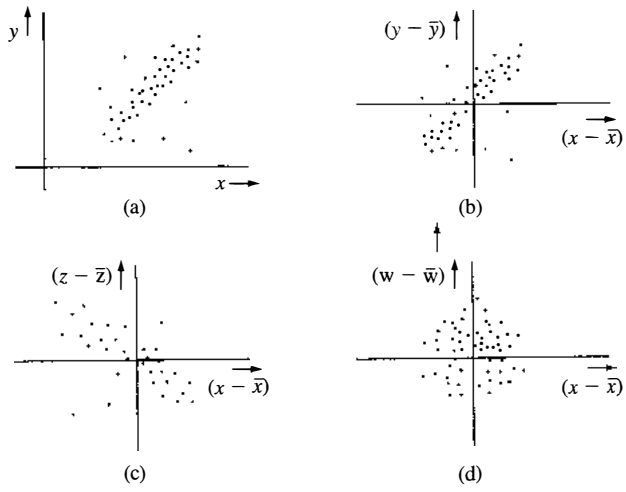


Figure 8.18 Scatter diagrams: (a), (b) positive correlation; (c) negative correlation; (d) zero correlation.

That is, if $x - \bar{x}$ is positive, $y - \bar{y}$ is likely to be positive, and if $x - \bar{x}$ is negative, $y - \bar{y}$ is more likely to be negative. Thus, the quantity $(x - \bar{x})(y - \bar{y})$ will be positive for most trials. We compute this product for every pair, add these products, and then divide by the number of trials. The result is the mean value of $(x - \bar{x})(y - \bar{y})$, that is, the covariance $\sigma_{xy} = \overline{(x - \bar{x})(y - \bar{y})}$. The covariance will be positive in the example under consideration. In such cases, we say that a positive correlation exists between variables x and y. We can conclude that a positive correlation implies variation of two variables in harmony (in the same direction, up or down).

Next, we consider the case of the two variables: x, the average daily temperature, and z, the sales volume of sweaters that day. It is reasonable to believe that as x (daily average temperature) increases, z (the sweater sales volume) tends to decrease. A hypothetical scatter diagram for this experiment is shown in Fig. 8.18c. Thus, if $x - \bar{x}$ is positive (above-average temperature), $z - \bar{z}$ is likely to be negative (below-average sweater sales). Similarly, when $x - \bar{x}$ is negative, $z - \bar{z}$ is likely to be positive. The product $(x - \bar{x})(z - \bar{z})$ will be negative for most of the trials, and the mean $\overline{(x - \bar{x})(z - \bar{z})} = \sigma_{xz}$ will be negative. In such a case, we say that negative correlation exists between x and y. It should be stressed here that negative correlation does not mean that x and y are unrelated. It means that they are dependent, but when one increases, the other decreases, and vice versa.

Last, consider the variables x (the average daily temperature) and w (the number of births). It is reasonable to expect that the daily temperature has little to do with the number of children born. A hypothetical scatter diagram for this case will appear as shown in Fig. 8.18d. If $x - \bar{x}$ is positive, $w - \bar{w}$ is equally likely to be positive or negative. The product $(x - \bar{x})(w - \bar{w})$ is therefore equally likely to be positive or negative, and the mean $\overline{(x - \bar{x})(w - \bar{w})} = \sigma_{xw}$ will be zero. In such a case, we say that RVs x and w are **uncorrelated.**

To reiterate, if σ_{xy} is positive (or negative), then x and y are said to have a positive (or negative) correlation, and if $\sigma_{xy} = 0$, then the variables x and y are said to be uncorrelated.

From this discussion, it appears that under suitable conditions, covariance can serve as a measure of the dependence of two variables. It often provides *some* information about the interdependence of the two RVs and proves useful in a number of applications.

The covariance σ_{xy} may be expressed in another way, as follows. By definition,

$$\begin{aligned}\sigma_{xy} &= \overline{(x - \bar{x})(y - \bar{y})} \\ &= \overline{xy} - \overline{\bar{x}y} - \overline{x\bar{y}} + \overline{\bar{x}\bar{y}} \\ &= \overline{xy} - \bar{x}\bar{y} - \bar{x}\bar{y} + \bar{x}\bar{y} \\ &= \overline{xy} - \bar{x}\bar{y}\end{aligned} \tag{8.77}$$

From Eq. (8.77) it follows that the variables x and y are uncorrelated ($\sigma_{xy} = 0$) if

$$\overline{xy} = \bar{x}\bar{y} \tag{8.78}$$

The correlation between x and y cannot be directly compared with the correlation between z and w. This is because different RVs may differ in strength. To be fair, the covariance value should be normalized appropriately. For this reason, the definition of **correlation coefficient** is particularly useful. **Correlation coefficient** ρ_{xy} is σ_{xy} normalized by $\sigma_x\sigma_y$,

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x\sigma_y} \tag{8.79}$$

Thus, if x and y are uncorrelated, then $\rho_{xy} = 0$. Also, it can be shown that (Prob. 8.5-5) that

$$-1 \leq \rho_{xy} \leq 1 \tag{8.80}$$

Independence vs. Uncorrelatedness

Note that for independent RVs [Eq. (8.64c)]

$$\overline{xy} = \bar{x}\bar{y} \qquad \text{and} \qquad \sigma_{xy} = 0$$

Hence, independent RVs are uncorrelated. This supports the heuristic argument presented earlier. It should be noted that whereas independent variables are uncorrelated, the converse is not necessarily true—uncorrelated variables are generally not independent (Prob. 8.5-3). Independence is, in general, a stronger and more restrictive condition than uncorrelatedness. For independent variables, we have shown [Eq. (8.64b)] that, when the expectations exist,

$$\overline{g_1(x)g_2(y)} = \overline{g_1(x)}\,\overline{g_2(y)}$$

for any functions $g_1(\cdot)$ and $g_2(\cdot)$, whereas for uncorrelatedness, the only requirement is that

$$\overline{xy} = \bar{x}\,\bar{y}$$

There is only one *special* case for which independence and uncorrelatedness are equivalent—when random variables x and y are jointly Gaussian. Note that when x and y are jointly Gaussian, individually x and y are also Gaussian.

Mean Square of the Sum of Uncorrelated Variables

If x and y are uncorrelated, then for $z = x + y$ we show that

$$\sigma_z^2 = \sigma_x^2 + \sigma_y^2 \tag{8.81}$$

That is, the variance of the sum is the sum of variances for uncorrelated RVs. We have proved this result earlier for independent variables x and y. Following the development after Eq. (8.71), we have

$$\begin{aligned}\sigma_z^2 &= \overline{[(x - \bar{x}) + (y - \bar{y})]^2} \\ &= \overline{(x - \bar{x})^2} + \overline{(y - \bar{y})^2} + 2\overline{(x - \bar{x})(y - \bar{y})} \\ &= \sigma_x^2 + \sigma_y^2 + 2\sigma_{xy}\end{aligned}$$

Because x and y are uncorrelated, $\sigma_{xy} = 0$, and Eq. (8.81) follows. If x and y have zero means, then z also has a zero mean, and the mean square values of these variables are equal to their variances. Hence,

$$\overline{(x + y)^2} = \overline{x^2} + \overline{y^2} \tag{8.82}$$

if x and y are uncorrelated and have zero means. Thus, Eqs. (8.81) and (8.82) are valid not only when x and y are independent, but also under the less restrictive condition that x and y be uncorrelated.

8.5 LINEAR MEAN SQUARE ESTIMATION

When two random variables x and y are related (or dependent), then a knowledge of one gives certain information about the other. Hence, it is possible to estimate the value of (parameter or signal) y from a knowledge of the value of x. The estimate of y will be another random variable $\hat{y}$. The estimated random variable $\hat{y}$ will in general be different from the actual y. One may choose various criteria of goodness for estimation. Minimum mean square error is one possible criterion. The optimum estimate in this case minimizes the mean square error $\overline{\epsilon^2}$ given by

$$\overline{\epsilon^2} = \overline{(y - \hat{y})^2}$$

In general, the optimum estimate $\hat{y}$ is a nonlinear function of x.* We simplify the problem by constraining the estimate $\hat{y}$ to be a linear function of x of the form

$$\hat{y} = ax$$

assuming that $\overline{x} = 0$.† In this case,

$$\begin{aligned}\overline{\epsilon^2} &= \overline{(y - \hat{y})^2} = \overline{(y - ax)^2} \\ &= \overline{y^2} + a^2\overline{x^2} - 2a\overline{xy}\end{aligned}$$

To minimize $\overline{\epsilon^2}$, we have

$$\frac{\partial \overline{\epsilon^2}}{\partial a} = 2a\overline{x^2} - 2\overline{xy} = 0$$

Hence,

$$a = \frac{\overline{xy}}{\overline{x^2}} = \frac{R_{xy}}{R_{xx}} \tag{8.83}$$

where $R_{xy} = \overline{xy}$, $R_{xx} = \overline{x^2}$, and $R_{yy} = \overline{y^2}$. Note that for this constant choice of a,

$$\epsilon = y - ax = y - \frac{R_{xy}}{R_{xx}}x$$

Hence,

$$\overline{x\epsilon} = \overline{x\left(y - \frac{R_{xy}}{R_{xx}}x\right)} = \overline{xy} - \frac{R_{xy}}{R_{xx}}\overline{x^2}$$

---

* It can be shown that[5] the optimum estimate $\hat{y}$ is the conditional mean of y when $x = x$, that is,

$$\hat{y} = E[y \mid x = x]$$

In general, this is a nonlinear function of x.

† Throughout the discussion, the variables x, y, ... will be assumed to have zero mean. This can be done without loss of generality. If the variables have nonzero means, we can form new variables $x' = x - \overline{x}$ and $y' = y - \overline{y}$, and so on. The new variables obviously have zero mean values.

Since by definition $\overline{xy} = R_{xy}$ and $\overline{xx} = \overline{x^2} = R_{xx}$, we have

$$\overline{x\epsilon} = R_{xy} - R_{xy} = 0 \tag{8.84}$$

The condition of Eq. (8.84) is known as the principle of orthogonality. The physical interpretation is that the data (x) used in estimation and the (minimum) error (ϵ) are orthogonal (implying uncorrelatedness in this case) when the mean square error is minimum.

Given the principle of orthogonality, the minimum mean square error is given by

$$\begin{aligned}\overline{\epsilon^2} &= \overline{(y - ax)^2} \\ &= \overline{(y - ax)y} - a \cdot \overline{\epsilon x} \\ &= \overline{(y - ax)y} \\ &= \overline{y^2} - a \cdot \overline{yx} \\ &= R_{yy} - aR_{xy}\end{aligned} \tag{8.85}$$

Using n Random Variables to Estimate a Random Variable

If a random variable x_0 is related to n RVs $x_1, x_2, \ldots, x_n$, then we can estimate x_0 using a linear combination* of $x_1, x_2, \ldots, x_n$:

$$\hat{x}_0 = a_1x_1 + a_2x_2 + \cdots + a_nx_n = \sum_{i=1}^{n} a_ix_i \tag{8.86}$$

The mean square error is given by

$$\overline{\epsilon^2} = \overline{[x_0 - (a_1x_1 + a_2x_2 + \cdots + a_nx_n)]^2}$$

To minimize $\overline{\epsilon^2}$, we must set

$$\frac{\partial\overline{\epsilon^2}}{\partial a_1} = \frac{\partial\overline{\epsilon^2}}{\partial a_2} = \cdots = \frac{\partial\overline{\epsilon^2}}{\partial a_n} = 0$$

that is,

$$\frac{\partial\overline{\epsilon^2}}{\partial a_i} = \frac{\partial}{\partial a_i}\overline{[x_0 - (a_1x_1 + a_2x_2 + \cdots + a_nx_n)]^2} = 0$$

Interchanging the order of differentiation and averaging, we have

$$\frac{\partial\overline{\epsilon^2}}{\partial a_i} = -2\overline{[x_0 - (a_1x_1 + a_2x_2 + \cdots + a_nx_n)]x_i} = 0 \tag{8.87a}$$

Equation (8.87a) can be written as

$$\overline{\epsilon \cdot x_i} = 0 \qquad i = 1, 2, \ldots, n \tag{8.87b}$$

* Throughout this section as before, we assume that all the random variables have zero mean values. This can be done without loss of generality.

It can be rewritten into Yule-Walker equations

$$R_{0i} = a_1 R_{i1} + a_2 R_{i2} + \cdots + a_n R_{in} \tag{8.88}$$

where

$$R_{ij} = \overline{x_i x_j}$$

Differentiating $\overline{\epsilon^2}$ with respect to $a_1, a_2, \ldots, a_n$ and equating to zero, we obtain n simultaneous equations of the form shown in Eq. (8.88). The desired constants $a_1, a_2, \ldots, a_n$ can be found from these equations by matrix inversion

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1n} \\ R_{21} & R_{22} & \cdots & R_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ R_{n1} & R_{n2} & \cdots & R_{nn} \end{bmatrix}^{-1} \begin{bmatrix} R_{01} \\ R_{02} \\ \vdots \\ R_{0n} \end{bmatrix} \tag{8.89}$$

Equation (8.87) shows that ϵ (the error) is orthogonal to data $(x_1, x_2, \ldots, x_n)$ for optimum estimation. This gives the more general form for the principle of orthogonality in mean square estimation. Consequently, the mean square error (under optimum conditions) is

$$\overline{\epsilon^2} = \overline{\epsilon\epsilon} = \overline{\epsilon[x_0 - (a_1 x_1 + a_2 x_2 + \cdots + a_n x_n)]}$$

Because $\overline{\epsilon x_i} = 0$ $(i = 1, 2, \ldots, n)$,

$$\begin{aligned} \overline{\epsilon^2} &= \overline{\epsilon x_0} \\ &= \overline{x_0[x_0 - (a_1 x_1 + a_2 x_2 + \cdots + a_n x_n)]} \\ &= R_{00} - (a_1 R_{01} + a_2 R_{02} + \cdots + a_n R_{0n}) \end{aligned} \tag{8.90}$$

Example 8.24 In differential pulse code modulation (DPCM), instead of transmitting sample values directly, we estimate (predict) the value of each sample from the knowledge of previous n samples. The estimation error ϵ_k, the difference between the actual value and the estimated value of the kth sample, is quantized and transmitted (Fig. 8.19). Because the estimation error ϵ_k is smaller than the sample value m_k, for the same number of quantization levels (the same number of PCM code bits), the SNR is increased. It was shown in Sec. 6.5 that the SNR improvement is equal to $\overline{m^2}/\overline{\epsilon^2}$, where $\overline{m^2}$ and $\overline{\epsilon^2}$ are the mean square values of the speech signal and the estimation error ϵ, respectively. In this example, we shall find the optimum linear second-order predictor and the corresponding SNR improvement.

The equation of a second-order estimator (predictor), shown in Fig. 8.19, is

$$\hat{m}_k = a_1 m_{k-1} + a_2 m_{k-2}$$

where $\hat{m}_k$ is the best linear estimate of m_k. The estimation error ϵ_k is given by

$$\epsilon_k = \hat{m}_k - m_k = a_1 m_{k-1} + a_2 m_{k-2} - m_k$$

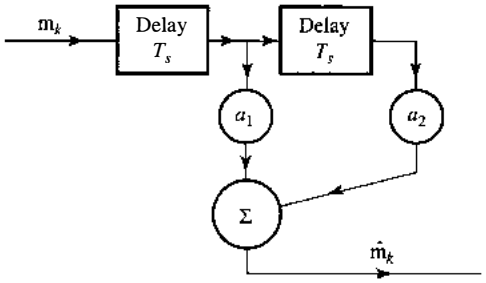


Figure 8.19 Second-order predictor in Example 8.24.

For speech signals, Jayant and Noll[5] give the values of correlations of various samples as:

$$\overline{m_k m_k} = \overline{m^2}, \quad \overline{m_k m_{k-1}} = 0.825\overline{m^2}, \quad \overline{m_k m_{k-2}} = 0.562\overline{m^2},$$
$$\overline{m_k m_{k-3}} = 0.308\overline{m^2}, \quad \overline{m_k m_{k-4}} = 0.004\overline{m^2}, \quad \overline{m_k m_{k-5}} = -0.243\overline{m^2}$$

Note that $R_{ij} = \overline{m_k m_{k-(j-i)}}$. Hence,

$$R_{11} = R_{22} = \overline{m^2}$$
$$R_{12} = R_{21} = R_{01} = 0.825\overline{m^2}$$
$$R_{02} = 0.562\overline{m^2}$$

The optimum values of a_1 and a_2 are found from Eq. (8.89) as $a_1 = 1.1314$ and $a_2 = -0.3714$, and the mean square error in the estimation is given by Eq. (8.90) as

$$\overline{\epsilon^2} = [1 - (0.825a_1 + 0.562a_2)]\overline{m^2} = 0.2753\overline{m^2} \tag{8.91}$$

The SNR improvement is $10 \log_{10} \overline{m^2}/0.2752\overline{m^2} = 5.6$ dB.

8.6 SUM OF RANDOM VARIABLES

In many applications, it is useful to characterize the RV z that is the sum of two RVs x and y:

$$z = x + y$$

Because $z = x + y$, $y = z - x$ regardless of the value of x. Hence, the event $z \leq z$ is the joint event [$y \leq z - x$ and x to have any value in the range $(-\infty, \infty)$]. Hence,

$$F_z(z) = P(z \leq z) = P(x \leq \infty, y \leq z - x)$$
$$= \int_{-\infty}^{\infty} \int_{-\infty}^{z-x} p_{xy}(x, y)\, dy\, dx$$
$$= \int_{-\infty}^{\infty} dx \int_{-\infty}^{z-x} p_{xy}(x, y)\, dy$$

and

$$p_z(z) = \frac{dF_z(z)}{dz} = \int_{-\infty}^{\infty} p_{xy}(x,\ z-x)\,dx$$

If x and y are independent RVs, then

$$p_{xy}(x,\ z-x) = p_x(x)p_y(z-x)$$

and

$$p_z(z) = \int_{-\infty}^{\infty} p_x(x)p_y(z-x)\,dx \tag{8.92}$$

The PDF $p_z(z)$ is then the convolution of PDFs $p_x(z)$ and $p_y(z)$. We can extend this result to a sum of n independent RVs $x_1,\ x_2,\ \ldots,\ x_n$. If

$$z = x_1 + x_2 + \cdots + x_n$$

then the PDF $p_z(z)$ will be the convolution of PDFs $p_{x_1}(x),\ p_{x_2}(x),\ \ldots,\ p_{x_n}(x)$, that is,

$$p_z(x) = p_{x_1}(x) * p_{x_2}(x) * \ \ldots\ * p_{x_n}(x) \tag{8.93}$$

Sum of Gaussian Random Variables

Gaussian random variables have several very important properties. For example, a Gaussian random variable x and its probability density function $p_x(x)$ are fully described by the mean μ_x and the variance σ_x^2. Furthermore, the sum of any number of jointly distributed Gaussian random variables is also a Gaussian random variable, regardless of their relationships (such as dependency). Again, note that when the members of a set of random variables $\{x_i\}$ are jointly Gaussian, each individual random variable x_i also has Gaussian distribution.

As an example, we will show that the sum of two independent, zero mean, Gaussian random variables is Gaussian. Let x_1 and x_2 be two zero mean and independent Gaussian random variables with probability density functions

$$p_{x_1}(x) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-x^2/(2\sigma_1^2)} \qquad \text{and} \qquad p_{x_2}(x) = \frac{1}{\sqrt{2\pi}\sigma_2} e^{-x^2/(2\sigma_2^2)}$$

Let

$$y = x_1 + x_2$$

The probability density function of y is therefore

$$p_y(y) = \int_{-\infty}^{\infty} p_{x_1}(x)p_{x_2}(y-x)\,dx$$

Upon carrying out this convolution (integration), we have

$$p_y(y) = \frac{1}{2\pi\sigma_1\sigma_2}\int_{-\infty}^{\infty}\exp\left(-\frac{x^2}{2\sigma_1^2}-\frac{(y-x)^2}{2\sigma_2}\right)dx$$

$$= \frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}}e^{-\frac{y^2}{2(\sigma_1^2+\sigma_2^2)}}\frac{1}{\sqrt{2\pi}\frac{\sigma_1\sigma_2}{\sqrt{\sigma_1^2+\sigma_1^2}}}\int_{-\infty}^{\infty}\exp\left(-\frac{1}{2\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}}\left[x-\frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}y\right]^2\right)dx \tag{8.94}$$

By a simple change of variable

$$w = \frac{\left[x-\frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}y\right]}{\frac{\sigma_1\sigma_2}{\sqrt{\sigma_1^2+\sigma_2^2}}}$$

we can rewrite the integral of Eq. (8.94) as

$$p_y(y) = \frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}}e^{-\frac{y^2}{2(\sigma_1^2+\sigma_2^2)}}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}e^{-\frac{1}{2}w^2}dw = \frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}}e^{-\frac{y^2}{2(\sigma_1^2+\sigma_2^2)}} \tag{8.95}$$

By examining Eq. (8.95), it can be seen that y is a Gaussian RV with zero mean and variance:

$$\sigma_y^2 = \sigma_1^2+\sigma_2^2$$

In fact, because x_1 and x_2 are independent, they must be uncorrelated. This relationship can be obtained from Eq. (8.81).

More generally,[5] if x_1 and x_2 are jointly Gaussian but not necessarily independent, then $y = x_1 + x_2$ is Gaussian RV with mean

$$\overline{y} = \overline{x_1} + \overline{x_2}$$

and variance

$$\sigma_y^2 = \sigma_{x_1}^2 + \sigma_{x_2}^2 + 2\sigma_{x_1x_2}$$

Based on induction, the sum of any number of jointly Gaussian distributed RV's is still Gaussian. More importantly, for any fixed constants $\{a_i,\ i = 1, \ldots, m\}$ and jointly Gaussian RVs $\{x_i,\ i = 1, \ldots, m\}$,

$$\sum_{i=1}^{m} a_i x_i$$

remains Gaussian. This result has important practical implications. For example, if x_k is a sequence of jointly Gaussian signal samples passing through a discrete time filter with impulse response $\{h_i\}$, then the filter output

$$y = \sum_{i=0}^{\infty} h_i x_{k-i} \tag{8.96}$$

will continue to be Gaussian. The fact that linear filter output to a Gaussian signal input will be a Gaussian signal is highly significant and is one of the most useful results in communication analysis.

8.7 CENTRAL LIMIT THEOREM

Under certain conditions, the sum of a large number of independent RVs tends to be a Gaussian random variable, independent of the probability densities of the variables added.* The rigorous statement of this tendency is what is known as the **central limit theorem.**† Proof of this theorem can be found in the Refs. 6 and 7. We shall give here only a simple plausibility argument.

The tendency toward a Gaussian distribution when a large number of functions are convolved in shown in Fig. 8.20. For simplicity, we assume all PDFs to be identical, that is, a gate function $0.5\,\Pi(x/2)$. Figure 8.20 shows the successive convolutions of gate functions. The tendency toward a bell-shaped density is evident.

This important result that the **distribution** of the sum of n independent Bernoulli random variables, when properly normalized, converges toward Gaussian distribution was established first by A. de Moivre in the early 1700s. The more general proof for an arbitrary distribution was credited to J. W. Lindenber and P. Lévy in the 1920s. Note that the "normalized sum" is the sample average (or sample mean) of n random variables.

Central Limit Theorem (for the sample mean):
Let $\mathrm{x}_1, \ldots, \mathrm{x}_n$ *be independent random samples from a given distribution with mean* μ *and variance* σ^2 *with* $0 < \sigma^2 < \infty$. *Then for any value x, we have*

$$\lim_{n\to\infty} P\left[\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\frac{\mathrm{x}_i - \mu}{\sigma} \le x\right] = \int_{-\infty}^{x}\frac{1}{\sqrt{2\pi}}e^{-v^2/2}\,dv \tag{8.97}$$

or equivalently,

$$\lim_{n\to\infty} P\left[\frac{\overset{\sim}{\mathrm{x}}_n - \mu}{\sigma/\sqrt{n}} > x\right] = Q(x) \tag{8.98}$$

Note that

$$\overset{\sim}{\mathrm{x}}_n = \frac{\mathrm{x}_1 + \cdots + \mathrm{x}_n}{n}$$

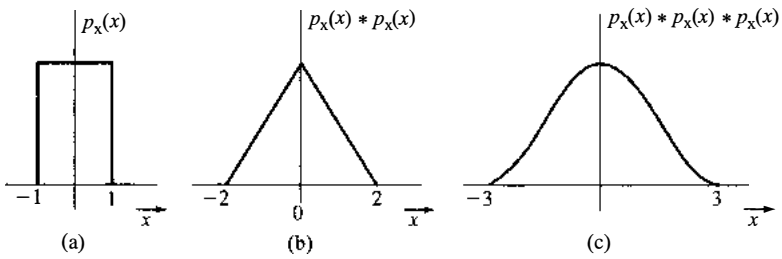


Figure 8.20 Demonstration of the central limit theorem.

* If the variables are Gaussian, this is true even if the variables are not independent.
† Actually, a group of theorems collectively called the central limit theorem.

is known as the sample mean. The interpretation is that the sample mean of any distribution with nonzero finite variance converges to Gaussian distribution with fixed mean μ and decreasing variance σ^2/n. In other words, regardless of the true distribution of x_i, $\sum_{i=1}^{n} x_i$ can be approximated by a Gaussian distribution with mean $n\mu$ and variance $n\sigma^2$.

Example 8.25 Consider a communication system that transmits a data packet of 1024 bits. Each bit can be in error with probability of 10^{-2}. Find the (approximate) probability that more than 30 of the 1024 bits are in error.

Define a random variable x_i such that $x_i = 1$ if the ith bit is in error and $x_i = 0$ if not. Hence

$$v = \sum_{i=1}^{1024} x_i$$

is the number of errors in the data packet. We would like to find $P(v > 30)$.

Since $P(x_i = 1) = 10^{-2}$ and $P(x_i = 0) = 1 - 10^{-2}$, strictly speaking we would need to find

$$P(v > 30) = \sum_{m=31}^{1024} \binom{1024}{m} \left(10^{-2}\right)^m \left(1 - 10^{-2}\right)^{1024-m}$$

This calculation is time-consuming. We now apply the central limit theorem to solve this problem approximately.

First, we find

$$\overline{x_i} = 10^{-2} \times (1) + (1 - 10^{-2}) \times (0) = 10^{-2}$$
$$\overline{x_i^2} = 10^{-2} \times (1)^2 + (1 - 10^{-2}) \times (0) = 10^{-2}$$

As a result,

$$\sigma_i^2 = \overline{x_i^2} - (\overline{x_i})^2 = 0.0099$$

Based on the central limit theorem, $v = \sum_i x_i$ is approximately Gaussian with mean of $1024 \cdot 10^{-2} = 10.24$ and variance $1024 \times 0.0099 = 10.1376$. Since

$$y = \frac{v - 10.24}{\sqrt{10.1376}}$$

is a standard Gaussian with zero mean and unit variance,

$$\begin{aligned} P(v > 30) &= P\left(y > \frac{30 - 10.24}{\sqrt{10.1376}}\right) \\ &= P(y > 6.20611) \\ &= Q(6.20611) \\ &\simeq 1.925 \times 10^{-10} \end{aligned}$$

Now is a good time to further relax the conditions in the central limit theorem for the sample mean. This highly important generalization is proved by the famous Russian mathematician A. Lyapunov in 1901.

Central Limit Theorem (for the sum of independent random variables): *Let random variables* $x_1, \ldots, x_n$ *be independent but not necessarily identically distributed. Each of the random variable* x_i *has mean* μ_i *and nonzero variance* $\sigma_i^2 < \infty$. *Furthermore, suppose that each third-order central moment*

$$\overline{|x_i - \mu_i|^3} < \infty, \qquad i = 1, \ldots, n$$

and suppose

$$\lim_{n\to\infty} \sum_{i=1}^{n} \overline{|x_i - \mu_i|^3} \left(\sum_{i=1}^{n} \sigma_i^2 \right)^{3/2} = 0$$

Then random variable

$$y(n) = \frac{\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} \mu_i}{\sqrt{\sum_{i=1}^{n} \sigma_i^2}}$$

converges to a standard Gaussian density as $n \to \infty$, *that is,*

$$\lim_{n\to\infty} P[y(n) > x] = Q(x) \tag{8.99}$$

The central limit theorem provides a plausible explanation for the well-known fact that many random variables in practical experiments are approximately Gaussian. For example, communication channel noise is the sum effect of many different random disturbance sources (e.g., sparks, lightning, static electricity). Based on the central limit theorem, noise as the sum of all these random disturbances should be approximately Gaussian.

REFERENCES

1. J. Singh, *Great Ideas of Modern Mathematics*, Dover, Boston, 1959.
2. M. Abramowitz and I. A. Stegun, Eds., *Handbook of Mathematical Functions*, National Bureau of Standards, Washington, DC, 1964, sec. 26.
3. The Chemical Rubber Co., *CRC Standard Mathematical Tables*, 26th ed., 1980.
4. J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, Wiley, New York, 1965, p. 83.
5. N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Prentice-Hall, 1984.
6. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed., McGraw-Hill, New York, 1995.
7. M. H. DeGroot, *Probabilities and Statistics*, 2nd ed., Addison Wesley, Reading, MA, 1987.

PROBLEMS

8.1-1 A card is drawn randomly from a regular deck of cards. Assign probability to the event that the card drawn is: **(a)** a red card; **(b)** a black queen; **(c)** a picture card (count an ace as a picture card); **(d)** a number card with number 7; **(e)** a number card with number ≤ 5.

8.1-2 Three regular dice are thrown. Assign probabilities to the following events: the sum of the points appearing on the three dice is **(a)** 4; **(b)** 9; **(c)** 15.

8.1-3 The probability that the number i appears on a throw of a certain loaded dice is k_i ($i = 1, 2, \ldots, 6$). Assign probabilities to all six outcomes.

8.1-4 A bin contains three oscillator microchips, marked O_1, O_2, and O_3, and two PLL microchips, marked P_1 and P_2. Two chips are picked randomly in succession without replacement.

(a) How many outcomes are possible (i.e., how many points are in the sample space)? List all the outcomes and assign probabilities to each of them.

(b) Express the following events as unions of the outcomes in part **(a)**: **(i)** one chip drawn is marked oscillator and the other PLL **(ii)** both chips are PLL; **(iii)** both chips are oscillators; and **(iv)** both chips are of the same kind. Assign probabilities to each of these events.

8.1-5 Use Eq. (8.12) to find the probabilities in Prob. 8.1-4, part **(b)**.

8.1-6 In Prob. 8.1-4, determine the probability that:

(a) The second pick is an oscillator chip given that the first pick is a PLL chip.

(b) The second pick is an oscillator chip given that the first pick is also an oscillator chip.

8.1-7 A binary source generates digits **1** and **0** randomly with equal probability. Assign probabilities to the following events with respect to 10 digits generated by the source: **(a)** there are exactly two **1**s and eight **0**s; **(b)** there are at least four **0**s.

8.1-8 In the California lottery (Lotto), a player chooses any 6 numbers out of 49 numbers (1 through 49). Six balls are drawn randomly (without replacement) from the 49 balls numbered 1 through 49.

(a) Find the probability of matching all 6 balls to the 6 numbers chosen by the player.

(b) Find the probability of matching exactly 5 balls.

(c) Find the probability of matching exactly 4 balls.

(d) Find the probability of matching exactly 3 balls.

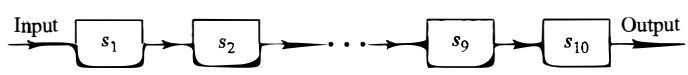
8.1-9 A network consists of 10 links $s_1, s_2, \ldots, s_{10}$ in cascade (Fig. P8.1-9). If any one of the links fails, the entire system fails. All links are independent, with equal probability of failure $p = 0.01$.

(a) What is the probability of failure of the network?

Hint: Consider the probability that none of the links fails.

(b) The reliability of a network is the probability of not failing. If the system reliability is required to be 0.99, what must be the failure probability of each link?

Figure P.8.1-9

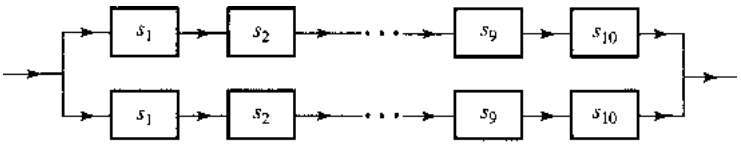


8.1-10 Network reliability improves when redundant links are used. The reliability of the network in Prob. 8.1-9 (Fig. P8.1-9) can be improved by building two subnetworks in parallel (Fig. P8.1-9). Thus, if one subnetwork fails, the other one will still connect.

(a) Using the data in Prob. 8.1-9, determine the reliability of the network in Fig. P8.1-10.

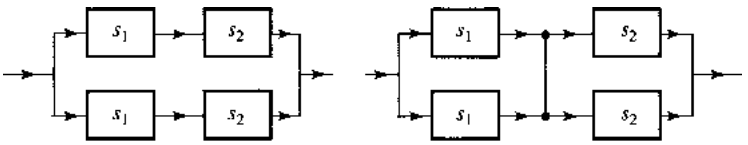
(b) If the reliability of this new network is required to be 0.999, what must be the failure probability of each link?

Figure P.8.1-10



8.1-11 Compare the reliability of the two networks in Fig. P8.1-11, given that the failure probability of links s_1 and s_2 is p each.

Figure P.8.1-11



8.1-12 In a poker game each player is dealt five cards from a regular deck of 52 cards. What is the probability that a player will get a flush (all five cards of the same suit)?

8.1-13 Two dice are thrown. One die is regular and the other is biased with the following probabilities:

$$P(1) = P(6) = \frac{1}{6}, \qquad P(2) = P(4) = 0, \qquad P(3) = P(5) = \frac{1}{3}$$

Determine the probabilities of obtaining a sum: **(a)** 4; **(b)** 5.

8.1-14 In Sec. 8.1, Example 8.5, determine:

(a) $P(B)$, the probability of drawing an ace in the second draw.

(b) $P(A|B)$, the probability that the first draw was a red ace given that the second draw is an ace.

Hint: Event B can occur in two ways: the first draw is a red ace and the second draw is an ace, or the first draw is not a red ace and the second draw is an ace. This is $A \cap B \cup A^c B$ (see Fig. 8.2).

8.1-15 A binary source generates digits **1** and **0** randomly with probabilities $P(1) = 0.8$ and $P(0) = 0.2$.

(a) What is the probability that exactly two **1**s will occur in a n-digit sequence?

(b) What is the probability that at least three **1**s will occur in a n-digit sequence?

8.1-16 In a binary communication channel, the receiver detects binary pulses with an error probability P_e. What is the probability that out of 100 received digits, no more than four digits are in error?

8.1-17 A PCM channel consists of 10 links, with a regenerative repeater at the end of each link. If the detection error probabilities of the 15 detectors are $p_1, p_2, \ldots, p_{15}$, determine the detection error probability of the entire channel if $p_i \ll 1$.

8.1-18 Example 8.8 considers the possibility of improving reliability by repeating a digit three times. Repeat this analysis for five repetitions.

8.1-19 A box contains nine bad microchips. A good microchip is thrown into the box by mistake. Someone is trying to retrieve the good chip. He draws a chip randomly and tests it. If the chip is bad, he throws it out and draws another chip randomly, repeating the procedure until he finds the good chip.

(a) What is the probability that he will find the good chip in the first trial?

(b) What is the probability that he will find the good chip in five trials?

8.1-20 One out of a group of 10 people is to be selected for a suicide mission by drawing straws. There are 10 straws: nine are of the same length and the tenth is shorter than the others. Each of the 10 people draws a straw, one by one. The person who draws the short straw is selected for the mission. Determine which position in the sequence favors the most and which favors the least drawing the short straw.

8.2-1 For a certain binary nonsymmetric channel it is given that

$$P_{\mathrm{y}|\mathrm{x}}(\mathbf{0}|\mathbf{1}) = 0.1 \qquad \text{and} \qquad \bar{P}_{\mathrm{y}|\mathrm{x}}(\mathbf{1}|\mathbf{0}) = 0.2$$

where x is the transmitted digit and y is the received digit. If $P_{\mathrm{x}}(\mathbf{0}) = 0.4$, determine $P_{\mathrm{y}}(\mathbf{0})$ and $P_{\mathrm{y}}(\mathbf{1})$.

8.2-2 A binary symmetric channel (see Example 8.13) has an error probability P_e. The probability of transmitting **1** is Q. If the receiver detects an incoming digit as **1**, what is the probability that the originally transmitted digit was: **(a) 1**; **(b) 0**?

Hint: If x is the transmitted digit and y is the received digit, you are given $P_{\mathrm{y}|\mathrm{x}}(\mathbf{0}|\mathbf{1}) = P_{\mathrm{y}|\mathrm{x}}(\mathbf{1}|\mathbf{0}) = P_e$. Now using Bayes' rule, find $P_{\mathrm{x}|\mathrm{y}}(\mathbf{1}|\mathbf{1})$ and $P_{\mathrm{x}|\mathrm{y}}(\mathbf{0}|\mathbf{1})$.

8.2-3 The PDF of amplitude x of a certain signal $x(t)$ is given by $p_{\mathrm{x}}(x) = 0.5|x|e^{-|x|}$.

(a) Find the probability that $\mathrm{x} \geq 1$.

(b) Find the probability that $-1 < \mathrm{x} \leq 2$.

(c) Find the probability that $\mathrm{x} \leq -2$.

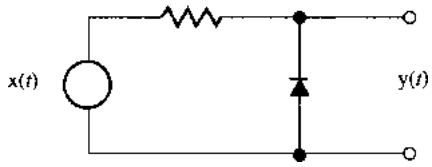
8.2-4 The PDF of an amplitude x of a Gaussian signal $\mathrm{x}(t)$ is given by

$$p_{\mathrm{x}}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2}$$

This signal is applied to the input of a half-wave rectifier circuit (Fig. P8.2-4).

Assuming an ideal diode, determine $F_{\mathrm{y}}(y)$ and $p_{\mathrm{y}}(y)$ of the output signal amplitude $\mathrm{y} = \mathrm{x} \cdot u(\mathrm{x})$. Notice that the probability of $\mathrm{x} = 0$ is not zero.

Figure P.8.2-4



8.2-5 The PDF of a Gaussian variable x is given by

$$p_{\mathrm{x}}(x) = \frac{1}{3\sqrt{2\pi}} e^{-(x-4)^2/18}$$

Determine: **(a)** $P(\mathrm{x} \geq 4)$; **(b)** $P(\mathrm{x} \leq 0)$; **(c)** $P(\mathrm{x} \geq -2)$.

8.2-6 For an RV x with PDF

$$p_{\mathrm{x}}(x) = \frac{1}{2\sqrt{2\pi}} e^{-x^2/32} u(x)$$

(a) Sketch $p_{\mathrm{x}}(x)$, and state (with reasons) if this is a Gaussian RV.

(b) Determine: **(i)** $P(\mathrm{x} \geq 1)$, **(ii)** $P(1 < \mathrm{x} \leq 2)$.

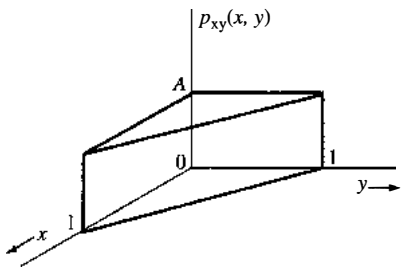
(c) How to generate RV x from another Gaussian RV? Show block diagram and explain.

8.2-7 The joint PDF of RVs x and y is shown in Fig. P8.2-7.

(a) Determine: **(i)** A; **(ii)** $p_{\mathrm{x}}(x)$; **(iii)** $p_{\mathrm{y}}(y)$; **(iv)** $P_{\mathrm{x|y}}(x|y)$; **(v)** $P_{\mathrm{y|x}}(y|x)$.

(b) Are x and y independent? Explain.

Figure P.8.2-7



8.2-8 The joint PDF $p_{\mathrm{xy}}(x, y)$ of two continuous RVs is given by

$$p_{\mathrm{xy}}(x, y) = xy\, e^{-(x^2)} e^{-y^2/2} u(x) u(y)$$

(a) Find $p_{\mathrm{x}}(x)$, $p_{\mathrm{y}}(y)$, $p_{\mathrm{x|y}}(x|y)$, and $p_{\mathrm{y|x}}(y|x)$.

(b) Are x and y independent?

8.2-9 RVs x and y are said to be jointly Gaussian if their joint PDF is given by

$$p_{\mathrm{xy}}(x, y) = \frac{1}{2\pi\sqrt{M}} e^{-(ax^2+by^2-2cxy)/2M}$$

where $M = ab - c^2$. Show that $p_{\mathrm{x}}(x)$, $p_{\mathrm{y}}(y)$, $p_{\mathrm{x}|\mathrm{y}}(x|y)$, and $p_{\mathrm{y}|\mathrm{x}}(y|x)$ are all Gaussian and that $\overline{\mathrm{x}^2} = b$, $\overline{\mathrm{y}^2} = a$, and $\overline{\mathrm{xy}} = c$.

Hint: Use

$$\int_{-\infty}^{\infty} e^{-px^2+qx}\,dx = \sqrt{\frac{\pi}{p}}e^{q^2/4p}$$

8.2-10 The joint PDF of RVs x and y is given by

$$p_{\mathrm{xy}}(x, y) = ke^{-(x^2+xy+y^2)}$$

Determine: **(a)** the constant k; **(b)** $p_{\mathrm{x}}(x)$; **(c)** $p_{\mathrm{y}}(y)$; **(d)** $p_{\mathrm{x}|\mathrm{y}}(x, y)$; **(e)** $p_{\mathrm{y}|\mathrm{x}}(y|x)$. Are x and y independent?

8.2-11 In the example on threshold detection (Example 8.16), it was assumed that the digits **1** and **0** were transmitted with equal probability. If $P_{\mathrm{x}}(\mathbf{1})$ and $P_{\mathrm{x}}(\mathbf{0})$, the probabilities of transmitting **1** and **0**, respectively, are not equal, show that the optimum threshold is not 0 but is a, where

$$a = \frac{\sigma_n^2}{2A_p}\ln\frac{P_{\mathrm{x}}(\mathbf{0})}{P_x(1)}$$

Hint: Assume that the optimum threshold is a, and write P_e in terms of the Q functions. For the optimum case, $dP_e/da = 0$. Use the fact that

$$Q(x) = 1 - \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-y^2/2}\,dy$$

and

$$\frac{dQ(x)}{dx} = -\frac{1}{\sqrt{2\pi}}e^{-x^2/2}$$

8.3-1 If an amplitude x of a Gaussian signal $\mathrm{x}(t)$ has a mean value of 2 and an RMS value of 3, determine its PDF.

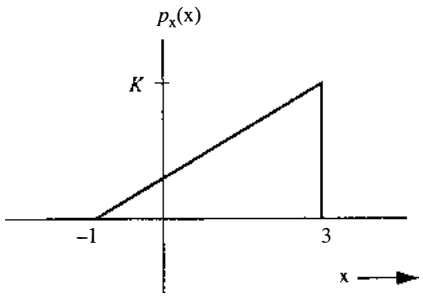
8.3-2 Determine the mean, the mean square, and the variance of the RV x in Prob. 8.2-3.

8.3-3 Determine the mean and the mean square value of RV x in Prob. 8.2-4.

8.3-4 Determine the mean and the mean square value of RV x in Prob. 8.2-6.

8.3-5 Find the mean, the mean square, and the variance of the RV x in Fig. P8.3-5.

Figure P.8.3-5



8.3-6 The sum of points on two tossed dice is a discrete RV x, as analyzed in Example 8.12. Determine the mean, the mean square, and the variance of the RV x.

8.3-7 For a Gaussian PDF $p_{\mathrm{x}}(x) = (1/\sigma_{\mathrm{x}}\sqrt{2\pi})e^{-x^2/2\sigma_{\mathrm{x}}^2}$, show that

$$\overline{\mathrm{x}^n} = \begin{cases} (1)(3)(5)\cdots(n-1)\sigma_{\mathrm{x}}^n & n \text{ even} \\ 0 & n \text{ odd} \end{cases}$$

Hint: See appropriate definite integrals in any standard mathematical table.

8.3-8 Ten regular dice are thrown. The sum of the numbers appearing on these 10 dice is an RV x. Find $\overline{\mathrm{x}}$, $\overline{\mathrm{x}^2}$, and σ_{x}^2.

Hint: Remember that the outcome of each die is independent.

8.5-1 Show that $|\rho_{\mathrm{xy}}| \leq 1$, where ρ_{xy} is the correlation coefficient [Eq. (8.79)] of RVs x and y.

Hint: For any real number a,

$$\overline{[a(\mathrm{x} - \bar{\mathrm{x}}) - (\mathrm{y} - \bar{\mathrm{y}})]^2} \geq 0$$

The discriminant of this quadratic in a is nonpositive.

8.5-2 Show that if two RVs x and y are related by

$$\mathrm{y} = k_1\mathrm{x} + k_2$$

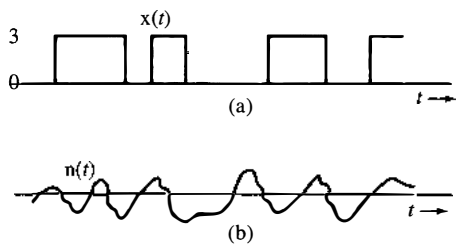
where k_1 and k_2 are arbitrary constants, the correlation coefficient $\rho_{\mathrm{xy}} = 1$ if k_1 is positive, and $\rho_{\mathrm{xy}} = -1$ if k_1 is negative.

8.5-3 Given $\mathrm{x} = \cos\Theta$ and $\mathrm{y} = \sin\Theta$, where Θ is an RV uniformly distributed in the range $(0, 2\pi)$, show that x and y are uncorrelated but are not independent.

8.6-1 The random binary signal $\mathrm{x}(t)$, shown in Fig. P8.6-1a, can take on only two values, 3 and 0, with equal probability. An exponential channel noise $\mathrm{n}(t)$ shown in Fig. P8.6-1b is added to this signal, giving the received signal $\mathrm{y}(t)$. The PDF of the noise amplitude n is exponential with a zero mean and a variance of 2. Determine and sketch the PDF of the amplitude y.

Hint: Use of Eq. (8.92) yields $p_{\mathrm{y}}(y) = p_{\mathrm{x}}(x) * p_{\mathrm{n}}(n)$.

Figure P.8.6-1



8.6-2 Repeat Prob. 8.6-1 if the amplitudes 3 and 0 of $\mathrm{x}(t)$ are not equiprobable but $P_{\mathrm{x}}(3) = 0.6$ and $P_{\mathrm{x}}(0) = 0.4$.

8.6-3 If $x(t)$ and $y(t)$ are both independent binary signals, each taking on values -1 and 1 only, with

$$P_x(1) = Q = 1 - P_x(-1)$$
$$P_y(1) = P = 1 - P_y(-1)$$

determine $P_z(z_i)$ where $z = x + y$.

8.6-4 If $z = x + y$, where x and y are independent Gaussian RVs with

$$p_x(x) = \frac{1}{\sigma_x\sqrt{2\pi}} e^{-(x-\bar{x})^2/2\sigma_x^2} \quad \text{and} \quad p_y(y) = \frac{1}{\sigma_y\sqrt{2\pi}} e^{-(y-\bar{y})^2/2\sigma_y^2}$$

then show that z is also Gaussian with

$$\bar{z} = \bar{x} + \bar{y} \quad \textit{and} \quad \sigma_z^2 = \sigma_x^2 + \sigma_y^2$$

Hint: Convolve $p_x(x)$ and $p_y(y)$. See pair 22 in Table 3.1.

8.6-5 In Example 8.24, design the optimum third-order predictor processor for speech signals and determine the SNR improvement. Values of various correlation coefficients for speech signals are given in Example 8.24.

9 RANDOM PROCESSES AND SPECTRAL ANALYSIS

The notion of a random process is a natural extension of the random variable (RV). Consider, for example, the temperature x of a certain city at noon. The temperature x is an RV and takes on different values every day. To get the complete statistics of x, we need to record values of x at noon over many days (a large number of trials). From this data, we can determine $p_{\mathrm{x}}(x)$, the PDF of the RV x (the temperature at noon).

But the temperature is also a function of time. At 1 p.m., for example, the temperature may have an entirely different distribution from that of the temperature at noon. Still, the two temperatures may be related, via a joint probability density function. Thus, this random temperature x is a function of time and can be expressed as $\mathrm{x}(t)$. If the random variable is defined for a time interval $t \in [t_a, t_b]$, then $\mathrm{x}(t)$ is a function of time and is random for every instant $t \in [t_a, t_b]$. An RV that is a function of time* is called a **random process**, or **stochastic process**. Thus, a random process is a collection of an infinite number of RVs. Communication signals as well as noises, typically random and varying with time, are well characterized by random processes. For this reason, random process is the subject of this chapter before we study the performance analysis of different communication systems.

9.1 FROM RANDOM VARIABLE TO RANDOM PROCESS

To specify an RV x, we run multiple trials of the experiment and from the outcomes estimate $p_{\mathrm{x}}(x)$. Similarly, to specify the random process $\mathrm{x}(t)$, we do the same thing for each time instant t. To continue with our example of the random process $\mathrm{x}(t)$, the temperature of the city, we need to record daily temperatures for each value of t (for each time of the day). This can be done by recording temperatures at every instant of the day, which gives a waveform $x(t, \zeta_i)$, where ζ_i indicates the day for which the record was taken. We need to repeat this procedure every day for a large number of days. The collection of all possible waveforms is known as the **ensemble** (corresponding to the sample space) of the random process $\mathrm{x}(t)$. A waveform in this collection is a **sample function** (rather than a sample point) of the random process (Fig. 9.1).

* Actually, to qualify as a random process, x could be a function of any practical variable, such as distance. In fact, a random process may also be a function of more than one variable.

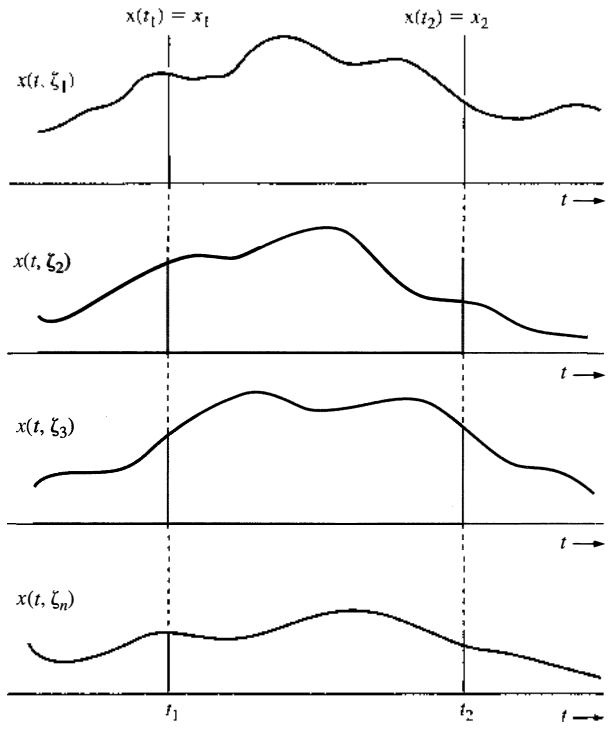


Figure 9.1 Random process to represent the temperature of a city.

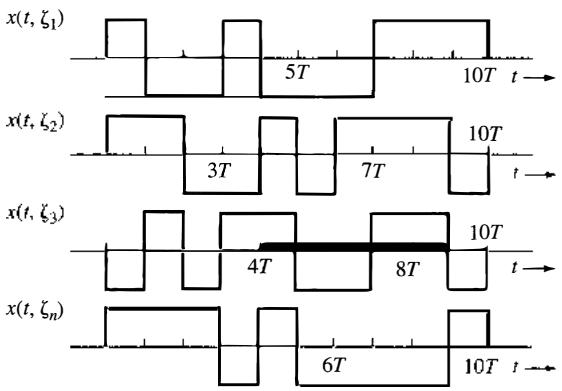


Figure 9.2 Ensemble with a finite number of sample functions.

Sample function amplitudes at some instant $t = t_1$ are the values taken by the RV $x(t_1)$ in various trials.

We can view a random process in another way. In the case of an RV, the outcome of each trial of the experiment is a number. We can view a random process also as the outcome of an experiment, where the outcome of each trial is a waveform (a sample function) that is a function of t. The number of waveforms in an ensemble may be finite or infinite. In the case of the random process $x(t)$ (the temperature of a city), the ensemble has infinitely many waveforms. On the other hand, if we consider the output of a binary signal generator (over the period 0 to $10T$), there are at most 2^{10} waveforms in this ensemble (Fig. 9.2).

One fine point that needs clarification is that the waveforms (or sample functions) in the ensemble are not random. They have occurred and are therefore deterministic. Randomness in this situation is associated not with the waveform but with the uncertainty as to which

waveform would occur in a given trial. This is completely analogous to the situation of an RV. For example, in the experiment of tossing a coin four times in succession (Example 8.4), 16 possible outcomes exist, all of which are known. The randomness in this situation is associated not with the outcomes but with the uncertainty as to which of the 16 outcomes will occur in a given trial. Indeed, the random process is basically an infinite long vector of random variables. Once an experiment is completed, the sampled vector is deterministic. However, since each element in the vector is random, the experimental outcome is also random, leading to uncertainty over what vector (or function) will be generated in each experiment.

Characterization of a Random Process

The next important question is how to characterize (describe) a random process. In some cases, we may be able to describe it analytically. Consider, for instance, a random process described by $\mathrm{x}(t) = A\cos(\omega_c t + \Theta)$, where Θ is an RV uniformly distributed over the range $(0, 2\pi)$. This analytical expression completely describes a random process (and its ensemble). Each sample function is a sinusoid of amplitude A and frequency ω_c. But the phase is random (see later, Fig. 9.5). It is equally likely to take any value in the range $(0, 2\pi)$. Such an analytical description requires well-defined models such that the random process is characterized by specific parameters that are random variables.

Unfortunately, it is not always possible to be able to describe a random process analytically. Without a specific model, we may have just an ensemble obtained experimentally. The ensemble has the complete information about the random process. From this ensemble, we must find some quantitative measure that will specify or characterize the random process. In this case, we consider the random process as an RV x that is a function of time. Thus, a random process is just a collection of an infinite number of RVs, which are generally dependent. We know that the complete information of several dependent RVs is provided by the joint PDF of those variables. Let x_i represent the RV $\mathrm{x}(t_i)$ generated by the amplitudes of the random process at instant $t = t_i$. Thus, x_1 is the RV generated by the amplitudes at $t = t_1$, and x_2 is the RV generated by the amplitudes at $t = t_2$, and so on, as shown in Fig. 9.1. The n RVs $\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \ldots, \mathrm{x}_n$ generated by the amplitudes at $t = t_1, t_2, t_3, \ldots, t_n$, respectively, are dependent in general. For the n samples, they are fully characterized by the n th-order joint probability density function or the n th-order joint cumulative distribution function (CDF)

$$F_{\mathrm{x}}(x_1, x_2, \ldots, x_n; t_1, t_2, \ldots, t_n) = P[\mathrm{x}(t) \le x_1; \mathrm{x}(t) \le x_2; \ldots; \mathrm{x}(t_n) \le x_n]$$

The definition of the joint CDF of the n random samples leads to the joint PDF

$$p_{\mathrm{x}}(x_1, x_2, \ldots, x_n; t_1, t_2, \ldots, t_n) = \frac{\partial^n}{\partial x_1 \partial x_2 \ldots \partial x_n} F_{\mathrm{x}}(x_1, x_2, \ldots, x_n; t_1, t_2, \ldots, t_n) \tag{9.1}$$

This discussion provides some good insight. It can be shown that the random process is completely described by the n th-order joint PDF (9.1) for all n (up to ∞) and for any choice of $t_1, t_2, t_3, \ldots, t_n$. Determining this PDF (of infinite order) is a formidable task. Fortunately, we shall soon see that when analyzing random signals and noises in conjunction with linear systems, we are often content with the specifications of the first- and second-order statistics.

A higher order PDF is the joint PDF of the random process at multiple time instants. Hence, we can always derive a lower order PDF from a higher order PDF by simple integration. For instance,

$$p_{\mathrm{x}}(x_1; t_1) = \int_{-\infty}^{\infty} p_{\mathrm{x}}(x_1, x_2; t_1, t_2)\, dx_2$$

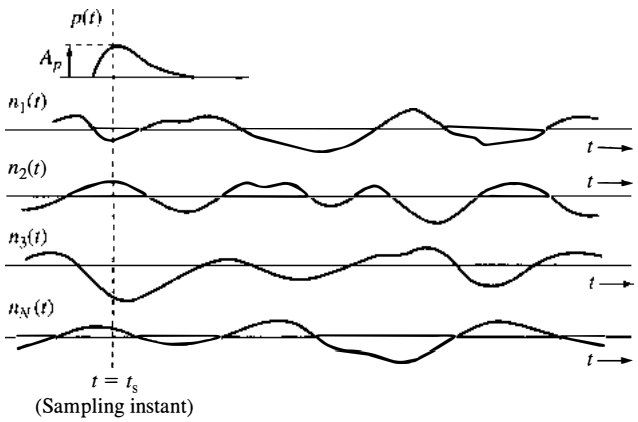


Figure 9.3 A random process to represent a channel noise.

Hence, when the nth-order PDF is available, there is no need to specify PDFs of order lower than n.

The mean $\overline{\mathrm{x}(t)}$ of a random process x(t) can be determined from the first-order PDF as

$$\overline{\mathrm{x}(t)} = \int_{-\infty}^{\infty} x\, p_{\mathrm{x}}(x;\ t)\, dx \tag{9.2}$$

which is typically a deterministic function of time t.

Why Do We Need Ensemble Statistics?

The preceding discussion shows that to specify a random process, we need ensemble statistics. For instance, to determine the PDF $p_{\mathrm{x}_1}(x_1)$, we need to find the values of all the sample functions at $t = t_1$. This is ensemble statistics. In the same way, the inclusion of all possible statistics in the specification of a random process necessitates some kind of ensemble statistics. In deterministic signals, we are used to studying the data of a waveform (or waveforms) as a function of time. Hence, the idea of investigating ensemble statistics makes us feel a bit uncomfortable at first. Theoretically, we may accept it, but does it have any practical significance? How is this concept useful in practice? We shall now answer this question.

To understand the necessity of ensemble statistics, consider the problem of threshold detection in Example 8.16. A **1** is transmitted by $p(t)$ and a **0** is transmitted by $-p(t)$ (polar signaling). The peak pulse amplitude is A_p. When **1** is transmitted, the received sample value is $A_p + \mathrm{n}$, where n is the noise. We would make a decision error if the noise value at the sampling instant t_s were less than $-A_p$, forcing the sum of signal and noise to fall below the threshold. To find this error probability, we repeat the experiment N times ($N \to \infty$) and see how many times the noise at $t = t_s$ is less than $-A_p$ (Fig. 9.3). This information is precisely one of ensemble statistics of the noise process $n(t)$ at instant t_s.

The importance of ensemble statistics is clear from this example. When we are dealing with a random process or processes, we do not know which sample function will occur in a given trial. Hence, for any statistical specification and characterization of the random process, we need to average over the entire ensemble. This is the basic physical reason for the appearance of ensemble statistics in random processes.

Autocorrelation Function of a Random Process

For the purpose of signal analysis, one of the most important (statistical) characteristics of a random process is its **autocorrelation function**, which leads to the spectral information of the

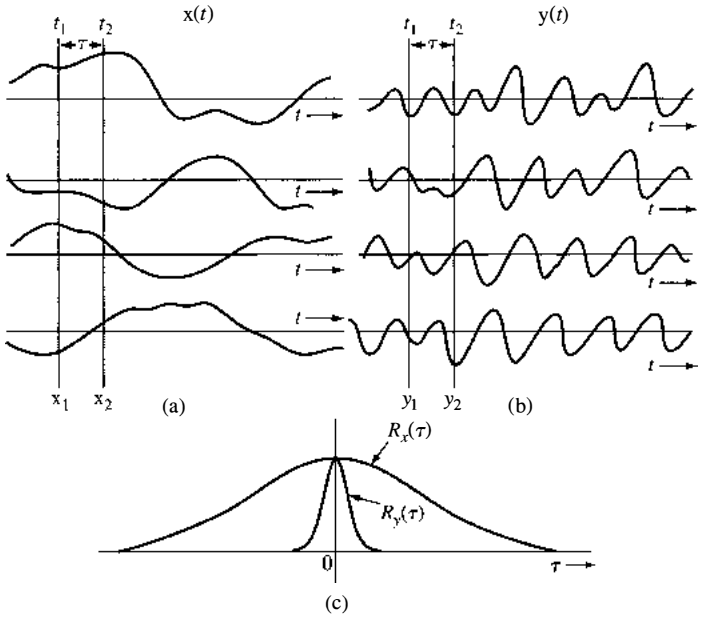


Figure 9.4 Autocorrelation functions for a slowly varying and a rapidly varying random process.

random process. The spectral content of a process depends on the rapidity of the amplitude change with time. This can be measured by correlating amplitudes at t_1 and $t_1 + \tau$. On average, the random process $x(t)$ in Fig. 9.4a is a slowly varying process in comparison to the process $y(t)$ in Fig. 9.4b. For $x(t)$, the amplitudes at t_1 and $t_1 + \tau$ are similar (Fig. 9.4a), that is, have stronger correlation. On the other hand, for $y(t)$, the amplitudes at t_1 and $t_1 + \tau$ have little resemblance (Fig. 9.4b), that is, have weaker correlation. Recall that correlation is a measure of the similarity of two RVs. Hence, we can use correlation to measure the similarity of amplitudes at t_1 and $t_2 = t_1 + \tau$. If the RVs $x(t_1)$ and $x(t_2)$ are denoted by x_1 and x_2, respectively, then for a real random process,* the autocorrelation function $R_x(t_1, t_2)$ is defined as

$$R_x(t_1, t_2) = \overline{x(t_1)x(t_2)} = \overline{x_1 x_2} \tag{9.3a}$$

This is the correlation of RVs $x(t_1)$ and $x(t_2)$, indicating the similarity between RVs $x(t_1)$ and $x(t_2)$. It is computed by multiplying amplitudes at t_1 and t_2 of a sample function and then averaging this product over the ensemble. It can be seen that for a small τ, the product x_1x_2 will be positive for most sample functions of $x(t)$, but the product y_1y_2 is equally likely to be positive or negative. Hence, $\overline{x_1x_2}$ will be larger than $\overline{y_1y_2}$. Moreover, x_1 and x_2 will show correlation for considerably larger values of τ, whereas y_1 and y_2 will lose correlation quickly, even for small τ, as shown in Fig. 9.4c. Thus, $R_x(t_1, t_2)$, the autocorrelation function of $x(t)$, provides valuable information about the frequency content of the process. In fact, we shall show that the PSD of $x(t)$ is the Fourier transform of its autocorrelation function, given by (for real processes)

$$\begin{aligned} R_x(t_1, t_2) &= \overline{x_1 x_2} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1 x_2 p_x(x_1, x_2; t_1, t_2)\, dx_1\, dx_2 \end{aligned} \tag{9.3b}$$

* For a complex random process $x(t)$, the autocorrelation function is defined as

$$R_x(t_1, t_2) = \overline{x^*(t_1)x(t_2)}$$

Hence, $R_x(t_1, t_2)$ can be derived from the joint PDF of x_1 and x_2, which is the second-order PDF.

9.2 CLASSIFICATION OF RANDOM PROCESSES

Random processes may be classified into the following broad categories.

Stationary and Nonstationary Random Processes

A random process whose statistical characteristics do not change with time is classified as a **stationary random process**. For a stationary process, we can say that a shift of time origin will be impossible to detect; the process will appear to be the same. Suppose we determine $p_x(x; t_1)$, then shift the origin by t_0, and again determine $p_x(x; t_1)$. The instant t_1 in the new frame of reference is $t_2 = t_1 + t_0$ in the old frame of reference. Hence, the PDFs of x at t_1 and $t_2 = t_1 + t_0$ must be the same; that is, $p_x(x; t_1)$ and $p_x(x; t_2)$ must be identical for a stationary random process. This is possible only if $p_x(x; t)$ is independent of t. Thus, the first-order density of a stationary random process can be expressed as

$$p_x(x; t) = p_x(x)$$

Similarly, for a stationary random process the autocorrelation function $R_x(t_1, t_2)$ must depend on t_1 and t_2 only through the difference $t_2 - t_1$. If not, we could determine a unique time origin. Hence, for a real stationary process,

$$R_x(t_1, t_2) = R_x(t_2 - t_1)$$

Therefore,

$$R_x(\tau) = \overline{x(t)x(t+\tau)} \tag{9.4}$$

For a stationary process, the joint PDF for x_1 and x_2 must also depend only on $t_2 - t_1$. Similarly, higher order PDFs are all independent of the choice of origin, that is,

$$\begin{aligned} p_x(x_1, x_2, \ldots, x_n; t_1, t_2, \ldots, t_n) &= p_x(x_1, x_2, \ldots, x_n; t_1 - t, t_2 - t, \ldots, t_n - t) \quad \forall t \\ &= p_x(x_1, x_2, \ldots, x_n; 0, t_2 - t_1, \ldots t_n - t_1) \end{aligned} \tag{9.5}$$

The random process $x(t)$ representing the temperature of a city is an example of a nonstationary random process because the temperature statistics (mean value, for example) depend on the time of the day. On the other hand, we can say that the noise process in Fig. 9.3 is stationary because its statistics (the mean and the mean square values, for example) do not change with time. In general, it is not easy to determine whether or not a process is stationary because the nth-order ($n = 1, 2, \ldots, \infty$) statistics must be investigated. In practice, we can ascertain stationarity if there is no change in the signal-generating mechanism. Such is the case for the noise process in Fig. 9.3.

Wide-Sense (or Weakly) Stationary Processes

A process that is not stationary in the strict sense, as discussed in the last subsection, may yet have a mean value and an autocorrelation function that are independent of the shift of time origin. This means

$$\overline{x(t)} = \text{constant}$$

and

$$R_x(t_1, t_2) = R_x(\tau) \qquad \tau = t_2 - t_1 \tag{9.6}$$

Such a process is known as a **wide-sense stationary**, or **weakly stationary**, process. Note that stationarity is a stronger condition than wide-sense stationarity. Stationary processes with well-defined autocorrelation functions are wide-sense stationary; exception for Gaussian random processes, however, the converse is not necessarily true.

Just as no sinusoidal signal exists in actual practice, no truly stationary process can occur in real life. All processes in practice are nonstationary because they must begin at some finite time and terminate at some finite time. A truly stationary process must start at $t = -\infty$ and go on forever. Many processes can be considered stationary for the time interval of interest, however, and the stationarity assumption allows a manageable mathematical model. The use of a stationary model is analogous to the use of a sinusoidal model in deterministic analysis.

Example 9.1 Show that the random process

$$\mathrm{x}(t) = A\cos(\omega_c t + \Theta)$$

where Θ is an RV uniformly distributed in the range $(0, 2\pi)$, is a wide-sense stationary process.

The ensemble (Fig. 9.5) consists of sinusoids of constant amplitude A and constant frequency ω_c, but the phase Θ is random. For any sample function, the phase is equally likely to have any value in the range $(0, 2\pi)$. Because Θ is an RV uniformly distributed over the range $(0, 2\pi)$, one can determine[1] $p_x(x, t)$ and, hence, $\overline{\mathrm{x}(t)}$, as in Eq. (9.2). For this particular case, however, $\overline{\mathrm{x}(t)}$ can be determined directly as a function of random variable Θ:

$$\overline{\mathrm{x}(t)} = \overline{A\cos(\omega_c t + \Theta)} = A\,\overline{\cos(\omega_c t + \Theta)}$$

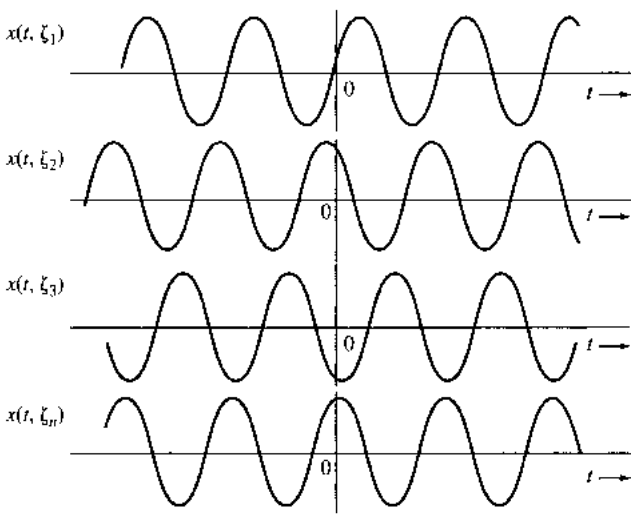


Figure 9.5 Ensemble for the random process $A\cos(\omega_c t + \Theta)$.

Because $\cos(\omega_c t + \Theta)$ is a function of an RV Θ, we have [see Eq. (8.61b)]

$$\overline{\cos(\omega_c t + \Theta)} = \int_0^{2\pi} \cos(\omega_c t + \theta) p_\Theta(\theta)\, d\theta$$

Because $p_\Theta(\theta) = 1/2\pi$ over $(0, 2\pi)$ and 0 outside this range,

$$\overline{\cos(\omega_c t + \Theta)} = \frac{1}{2\pi}\int_0^{2\pi} \cos(\omega_c t + \theta)\, d\theta = 0$$

Hence,

$$\overline{x(t)} = 0 \tag{9.7a}$$

Thus, the ensemble mean of sample function amplitudes at any instant t is zero.
The autocorrelation function $R_x(t_1, t_2)$ for this process also can be determined directly from Eq. (9.3a),

$$\begin{aligned} R_x(t_1, t_2) &= \overline{A^2 \cos(\omega_c t_1 + \Theta)\cos(\omega_c t_2 + \Theta)} \\ &= A^2\, \overline{\cos(\omega_c t_1 + \Theta)\cos(\omega_c t_2 + \Theta)} \\ &= \frac{A^2}{2}\left\{\overline{\cos[\omega_c(t_2 - t_1)]} + \overline{\cos[\omega_c(t_2 + t_1) + 2\Theta]}\right\} \end{aligned}$$

The first term on the right-hand side contains no RV. Hence, $\overline{\cos[\omega_c(t_2 - t_1)]}$ is $\cos[\omega_c(t_2 - t_1)]$ itself. The second term is a function of the uniform RV Θ, and its mean is

$$\overline{\cos[\omega_c(t_2 + t_1) + 2\Theta]} = \frac{1}{2\pi}\int_0^{2\pi} \cos[\omega_c(t_2 + t_1) + 2\theta]\, d\theta = 0$$

Hence,

$$R_x(t_1, t_2) = \frac{A^2}{2}\cos[\omega_c(t_2 - t_1)] \tag{9.7b}$$

or

$$R_x(\tau) = \frac{A^2}{2}\cos\omega_c\tau \qquad \tau = t_2 - t_1 \tag{9.7c}$$

From Eqs. (9.7a) and (9.7b) it is clear that $x(t)$ is a wide-sense stationary process.

Ergodic Wide-Sense Stationary Processes

We have studied the mean and the autocorrelation function of a random process. These are ensemble averages. For example, $\overline{x(t)}$ is the ensemble average of sample function amplitudes at t, and $R_x(t_1, t_2) = \overline{x_1 x_2}$ is the ensemble average of the product of sample function amplitudes $x(t_1)$ and $x(t_2)$.

We can also define time averages for each sample function. For example, a time mean $x(t)$ of a sample function $x(t)$ is*

$$\overline{x(t)} = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} x(t)\, dt \tag{9.8a}$$

* Here a sample function $x(t, \zeta_i)$ is represented by $x(t)$ for convenience.

Similarly, the time autocorrelation function $\mathcal{R}_x(\tau)$ defined in Eq. (3.82b) is

$$\mathcal{R}_x(\tau) = \overbrace{x(t)x(t+\tau)} = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t)x(t+\tau)\, dt \tag{9.8b}$$

For **ergodic (wide-sense) stationary processes**, ensemble averages are equal to the time averages of any sample function. Thus, for an ergodic process $x(t)$,

$$\overline{x(t)} = \overbrace{x(t)} \tag{9.9a}$$

$$R_x(\tau) = \mathcal{R}_x(\tau) \tag{9.9b}$$

These are the two averages for ergodic wide-sense stationary processes. For the broader definition of an ergodic process, all possible ensemble averages are equal to the corresponding time averages of one of its sample functions. Figure 9.6 illustrates the relationship among different classes of (ergodic) processes. In the coverage of this book, our focus lies in the class of ergodic wide-sense stationary processes.

It is difficult to test whether a process is ergodic or not, because we must test all possible orders of time and ensemble averages. Nevertheless, in practice many of the stationary processes are ergodic with respect to at least low-order statistics, such as the mean and the autocorrelation. For the process in Example 9.1 (Fig. 9.5), we can show that $x(t) = 0$ and $\mathcal{R}_x(\tau) = (A^2/2) \cos \omega_c \tau$ (see Prob. 3.8-1). Therefore, this process is ergodic at least with respect to the first- and second-order averages.

The ergodicity concept can be explained by a simple example of traffic lights in a city. Suppose the city is well planned, with all its streets in E–W and N–S directions only and with traffic lights at each intersection. Assume that each light stays green for 0.75 second in the E–W direction and 0.25 second in the N–S direction and that switching of any light is independent of the other lights. For the sake of simplicity, we ignore the orange light.

If we consider a certain person driving a car arriving at any traffic light randomly in the E–W direction, the probability that the person will have a green light is 0.75; that is, on the average, 75% of the time the person will observe a green light. On the other hand, if we consider a large number of drivers arriving at a traffic light in the E–W direction at some instant t, then 75% of the drivers will have a green light, and the remaining 25% will have a red light. Thus, the experience of a single driver arriving randomly many times at a traffic light will contain the same statistical information (sample function statistics) as that of a large number of drivers arriving simultaneously at various traffic lights (ensemble statistics) at one instant.

The ergodicity notion is extremely important because we do not have a large number of sample functions available in practice from which to compute ensemble averages. If the process is known to be ergodic, then we need only one sample function to compute ensemble averages. As mentioned earlier, many of the stationary processes encountered in practice are ergodic with

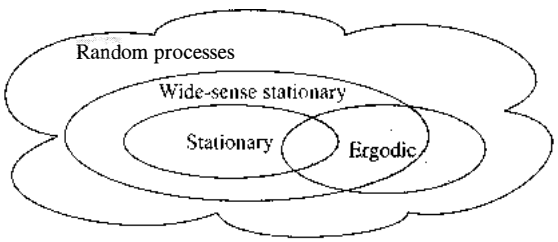


Figure 9.6 Classification of random processes.

respect to at least second-order averages. As we shall see in dealing with stationary processes in conjunction with linear systems, we need only the first- and second-order averages. This means that in most cases we can get by with a single sample function, as is often the case in practice.

9.3 POWER SPECTRAL DENSITY

An electrical engineer instinctively thinks of signals and linear systems in terms of their frequency domain descriptions. Linear systems are characterized by their frequency response (the transfer function), and signals are expressed in terms of the relative amplitudes and phases of their frequency components (the Fourier transform). From a knowledge of the input spectrum and transfer function, the response of a linear system to a given signal can be obtained in terms of the frequency content of that signal. This is an important analytical procedure for deterministic signals. We may wonder if similar methods may be found for random processes. Ideally, all the sample functions of a random process are assumed to exist over the entire time interval $(-\infty, \infty)$ and, thus, are power signals.* We therefore inquire about the existence of a power spectral density (PSD). Superficially, the concept of a random process having a PSD may appear ridiculous for the following reasons. In the first place, we may not be able to describe a sample function analytically. Second, for a given process, every sample function may be different from another one. Hence, even if a PSD does exist for each sample function, it may be different for different sample functions. Fortunately, both problems can be neatly resolved, and it is possible to define a meaningful PSD for a stationary (at least in the wide sense) random process. For nonstationary processes, the PSD may not exist.

Whenever randomness is involved, our inquiries can at best provide answers in terms of averages. When tossing a coin, for instance, the most we can say about the outcome is that on the average we will obtain heads in about half the trials and tails in the remaining half of the trials. For random signals or RVs, we do not have enough information to predict the outcome with certainty, and we must accept answers in terms of averages. It is not possible to transcend this limit of knowledge because of our fundamental ignorance of the process. It seems reasonable to define the PSD of a random process as a weighted mean of the PSDs of all sample functions. This is the only sensible solution, since we do not know exactly which of the sample functions may occur in a given trial. We must be prepared for any sample function. Consider, for example, the problem of filtering a certain random process. We would not want to design a filter with respect to any one particular sample function because any of the sample functions in the ensemble may be present at the input. A sensible approach is to design the filter with respect to the mean parameters of the input process. In designing a system to perform certain operations, one must design it with respect to the whole ensemble. We are therefore justified in defining the PSD $S_x(f)$ of a random process $x(t)$ as the ensemble average of the PSDs of all sample functions. Thus [see Eq. (3.80)],

$$S_x(f) = \lim_{T\to\infty} \overline{\left[\frac{|X_T(f)|^2}{T}\right]} \qquad \text{W/Hz} \tag{9.10a}$$

where $X_T(f)$ is the Fourier transform of the time-truncated random process

$$x_T(t) = x(t)\,\Pi(t/T)$$

* As we shall soon see, for the PSD to exist, the process must be stationary (at least in the wide sense). Stationary processes, because their statistics do not change with time, are power signals.

and the bar atop represents ensemble average. Note that ensemble averaging is done before the limiting operation. We shall now show that the PSD as defined in Eq. (9.10a) is the Fourier transform of the autocorrelation function $R_x(\tau)$ of the process $x(t)$; that is,

$$R_x(\tau) \Longleftrightarrow S_x(f) \tag{9.10b}$$

This can be proved as follows:

$$X_T(f) = \int_{-\infty}^{\infty} x_T(t)e^{-j2\pi ft}\,dt = \int_{-T/2}^{T/2} x(t)e^{-j2\pi ft}\,dt \tag{9.11}$$

Thus, for real $x(t)$,

$$\begin{aligned} |X_T(f)|^2 &= X_T(-f)X_T(f) \\ &= \int_{-T/2}^{T/2} x(t_1)e^{j2\pi ft_1}\,dt_1 \int_{-T/2}^{T/2} x(t_2)e^{-j2\pi ft_2}\,dt_2 \\ &= \int_{-T/2}^{T/2}\int_{-T/2}^{T/2} x(t_1)x(t_2)e^{-j2\pi f(t_2-t_1)}\,dt_1 dt_2 \end{aligned}$$

and

$$\begin{aligned} S_x(f) &= \lim_{T\to\infty}\left[\overline{\frac{|X_T(f)|^2}{T}}\right] \\ &= \lim_{T\to\infty}\left[\overline{\frac{1}{T}\int_{-T/2}^{T/2}\int_{-T/2}^{T/2} x(t_1)x(t_2)e^{-j2\pi f(t_2-t_1)}\,dt_1\,dt_2}\right] \end{aligned} \tag{9.12}$$

Interchanging the operation of integration and ensemble averaging,* we get

$$\begin{aligned} S_x(f) &= \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2}\int_{-T/2}^{T/2}\overline{x(t_1)x(t_2)}e^{-j2\pi f(t_2-t_1)}\,dt_1\,dt_2 \\ &= \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2}\int_{-T/2}^{T/2} R_x(t_2-t_1)e^{-j2\pi f(t_2-t_1)}\,dt_1\,dt_2 \end{aligned}$$

Here we are assuming that the process $x(t)$ is at least wide-sense stationary, so that $\overline{x(t_1)x(t_2)} = R_x(t_2 - t_1)$. For convenience, let

$$R_x(t_2 - t_1)e^{-j2\pi f(t_2-t_1)} = \varphi(t_2 - t_1) \tag{9.13}$$

Then,

$$S_x(f) = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2}\int_{-T/2}^{T/2}\varphi(t_2 - t_1)\,dt_1\,dt_2 \tag{9.14}$$

* The operation of ensemble averaging is also an operation of integration. Hence, interchanging integration with ensemble averaging is equivalent to interchanging the order of integration.

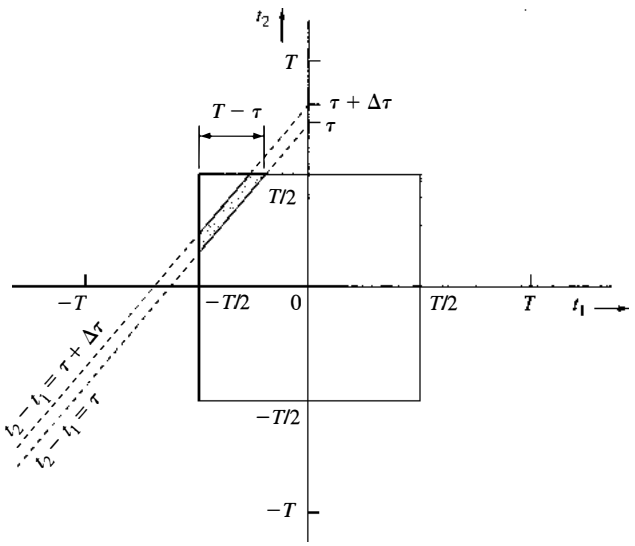


Figure 9.7 Derivation of the Wiener-Khintchine theorem.

The integral on the right-hand side is a double integral over the range $(-T/2,\ T/2)$ for each of the variables t_1 and t_2. The square region of integration in the t_1–t_2 plane is shown in Fig. 9.7. The integral in Eq. (9.14) is a volume under the surface $\varphi(t_2 - t_1)$ over the square region in Fig. 9.7. The double integral in Eq. (9.14) can be converted to a single integral by observing that $\varphi(t_2 - t_1)$ is constant along any line $t_2 - t_1 = \tau$ (a constant) in the t_1–t_2 plane (Fig. 9.7).

Let us consider two such lines, $t_2 - t_1 = \tau$ and $t_2 - t_1 = \tau + \Delta\tau$. If $\Delta\tau \to 0$, $\varphi(t_2 - t_1) \simeq \varphi(\tau)$ over the shaded region whose area is $(T - \tau)\ \Delta\tau$. Hence, the volume under the surface $\varphi(t_2 - t_1)$ over the shaded region is $\varphi(\tau)(T - \tau)\ \Delta\tau$. If τ were negative, the volume would be $\varphi(\tau)(T + \tau)\ \Delta\tau$. Hence, in general, the volume over the shaded region is $\varphi(\tau)(T - |\tau|)\ \Delta\tau$. The desired volume over the square region in Fig. 9.7 is the sum of the volumes over the shaded strips and is obtained by integrating $\varphi(\tau)(T - |\tau|)$ over the range of τ, which is $(-T,\ T)$ (see Fig. 9.7). Hence,

$$\begin{aligned} S_x(f) &= \lim_{T\to\infty} \frac{1}{T} \int_{-T}^{T} \varphi(\tau)(T - |\tau|)\, d\tau \\ &= \lim_{T\to\infty} \int_{-T/2}^{T/2} \varphi(\tau) \left(1 - \frac{|\tau|}{T}\right) d\tau \\ &= \int_{-\infty}^{\infty} \varphi(\tau)\, d\tau \end{aligned}$$

provided $\int_{-\infty}^{\infty} |\tau|\varphi(\tau)\, d\tau$ is bounded. Substituting Eq. (9.13) into this equation, we have

$$S_x(f) = \int_{-\infty}^{\infty} R_x(\tau) e^{-j2\pi f\tau}\, d\tau \tag{9.15}$$

provided $\int_{-\infty}^{\infty} |\tau| R_x(\tau) e^{-j2\pi f\tau}\, d\tau$ is bounded. Thus, the PSD of a wide-sense stationary random process is the Fourier transform of its autocorrelation function,*

$$R_x(\tau) \Longleftrightarrow S_x(f) \tag{9.16}$$

This is the well-known **Wiener-Khintchine theorem**, first presented in Chapter 3.

From the discussion thus far, the autocorrelation function emerges as one of the most significant entities in the spectral analysis of a random process. Earlier we showed heuristically how the autocorrelation function is connected with the frequency content of a random process.

The autocorrelation function $R_x(\tau)$ for real processes is an even function of τ. This can be proved in two ways. First, because $|X_T(f)|^2 = |X_T(f)X_T^*(f)| = |X_T(f)X_T(-f)|$ is an even function of f, $S_x(f)$ is also an even function of f, and $R_x(\tau)$, its inverse transform, is also an even function of τ (see Prob. 3.1-1). Alternately, we may argue that

$$R_x(\tau) = \overline{x(t)x(t+\tau)} \qquad \text{and} \qquad R_x(-\tau) = \overline{x(t)x(t-\tau)}$$

Letting $t - \tau = \sigma$, we have

$$R_x(-\tau) = \overline{x(\sigma)x(\sigma+\tau)} = R_x(\tau) \tag{9.17}$$

The PSD $S_x(f)$ is also a real and even function of f.

The mean square value $\overline{x^2(t)}$ of the random process $x(t)$ is $R_x(0)$,

$$R_x(0) = \overline{x(t)x(t)} = \overline{x^2(t)} = \overline{x^2} \tag{9.18}$$

The mean square value $\overline{x^2}$ is not the time mean square of a sample function but the ensemble average of the squares of all sample function amplitudes at any instant t.

The Power of a Random Process

The power P_x (average power) of a wide-sense random process $x(t)$ is its mean square value $\overline{x^2}$. From Eq. (9.16),

$$R_x(\tau) = \int_{-\infty}^{\infty} S_x(f) e^{j2\pi f\tau}\, df$$

Hence, from Eq. (9.18),

$$P_x = \overline{x^2} = R_x(0) = \int_{-\infty}^{\infty} S_x(f)\, df \tag{9.19a}$$

Because $S_x(f)$ is an even function of f, we have

$$P_x = \overline{x^2} = 2\int_0^{\infty} S_x(f)\, df \tag{9.19b}$$

where f is the frequency in hertz. This is the same relationship as that derived for deterministic signals in Chapter 3 [Eq. (3.81)]. The power P_x is the area under the PSD. Also, $P_x = \overline{x^2}$ is the ensemble mean of the square amplitudes of the sample functions at any instant.

* It can be shown that Eq. (9.15) holds also for complex random processes, for which we define $R_x(\tau) = \overline{x^*(t)x(t+\tau)}$.

It is helpful to repeat here, once again, that the PSD may not exist for processes that are not wide-sense stationary. Hence, in our future discussion, random processes will be assumed to be at least wide-sense stationary unless specifically stated otherwise.

Example 9.2 Determine the autocorrelation function $R_x(\tau)$ and the power P_x of a low-pass random process with a white noise PSD $S_x(f) = \mathcal{N}/2$ (Fig. 9.8a).

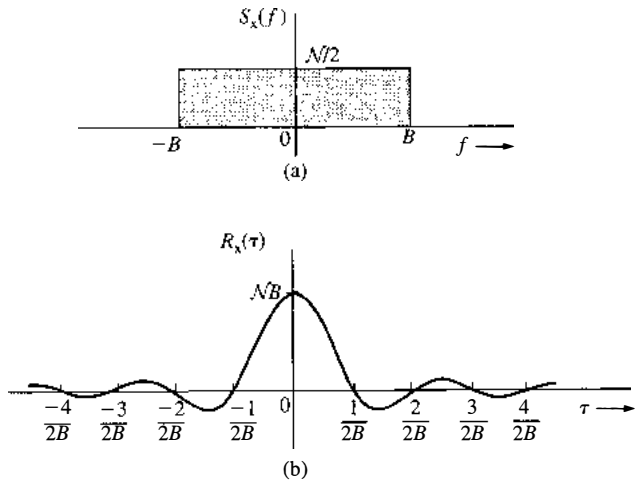


Figure 9.8 Bandpass white noise PSD and its autocorrelation function.

We have

$$S_x(f) = \frac{\mathcal{N}}{2}\Pi\left(\frac{f}{2B}\right) \tag{9.20a}$$

Hence, from Table 3.1 (pair 18),

$$R_x(\tau) = \mathcal{N}B\,\text{sinc}\,(2\pi B\tau) \tag{9.20b}$$

This is shown in Fig. 9.8b. Also,

$$P_x = \overline{x^2} = R_x(0) = \mathcal{N}B \tag{9.20c}$$

Alternately,

$$\begin{aligned} P_x &= 2\int_0^\infty S_x(f)\,df \\ &= 2\int_0^B \frac{\mathcal{N}}{2}\,df \\ &= \mathcal{N}B \end{aligned} \tag{9.20d}$$

Example 9.3 Determine the PSD and the mean square value of a random process

$$\mathrm{x}(t) = A\cos(\omega_c t + \Theta) \tag{9.21a}$$

where Θ is an RV uniformly distributed over $(0,\ 2\pi)$.

For this case $R_{\mathrm{x}}(\tau)$ is already determined [Eq. (9.7c)],

$$R_{\mathrm{x}}(\tau) = \frac{A^2}{2}\cos\ \omega_c\tau \tag{9.21b}$$

Hence,

$$S_{\mathrm{x}}(f) = \frac{A^2}{4}[\delta(f + f_c) + \delta(f - f_c)] \tag{9.21c}$$

$$P_{\mathrm{x}} = \overline{\mathrm{x}^2} = R_{\mathrm{x}}(0) = \frac{A^2}{2} \tag{9.21d}$$

Thus, the power, or the mean square value, of the process $\mathrm{x}(t) = A\cos(\omega_c t + \Theta)$ is $A^2/2$. The power P_{x} can also be obtained by integrating $S_{\mathrm{x}}(f)$ with respect to f.

Example 9.4 Amplitude Modulation

Determine the autocorrelation function and the PSD of the DSB-SC-modulated process $\mathrm{m}(t)\cos(\omega_c t + \Theta)$, where $\mathrm{m}(t)$ is a wide-sense stationary random process, and Θ is an RV uniformly distributed over $(0, 2\pi)$ and independent of $\mathrm{m}(t)$.

Let

$$\varphi(t) = \mathrm{m}(t)\cos(\omega_c t + \Theta)$$

Then

$$R_\varphi(\tau) = \overline{\mathrm{m}(t)\cos(\omega_c t + \Theta)\cdot \mathrm{m}(t+\tau)\cos[\omega_c(t+\tau) + \Theta]}$$

Because $\mathrm{m}(t)$ and Θ are independent, we can write [see Eqs. (8.64b) and (9.7c)]

$$\begin{aligned} R_\varphi(\tau) &= \overline{\mathrm{m}(t)\mathrm{m}(t+\tau)}\ \overline{\cos(\omega_c t + \Theta)\cos[\omega_c(t+\tau) + \Theta]} \\ &= \frac{1}{2}R_m(\tau)\cos\ \omega_c\tau \end{aligned} \tag{9.22a}$$

Consequently,*

$$S_\varphi(f) = \frac{1}{4}[S_{\mathrm{m}}(f + f_c) + S_{\mathrm{m}}(f - f_c)] \tag{9.22b}$$

From Eq. (9.22a) it follows that

$$\overline{\varphi^2(t)} = R_\varphi(0) = \frac{1}{2}R_{\mathrm{m}}(0) = \frac{1}{2}\overline{\mathrm{m}^2(t)} \tag{9.22c}$$

* We obtain the same result even if $\varphi(t) = \mathrm{m}(t)\sin(\omega_c t + \Theta)$.

Hence, the power of the DSB-SC-modulated signal is half the power of the modulating signal. We derived the same result earlier [Eq. (3.93)] for deterministic signals.

We note that, without the random phase Θ, a DSB-SC amplitude-modulated signal $m(t)\cos(\omega_c t)$ is in fact not wide-sense stationary. To find its PSD, we can resort to the time autocorrelation concept of Chapter 3.

Example 9.5 Random Binary Process

In this example we shall consider a random binary process for which a typical sample function is shown in Fig. 9.9a. The signal can assume only two states (values), 1 or -1, with equal probability. The transition from one state to another can take place only at node points, which occur every T_b seconds. The probability of a transition from one state to the other is 0.5. The first node is equally likely to be situated at any instant within the interval 0 to T_b from the origin. Analytically, we can represent $x(t)$ as

$$x(t) = \sum_n a_n p(t - nT_b - \alpha)$$

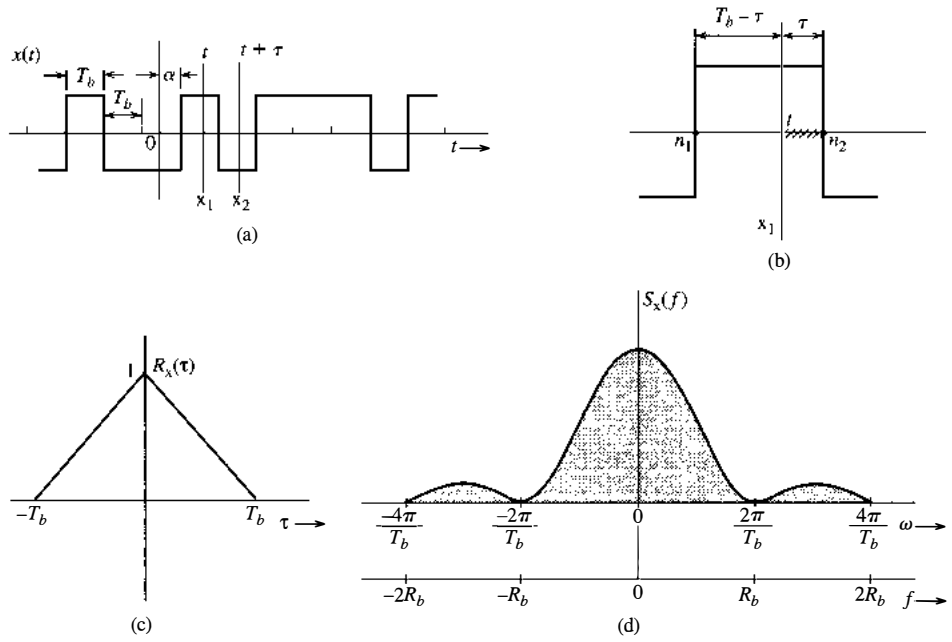


Figure 9.9 Derivation of autocorrelation function and PSD of a random binary process.

where α is an RV uniformly distributed over the range $(0, T_b)$ and $p(t)$ is the basic pulse (in this case $\Pi[(t - T_b/2)/T_b]$). Note that α is the distance of the first node from the origin, and it varies randomly from sample function to sample function. In addition, a_n is random, taking values 1 or -1 with equal probability. The amplitudes at t represent RV x_1, and those at $t+\tau$ represent RV x_2. Note that x_1 and x_2 are discrete and each can assume only

two values, -1 and 1. Hence,

$$R_X(\tau) = \overline{x_1 x_2} = \sum_{x_1} \sum_{x_2} x_1 x_2 P_{x_1 x_2}(x_1, x_2)$$

$$= P_{x_1 x_2}(1, 1) + P_{x_1 x_2}(-1, -1) - P_{x_1 x_2}(-1, 1) - P_{x_1 x_2}(1, -1) \tag{9.23a}$$

By symmetry, the first two terms and the last two terms on the right-hand side are equal. Therefore,

$$R_X(\tau) = 2[P_{x_1 x_2}(1, 1) - P_{x_1 x_2}(1, -1)] \tag{9.23b}$$

From Bayes' rule, we have

$$R_X(\tau) = 2P_{x_1}(1)[P_{x_2|x_1}(1|1) - P_{x_2|x_1}(-1|1)]$$
$$= P_{x_2|x_1}(1|1) - P_{x_2|x_1}(-1|1) \tag{9.23c}$$

Moreover,

$$P_{x_2|x_1}(1|1) = 1 - P_{x_2|x_1}(-1|1)$$

Hence,

$$R_X(\tau) = 1 - 2P_{x_2|x_1}(-1|1)$$

It is helpful to compute $R_X(\tau)$ for small values of τ first. Let us consider the case of $\tau < T_b$, where, at most, one node is in the interval t to $t + \tau$. In this case, the event $x_2 = -1$ given $x_1 = 1$ is a joint event $A \cap B$, where the event A is "a node in the interval $(t, t + \tau)$" and B is "the state change at this node." Because A and B are independent events,

$$P_{x_2|x_1}(-1|1) = P(\text{a node lies in } t \text{ to } t + \tau)P(\text{state change})$$
$$= \tfrac{1}{2}P(\text{a node lies in } t \text{ to } t + \tau)$$

Figure 9.9b shows adjacent nodes n_1 and n_2, between which t lies. We mark off the interval τ from the node n_2. If t lies anywhere in this interval (sawtooth line), the node n_2 lies within t and $t + \tau$. But because the instant t is chosen arbitrarily between nodes n_1 and n_2, it is equally likely to be at any instant over the T_b seconds between n_1 and n_2, and the probability that t lies in the shaded interval is simply τ/T_b. Therefore,

$$P_{x_2|x_1}(-1|1) = \frac{1}{2}\left(\frac{\tau}{T_b}\right) \tag{9.24}$$

and

$$R_X(\tau) = 1 - \frac{\tau}{T_b} \qquad \tau < T_b \tag{9.25}$$

Because $R_X(\tau)$ is an even function of τ, we have

$$R_X(\tau) = 1 - \frac{|\tau|}{T_b} \qquad |\tau| < T_b \tag{9.26}$$

Next, consider the range $\tau > T_b$. In this case at least one node lies in the interval t to $t + \tau$. Hence, x_1 and x_2 become independent, and

$$R_X(\tau) = \overline{x_1 x_2} = \overline{x_1}\,\overline{x_2} = 0 \qquad \tau > T_b$$

where, by inspection, we observe that $\bar{x}_1 = \bar{x}_2 = 0$ (Fig. 9.9a). This result can also be obtained by observing that for $|\tau| > T_b$, x_1 and x_2 are independent, and it is equally likely that $x_2 = 1$ or -1 given that $x_1 = 1$ (or -1). Hence, all four probabilities in Eq. (9.23a) are equal to 1/4, and

$$R_x(\tau) = 0 \qquad \tau > T_b$$

Therefore,

$$R_x(\tau) = \begin{cases} 1 - |\tau|/T_b & |\tau| < T_b \\ 0 & |\tau| > T_b \end{cases} \tag{9.27a}$$

and

$$S_x(f) = T_b \operatorname{sinc}^2(\pi f T_b) \tag{9.27b}$$

The autocorrelation function and the PSD of this process are shown in Fig. 9.9c and d. Observe that $\overline{x^2} = R_x(0) = 1$, as expected.

The random binary process described in Example 9.5 is sometimes known as the telegraph signal. This process also coincides with the polar signaling of Sec. 7.2.2 when the pulse shape is a rectangular NRZ pulse (Fig. 7.2). For wide-sense stationarity, the signal's initial starting point α is randomly distributed.

Let us now consider a more general case of the pulse train $y(t)$, discussed in Sec. 7.2 (Fig. 7.4). From the knowledge of the PSD of this train, we can derive the PSD of on-off, polar, bipolar, duobinary, split-phase, and many more important digital signals.

Example 9.6 Random PAM Pulse Train

Digital data is transmitted by using a basic pulse $p(t)$, as shown in Fig. 9.10a. The successive pulses are separated by T_b seconds, and the kth pulse is $a_k p(t)$, where a_k is an RV. The distance α of the first pulse (corresponding to $k = 0$) from the origin is equally likely to be any value in the range $(0, T_b)$. Find the autocorrelation function and the PSD of such a random pulse train $y(t)$ whose sample function is shown in Fig. 9.10b. The random process $y(t)$ can be described as

$$y(t) = \sum_{k=-\infty}^{\infty} a_k p(t - kT_b - \alpha)$$

where α is an RV uniformly distributed in the interval $(0, T_b)$. Thus, α is different for each sample function. Note that $p(\alpha) = 1/T_b$ over the interval $(0, T_b)$ and is zero everywhere else.* It can be shown that $\overline{y(t)} = (\overline{a_k}/T_b) \int_{-\infty}^{\infty} p(t)\, dt$ is a constant.†

---

* If $\alpha = 0$, the process can be expressed as $y(t) = \sum_{k=-\infty}^{\infty} a_k p(t - kT_b)$. In this case $\overline{y(t)} = \overline{a_k} \sum_{k=-\infty}^{\infty} p(t - kT_b)$ is not constant, but is periodic with period T_b. Similarly, we can show that the autocorrelation function is periodic with the same period T_b. This is an example of a **cyclostationary**, or periodically stationary, process (a process whose statistics are invariant to a shift of the time origin by integral multiples of a constant T_b). Cyclostationary processes, as seen here, are clearly not wide-sense stationary. But they can be made wide-sense stationary with slight modification by adding the RV α in the expression of $y(t)$, as in this example.

† Using exactly the same approach, as seen shortly in the derivation of Eq. (9.28), we can show that $\overline{y(t)} = (\overline{a_k}/T_b) \int_{-\infty}^{\infty} p(t)\, dt$.

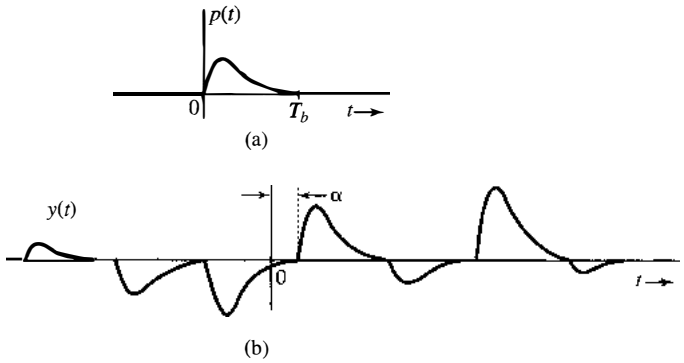


Figure 9.10 Random PAM process.

We have the expression

$$\begin{aligned} R_y(\tau) &= \overline{y(t)y(t+\tau)} \\ &= \overline{\sum_{k=-\infty}^{\infty} a_k p(t - kT_b - \alpha) \sum_{m=-\infty}^{\infty} a_m p(t + \tau - mT_b - \alpha)} \\ &= \sum_{k=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \overline{a_k a_m p(t - kT_b - \alpha) p(t + \tau - mT_b - \alpha)} \end{aligned}$$

Because a_k and a_m are independent of α,

$$R_y(\tau) = \sum_{k=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \overline{a_k a_m} \cdot \overline{p(t - kT_b - \alpha) \cdot p(t + \tau - mT_b - \alpha)}$$

Both k and m are integers. Letting $m = k + n$, this expression can be written

$$R_y(\tau) = \sum_{k=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \overline{a_k a_{k+n}} \cdot \overline{p(t - kT_b - \alpha) \cdot p(t + \tau - [k+n]T_b - \alpha)}$$

The first term under the double sum is the correlation of RVs a_k and a_{k+n} and will be denoted by $\mathcal{R}_n$. The second term, being a mean with respect to the RV α, can be expressed as an integral. Thus,

$$R_y(\tau) = \sum_{n=-\infty}^{\infty} \mathcal{R}_n \sum_{k=-\infty}^{\infty} \int_0^{T_b} p(t - kT_b - \alpha) p(t + \tau - [k+n]T_b - \alpha) p(\alpha)\, d\alpha$$

Recall that α is uniformly distributed over the interval 0 to T_b. Hence, $p(\alpha) = 1/T_b$ over the interval $(0, T_b)$, and is zero otherwise. Therefore,

$$\begin{aligned} R_y(\tau) &= \sum_{n=-\infty}^{\infty} \mathcal{R}_n \sum_{k=-\infty}^{\infty} \frac{1}{T_b} \int_0^{T_b} p(t - kT_b - \alpha) p(t + \tau - [k+n]T_b - \alpha)\, d\alpha \\ &= \frac{1}{T_b} \sum_{n=-\infty}^{\infty} \mathcal{R}_n \sum_{k=-\infty}^{\infty} \int_{t-(k+1)T_b}^{t-kT_b} p(\beta) p(\beta + \tau - nT_b)\, d\beta \\ &= \frac{1}{T_b} \sum_{n=-\infty}^{\infty} \mathcal{R}_n \int_{-\infty}^{\infty} p(\beta) p(\beta + \tau - nT_b)\, d\beta \end{aligned}$$

The integral on the right-hand side is the time autocorrelation function of the pulse $p(t)$ with the argument $\tau - nT_b$. Thus,

$$R_y(\tau) = \frac{1}{T_b} \sum_{n=-\infty}^{\infty} \mathcal{R}_n \psi_p(\tau - nT_b) \tag{9.28}$$

where

$$\mathcal{R}_n = \overline{a_k a_{k+n}} \tag{9.29}$$

and

$$\psi_p(\tau) = \int_{-\infty}^{\infty} p(t) p(t + \tau)\, dt \tag{9.30}$$

As seen in Eq. (3.74), if $p(t) \Longleftrightarrow P(f)$, then $\psi_p(\tau) \Longleftrightarrow |P(f)|^2$. Therefore, the PSD of $y(t)$, which is the Fourier transform of $R_y(\tau)$, is given by

$$\begin{aligned} S_y(f) &= \frac{1}{T_b} \sum_{n=-\infty}^{\infty} \mathcal{R}_n |P(f)|^2 e^{-jn2\pi fT_b} \\ &= \frac{|P(f)|^2}{T_b} \sum_{n=-\infty}^{\infty} \mathcal{R}_n e^{-jn2\pi fT_b} \end{aligned} \tag{9.31}$$

This result is similar to that found in Eq. (7.11b). The only difference is the use of the ensemble average in defining $\mathcal{R}_n$ in this chapter, whereas R_n in Chapter 7 is the time average.

Example 9.7 Find the PSD $S_y(f)$ for a polar binary random signal where **1** is transmitted by a pulse $p(t)$ (Fig. 9.11) whose Fourier transform is $P(f)$, and **0** is transmitted by $-p(t)$. The digits **1** and **0** are equally likely, and one digit is transmitted every T_b seconds. Each digit is independent of the other digits.

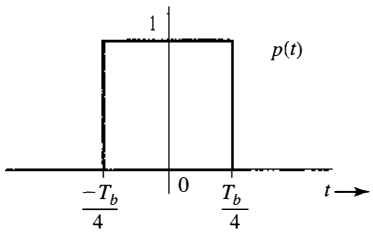


Figure 9.11 Basic pulse for a random binary polar signal.

In this case, a_k can take on values 1 and -1 with probability 1/2 each. Hence,

$$\overline{a_k} = \sum_{k=1,-1} a_k P(a_k) = (1)P_{a_k}(1) + (-1)P_{a_k}(-1)$$
$$= \frac{1}{2} - \frac{1}{2} = 0$$
$$\mathcal{R}_0 = \overline{a_k^2} = \sum_{k=1,-1} a_k^2 P(a_k) = (1)^2 P_{a_k}(1) + (-1)^2 P_{a_k}(-1)$$
$$= \frac{1}{2}(1)^2 + \frac{1}{2}(-1)^2 = 1$$

and because each digit is independent of the remaining digits,

$$\mathcal{R}_n = \overline{a_k a_{k+n}} = \overline{a_k}\ \overline{a_{k+n}} = 0 \qquad n \geq 1$$

Hence, from Eq. (9.31),

$$S_y(f) = \frac{|P(f)|^2}{T_b}$$

We already found this result in Eq. (7.13), where we used time averaging instead of ensemble averaging. When a process is ergodic of second order (or higher), the ensemble and time averages yield the same result. Note that Example 9.5 is a special case of this result, where $p(t)$ is a full-width rectangular pulse $\Pi(t/T_b)$ with $P(f) = T_b \operatorname{sinc}(\pi f T_b)$, and

$$S_y(f) = \frac{|P(f)|^2}{T_b} = T_b \operatorname{sinc}^2(\pi f T_b)$$

Example 9.8 Find the PSD $S_y(f)$ for on-off and bipolar random signals which use a basic pulse for $p(t)$, as shown in Fig. 9.11. The digits **1** and **0** are equally likely, and digits are transmitted every T_b seconds. Each digit is independent of the remaining digits. All these line codes are described in Sec. 7.2.

In each case we shall first determine $\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n$.

(a) *On-off signaling:* In this case, a_n can take on values 1 and 0 with probability 1/2 each. Hence,

$$\overline{a_k} = (1)P_{a_k}(1) + (0)P_{a_k}(0) = \frac{1}{2}(1) + \frac{1}{2}(0) = \frac{1}{2}$$

$$\mathcal{R}_0 = \overline{a_k^2} = (1)^2 P_{a_k}(1) + (0)^2 P_{a_k}(0) = \frac{1}{2}(1)^2 + \frac{1}{2}(0)^2 = \frac{1}{2}$$

and because each digit is independent of the remaining digits,

$$\mathcal{R}_n = \overline{a_k a_{k+n}} = \overline{a_k}\,\overline{a_{k+n}} = \left(\frac{1}{2}\right)\left(\frac{1}{2}\right) = \frac{1}{4} \qquad n \geq 1$$

Therefore, from Eq. (9.31),

$$S_y(f) = \frac{|P(f)|^2}{T_b}\left[\frac{1}{2} + \frac{1}{4}\sum_{\substack{n=-\infty \\ n\neq 0}}^{\infty} e^{-jn2\pi fT_b}\right] \tag{9.32a}$$

$$= \frac{|P(f)|^2}{T_b}\left[\frac{1}{4} + \frac{1}{4}\sum_{n=-\infty}^{\infty} e^{-jn2\pi fT_b}\right] \tag{9.32b}$$

Equation (9.32b) is obtained from Eq. (9.32a) by splitting the term 1/2 corresponding to $\mathcal{R}_0$ into two: 1/4 outside the summation and 1/4 inside the summation (corresponding to $n = 0$). This result is identical to Eq. (7.18b) found earlier by using time averages.

We now use a Poisson summation formula,*

$$\sum_{n=-\infty}^{\infty} e^{-jn2\pi fT_b} = \frac{1}{T_b}\sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_b}\right)$$

Substitution of this result into Eq. (9.32b) yields

$$S_y(f) = \frac{|P(f)|^2}{4T_b}\left[1 + \frac{1}{T_b}\sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_b}\right)\right] \tag{9.32c}$$

Note that the spectrum $S_y(f)$ consists of both a discrete and a continuous part. A discrete component of clock frequency ($R_b = 1/T_b$) is present in the spectrum. The continuous component of the spectrum is $|P(f)|^2/4T_b$ is identical (except for a scaling factor 1/4) to the spectrum of the polar signal in Example 9.7. This is a logical result because as Fig. 7.3 shows, an on-off signal can be expressed as a sum of a

* The impulse train in Fig. 3.23a is $\delta_{T_b}(t)$ can be expressed as $\delta_{T_b}(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_b)$. Also $\delta(t - nT_b) \Leftrightarrow e^{-jn2\pi fT_b}$. Hence, the Fourier transform of this impulse train is $\sum_{n=-\infty}^{\infty} e^{-jn2\pi fT_b}$. But we found the alternate form of the Fourier transform of this train in Eq. (3.43) (Example 3.11). Hence,

$$\sum_{n=-\infty}^{\infty} e^{jn2\pi fT_b} = \frac{1}{T_b}\sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_b}\right)$$

polar and a periodic component. The polar component is exactly half the polar signal discussed earlier. Hence, the PSD of this component is one-fourth of the PSD of the polar signal. The periodic component is of clock frequency R_b, and consists of discrete components of frequency R_b and its harmonics.

(b) *Bipolar signaling:* In this case, a_k can take on values 0, 1, and -1 with probabilities 1/2, 1/4, and 1/4, respectively. Hence,

$$\begin{aligned}\overline{\mathrm{a}_k} &= (0)P_{\mathrm{a}_k}(0) + (1)P_{\mathrm{a}_k}(1) + (-1)P_{\mathrm{a}_k}(-1)\\ &= \frac{1}{2}(0) + \frac{1}{4}(1) + \frac{1}{4}(-1) = 0\\ \mathcal{R}_0 = \overline{\mathrm{a}_k^2} &= (0)^2P_{\mathrm{a}_k}(0) + (1)^2P_{\mathrm{a}_k}(1) + (-1)^2P_{\mathrm{a}_k}(-1)\\ &= \frac{1}{2}(0)^2 + \frac{1}{4}(1)^2 + \frac{1}{4}(-1)^2 = \frac{1}{2}\end{aligned}$$

Also,

$$\mathcal{R}_1 = \overline{\mathrm{a}_k\,\mathrm{a}_{k+1}} = \sum_k \sum_{k+1} a_k a_{k+1} P_{\mathrm{a}_k \mathrm{a}_{k+1}}(a_k a_{k+1})$$

Because a_k and a_{k+1} can take three values each, the sum on the right-hand side has nine terms, of which only four terms (corresponding to values ± 1 for a_k and a_{k+1}) are nonzero. Thus,

$$\begin{aligned}\mathcal{R}_1 = {} & (1)(1)P_{\mathrm{a}_k\mathrm{a}_{k+1}}(1,\,1) + (-1)(1)P_{\mathrm{a}_k\mathrm{a}_{k+1}}(-1,\,1)\\ & + (1)(-1)P_{\mathrm{a}_k\mathrm{a}_{k+1}}(1,\,-1) + (-1)(-1)P_{\mathrm{a}_k\mathrm{a}_{k+1}}(-1,\,-1)\end{aligned}$$

Because of the bipolar rule,

$$P_{\mathrm{a}_k\mathrm{a}_{k+1}}(1,\,1) = P_{\mathrm{a}_k\mathrm{a}_{k+1}}(-1,\,-1) = 0$$

and

$$P_{\mathrm{a}_k\mathrm{a}_{k+1}}(-1,\,1) = P_{\mathrm{a}_k}(-1)P_{\mathrm{a}_{k+1}|\mathrm{a}_k}(1|-1) = \left(\frac{1}{4}\right)\left(\frac{1}{2}\right) = \frac{1}{8}$$

Similarly, we find $P_{\mathrm{a}_k\mathrm{a}_{k+1}}(1,\,-1) = 1/8$. Substitution of these values in $\mathcal{R}_1$ yields

$$\mathcal{R}_1 = -\frac{1}{4}$$

For $n \geq 2$, the pulse strengths a_k and a_{k+1} become independent. Hence,

$$\mathcal{R}_n = \overline{\mathrm{a}_k\,\mathrm{a}_{k+n}} = \overline{\mathrm{a}_k}\;\overline{\mathrm{a}_{k+n}} = (0)(0) = 0 \qquad n \geq 2$$

Substitution of these values in Eq. (9.31) and noting that $\mathcal{R}_n$ is an even function of n, yields

$$S_{\mathrm{y}}(f) = \frac{|P(f)|^2}{T_b}\sin^2(\pi f T_b)$$

This result is identical to Eq. (7.21b) found earlier by using time averages.

9.4 MULTIPLE RANDOM PROCESSES

For two real random processes $x(t)$ and $y(t)$, we define the **cross-correlation function*** $R_{xy}(t_1, t_2)$ as

$$R_{xy}(t_1, t_2) = \overline{x(t_1)y(t_2)} \tag{9.33a}$$

The two processes are said to be **jointly stationary** (in the wide sense) if each of the processes is individually wide-sense stationary and if

$$\begin{aligned} R_{xy}(t_1, t_2) &= R_{xy}(t_2 - t_1) \\ &= R_{xy}(\tau) \end{aligned} \tag{9.33b}$$

Uncorrelated, Orthogonal (Incoherent), and Independent Processes

Two processes $x(t)$ and $y(t)$ are said to be **uncorrelated** if their cross-correlation function is equal to the product of their means; that is,

$$R_{xy}(\tau) = \overline{x(t)y(t+\tau)} = \bar{x}\bar{y} \tag{9.34}$$

This implies that RVs $x(t)$ and $y(t + \tau)$ are uncorrelated for all t and τ.

Processes $x(t)$ and $y(t)$ are said to be **incoherent**, or **orthogonal**, if

$$R_{xy}(\tau) = 0 \tag{9.35}$$

Incoherent, or orthogonal, processes are uncorrelated processes with $\bar{x}$ and/or $\bar{y} = 0$.

Processes $x(t)$ and $y(t)$ are **independent** random processes if the random variables $x(t_1)$ and $y(t_2)$ are independent for all possible choices of t_1 and t_2,

Cross-Power Spectral Density

We define the **cross-power spectral density** $S_{xy}(f)$ for two random processes $x(t)$ and $y(t)$ as

$$S_{xy}(f) = \lim_{T\to\infty} \frac{\overline{X_T^*(f)Y_T(f)}}{T} \tag{9.36}$$

where $X_T(f)$ and $Y_T(f)$ are the Fourier transforms of the truncated processes $x(t)\,\Pi(t/T)$ and $y(t)\,\Pi(t/T)$, respectively. Proceeding along the lines of the derivation of Eq. (9.16), it can be shown that†

$$R_{xy}(\tau) \Longleftrightarrow S_{xy}(f) \tag{9.37a}$$

It can be seen from Eqs. (9.33) that for real random processes $x(t)$ and $y(t)$,

$$R_{xy}(\tau) = R_{yx}(-\tau) \tag{9.37b}$$

Therefore,

$$S_{xy}(f) = S_{yx}(-f) \tag{9.37c}$$

---

* For complex random processes, the cross-correlation function is defined as

$$R_{xy}(t_1, t_2) = \overline{x^*(t_1)y(t_2)}$$

† Equation (9.37a) is valid for complex processes as well.

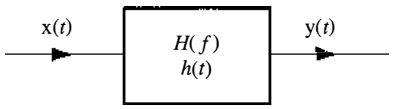


Figure 9.12 Transmission of a random process through a linear time-invariant system.

9.5 TRANSMISSION OF RANDOM PROCESSES THROUGH LINEAR SYSTEMS

If a random process $x(t)$ is applied at the input of a *stable* linear time-invariant system (Fig. 9.12) with transfer function $H(f)$, we can determine the autocorrelation function and the PSD of the output process $y(t)$. We now show that

$$R_y(\tau) = h(\tau) * h(-\tau) * R_x(\tau) \tag{9.38}$$

and

$$S_y(f) = |H(f)|^2 S_x(f) \tag{9.39}$$

To prove this, we observe that

$$y(t) = \int_{-\infty}^{\infty} h(\alpha)x(t-\alpha)\,d\alpha$$

and

$$y(t+\tau) = \int_{-\infty}^{\infty} h(\alpha)x(t+\tau-\alpha)\,d\alpha$$

Hence,*

$$\begin{aligned} R_y(\tau) = \overline{y(t)y(t+\tau)} &= \overline{\int_{-\infty}^{\infty} h(\alpha)x(t-\alpha)\,d\alpha \int_{-\infty}^{\infty} h(\beta)x(t+\tau-\beta)\,d\beta} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(\alpha)h(\beta)\overline{x(t-\alpha)x(t+\tau-\beta)}\,d\alpha\,d\beta \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(\alpha)h(\beta)R_x(\tau+\alpha-\beta)\,d\alpha\,d\beta \end{aligned}$$

This double integral is precisely the double convolution $h(\tau)*h(-\tau)*R_x(\tau)$. Hence, Eqs. (9.38) and (9.39) follow.

Example 9.9 Thermal Noise

Random thermal motion of electrons in a resistor R causes a random voltage across its terminals. This voltage $n(t)$ is known as the **thermal noise**. Its PSD $S_n(f)$ is practically flat over a very large band (up to 1000 GHz at room temperature) and is given by[1]

$$S_n(f) = 2kTR \tag{9.40}$$

* In this development, we interchange the operations of averaging and integrating. Because averaging is really an operation of integration, we are really changing the order of integration, and we assume that such a change is permissible.

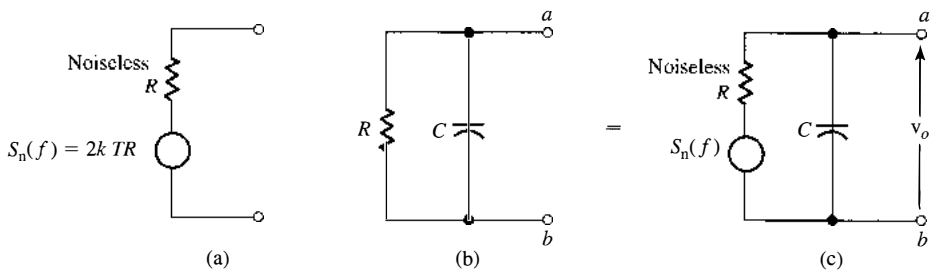


Figure 9.13 Thermal noise representation in a resistor.

where k is the Boltzmann constant (1.38×10^{-23}) and T is the ambient temperature in kelvins. A resistor R at a temperature T kelvin can be represented by a noiseless resistor R in series with a random white-noise voltage source (thermal noise) having a PSD of $2kTR$ (Fig. 9.13a). Observe that the thermal noise power over a band Δf is $(2kTR)\,2\Delta f = 4kTR\Delta f$.

Let us calculate the thermal noise voltage (rms value) across the simple RC circuit in Fig. 9.13b. The resistor R is replaced by an equivalent noiseless resistor in series with the thermal noise voltage source. The transfer function $H(f)$ relating the voltage v_o at terminals a–b to the thermal noise voltage is given by

$$H(f) = \frac{1/j2\pi fC}{R + 1/j2\pi fC} = \frac{1}{1 + j2\pi fRC}$$

If $S_0(f)$ is the PSD of the voltage v_o, then from Eq. (9.39) we have

$$S_0(f) = \left|\frac{1}{1 + j2\pi fRC}\right|^2 2kTR$$
$$= \frac{2kTR}{1 + 4\pi^2 f^2 R^2 C^2}$$

The mean square value $\overline{\mathrm{v}_o^2}$ is given by

$$\overline{\mathrm{v}_o^2} = \int_{-\infty}^{\infty} \frac{2kTR}{1 + 4\pi^2 f^2 R^2 C^2}\, df$$
$$= \frac{kT}{C} \tag{9.41}$$

Hence, the rms thermal noise voltage across the capacitor is $\sqrt{kT/C}$.

Sum of Random Processes

If two stationary processes (at least in the wide sense) $\mathrm{x}(t)$ and $\mathrm{y}(t)$ are added to form a process $\mathrm{z}(t)$, the statistics of $\mathrm{z}(t)$ can be determined in terms of those of $\mathrm{x}(t)$ and $\mathrm{y}(t)$. If

$$\mathrm{z}(t) = \mathrm{x}(t) + \mathrm{y}(t) \tag{9.42a}$$

then

$$R_{\mathrm{z}}(\tau) = \overline{\mathrm{z}(t)\mathrm{z}(t+\tau)} = \overline{[\mathrm{x}(t) + \mathrm{y}(t)][\mathrm{x}(t+\tau) + \mathrm{y}(t+\tau)]}$$
$$= R_{\mathrm{x}}(\tau) + R_{\mathrm{y}}(\tau) + R_{\mathrm{xy}}(\tau) + R_{\mathrm{yx}}(\tau) \tag{9.42b}$$

If $x(t)$ and $y(t)$ are uncorrelated, then from Eq. (9.34),

$$R_{xy}(\tau) = R_{yx}(\tau) = \bar{x}\bar{y}$$

and

$$R_z(\tau) = R_x(\tau) + R_y(\tau) + 2\bar{x}\bar{y} \tag{9.43}$$

Most processes of interest in communication problems have zero means. If processes $x(t)$ and $y(t)$ are uncorrelated with either $\bar{x}$ or $\bar{y} = 0$ [i.e., if $x(t)$ and $y(t)$ are incoherent], then

$$R_z(\tau) = R_x(\tau) + R_y(\tau) \tag{9.44a}$$

and

$$S_z(f) = S_x(f) + S_y(f) \tag{9.44b}$$

It also follows from Eqs. (9.44a) and (9.19) that

$$\overline{z^2} = \overline{x^2} + \overline{y^2} \tag{9.44c}$$

Hence, the mean square of a sum of incoherent (or orthogonal) processes is equal to the sum of the mean squares of these processes.

Example 9.10 Two independent random voltage processes $x_1(t)$ and $x_2(t)$ are applied to an *RC* network, as shown in Fig. 9.14. It is given that

$$S_{x_1}(f) = K \qquad S_{x_2}(f) = \frac{2\alpha}{\alpha^2 + (2\pi f)^2}$$

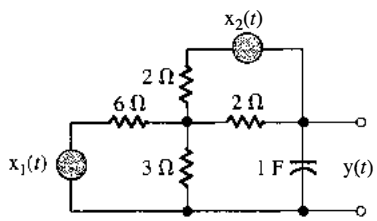


Figure 9.14 Noise calculations in a resistive circuit.

Determine the PSD and the power P_y of the output random process $y(t)$. Assume that the resistors in the circuit contribute negligible thermal noise (i.e., assume that they are noiseless).

Because the network is linear, the output voltage $y(t)$ can be expressed as

$$y(t) = y_1(t) + y_2(t)$$

where $y_1(t)$ is the output from input $x_1(t)$ [assuming $x_2(t) = 0$] and $y_2(t)$ is the output from input $x_2(t)$ [assuming $x_1(t) = 0$]. The transfer functions relating $y(t)$ to $x_1(t)$ and $x_2(t)$ are $H_1(f)$ and $H_2(f)$, respectively, given by

$$H_1(f) = \frac{1}{3(3 \cdot j2\pi f + 1)}, \qquad H_2(f) = \frac{1}{2(3 \cdot j2\pi f + 1)}$$

Hence,

$$S_{y_1}(f) = |H_1(f)|^2 S_{x_1}(f) = \frac{K}{9[9(2\pi f)^2 + 1]}$$

and

$$S_{y_2}(f) = |H_2(f)|^2 S_{x_2}(f) = \frac{\alpha}{2[9(2\pi f)^2 + 1][\alpha^2 + (2\pi f)^2]}$$

Because the input processes $x_1(t)$ and $x_2(t)$ are independent, the outputs $y_1(t)$ and $y_2(t)$ generated by them will also be independent. Also, the PSDs of $y_1(t)$ and $y_2(t)$ have no impulses at $f = 0$, implying that they have no dc components [i.e., $\overline{y_1(t)} = \overline{y_2(t)} = 0$]. Hence, $y_1(t)$ and $y_2(t)$ are incoherent, and

$$\begin{aligned} S_y(f) &= S_{y_1}(f) + S_{y_2}(f) \\ &= \frac{2K[\alpha^2 + (2\pi f)^2] + 9\alpha}{18[9(2\pi f)^2 + 1][\alpha^2 + (2\pi f)^2]} \end{aligned}$$

The power P_y (or the mean square value $\overline{y^2}$) can be determined in two ways. We can find $R_y(\tau)$ by taking the inverse transforms of $S_{y_1}(f)$ and $S_{y_2}(f)$ as

$$R_y(\tau) = \underbrace{\frac{K}{54} e^{-|\tau|/3}}_{R_{y_1}(\tau)} + \underbrace{\frac{3\alpha - e^{-\alpha|\tau|}}{4(9\alpha^2 - 1)}}_{R_{y_2}(\tau)}$$

and

$$P_y = \overline{y^2} = R_y(0) = \frac{K}{54} + \frac{3\alpha - 1}{4(9\alpha^2 - 1)}$$

Alternatively, we can determine $\overline{y^2}$ by integrating $S_y(f)$ with respect to f (or f) [see Eq. (9.19)].

9.6 APPLICATION: OPTIMUM FILTERING (WIENER-HOPF FILTER)

When a desired signal is mixed with noise, the SNR can be improved by passing it through a filter that suppresses frequency components where the signal is weak but the noise is strong. The SNR improvement in this case can be explained qualitatively by considering a case of white noise mixed with a signal $m(t)$ whose PSD decreases at high frequencies. If the filter attenuates higher frequencies more, the signal will be reduced—in fact, distorted. The distortion component $m_\epsilon(t)$ may be considered as bad as added noise. Thus, attenuation of higher frequencies will cause additional noise (from signal distortion), but, in compensation, it will reduce the channel noise, which is strong at high frequencies. Because at higher frequencies the signal has a small power content, the distortion component will be small in comparison to the reduction in channel noise, and the total distortion may be smaller than before.

Let $H_{opt}(f)$ be the optimum filter (Fig. 9.15a). This filter, not being ideal, will cause signal distortion. The distortion signal $m_\epsilon(t)$ can be found from Fig. 9.15b. The distortion signal power N_D appearing at the output is given by

$$N_D = \int_{-\infty}^{\infty} S_m(f)|H_{opt}(f) - 1|^2 \, df$$

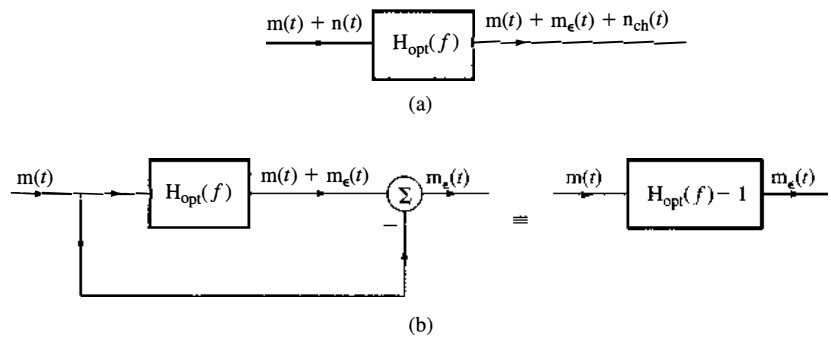


Figure 9.15
Wiener-Hopf filter calculations.

where $S_m(f)$ is the signal PSD at the input of the receiving filter. The channel noise power N_{ch} appearing at the filter output is given by

$$N_{ch} = \int_{-\infty}^{\infty} S_n(f)|H_{opt}(f)|^2\, df$$

where $S_n(f)$ is the noise PSD appearing at the input of the receiving filter. The distortion component acts as a noise. Because the signal and the channel noise are incoherent, the total noise N_o at the receiving filter output is the sum of the channel noise N_{ch} and the distortion noise N_D,

$$\begin{aligned} N_o &= N_{ch} + N_D \\ &= \int_{-\infty}^{\infty} \left[|H_{opt}(f)|^2 S_n(f) + |H_{opt}(f) - 1|^2 S_m(f) \right] df \end{aligned} \tag{9.45a}$$

Using the fact that $|A + B|^2 = (A + B)(A^* + B^*)$, and noting that both $S_m(f)$ and $S_n(f)$ are real, we can rearrange Eq. (9.45a) as

$$N_o = \int_{-\infty}^{\infty} \left[\left| H_{opt}(f) - \frac{S_m(f)}{S_r(f)} \right|^2 S_r(f) + \frac{S_m(f)S_n(f)}{S_r(f)} \right] df \tag{9.45b}$$

where $S_r(f) = S_m(f) + S_n(f)$. The integrand on the right-hand side of Eq. (9.45b) is nonnegative. Moreover, it is a sum of two nonnegative terms. Hence, to minimize N_o, we must minimize each term. Because the second term $S_m(f)S_n(f)/S_r(f)$ is independent of $H_{opt}(f)$, only the first term can be minimized. From Eq. (9.45b) it is obvious that this term is minimum at zero when

$$\begin{aligned} H_{opt}(f) &= \frac{S_m(f)}{S_r(f)} \\ &= \frac{S_m(f)}{S_m(f) + S_n(f)} \end{aligned} \tag{9.46a}$$

For this optimum choice, the output noise power N_o is given by

$$\begin{aligned} N_o &= \int_{-\infty}^{\infty} \frac{S_m(f)S_n(f)}{S_r(f)}\, df \\ &= \int_{-\infty}^{\infty} \frac{S_m(f)S_n(f)}{S_m(f) + S_n(f)}\, df \end{aligned} \tag{9.46b}$$

The optimum filter is known as the **Wiener-Hopf filter** in the literature. Equation (9.46a) shows that $H_{\text{opt}}(f) \approx 1$ (no attenuation) when $S_{\text{m}}(f) \gg S_{\text{n}}(f)$. But when $S_{\text{m}}(f) \ll S_{\text{n}}(f)$, the filter has high attenuation. In other words, the optimum filter attenuates heavily the band where noise is relatively stronger. This causes some signal distortion, but at the same time it attenuates the noise more heavily so that the overall SNR is improved.

Comments on the Optimum Filter

If the SNR at the filter input is reasonably large—for example, $S_{\text{m}}(f) > 100S_{\text{n}}(f)$ (SNR of 20 dB)—the optimum filter [Eq. (9.46a)] in this case is practically an ideal filter, and N_o [Eq. (9.46b)] is given by

$$N_o \simeq \int_{-\infty}^{\infty} S_{\text{n}}(f)\, df$$

Hence for a large input SNR, optimization yields insignificant improvement. The Wiener-Hopf filter is therefore practical only when the input SNR is small (large-noise case).

Another issue is the realizability of the optimum filter in Eq. (9.46a). Because $S_{\text{m}}(f)$ and $S_{\text{n}}(f)$ are both even functions of f, the optimum filter $H_{\text{opt}}(f)$ is an even function of f. Hence, the unit impulse response $h_{\text{opt}}(t)$ is an even function of t (see Prob. 3.1-1). This makes $h_{\text{opt}}(t)$ noncausal and the filter unrealizable. As noted earlier, such a filter can be realized approximately if we are willing to tolerate some delay in the output. If delay cannot be tolerated, the derivation of $H_{\text{opt}}(f)$ must be repeated with a realizability constraint. Note that the realizable optimum filter can never be superior to the unrealizable optimum filter [Eq. (9.46a)]. Thus, the filter in Eq. (9.46a) gives the upper bound on performance (output SNR). Discussion of realizable optimum filters can be readily found in the literature[1,2].

Example 9.11 A random process $\text{m}(t)$ (the signal) is mixed with a white channel noise $\text{n}(t)$. Given

$$S_{\text{m}}(2f) = \frac{2\alpha}{\alpha^2 + (2\pi f)^2} \qquad \text{and} \qquad S_{\text{n}}(f) = \frac{\mathcal{N}}{2}$$

find the Wiener-Hopf filter to maximize the SNR. Find the resulting output noise power N_o.

From Eq. (9.46a),

$$H_{\text{opt}}(f) = \frac{4\alpha}{4\alpha + \mathcal{N}[\alpha^2 + (2\pi f)^2]}$$

$$= \frac{4\alpha}{\mathcal{N}[\beta^2 + (2\pi f)^2]} \qquad \beta^2 = \frac{4\alpha}{\mathcal{N}} + \alpha^2 \tag{9.47a}$$

Hence,

$$h_{\text{opt}}(t) = \frac{2\alpha}{\mathcal{N}\beta} e^{-\beta|t|} \tag{9.47b}$$

Figure 9.16a shows $h_{\text{opt}}(t)$. It is evident that this is an unrealizable filter. However, a delayed version (Fig. 9.16b) of this filter, that is, $h_{\text{opt}}(t - t_0)$, is closely realizable if we make $t_0 \geq 3/\beta$ and eliminate the tail for $t < 0$ (Fig. 9.16c).

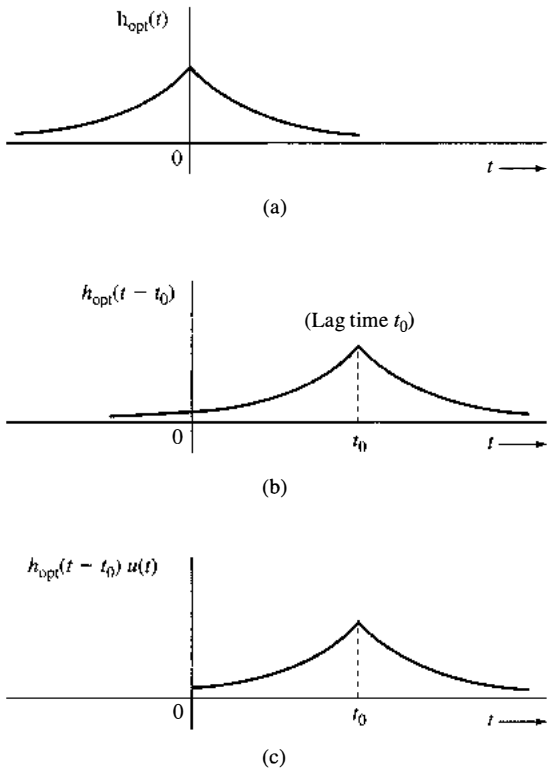


Figure 9.16 Close realization of an unrealizable filter using delay.

The output noise power N_o is [Eq. (9.46b)]

$$N_o = \int_0^\infty \frac{2\alpha}{\beta^2 + (2\pi f)^2}\, df = \frac{\alpha}{\beta} = \frac{\alpha}{\sqrt{\alpha^2 + (4\alpha/\mathcal{N})}} \tag{9.48}$$

9.7 APPLICATION: PERFORMANCE ANALYSIS OF BASEBAND ANALOG SYSTEMS

We now apply the concept of power spectral density (PSD) to analyze the performance of baseband analog communication systems. In analog signals, the SNR is basic in specifying the signal quality. For voice signals, an SNR of 5 to 10 dB at the receiver implies a barely intelligible signal. Telephone-quality signals have an SNR of 25 to 35 dB, whereas for television, an SNR of 45 to 55 dB is required.

Figure 9.17 shows a simple communication system in which analog signal $m(t)$ is transmitted at power S_T through a channel (representing a transmission medium). The transmitted signal is corrupted by additive channel noise during transmission. The channel also attenuates (and may also distort) the signal. At the receiver input, we have a signal mixed with noise. The signal and noise powers at the receiver input are S_i and N_i, respectively.

The receiver processes (filters) the signal to yield the output $s_o(t) + n_o(t)$. The noise component $n_o(t)$ came from the processing of $n(t)$ by the receiver, while the signal component $s_o(t)$ came from the message $m(t)$. The signal and noise powers at the receiver output are S_o

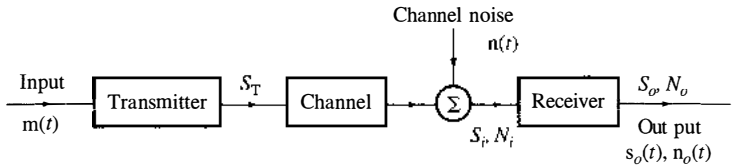


Figure 9.17 Communication system model.

and N_o, respectively. In analog systems, the quality of the received signal is determined by S_o/N_o, the output SNR. Hence, we shall focus our attention on this figure of merit under either a fixed transmission power S_T or for a given S_i.

In baseband systems, the signal is transmitted directly without any modulation. This mode of communication is suitable over a pair of twisted wires or coaxial cables. It is mainly used in short-haul links. For a baseband system, the transmitter and the receiver are ideal baseband filters. The ideal low-pass transmitter limits the input signal spectrum to a given bandwidth, whereas the low-pass receiver eliminates the out-of-band noise and other channel interference. (More elaborate transmitter and receiver filters can be used, as shown in the next section.)

The baseband signal $\mathrm{m}(t)$ is assumed to be a zero mean, wide-sense stationary random process band-limited to B Hz. We consider the case of ideal low-pass (or baseband) filters with bandwidth B at the transmitter and the receiver (Fig. 9.17). The channel is assumed to be distortionless. The power, or the mean square value, of $\mathrm{m}(t)$ is $\overline{\mathrm{m}^2}$, given by

$$S_i = \overline{\mathrm{m}^2} = 2\int_0^B S_{\mathrm{m}}((f))\,df \tag{9.49}$$

For this case,

$$S_o = S_i \tag{9.50a}$$

and

$$N_o = 2\int_0^B S_{\mathrm{n}}(f)\,df \tag{9.50b}$$

where $S_{\mathrm{n}}(f)$ is the PSD of the channel noise. For the case of a white noise, $S_{\mathrm{n}}(f) = \mathcal{N}/2$, and

$$N_o = 2\int_0^B \frac{\mathcal{N}}{2}\,df = \mathcal{N}B \tag{9.50c}$$

and

$$\frac{S_o}{N_o} = \frac{S_i}{\mathcal{N}B} \tag{9.50d}$$

We define a parameter γ as

$$\gamma = \frac{S_i}{\mathcal{N}B} \tag{9.51}$$

From Eqs. (9.50d) and (9.51) we have

$$\frac{S_o}{N_o} = \gamma \tag{9.52}$$

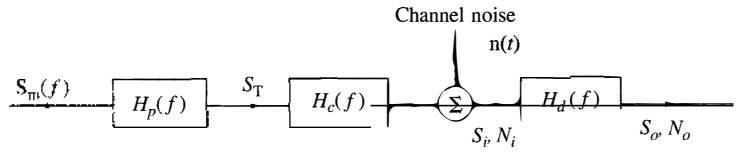


Figure 9.18 Optimum preemphasis and deemphasis filters in baseband systems.

The parameter γ is directly proportional to S_i and, therefore, directly proportional to S_T. Hence, a given S_T (or S_i) implies a given γ. Equation (9.52) is precisely the result we are looking for. It gives the receiver output SNR for a given S_T (or S_i).

The value of the SNR in Eq. (9.52) often serves as a benchmark against which the output SNRs of other modulation systems are measured in practice.

9.8 APPLICATION: OPTIMUM PREEMPHASIS-DEEMPHASIS SYSTEMS

It is possible to increase the output SNR by deliberate distortion of the transmitted signal (preemphasis) and the corresponding compensation (deemphasis) at the receiver. For an intuitive understanding of this process, consider a case of white channel noise and a signal $\mathrm{m}(t)$ whose PSD decreases with frequency. In this case, we can boost the high-frequency components of $\mathrm{m}(t)$ at the transmitter (preemphasis). Because the signal has relatively less power at high frequencies, this preemphasis will require only a small increase in transmitted power.* At the receiver, the high-frequency components are attenuated (or deemphasized) to undo the preemphasis at the transmitter. This will restore the useful signal to its original form. The channel noise receives an entirely different treatment. Because the noise is added after the transmitter, it does not undergo preemphasis. At the receiver, however, it does undergo deemphasis (i.e., attenuation of high-frequency components). Thus, at the receiver output, the signal power is restored but the noise power is reduced. The output SNR is therefore increased.

In this section, we consider a baseband system. The extension of preemphasis and deemphasis to modulated systems is straightforward. A baseband system with a preemphasis filter $H_p(f)$ at the transmitter and the corresponding complementary deemphasis filter $H_d(f)$ at the receiver is shown in Fig. 9.18. The channel transfer function is $H_c(f)$, and the PSD of the input signal $\mathrm{m}(t)$ is $S_{\mathrm{m}}(f)$. We shall determine the optimum preemphasis-deemphasis (PDE) filters $H_p(f)$ and $H_d(f)$ required for distortionless transmission of the signal $\mathrm{m}(t)$.

For distortionless transmission,

$$|H_p(f)H_c(f)H_d(f)| = G \quad \text{(a constant)} \tag{9.53a}$$

and

$$\theta_p(f) + \theta_c(f) + \theta_d(f) = -2\pi f t_d \tag{9.53b}$$

We want to maximize the output SNR, S_o/N_o, for a given transmitted power S_T.

Referring to Fig. 9.18, we have

$$S_T = \int_{-\infty}^{\infty} S_{\mathrm{m}}(f)|H_p(f)|^2\,df \tag{9.54a}$$

* Actually, the transmitted power is maintained constant by attenuating the preemphasized signal slightly.

Because $H_p(f)H_c(f)H_d(f) = G\exp(-j2\pi ft_d)$, the signal power S_o at the receiver output is

$$S_o = G^2 \int_{-\infty}^{\infty} S_{\mathrm{m}}(f)\, df \tag{9.54b}$$

The noise power N_o at the receiver output is

$$N_o = \int_{-\infty}^{\infty} S_{\mathrm{n}}(f)|H_d(f)|^2\, df \tag{9.54c}$$

Thus,

$$\frac{S_o}{N_o} = \frac{G^2 \int_{-\infty}^{\infty} S_{\mathrm{m}}(f)\, df}{\int_{-\infty}^{\infty} S_{\mathrm{n}}(f)|H_d(f)|^2\, df} \tag{9.55}$$

We wish to maximize this ratio subject to the condition in Eq. (9.54a) with S_T as a given constant. Applying this power limitation makes the design of $H_p(f)$ a well-posed problem, for otherwise filters with larger gains will always be better. We can include this constraint by multiplying the numerator and the denominator of the right-hand side of Eq. (9.55) by the left-hand side and the right-hand side, respectively, of Eq. (9.54a). This gives

$$\frac{S_o}{N_o} = \frac{G^2 S_T \int_{-\infty}^{\infty} S_{\mathrm{m}}(f)\, df}{\int_{-\infty}^{\infty} S_{\mathrm{n}}(f)|H_d(f)|^2\, df \int_{-\infty}^{\infty} S_{\mathrm{m}}(f)|H_p(f)|^2\, df} \tag{9.56}$$

The numerator of the right-hand side of Eq. (9.56) is fixed and *unaffected* by the PDE filters. Hence, to maximize S_o/N_o, we need only minimize the denominator of the right-hand side of Eq. (9.56). To do this, we use the Cauchy-Schwarz inequality (Appendix B),

$$\int_{-\infty}^{\infty} S_{\mathrm{m}}(f)|H_p(f)|^2\, df \int_{-\infty}^{\infty} S_{\mathrm{n}}(f)|H_d(f)|^2\, df \geq \left| \int_{-\infty}^{\infty} [S_{\mathrm{m}}(f)S_{\mathrm{n}}(f)]^{1/2}|H_p(f)H_d(f)|\, df \right|^2 \tag{9.57}$$

The equality holds if and only if

$$S_{\mathrm{m}}(f)|H_p(f)|^2 = K^2 S_{\mathrm{n}}(f)|H_d(f)|^2 \tag{9.58}$$

where K is an arbitrary constant. Thus to maximize S_o/N_o, Eq. (9.58) must be satisfied. Substitution of Eq. (9.53a) into Eq. (9.58) yields

$$|H_p(f)|^2_{\mathrm{opt}} = GK \frac{\sqrt{S_{\mathrm{n}}(f)/S_{\mathrm{m}}(f)}}{|H_c(f)|} \tag{9.59a}$$

$$|H_d(f)|^2_{\mathrm{opt}} = \frac{G}{K} \frac{\sqrt{S_{\mathrm{m}}(f)/S_{\mathrm{n}}(f)}}{|H_c(f)|} \tag{9.59b}$$

The constant K is found by substituting Eq. (9.59a) into the power constraint of Eq. (9.54a) as

$$K = \frac{S_T}{G \int_{-\infty}^{\infty} [\sqrt{S_{\mathrm{m}}(f)S_{\mathrm{n}}(f)}/|H_c(f)|]\, df} \tag{9.59c}$$

Substitution of this value of K into Eqs. (9.59a) and (9.59b) yields

$$|H_p(f)|^2_{\text{opt}} = \frac{S_T\sqrt{S_n(f)/S_m(f)}}{|H_c(f)| \int_{-\infty}^{\infty}[\sqrt{S_m(f)S_n(f)}/|H_c(f)|]\,df} \tag{9.60a}$$

$$|H_d(f)|^2_{\text{opt}} = \frac{G^2 \int_{-\infty}^{\infty}[\sqrt{S_m(f)S_n(f)}/|H_c(f)|]\,df}{S_T|H_c(f)|\sqrt{S_n(f)/S_m(f)}} \tag{9.60b}$$

The output SNR under optimum conditions is given by Eq. (9.56) with its denominator replaced by the right-hand side of Eq. (9.57). Finally, substituting $|H_p(f)H_d(f)| = G/|H_c(f)|$ leads to

$$\left(\frac{S_o}{N_o}\right)_{\text{opt}} = \frac{S_T \int_{-\infty}^{\infty} S_m(f)\,df}{\left(\int_{-\infty}^{\infty}[\sqrt{S_m(f)S_n(f)}/|H_c(f)|]\,df\right)^2} \tag{9.60c}$$

Equations (9.60a) and (9.60b) give the magnitudes of the optimum filters $H_p(f)$ and $H_d(f)$. The phase functions must be chosen to satisfy the condition of distortionless transmission [Eq. (9.53b)].

Observe that the preemphasis filter in Eq. (9.59a) boosts frequency components where the signal is weak and suppresses frequency components where the signal is strong. The deemphasis filter in Eq. (9.59b) does exactly the opposite. Thus, the signal is unchanged but the noise is reduced.

Example 9.12 Consider the case with $\alpha = 1400\pi$,

$$S_m(f) = \begin{cases} \frac{C}{(2\pi f)^2+\alpha^2} & |f| \le 4000 \\ 0 & |f| \ge 4000 \end{cases} \tag{9.61a}$$

The channel noise is white with PSD

$$S_n(f) = \frac{\mathcal{N}}{2} \tag{9.61b}$$

The channel is assumed to be ideal [$H_c(f) = 1$ and $G = 1$] over the band of interest (0–4000 Hz).

Without preemphasis-deemphasis, we have

$$\begin{aligned} S_o &= \int_{-4000}^{4000} S_m(f)\,df \\ &= 2\int_0^{4000} \frac{C}{(2\pi f)^2+\alpha^2}\,df \qquad \alpha = 1400\pi \\ &= 10^{-4}C \end{aligned}$$

Also, because $G = 1$, the transmitted power $S_T = S_o$,

$$S_o = S_T = 10^{-4}C$$

and the noise power without preemphasis-deemphasis is

$$N_o = \mathcal{N}B = 4000\mathcal{N}$$

Therefore,

$$\frac{S_o}{N_o} = 2.5 \times 10^{-8}\frac{C}{\mathcal{N}} \tag{9.62}$$

The optimum transmitting and receiving filters are given [Eqs. (9.60a) and (9.60b)] by

$$|H_p(f)|^2 = \frac{10^{-4}\sqrt{(2\pi f)^2 + \alpha^2}}{\int_{-\infty}^{\infty}\left(1/\sqrt{(2\pi f)^2 + \alpha^2}\right) df} = \frac{1.286\sqrt{(2\pi f)^2 + \alpha^2}}{10^4} \qquad |f| \le 4000 \tag{9.63a}$$

$$|H_d(f)|^2 = \frac{10^4 \int_{-\infty}^{\infty}\left(1/\sqrt{(2\pi f)^2 + \alpha^2}\right) df}{\sqrt{(2\pi f)^2 + \alpha^2}} = \frac{0.778 \times 10^4}{\sqrt{(2\pi f)^2 + \alpha^2}} \qquad |f| \le 4000 \tag{9.63b}$$

The output SNR using optimum preemphasis and deemphasis in found from Eq. (9.60c) as

$$\left(\frac{S_o}{N_o}\right)_{\text{opt}} = \frac{(10^{-4}C)^2}{(\mathcal{N}C/2)\left[\int_{-4000}^{4000}\left[1/\sqrt{4\pi^2 f^2 + (1400\pi)^2}\right] df\right]^2}$$

$$= 3.3 \times 10^{-8}\frac{C}{\mathcal{N}} \tag{9.64}$$

Comparison of Eq. (9.62) with Eq. (9.64) shows that preemphasis-deemphasis has increased the output SNR by a factor of 1.32.

9.9 BANDPASS RANDOM PROCESSES

If the PSD of a random process is confined to a certain passband (Fig. 9.19), the process is a **bandpass** random process. Bandpass random processes can be used effectively to model modulated communication signals and bandpass noises. Just as a bandpass signal can be represented in terms of quadrature components [see Eq. (3.39)], we can express a bandpass random process $\mathrm{x}(t)$ in terms of quadrature components as follows:

$$\mathrm{x}(t) = \mathrm{x}_c(t)\cos\,\omega_c t + \mathrm{x}_s(t)\sin\,\omega_c t \tag{9.65}$$

In this representation, $\mathrm{x}_c(t)$ is known as the **in-phase** component and $\mathrm{x}_s(t)$ is known as the **quadrature** component of the bandpass random process.

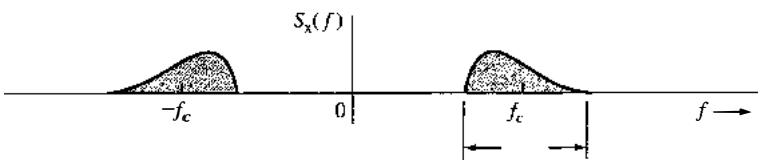


Figure 9.19 PSD of a bandpass random process.

This can be proven by considering the system in Fig. 9.20a, where $H_0(f)$ is an ideal low-pass filter (Fig. 9.20b) with unit impulse response $h_0(t)$. First we show that the system in Fig. 9.20a is an ideal bandpass filter with the transfer function $H(f)$ shown in Fig. 9.20c. This can be conveniently done by computing the response $h(t)$ to the unit impulse input $\delta(t)$. Because the system contains time-varying multipliers, however, we must also test whether it is a time-varying or a time-invariant system. It is therefore appropriate to consider the system response to an input $\delta(t-\alpha)$. This is an impulse at $t=\alpha$. Using the fact that [see Eq. (2.10b)] $f(t)\,\delta(t-\alpha) = f(\alpha)\delta(t-\alpha)$, we can express the signals at various points as follows:

$$
\begin{aligned}
\text{Signal at } a_1: &\quad \cos(\omega_c\alpha+\theta)\,\delta(t-\alpha)\\
a_2: &\quad \sin(\omega_c\alpha+\theta)\,\delta(t-\alpha)\\
b_1: &\quad \cos(\omega_c\alpha+\theta)h_0(t-\alpha)\\
b_2: &\quad \sin(\omega_c\alpha+\theta)h_0(t-\alpha)\\
c_1: &\quad \cos(\omega_c\alpha+\theta)\cos(\omega_c t+\theta)h_0(t-\alpha)\\
c_2: &\quad \sin(\omega_c\alpha+\theta)\sin(\omega_c t+\theta)h_0(t-\alpha)\\
d: &\quad h_0(t-\alpha)\,[\cos(\omega_c\alpha+\theta)\cos(\omega_c t+\theta)+\sin(\omega_c\alpha+\theta)\sin(\omega_c t+\theta)]\\
&= 2h_0(t-\alpha)\cos[\omega_c(t-\alpha)]
\end{aligned}
$$

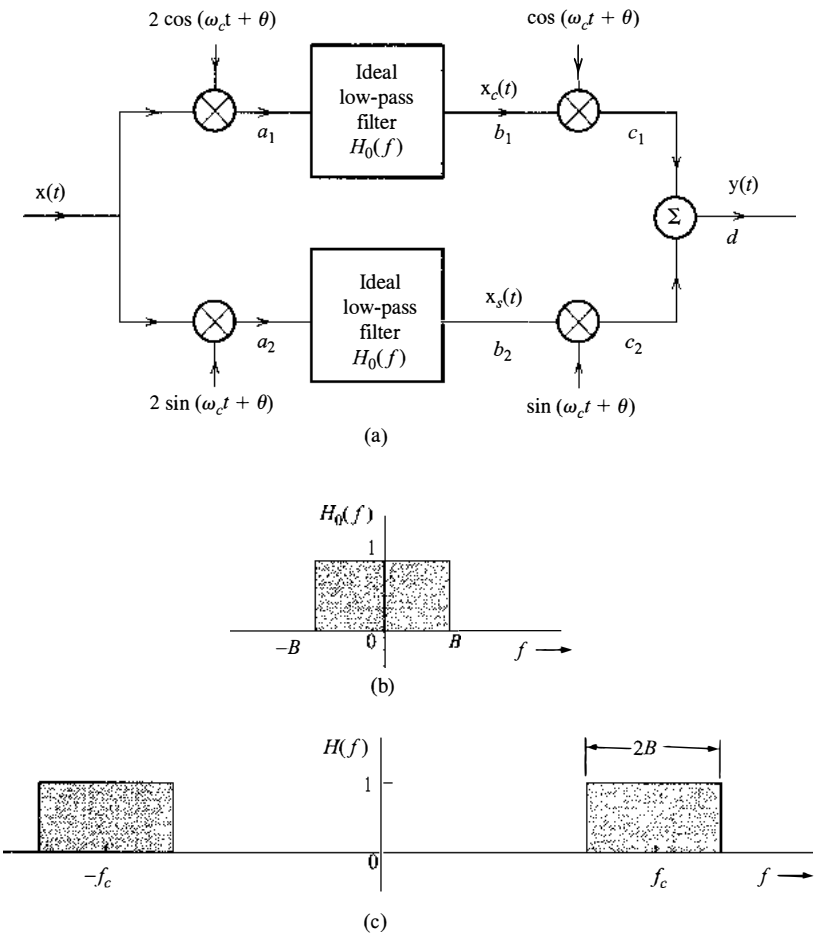


Figure 9.20 (a) Equivalent circuit of an ideal bandpass filter. (b) Ideal low-pass filter frequency response. (c) Ideal bandpass filter frequency response.

Thus, the system response to the input $\delta(t-\alpha)$ is $2h_0(t-\alpha)\cos[\omega_c(t-\alpha)]$. Clearly, this means that the underlying system is linear time invariant, with impulse response

$$h(t) = 2h_0(t)\cos\omega_c t$$

and transfer function

$$H(f) = H_0(f+f_c) + H_0(f-f_c)$$

The transfer function $H(f)$ (Fig. 9.20c) represents an ideal bandpass filter.

If we apply the bandpass process $\mathrm{x}(t)$ (Fig. 9.19) to the input of this system, the output $\mathrm{y}(t)$ at d will remain the same as $\mathrm{x}(t)$. Hence, the output PSD will be the same as the input PSD

$$|H(f)|^2 S_x(f) = S_x(f)$$

If the processes at points b_1 and b_2 (low-pass filter outputs) are denoted by $\mathrm{x}_c(t)$ and $\mathrm{x}_s(t)$, respectively, then the output $\mathrm{x}(t)$ can be written as

$$\mathrm{x}(t) = \mathrm{x}_c(t)\cos(\omega_c t+\theta) + \mathrm{x}_s(t)\sin(\omega_c t+\theta) \tag{9.66}$$

where $\mathrm{x}_c(t)$ and $\mathrm{x}_s(t)$ are low-pass random processes band-limited to B Hz (because they are the outputs of low-pass filters of bandwidth B). Because Eq. (9.66) is valid for any value of θ, by substituting $\theta = 0$, we get the desired representation in Eq. (9.65).

To characterize $\mathrm{x}_c(t)$ and $\mathrm{x}_s(t)$, consider once again Fig. 9.20a with the input $\mathrm{x}(t)$. Let θ be an RV uniformly distributed over the range $(0, 2\pi)$, that is, for a sample function, θ is equally likely to take on any value in the range $(0, 2\pi)$. In this case $\mathrm{x}(t)$ is represented as in Eq. (9.66). We observe that $\mathrm{x}_c(t)$ is obtained by multiplying $\mathrm{x}(t)$ by $2\cos(\omega_c t+\theta)$, and then passing the result through a low-pass filter. The PSD of $2\mathrm{x}(t)\cos(\omega_c t+\theta)$ is [see Eq. (9.22b)]

$$4\times\frac{1}{4}[S_x(f+f_c)+S_x(f-f_c)]$$

This PSD is $S_x(f)$ shifted up and down by f_c, as shown in Fig. 9.21a. When this is passed through a low-pass filter, the resulting PSD of $\mathrm{x}_c(t)$ is as shown in Fig. 9.21b. It is clear that

$$S_{x_c}(f) = \begin{cases} S_x(f+f_c)+S_x(f-f_c) & |f|\le B \\ 0 & |f|>B \end{cases} \tag{9.67a}$$

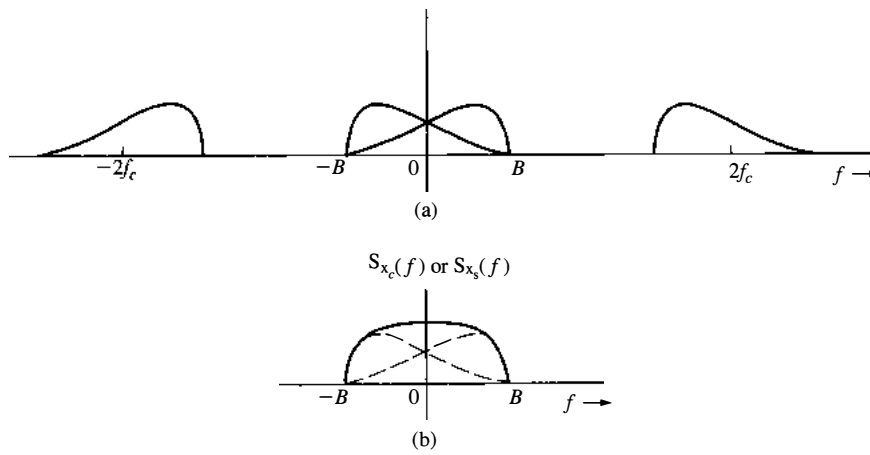


Figure 9.21 Derivation of PSDs of quadrature components of a bandpass random process.

We can obtain $S_{x_s}(f)$ in the same way. As far as the PSD is concerned, multiplication by $\cos(\omega_c t + \theta)$ or $\sin(\omega_c t + \theta)$ makes no difference [see footnote following Eq. (9.22a)], and we get

$$S_{x_c}(f) = S_{x_s}(f) = \begin{cases} S_x(f + f_c) + S_x(f - f_c), & |f| \le B \\ 0, & |f| > B \end{cases} \tag{9.67b}$$

From Figs. 9.19 and 9.21b, we make the interesting observation that the areas under the PSDs $S_x(f)$, $S_{x_c}(f)$, and $S_{x_s}(f)$ are equal. Hence, it follows that

$$\overline{x_c^2(t)} = \overline{x_s^2(t)} = \overline{x^2(t)} \tag{9.67c}$$

Thus, the mean square values (or powers) of $x_c(t)$ and $x_s(t)$ are identical to that of $x(t)$.

These results are derived by assuming Θ to be an RV. For the representation in Eq. (9.65), $\Theta = 0$, and Eqs. (9.67b) and (9.67c) may not be true. Fortunately, those equations hold even for the case of $\Theta = 0$. The proof is rather long and cumbersome and will not be given here.[1–3] It can also be shown[1–3] that

$$\overline{x_c(t)x_s(t)} = R_{x_c x_s}(0) = 0 \tag{9.68}$$

That is, the amplitudes x_c and x_s at any given instant are uncorrelated. Moreover, if $S_x(f)$ is symmetrical about ω_c (as well as $-\omega_c$), then

$$R_{x_c x_s}(\tau) = 0 \tag{9.69}$$

Example 9.13 The PSD of a bandpass white noise $n(t)$ is $\mathcal{N}/2$ (Fig. 9.22a). Represent this process in terms of quadrature components. Derive $S_{n_c}(f)$ and $S_{n_s}(f)$, and verify that $\overline{n_c^2} = \overline{n_s^2} = \overline{n^2}$.

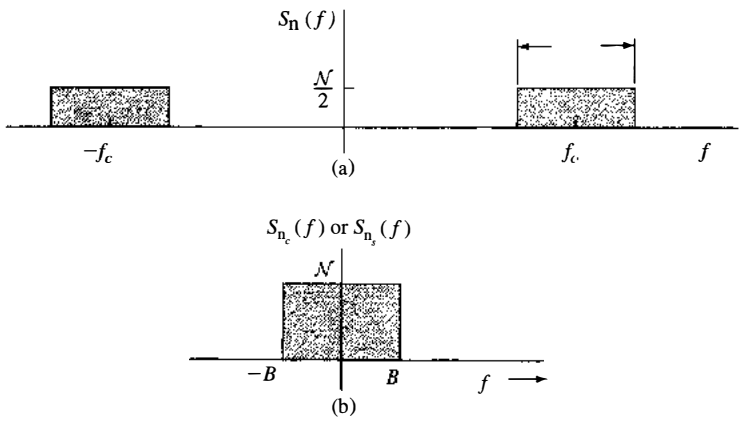


Figure 9.22 (a) PSD of a bandpass white noise process. (b) PSD of its quadrature components.

We have the expression

$$n(t) = n_c(t)\cos\omega_c t + n_s(t)\sin\omega_c t \tag{9.70}$$

where

$$S_{n_c}(f) = S_{n_s}(f) = \begin{cases} S_n(f + f_c) + S_n(f - f_c) & |f| \le B \\ 0 & |f| > B \end{cases}$$

It follows from this equation and from Fig. 9.22 that

$$S_{n_c}(f) = S_{n_s}(f) = \begin{cases} \mathcal{N} & |f| \le B \\ 0 & |f| > B \end{cases} \tag{9.71}$$

Also,

$$\overline{n^2} = 2\int_{f_c-B}^{f_c+B} \frac{\mathcal{N}}{2}\, df = 2\mathcal{N}B \tag{9.72a}$$

From Fig. 9.22b it follows that

$$\overline{n_c^2} = \overline{n_s^2} = 2\int_0^B \mathcal{N}\, df = 2\mathcal{N}B \tag{9.72b}$$

Hence,

$$\overline{n_c^2} = \overline{n_s^2} = \overline{n^2} = 2\mathcal{N}B \tag{9.72c}$$

Nonuniqueness of the Quadrature Representation

No unique center frequency exists for a bandpass signal. For the spectrum in Fig. 9.23a, for example, we may consider the spectrum to have a bandwidth $2B$ centered at ω_c. The same spectrum can be considered to have a bandwidth $2B'$ centered at ω_1, as also shown in Fig. 9.23a. The quadrature representation [Eq. (9.65)] is also possible for center frequency ω_1:

$$x(t) = x_{c_1}(t)\cos\omega_1 t + x_{s_1}(t)\sin\omega_1 t$$

where

$$S_{x_{c_1}}(f) = S_{x_{s_1}}(f) = \begin{cases} S_x(f+f_1) + S_x(f-f_1) & |f| \le B' \\ 0 & |f| > B' \end{cases} \tag{9.73}$$

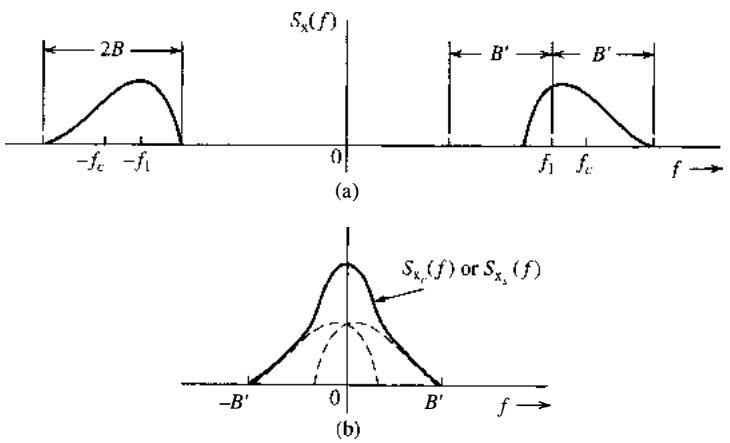


Figure 9.23 Nonunique nature of quadrature component representation of a bandpass process.

This is shown in Fig. 9.23b. Thus, the quadrature representation of a bandpass process is not unique. An infinite number of possible choices exist for the center frequency, and **corresponding to each center frequency is a distinct quadrature representation.**

Example 9.14 A bandpass white noise PSD of an SSB channel (lower sideband) is shown in Fig. 9.24a. Represent this signal in terms of quadrature components with the carrier frequency ω_c.

The true center frequency of this PSD is not ω_c; but we can still use ω_c as the center frequency, as discussed earlier,

$$\mathrm{n}(t) = \mathrm{n}_c(t)\cos\omega_c t + \mathrm{n}_s(t)\sin\omega_c t \tag{9.74}$$

The PSD $S_{\mathrm{n}_c}(f)$ or $S_{\mathrm{n}_s}(f)$ obtained by shifting $S_{\mathrm{n}}(f)$ up and down by f_c [see Eq. (9.73)] is shown in Fig. 9.24b,

$$S_{\mathrm{n}_c}(f) = S_{\mathrm{n}_s}(f) = \begin{cases} \frac{\mathcal{N}}{2} & |f| \le B \\ 0 & |f| > B \end{cases} \tag{9.75}$$

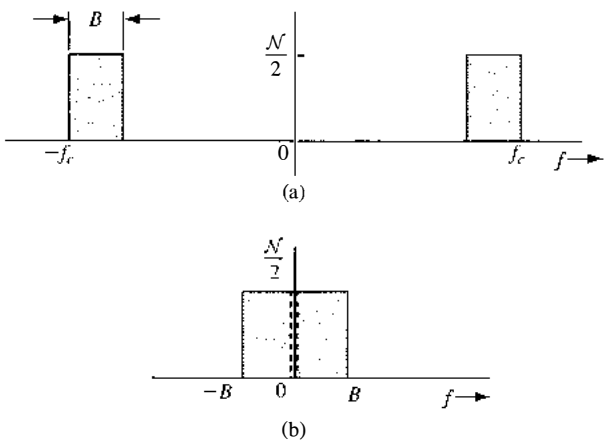


Figure 9.24 A possible form of quadrature component representation of noise in SSB.

From Fig. 9.24a it follows that

$$\overline{\mathrm{n}^2} = \mathcal{N}B \tag{9.76a}$$

Similarly, from Fig. 9.24b we have

$$\overline{\mathrm{n}_c^2} = \overline{\mathrm{n}_s^2} = \mathcal{N}B \tag{9.76b}$$

Hence,

$$\overline{\mathrm{n}_c^2} = \overline{\mathrm{n}_s^2} = \overline{\mathrm{n}^2} = \mathcal{N}B \tag{9.76c}$$

Bandpass "White" Gaussian Random Process

Thus far we have avoided defining a Gaussian random process. The Gaussian random process is perhaps the single most important random process in the area of communication. A careful and unhurried discussion, however, is beyond our scope. All we need to know here is that an RV $\mathrm{x}(t)$ formed by sample function amplitudes at instant t of a Gaussian process is Gaussian, with a PDF of the form of Eq. (8.39).

A Gaussian random process with a uniform PSD is called a white Gaussian random process. The term *bandpass "white" Gaussian process* is actually a misnomer. However, it is a popular notion to represent a random process $\mathrm{n}(t)$ with uniform PSD $\mathcal{N}/2$ centered at ω_c and with a bandwidth $2B$ (Fig. 9.22a). Utilizing the quadrature representation, it can be expressed as

$$\mathrm{n}(t) = \mathrm{n}_c(t) \cos\ \omega_c t + \mathrm{n}_s(t) \sin\ \omega_c t \tag{9.77}$$

where, from Eq. (9.71), we have

$$S_{\mathrm{n}_c}(f) = S_{\mathrm{n}_s}(f) = \begin{cases} \mathcal{N} & |f| \leq B \\ 0 & |f| > B \end{cases}$$

Also, from Eq. (9.72c),

$$\overline{\mathrm{n}_c^2} = \overline{\mathrm{n}_s^2} = \overline{\mathrm{n}^2} = 2\mathcal{N}B \tag{9.78}$$

The bandpass signal can also be expressed in polar form [see Eq. (3.40)]:

$$\mathrm{n}(t) = \mathrm{E}(t) \cos\ (\omega_c t + \Theta) \tag{9.79a}$$

where the random envelope and random phase are defined by

$$\mathrm{E}(t) = \sqrt{\mathrm{n}_c^2(t) + \mathrm{n}_s^2(t)} \tag{9.79b}$$

$$\Theta(t) = -\tan^{-1} \frac{\mathrm{n}_s(t)}{\mathrm{n}_c(t)} \tag{9.79c}$$

The RVs $\mathrm{n}_c(t)$ and $\mathrm{n}_s(t)$ are uncorrelated [see Eq. (9.68)] Gaussian RVs with zero means and variance $2\mathcal{N}B$ [Eq. (9.78)]. Hence, their PDFs are identical:

$$p_{\mathrm{n}_c}(\alpha) = p_{\mathrm{n}_s}(\alpha) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\alpha^2/2\sigma^2} \tag{9.80a}$$

where

$$\sigma^2 = 2\mathcal{N}B \tag{9.80b}$$

It has been shown in Prob. 8.2-10 that if two Gaussian RVs are uncorrelated, they are independent. In such a case, as shown in Example 8.17, $\mathrm{E}(t)$ has a Rayleigh density

$$p_{\mathrm{E}}(E) = \frac{E}{\sigma^2} e^{-E^2/2\sigma^2} u(E), \qquad \sigma^2 = 2\mathcal{N}B \tag{9.81}$$

and Θ in Eq. (9.79a) is uniformly distributed over $(0,\ 2\pi)$.

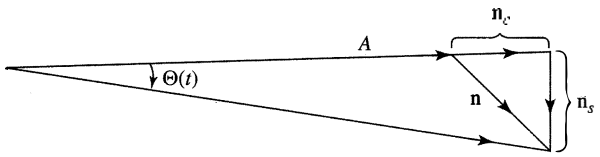


Figure 9.25 Phasor representation of a sinusoid and a narrowband Gaussian noise.

Sinusoidal Signal in Noise

Another case of interest is a sinusoid plus a narrowband Gaussian noise. If $A\cos(\omega_c t + \varphi)$ is a sinusoid mixed with $n(t)$, a Gaussian bandpass noise centered at ω_c, then the sum $y(t)$ is given by

$$y(t) = A\cos(\omega_c t + \varphi) + n(t)$$

Using Eq. (9.66) to represent the bandpass noise, we have

$$y(t) = [A + n_c(t)]\cos(\omega_c t + \varphi) + n_s(t)\sin(\omega_c t + \varphi) \tag{9.82a}$$

$$= E(t)\cos[\omega_c t + \Theta(t) + \varphi] \tag{9.82b}$$

where $E(t)$ is the envelope [$E(t) > 0$] and $\Theta(t)$ is the angle shown in Fig. 9.25,

$$E(t) = \sqrt{[A + n_c(t)]^2 + n_s^2(t)} \tag{9.83a}$$

$$\Theta(t) = -\tan^{-1}\frac{n_s(t)}{A + n_c(t)} \tag{9.83b}$$

Both $n_c(t)$ and $n_s(t)$ are Gaussian, with variance σ^2. For white Gaussian noise, $\sigma^2 = 2\mathcal{N}B$ [Eq. (9.80b)]. Arguing in a manner analogous to that used in deriving Eq. (8.57), and observing that

$$\begin{aligned} n_c^2 + n_s^2 &= E^2 - A^2 - 2An_c \\ &= E^2 - 2A(A + n_c) + A^2 \\ &= E^2 - 2AE\cos\Theta(t) + A^2 \end{aligned}$$

we have

$$p_{E\Theta}(E, \theta) = \frac{E}{2\pi\sigma^2}e^{-(E^2 - 2AE\cos\theta + A^2)/2\sigma^2} \tag{9.84}$$

where σ^2 is the variance of n_c (or n_s) and is equal to $2\mathcal{N}B$ for white noise. From Eq. (9.84) we have

$$\begin{aligned} p_E(E) &= \int_{-\pi}^{\pi} p_{E\Theta}(E, \theta)\, d\theta \\ &= \frac{E}{\sigma^2}e^{-(E^2 + A^2)/2\sigma^2}\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} e^{(AE/\sigma^2)\cos\theta}\, d\theta\right] \end{aligned} \tag{9.85}$$

The bracketed term on the right-hand side of Eq. (9.85) defines $I_0(AE/\sigma^2)$, where I_0 is the **modified zero-order Bessel function** of the first kind. Thus,

$$p_E(E) = \frac{E}{\sigma^2}e^{-(E^2 + A^2)/2\sigma^2} I_0\left(\frac{AE}{\sigma^2}\right) \tag{9.86a}$$

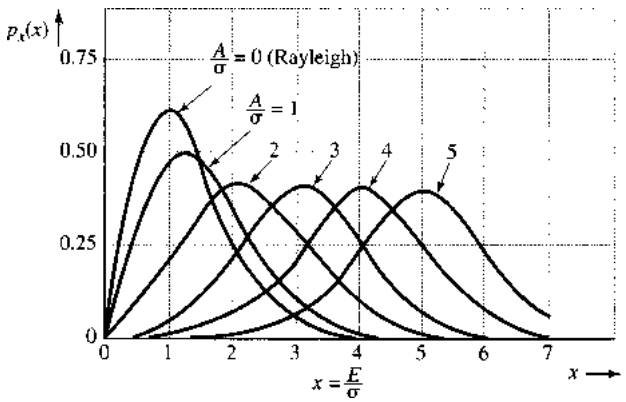


Figure 9.26
Rican PDF.

This is known as the **Rice density**, or **Ricean density**. For a large sinusoidal signal ($A \gg \sigma$), it can be shown that[4]

$$I_0\left(\frac{AE}{\sigma^2}\right) \simeq \sqrt{\frac{\sigma^2}{2\pi AE}} e^{AE/\sigma^2}$$

and

$$P_{\mathrm{E}}(E) \simeq \sqrt{\frac{E}{2\pi A\sigma^2}} e^{-(E-A)^2/2\sigma^2} \tag{9.86b}$$

Because $A \gg \sigma$, $E \simeq A$, and $p_{\mathrm{E}}(E)$ in Eq. (9.86b) is very nearly a Gaussian density with mean A and variance σ,

$$p_{\mathrm{E}}(E) \simeq \frac{1}{\sigma\sqrt{2\pi}} e^{-(E-A)^2/2\sigma^2} \tag{9.86c}$$

Figure 9.26 shows the PDF of the normalized RV E/σ. Note that for $A/\sigma = 0$, we obtain the Rayleigh density.

From the joint PDF $p_{\mathrm{E}\Theta}(E, \theta)$, we can also obtain $p_{\Theta}(\theta)$, the PDF of the phase Θ, by integrating the joint PDF with respect to E,

$$p_{\Theta}(\theta) = \int_0^{\infty} p_{\mathrm{E}\Theta}(E,\ \theta)\, dE$$

Although the integration is straightforward, there are a number of involved steps, and for this reason it will not be repeated here. The final result is

$$p_{\Theta}(\theta) = \frac{1}{2\pi} e^{-A^2/2\sigma^2} \left\{1 + \frac{A}{\sigma}\sqrt{2\pi}\cos\theta e^{A^2\cos^2\theta/2\sigma^2}\left[1 - Q\left(\frac{A\cos\theta}{\sigma}\right)\right]\right\} \tag{9.86d}$$

REFERENCES

1. B. P. Lathi, *An Introduction to Random Signals and Communication Theory*, International Textbook Co., Scranton, PA, 1968.
2. J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, Wiley, New York, 1965.

3. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 2nd ed., McGraw-Hill, New York, 1984.
4. S. O. Rice, "Mathematical Analysis of Random Noise," *Bell. Syst. Tech. J.*, vol. 23, pp. 282–332, July 1944; vol. 24, pp. 46–156, Jan. 1945.

PROBLEMS

9.1-1 **(a)** Sketch the ensemble of the random process

$$\mathrm{x}(t) = \mathrm{a}\cos(\omega_c t + \Theta)$$

where ω_c and Θ are constants and a is an RV uniformly distributed in the range $(-A, A)$.

(b) Just by observing the ensemble, determine whether this is a stationary or a nonstationary process. Give your reasons.

9.1-2 Repeat part **(a)** of Prob. 9.1-1 if a and Θ are constants but ω_c is an RV uniformly distributed in the range (0, 100).

9.1-3 **(a)** Sketch the ensemble of the random process

$$\mathrm{x}(t) = \mathrm{a}t + b$$

where b is a constant and a is an RV uniformly distributed in the range $(-2, 2)$.

(b) Just by observing the ensemble, state whether this is a stationary or a nonstationary process.

9.1-4 Determine $\overline{\mathrm{x}(t)}$ and $R_{\mathrm{x}}(t_1, t_2)$ for the random process in Prob. 9.1-1, and determine whether this is a wide-sense stationary process.

9.1-5 Repeat Prob. 9.1-4 for the process $\mathrm{x}(t)$ in Prob. 9.1-2.

9.1-6 Repeat Prob. 9.1-4 for the process $\mathrm{x}(t)$ in Prob. 9.1-3.

9.1-7 Given a random process $\mathrm{x}(t) = \mathrm{k}t$, where k is an RV uniformly distributed in the range $(-1, 1)$.

(a) Sketch the ensemble of this process.

(b) Determine $\overline{\mathrm{x}(t)}$.

(c) Determine $R_{\mathrm{x}}(t_1, t_2)$.

(d) Is the process wide-sense stationary?

(e) Is the process ergodic?

(f) If the process is wide-sense stationary, what is its power P_s [that is, its mean square value $\overline{\mathrm{x}^2(t)}$?

9.1-8 Repeat Prob. 9.1-7 for the random process

$$\mathrm{x}(t) = \mathrm{a}\cos(\omega_c t + \Theta)$$

where ω_c is a constant and a and Θ are independent RVs uniformly distributed in the ranges $(-1, 1)$ and $(0, 2\pi)$, respectively.

9.2-1 For each of the following functions, state whether it can be a valid PSD of a real random process.

(a) $\dfrac{(2\pi f)^2}{(2\pi f)^2 + 16}$

(b) $\dfrac{1}{(2\pi f)^2 - 16}$

(c) $\dfrac{(2\pi f)}{(2\pi f)^2 + 16}$

(d) $\delta(2\pi f) + \dfrac{1}{(2\pi f)^2 + 16}$

(e) $\delta[2\pi(f + f_0)] - \delta[2\pi(f - f_0)]$

(f) $j[\delta(f + f_0) + \delta(f - f_0)]$

(g) $\dfrac{j(2\pi f)^2}{(2\pi f)^2 + 16}$

9.2-2 Show that for a wide-sense stationary process $\mathrm{x}(t)$,

(a) $R_{\mathrm{x}}(0) \geq |R_{\mathrm{x}}(\tau)| \qquad \tau \neq 0$

Hint: $\overline{(\mathrm{x}_1 \pm \mathrm{x}_2)^2} = \overline{\mathrm{x}_1^2} + \overline{\mathrm{x}_2^2} \pm \overline{2\mathrm{x}_1\mathrm{x}_2} \geq 0$. Let $\mathrm{x}_1 = \mathrm{x}(t_1)$ and $\mathrm{x}_2 = \mathrm{x}(t_2)$.

(b) $\lim_{\tau \to \infty} R_{\mathrm{x}}(\tau) = \bar{\mathrm{x}}^2$

Hint: As $\tau \to \infty$, x_1 and x_2 tend to become independent.

9.2-3 Show that if the PSD of a random process $\mathrm{x}(t)$ is band-limited, and if

$$R_x\left(\frac{n}{2B}\right) = \begin{cases} 1 & n = 0 \\ 0 & n = \pm 1,\ \pm 2,\ \pm 3,\ \ldots \end{cases}$$

then the minimum bandwidth process $\mathrm{x}(t)$ that can exhibit this autocorrelation function is a white band-limited process; that is, $S_{\mathrm{x}}(f) = k\,\Pi(f/2W)$.

Hint: Use the sampling theorem to reconstruct $R_{\mathrm{x}}(\tau)$.

9.2-4 For the random binary process in Example 9.5 (Fig. 9.9a), determine $R_{\mathrm{x}}(\tau)$ and $S_{\mathrm{x}}(f)$ if the probability of transition (from 1 to -1 or vice versa) at each node is p instead of 0.5.

9.2-5 A wide-sense stationary white process $\mathrm{m}(t)$ band-limited to B Hz is sampled at the Nyquist rate. Each sample is transmitted by a basic pulse $p(t)$ multiplied by the sample value. This is a PAM signal. Show that the PSD of the PAM signal is $2BR_{\mathrm{m}}(0)|P(f)|^2$.

Hint: Use Eq. (9.31). Show that Nyquist samples a_k and a_{k+n} $(n \geq 1)$ are uncorrelated.

9.2-6 A duobinary line code proposed by Lender is a ternary scheme similar to bipolar that requires only half the bandwidth of the latter. In this code, **0** is transmitted by no pulse, and **1** is transmitted by pulse $p(t)$ or $-p(t)$ using the following rule: A **1** is encoded by the same pulse as that used to encode the preceding **1** if the two **1**s are separated by an even number of **0**s. It is encoded by the negative of the pulse used to encode the preceding **1** if the two **1**s are separated by an odd number of **0**s. Random binary digits are transmitted every T_b seconds. Assuming $P(\mathbf{0}) = P(\mathbf{1}) = 0.5$, show that

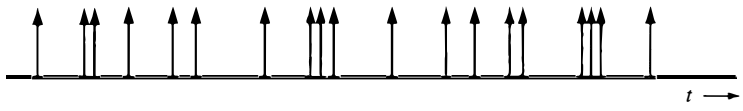
$$S_y(f) = \frac{|P(f)|^2}{T_b} \cos^2(\pi f T_b)$$

Find $S_{\mathrm{y}}(f)$ if $p(t)$, the basic pulse used, is a half-width rectangular pulse $\Pi(2t/T_b)$.

9.2-7 Determine $S_{\mathrm{y}}(f)$ for polar signaling if $P(\mathbf{1}) = Q$ and $P(\mathbf{0}) = 1 - Q$.

9.2-8 An impulse noise $\mathrm{x}(t)$ can be modeled by a sequence of unit impulses located at random instants (Fig. P9.2-8). There are an average of α impulses per second, and the location of any impulse is independent of the locations of other impulses. Show that $R_{\mathrm{x}}(\tau) = \alpha\,\delta(\tau) + \alpha^2$.

Figure P.9.2-8



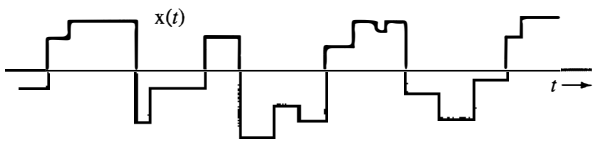
9.2-9 Repeat Prob. 9.2-8 if the impulses are equally likely to be positive and negative.

9.2-10 A sample function of a random process $\mathrm{x}(t)$ is shown in Fig. P9.2-10. The signal $\mathrm{x}(t)$ changes abruptly in amplitude at random instants. There are an average of β amplitude changes (or shifts) per second. The probability that there will be no amplitude shift in τ seconds is given by $P_0(\tau) = e^{-\beta\tau}$. The amplitude after a shift is independent of the amplitude before the shift. The amplitudes are randomly distributed, with a PDF $p_{\mathrm{x}}(x)$. Show that

$$R_{\mathrm{x}}(\tau) = \overline{\mathrm{x}^2} e^{-\beta|\tau|} \qquad \text{and} \qquad S_{\mathrm{x}}(f) = \frac{2\beta\overline{\mathrm{x}^2}}{\beta^2 + (2\pi f)^2}$$

This process represents a model for thermal noise.[1]

Figure P.9.2-10



9.3-1 Show that for jointly wide-sense stationary, real, random processes $\mathrm{x}(t)$ and $\mathrm{y}(t)$,

$$|R_{\mathrm{xy}}(\tau)| \leq [R_{\mathrm{x}}(0)R_{\mathrm{y}}(0)]^{1/2}$$

Hint: For any real number a, $\overline{(a\mathrm{x} - \mathrm{y})^2} \geq 0$.

9.3-2 If $\mathrm{x}(t)$ and $\mathrm{y}(t)$ are two incoherent random processes, and two new processes $\mathrm{u}(t)$ and $\mathrm{v}(t)$ are formed as follows:

$$\mathrm{u}(t) = 2\mathrm{x}(t) - \mathrm{y}(t) \qquad \mathrm{v}(t) = \mathrm{x}(t) + 3\mathrm{y}(t)$$

find $R_{\mathrm{u}}(\tau)$, $R_{\mathrm{v}}(\tau)$, $R_{\mathrm{uv}}(\tau)$, and $R_{\mathrm{vu}}(\tau)$ in terms of $R_{\mathrm{x}}(\tau)$ and $R_{\mathrm{y}}(\tau)$.

9.3-3 Two random processes $\mathrm{x}(t)$ and $\mathrm{y}(t)$ are

$$\mathrm{x}(t) = A\cos(\omega_0 t + \varphi) \qquad \text{and} \qquad \mathrm{y}(t) = B\sin(n\omega_0 t + n\varphi + \psi)$$

where n = integer $\neq 0$ and A, B, ψ, and ω_0 are constants and φ is an RV uniformly distributed in the range $(0, 2\pi)$. Show that the two processes are incoherent.

9.3-4 A sample signal is a periodic random process $\mathrm{x}(t)$ shown in Fig. P9.3-4. The initial delay b where the first pulse begins is an RV uniformly distributed in the range $(0, T_b)$.

(a) Show that the sample signal can be written as

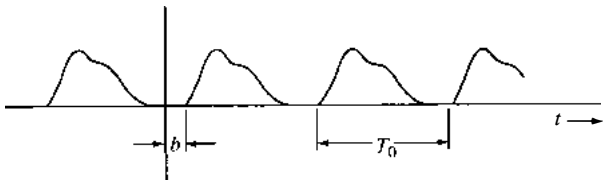
$$\mathrm{x}(t) = C_0 + \sum_{n=1}^{\infty} C_n \cos[n\omega_0(t - \mathrm{b}) + \theta_n]$$

by first finding its trigonometric Fourier series when b = 0.

(b) Show that

$$R_x(\tau) = C_0^2 + \frac{1}{2}\sum_{n=1}^{\infty} C_n^2 \cos n\omega_0\tau \qquad \omega_0 = \frac{2\pi}{T_0}$$

Figure P.9.3-4

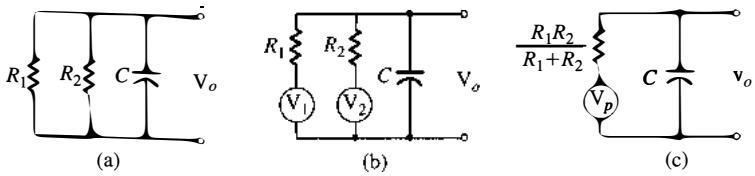


9.4-1 A simple *RC* circuit has two resistors R_1 and R_2 in parallel (Fig. P9.4-1a). Calculate the rms value of the thermal noise voltage v_o across the capacitor in two ways:

(a) Consider resistors R_1 and R_2 as two separate resistors, with respective thermal noise voltages of PSD $2kTR_1$ and $2kTR_2$ (Fig. P9.4-1b). Note that the two sources are independent.

(b) Consider the parallel combination of R_1 and R_2 as a single resistor of value $R_1R_2/(R_1+R_2)$, with its thermal-noise voltage source of PSD $2kTR_1R_2/(R_1+R_2)$ (Fig. P9.4-1c). Comment.

Figure P.9.4-1



9.4-2 Show that $R_{xy}(\tau)$, the cross-correlation function of the input process $x(t)$ and the output process $y(t)$ in Fig. 9.12, is

$$R_{xy}(\tau) = h(\tau) * R_x(\tau) \qquad \text{and} \qquad S_{xy}(f) = H(f)S_x(f)$$

Hence, show that for the thermal noise $n(t)$ and the output $v_o(t)$ in Fig. 9.13 (Example 9.9),

$$S_{nv_o}(f) = \frac{2kTR}{1+j2\pi fRC} \qquad \text{and} \qquad R_{nv_o}(\tau) = \frac{2kT}{C}e^{-\tau/RC}u(\tau)$$

9.4-3 A shot noise is similar to impulse noise described in Prob. 9.2-8 except that instead of random impulses, we have pulses of finite width. If we replace each impulse in Fig. P9.2-8 by a pulse $h(t)$ whose width is large in comparison to $1/\alpha$, so that there is a considerable overlapping of pulses, we get shot noise. The result of pulse overlapping is that the signal looks like a continuous random signal, as shown in Fig. P9.4-3.

(a) Derive the autocorrelation function and the PSD of such a random process.

Hint: Shot noise results from passing impulse noise through a suitable filter. First derive the PSD of the shot noise and then obtain the autocorrelation function from the PSD. The answers will be in terms of α, $h(t)$, or $H(f)$.

(b) The shot noise in transistors can be modeled by

$$h(t) = \frac{q}{T} e^{-t/T} u(t)$$

where q is the charge on an electron and T is the electron transit time. Determine and sketch the autocorrelation function and the PSD of the transistor shot noise.

Figure P.9.4-3



9.6-1 A signal process $\mathrm{m}(t)$ is mixed with a channel noise $\mathrm{n}(t)$. The respective PSDs are

$$S_{\mathrm{m}}(f) = \frac{6}{9 + (2\pi f)^2} \qquad \text{and} \qquad S_{\mathrm{n}}(f) = 6$$

(a) Find the optimum Wiener-Hopf filter.

(b) Sketch its unit impulse response.

(c) Estimate the amount of delay necessary to make this filter closely realizable (causal).

(d) Compute the noise power at the input and the output of the filter.

9.6-2 Repeat Prob. 9.6-1 if

$$S_{\mathrm{m}}(f) = \frac{4}{4 + (2\pi f)^2} \qquad \text{and} \qquad S_{\mathrm{n}}(f) = \frac{32}{64 + (2\pi f)^2}$$

9.7-1 A message signal $\mathrm{m}(t)$ with

$$S_{\mathrm{m}}(f) = \left[\frac{\alpha^2}{(2\pi f)^2 + \alpha^2}\right]^2 \qquad (\alpha = 3000\pi)$$

DSB-SC modulates a carrier of 100 kHz. Assume an ideal channel with $H_c(f) = 10^{-3}]$ and the channel noise PSD $S_{\mathrm{n}}(f) = 2 \times 10^{-9}$. The transmitted power is required to be 1 kW, and $G = 10^{-2}]$.

(a) Determine transfer functions of optimum preemphasis and deemphasis filters.

(b) Determine the output signal power, the noise power, and the output SNR.

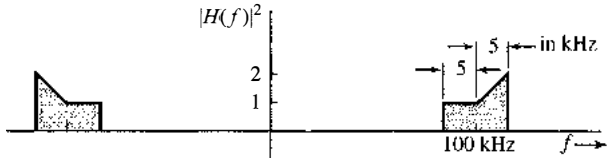
(c) Determine γ at the demodulator input.

9.7-2 Repeat Prob. 9.7-1 for the SSB (USB) case.

9.7-3 It was shown in the text that when the baseband $\mathrm{m}(t)$ is band-limited with a uniform PSD, PM and FM have identical performance from the SNR point of view. For such $\mathrm{m}(t)$, show that optimum PDE filters in angle modulation can improve the output SNR by a factor of 4/3 (or 1.3 dB) only. Find the optimum PDE filter transfer functions.

9.8-1 A white process of PSD $\mathcal{N}/2$ is transmitted through a bandpass filter $H(f)$ (Fig. P9.8-1). Represent the filter output $\mathrm{n}(t)$ in terms of quadrature components, and determine $S_{\mathrm{n_c}}(f)$, $S_{\mathrm{n_s}}(f)$, $\overline{\mathrm{n}_c^2}$, $\overline{\mathrm{n}_s^2}$, and $\overline{\mathrm{n}^2}$ when the center frequency used in this representation is 100 kHz (i.e., $f_c = 100 \times 10^3$).

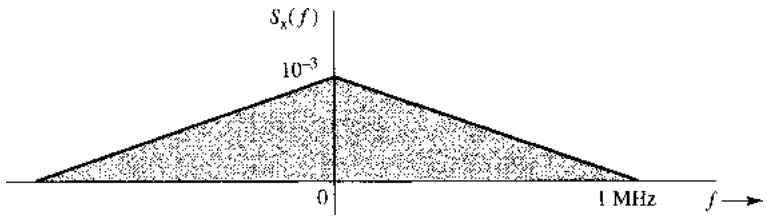
Figure P.9.8-1



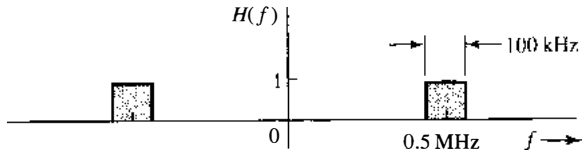
9.8-2 Repeat Prob. 9.8-1 if the center frequency f_c used in the representation is not a true center frequency. Consider three cases: **(a)** $f_c = 105$ kHz; **(b)** $f_c = 95$ kHz; **(c)** $f_c = 120$ kHz.

9.8-3 A random process $\mathrm{x}(t)$ with the PSD shown in Fig. P9.8-3a is passed through a bandpass filter (Fig. P9.8-3b). Determine the PSDs and mean square values of the quadrature components of the output process. Assume the center frequency in the representation to be 0.5 MHz.

Figure P.9.8-3



(a)



(b)

10 PERFORMANCE ANALYSIS OF DIGITAL COMMUNICATION SYSTEMS

In analog communications, the user objective is to achieve high fidelity in waveform reproduction; Hence, the suitable performance criterion is the output signal-to-noise ratio. The choice of this criterion indicates that the signal-to-noise ratio reflects the quality of the message and is related to the ability of a listener to interpret a message.

In digital communication systems, the transmitter input is chosen from a finite set of possible symbols. The objective at the receiver is not to reproduce the waveform that carries the symbol with fidelity; instead, the receiver aims to accurately determine which particular symbol was transmitted among the set of possible ones. Because each symbol is represented by a particular waveform at the transmitter, our goal is to decide, from the noisy received signal, which particular waveform was originally transmitted. Logically, the appropriate figure of merit in a digital communication system is the probability of error in this decision at the receiver. In particular, the probability of bit error, also known as the bit error rate (BER), is a direct quality measure of the communication system. Not only is the BER important to digital signal sources, it is also directly related to the quality of signal reproduction for analog signal sources.

In this chapter, we present two important aspects in the performance analysis of digital communication systems. The first part focuses on the error analysis of several specific binary detection receivers. The goal is for students to learn how to apply the fundamental tools of probability theory and random processes for BER performance analysis. Our second focus is to illustrate detailed derivation of *optimum detection receivers* for general digital communication systems such that the receiver BER can be minimized.

10.1 OPTIMUM LINEAR DETECTOR FOR BINARY POLAR SIGNALING

In binary communication systems, the information is transmitted as **0** or **1** in each time interval T_o. To begin, we consider the binary polar signaling system of Fig. 10.1a, in which the source signal bit **1** and **0** are represented by $\pm p(t)$, respectively. Having passed a distortionless, but

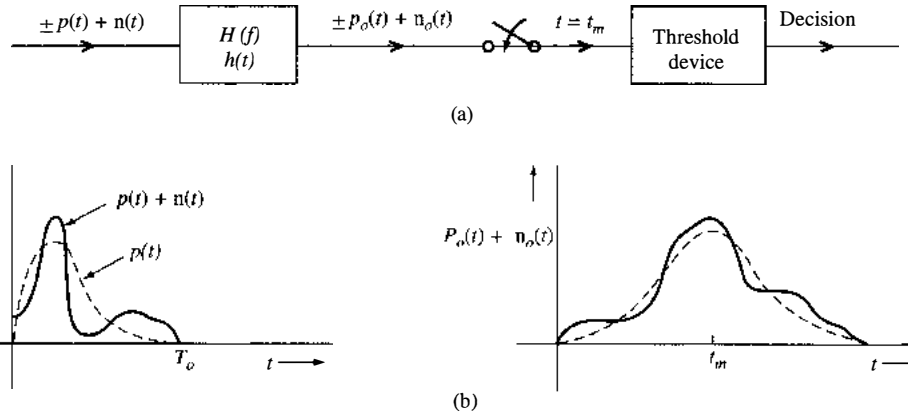


Figure 10.1 Typical binary polar signaling and linear receiver.

noisy, channel, the received signal waveform is

$$y(t) = \pm p(t) + \mathrm{n}(t) \qquad 0 \le t \le T_0 \tag{10.1}$$

where $\mathrm{n}(t)$ is a Gaussian channel noise.

10.1.1 Binary Threshold Detection

Given the received waveform of Eq. (10.1), the binary receiver must decide whether the transmission was originally a **1** or a **0**. Thus, the received signal $y(t)$ must be processed to produce a decision variable for each symbol. The linear receiver for binary signaling, as shown in Fig. 10.1a, has a general architecture that can be optimum (to be shown later in Section 10.6). Given the receiver filter $H(f)$ or $h(t)$, its output signal for $0 \le t \le T_0$ is simply

$$y(t) = \pm \underbrace{p(t) * h(t)}_{p_o(t)} + \underbrace{\mathrm{n}(t) * h(t)}_{\mathrm{n}_o(t)} = \pm p_o(t) + \mathrm{n}_o(t) \tag{10.2}$$

The decision variable of this linear binary receiver is the sample of the receiver filter output at $t = t_m$:

$$r(t_m) = \pm p_o(t_m) + \mathrm{n}_o(t_m) \tag{10.3}$$

Based on the properties of Gaussian variables in Section 8.6,

$$\mathrm{n}_o(t) = \int_0^t \mathrm{n}(\tau) h(t - \tau)\, d\tau$$

is Gaussian with zero mean so long as $\mathrm{n}(t)$ is a zero mean Gaussian noise. If we define

$$A_p = p_o(t_m) \tag{10.4a}$$

$$\sigma_{\mathrm{n}}^2 = E\{\mathrm{n}_o(t_m)^2\} \tag{10.4b}$$

then this binary detection problem is exactly the same as the threshold detection of Example 8.16. We have shown in Example 8.16 that, if the binary data are equally likely

to be **0** or **1**, then the optimum threshold detection is

$$\text{dec}\{r(t_m)\} = \begin{cases} \mathbf{1} & \text{if } r(t_m) \geq 0 \\ \mathbf{0} & \text{if } r(t_m) < 0 \end{cases} \tag{10.5a}$$

whereas the probability of (bit) error is

$$P_e = Q(\rho) \tag{10.5b}$$

in which

$$\rho = \frac{A_p}{\sigma_n} \tag{10.5c}$$

To minimize P_e, we need to maximize ρ because $Q(\rho)$ decreases monotonically with ρ.

10.1.2 Optimum Receiver Filter—Matched Filter

Let the received pulse $p(t)$ be time-limited to T_o (Fig. 10.1). We shall keep the discussion as general as possible at this point. To minimize the BER or P_e, we should determine the best receiver filter $H(f)$ and the corresponding sampling instant t_m such that $Q(\rho)$ is minimized. In other words, we seek a filter with a transfer function $H(f)$ that maximizes

$$\rho^2 = \frac{p_o^2(t_m)}{\sigma_n^2} \tag{10.6}$$

which is coincidentally also the signal-to-noise ratio at time instant $t = t_m$.

First, denote the Fourier transform of $p(t)$ as $P(f)$ and the PSD of the channel noise $n(t)$ as $S_n(f)$. We will determine the optimum receiver filter in the frequency domain. Starting with

$$\begin{aligned} p_o(t) &= \mathcal{F}^{-1}[P(f)H(f)] \\ &= \int_{-\infty}^{\infty} P(f)H(f)e^{j2\pi ft}\, df \end{aligned}$$

we have the sample value at $t = t_m$

$$p_o(t_m) = \int_{-\infty}^{\infty} P(f)H(f)e^{j2\pi ft_m}\, df \tag{10.7}$$

On the other hand, the filtered noise has zero mean

$$\overline{n_o(t)} = \overline{\int_0^t n(\tau)h(t-\tau)\, d\tau} = \int_0^t \overline{n(\tau)}h(t-\tau)\, d\tau = 0$$

while its variance is given by

$$\sigma_n^2 = \overline{n_o^2(t)} = \int_{-\infty}^{\infty} S_n(f)|H(f)|^2\, df \tag{10.8}$$

Hence, the signal-to-noise ratio is given in the frequency domain as

$$\rho^2 = \frac{\left|\int_{-\infty}^{\infty} H(f)P(f)e^{j2\pi ft_m}\,df\right|^2}{\int_{-\infty}^{\infty} S_n(f)|H(f)|^2\,df} \tag{10.9}$$

The Cauchy-Schwarz inequality (Appendix B) is a very powerful tool for finding the optimum filter $H(f)$. We can simply identify

$$X(f) = H(f)\sqrt{S_n(f)} \qquad Y(f) = \frac{P(f)e^{j2\pi ft_m}}{\sqrt{S_n(f)}}$$

Then by applying the Cauchy-Schwarz inequality to the numerator of Eq. (10.9), we have

$$\begin{aligned}
\rho^2 &= \frac{\left|\int_{-\infty}^{\infty} X(f)Y(f)\,df\right|^2}{\int_{-\infty}^{\infty} |X(f)|^2\,df} \\
&\le \frac{\int_{-\infty}^{\infty} |X(f)|^2\,df \cdot \int_{-\infty}^{\infty} |Y(f)|^2\,df}{\int_{-\infty}^{\infty} |X(f)|^2\,df} \\
&= \int_{-\infty}^{\infty} |Y(f)|^2\,df \\
&= \int_{-\infty}^{\infty} \frac{|P(f)|^2}{S_n(f)}\,df
\end{aligned} \tag{10.10a}$$

with equality if and only if $X(f) = k[Y(f)]^*$ or

$$H(f)\sqrt{S_n(f)} = k\left[\frac{P(f)e^{j2\pi ft_m}}{\sqrt{S_n(f)}}\right]^* = \frac{kP(-f)e^{-j2\pi ft_m}}{\sqrt{S_n(f)}}$$

Hence, the SNR is maximized if and only if

$$H(f) = k\frac{P(-f)e^{-j2\pi ft_m}}{S_n(f)} \tag{10.10b}$$

where k is an arbitrary constant. This optimum receiver filter is known as the **matched filter**. This optimum result states that the **best filter** at the binary linear receiver depends on several important factors: (1) the noise PSD $S_n(f)$, (2) the sampling instant t_m, and (3) the pulse shape $P(f)$. It is independent of the gain at the receiver k, since the same gain would apply to both the signal and the noise without affecting the SNR.

For white channel noise $S_n(f) = \mathcal{N}/2$, Eq. (10.10a) becomes

$$\rho^2 \le \rho_{\max}^2 = \frac{2}{\mathcal{N}}\int_{-\infty}^{\infty} |P(f)|^2\,df = \frac{2E_p}{\mathcal{N}} \tag{10.11a}$$

where E_p is the energy of $p(t)$, and the matched filter is simply

$$H(f) = k'P(-f)e^{-j2\pi ft_m} \tag{10.11b}$$

where $k' = 2k/\mathcal{N}$ is an arbitrary constant.

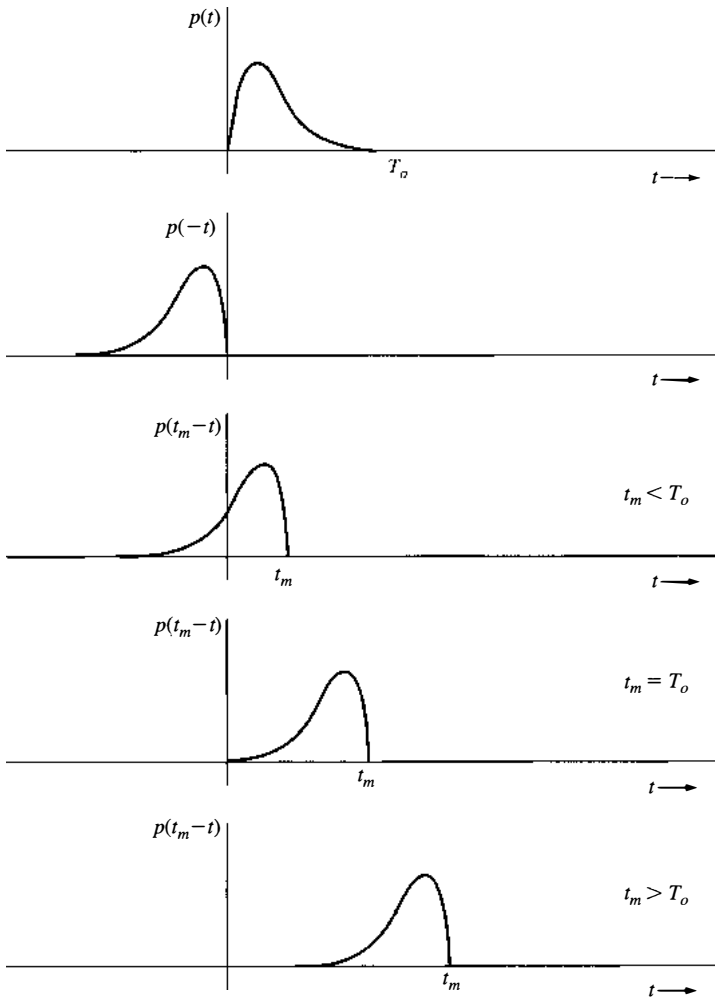


Figure 10.2 Optimum choice for sampling instant.

The unit impulse response $h(t)$ of the optimum filter is obtained from the inverse Fourier transform

$$h(t) = \mathcal{F}^{-1}[k'P(-f)e^{-j2\pi ft_m}]$$

Note that $p(-t) \Longleftrightarrow P(-f)$ and $e^{-j2\pi ft_m}$ represents the time delay of t_m seconds. Hence,

$$h(t) = k'p(t_m - t) \tag{10.11c}$$

The response $p(t_m - t)$ is the signal pulse $p(-t)$ delayed by t_m. Three cases, $t_m < T_o$, $t_m = T_o$, and $t_m > T_o$, are shown in Fig. 10.2. The first case, $t_m < T_o$, yields a noncausal impulse response, which is unrealizable.* Although the other two cases yield physically realizable filters, the last case, $t_m > T_o$, delays the decision-making instant t_m unnecessarily. The case

* The filter unrealizability can be readily understood intuitively when the decision-making instant is $t_m < T_o$. In this case, we are forced to make a decision before the full pulse has been fed to the filter ($t_m < T_o$). This calls for a prophetic filter, which can respond to inputs before they are applied. As we know, only unrealizable (noncausal) filters can do this job.

$t_m = T_o$ gives the minimum delay for decision making using a realizable filter. In our future discussion, we shall assume $t_m = T_o$, unless otherwise specified.

Observe that both $p(t)$ and $h(t)$ have a width of T_o seconds. Hence, $p_o(t)$, which is a convolution of $p(t)$ and $h(t)$, has a width of $2T_o$ seconds, with its peak occurring at $t = T_o$ where the decision sample is taken. Also, because $P_o(f) = P(f)H(f) = k'|P(f)|^2 e^{-j2\pi f T_o}$, $p_o(t)$ is symmetrical about $t = T_o$.*

Since the gain k' does not affect the SNR ρ, we choose $k' = 1$. This gives the matched filter under white noise

$$h(t) = p\,(T_o - t) \tag{10.12a}$$

or equivalently

$$H(f) = P(-f)e^{-j2\pi f T_o} \tag{10.12b}$$

for which the signal to noise ratio is maximum at the decision-making instant $t = T_o$.

The matched filter is optimum in the sense that it maximizes the signal-to-noise ratio at the decision-making instant. Although it is reasonable to assume that maximization of this particular signal-to-noise ratio will minimize the detection error probability, we have not proven that the original structure of linear receiver with threshold detection (sample and decide) is the optimum structure. The optimality of the matched filter receiver under white Gaussian noise will be shown later (Section 10.6).

Given the matched filter under white Gaussian noise, the matched filter receiver leads to $\rho_{\max}$ of Eq. (10.11a) as well as the minimum BER of

$$P_e = Q(\rho_{\max}) = Q\left(\sqrt{\frac{2E_p}{\mathcal{N}}}\right) \tag{10.13}$$

Equation (10.13) is quite remarkable. It shows that, as far as the system performance is concerned, when the matched filter receiver is used, various waveforms used for $p(t)$ are equivalent as long as they have the same energy

$$E_p = \int_{-\infty}^{\infty} |P(f)|^2\,df = \int_0^{T_o} |p(t)|^2\,dt$$

The matched filter may also be implemented by the alternative arrangement shown in Fig. 10.3. If the input to the matched filter is $\mathrm{y}(t)$, then the output $\mathrm{r}(t)$ is given by

$$\mathrm{r}(t) = \int_{-\infty}^{\infty} \mathrm{y}(x)h(t-x)\,dx \tag{10.14}$$

where $h(t) = p\,(T_o - t)$ and

$$h(t-x) = p\,[T_o - (t-x)] = p\,(x + T_o - t) \tag{10.15}$$

* This follows from the fact that because $|P(f)|^2$ is an even function of f, its inverse transform is symmetrical about $t = 0$ (see Prob. 3.1-1). The output from the previous input pulse terminates and has a zero value at $t = T_o$. Similarly, the output from the following pulse starts and has a zero value at $t = T_o$. Hence, at the decision-making instant T_o, no intersymbol interference occurs.

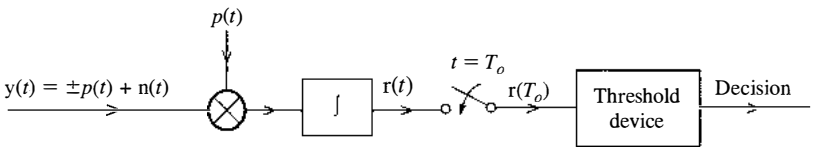


Figure 10.3 Correlation detector.

Hence,

$$\mathrm{r}(t) = \int_{-\infty}^{\infty} \mathrm{y}(x) p\,(x + T_o - t)\,dx \tag{10.16a}$$

At the decision-making instant $t = T_o$, we have

$$\mathrm{r}(T_o) = \int_{-\infty}^{\infty} \mathrm{y}(x)\,p\,(x)\,dx \tag{10.16b}$$

Because the input $\mathrm{y}(x)$ is assumed to start at $x = 0$ and $p\,(x) = 0$ for $x > T_o$, we have the decision variable

$$\mathrm{r}(T_o) = \int_{0}^{T_o} \mathrm{y}(x)\,p\,(x)\,dx \tag{10.16c}$$

We can implement Eqs. (10.16) as shown in Fig. 10.3. This type of arrangement, known as the correlation receiver, is equivalent to the matched filter receiver.

The right-hand side of Eq. (10.16c) is the cross-correlation of the received pulse with $p(t)$. Recall that correlation basically measures the similarity of signals (Sec. 2.7). Thus, the optimum detector measures the similarity of the received signal with the pulse $p(t)$. Based on this similarity measure, the sign of the correlation decides whether $p(t)$ or $-p(t)$ was transmitted.

Thus far we have discussed polar signaling in which only one basic pulse $p(t)$ of opposite signs is used. Generally, in binary communication, we use two distinct pulses $p(t)$ and $q(t)$ to represent the two symbols. The optimum receiver for such a case will now be discussed.

10.2 GENERAL BINARY SIGNALING

10.2.1 Optimum Linear Receiver Analysis

In a binary scheme where symbols are transmitted every T_b seconds, the more general transmission scheme may use two pulses $p(t)$ and $q(t)$ to transmit **1** and **0**. The optimum linear receiver structure under consideration is shown in Fig. 10.4a. The received signal is

$$y(t) = \begin{cases} p(t) + \mathrm{n}(t) & 0 \le t \le T_b \quad \text{for data symbol } \mathbf{1} \\ q(t) + \mathrm{n}(t) & 0 \le t \le T_b \quad \text{for data symbol } \mathbf{0} \end{cases}$$

The incoming signal $y(t)$ is transmitted through a filter $H(f)$, and the output $\mathrm{r}(t)$ is sampled at T_b. The decision of whether **0** or **1** was present at the input depends on whether is or is not $\mathrm{r}(T_b)$ is less than a_o, where a_o is the optimum threshold.

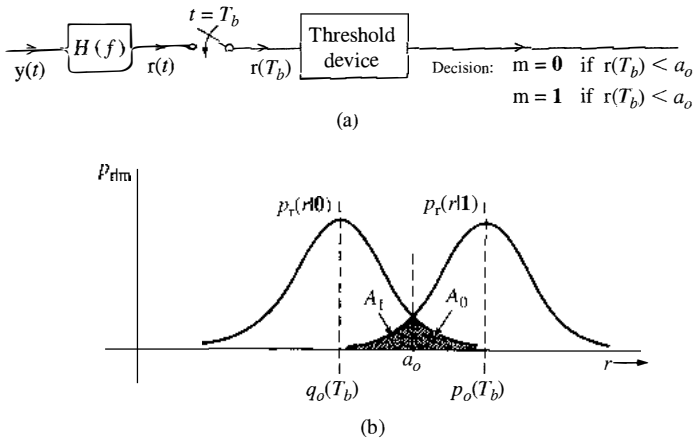


Figure 10.4 Optimum binary threshold detection.

Let $p_o(t)$ and $q_o(t)$ be the response of $H(f)$ to inputs $p(t)$ and $q(t)$, respectively. From Eq. (10.7) it follows that

$$p_o(T_b) = \int_{-\infty}^{\infty} P(f)H(f)e^{j2\pi fT_b}\,df \tag{10.17a}$$

$$q_o(T_b) = \int_{-\infty}^{\infty} Q(f)H(f)e^{j2\pi fT_b}\,df \tag{10.17b}$$

and σ_n^2, the variance, or power, of the noise at the filter output, is

$$\sigma_n^2 = \int_{-\infty}^{\infty} S_n(f)|H(f)|^2\,df \tag{10.17c}$$

Without loss of generality, we let $P_o(T_b) > P_o(T_b)$. Denote n as the noise output at T_b. Then the sampler output $\mathrm{r}(T_b) = q_o(T_b) + \mathrm{n}$ or $p_o(T_b) + \mathrm{n}$, depending on whether $\mathrm{m} = \mathbf{0}$ or $\mathrm{m} = \mathbf{1}$, is received. Hence, r is a Gaussian RV of variance σ_n^2 with mean $q_o(T_b)$ or $p_o(T_b)$, depending on whether $\mathrm{m} = \mathbf{0}$ or $\mathbf{1}$. Thus, the conditional PDFs of the sampled output $\mathrm{r}(T_b)$ are

$$p_{\mathrm{r}|\mathrm{m}}(r|\mathbf{0}) = \frac{1}{\sigma_n\sqrt{2\pi}}\exp\left(-\frac{[r - q_o(T_b)]^2}{2\sigma_n^2}\right)$$

$$p_{\mathrm{r}|\mathrm{m}}(r|\mathbf{1}) = \frac{1}{\sigma_n\sqrt{2\pi}}\exp\left(-\frac{[r - p_o(T_b)]^2}{2\sigma_n^2}\right)$$

Optimum Threshold

The two PDFs are shown in Fig. 10.4b. If a_o is the optimum threshold of detection, then the decision rule is

$$\mathrm{m} = \begin{cases} \mathbf{0} & \text{if } \mathrm{r} < a_o \\ \mathbf{1} & \text{if } \mathrm{r} > a_o \end{cases}$$

The conditional error probability $P(\epsilon \mid \mathrm{m} = \mathbf{0})$ is the probability of making a wrong decision when $\mathrm{m} = \mathbf{0}$. This is simply the area A_0 under $p_{\mathrm{r}|\mathrm{m}}(r|\mathbf{0})$ from a_o to ∞. Similarly, $P(\epsilon|\mathrm{m} = \mathbf{1})$

is the area A_1 under $p_{r|m}(r|\mathbf{1})$ from $-\infty$ to a_o (Fig. 10.4b), and

$$P_e = \sum_i P(\epsilon|m_i)P(m_i) = \frac{1}{2}(A_0 + A_1)$$
$$= \frac{1}{2}\left[Q\left(\frac{a_o - q_o(T_o)}{\sigma_n}\right) + Q\left(\frac{p_o(T_o) - a_o}{\sigma_n}\right)\right] \tag{10.18}$$

assuming $P_m(0) = P_m(1) = 0.5$. From Fig. 10.4b it can be seen that the sum $A_0 + A_1$ of the shaded areas is minimized by choosing a_o at the intersection of the two PDFs. This optimum threshold can also be determined directly by setting to zero the derivative of P_e in Eq. (10.18) with respect to a_o such that

$$\frac{\partial P_e}{\partial a_o} = \frac{1}{2}\left[Q'\left(\frac{a_o - q_o(T_o)}{\sigma_n}\right)\frac{1}{\sigma_n} - Q'\left(\frac{p_o(T_o) - a_o}{\sigma_n}\right)\frac{1}{\sigma_n}\right]$$
$$= \frac{1}{2\sigma_n}\left[\frac{1}{\sigma_n\sqrt{2\pi}}\exp\left[-\frac{[a_o - q_o(T_b)]^2}{2\sigma_n^2}\right] - \frac{1}{\sigma_n\sqrt{2\pi}}\exp\left(-\frac{[p_o(T_b) - a_o]^2}{2\sigma_n^2}\right)\right]$$
$$= 0$$

Thus, the optimum a_o is

$$a_o = \frac{p_o(T_b) + q_o(T_b)}{2} \tag{10.19a}$$

and the corresponding P_e is

$$P_e = P(\epsilon|\mathbf{0}) = P(\epsilon|\mathbf{1})$$
$$= \frac{1}{\sigma_n\sqrt{2\pi}}\int_{a_o}^{\infty}\exp\left(-\frac{[r - q_o(T_b)]^2}{2\sigma_n^2}\right)dr$$
$$= Q\left[\frac{a_o - q_o(T_b)}{\sigma_n}\right]$$
$$= Q\left[\frac{p_o(T_b) - q_o(T_b)}{2\sigma_n}\right] \tag{10.19b}$$
$$= Q\left(\frac{\beta}{2}\right) \tag{10.19c}$$

where we define

$$\beta = \frac{p_o(T_b) - q_o(T_b)}{\sigma_n} \tag{10.20}$$

Substituting Eq. (10.17) into Eq. (10.20), we get

$$\beta^2 = \frac{\left|\int_{-\infty}^{\infty}[P(f) - Q(f)]H(f)e^{j2\pi fT_b}\,df\right|^2}{\int_{-\infty}^{\infty}S_n(f)\,|H(f)|^2\,df}$$

This equation is of the same form as Eq. (10.9) with $P(f)$ replaced by $P(f) - Q(f)$. Hence, Cauchy-Schwarz inequality can again be applied to show

$$\beta_{\max}^2 = \int_{-\infty}^{\infty} \frac{|P(f) - Q(f)|^2}{S_n(f)}\, df \tag{10.21a}$$

and the optimum filter $H(f)$ is given by

$$H(f) = k\frac{[P(-f) - Q(-f)]e^{-j2\pi fT_b}}{S_n(f)} \tag{10.21b}$$

where k is an arbitrary constant.

The Special Case of White Gaussian Noise

For white noise $S_n(f) = \mathcal{N}/2$, and the optimum filter $H(f)$ is given by*

$$H(f) = [P(-f) - Q(-f)]e^{-j2\pi fT_b} \tag{10.22a}$$

and

$$h(t) = p(T_b - t) - q(T_b - t) \tag{10.22b}$$

This is a filter matched to the pulse $p(t) - q(t)$. The corresponding β is [Eq. (10.21a)]

$$\beta_{\max}^2 = \frac{2}{\mathcal{N}} \int_{-\infty}^{\infty} |P(f) - Q(f)|^2\, df \tag{10.23a}$$

$$= \frac{2}{\mathcal{N}} \int_0^{T_b} [p(t) - q(t)]^2\, dt \tag{10.23b}$$

$$= \frac{E_p + E_q - 2E_{pq}}{\mathcal{N}/2} \tag{10.23c}$$

where E_p and E_q are the energies of $p(t)$ and $q(t)$, respectively, and

$$E_{pq} = \int_0^{T_b} p(t)q(t)\, dt \tag{10.24}$$

So far, we have been using the notation P_e to denote error probability. In the binary case, this error probability is the **bit error probability** or **bit error rate** (BER) and will be denoted by P_b (rather than P_e). Thus, from Eqs. (10.19c) and (10.23c),

$$P_b = Q\left(\frac{\beta_{\max}}{2}\right) \tag{10.25a}$$

$$= Q\left(\sqrt{\frac{E_p + E_q - 2E_{pq}}{2\mathcal{N}}}\right) \tag{10.25b}$$

* Because k in Eq. (10.21b) is arbitrary, we choose $k = \mathcal{N}/2$ for convenience.

The optimum threshold a_o is obtained by substituting Eqs. (10.17a, b) and (10.22a) into Eq. (10.19a) and recognizing (via variable substitution) that

$$\int_{-\infty}^{\infty} P(f)Q(-f)\,df = \int_{-\infty}^{\infty} P(-f)Q(f)\,df = E_{pq} \tag{10.26}$$

This gives

$$a_o = \tfrac{1}{2}(E_p - E_q) \tag{10.27}$$

In deriving the optimum binary receiver, we assumed a certain receiver structure (the threshold detection receiver in Fig. 10.4). It is not clear yet whether there exists another structure that may have better performance than that in Fig. 10.4. It will be shown later (in Sec. 10.6) that for a Gaussian noise, the receiver derived here is the definite optimum. Equation (10.25b) gives P_b for the optimum receiver when the channel noise is white Gaussian. For the case of nonwhite noise, P_b is obtained by substituting $\beta_{\max}$ from Eq. (10.21a) into Eq. (10.25a).

Equivalent Optimum Binary Receivers

For the optimum receiver in Fig. 10.4a,

$$H(f) = P(-f)e^{-j2\pi fT_b} - Q(-f)e^{-j2\pi fT_b}$$

This filter can be realized as a parallel combination of two filters matched to $p(t)$ and $q(t)$, respectively, as shown in Fig. 10.5a. Yet another equivalent form is shown in Fig. 10.5b. Because the threshold is $(E_p - E_q)/2$, we subtract $E_p/2$ and $E_q/2$, respectively, from the two matched filter outputs. This is equivalent to shifting the threshold to 0. In the case of $E_p = E_q$, we need not subtract $E_p/2$ and $E_q/2$ from the two outputs, and the receiver simplifies to that shown in Fig. 10.5c.

10.2.2 Performance Analysis of General Binary Systems

In this section, we analyze the performance of several typical binary digital communication systems by applying the techniques derived in the last section for general binary receivers.

Polar Signaling

For the case of polar signaling, $q(t) = -p(t)$. Hence,

$$E_p = E_q \qquad \text{and} \qquad E_{pq} = -\int_{-\infty}^{\infty} p^2(t)\,dt = -E_p \tag{10.28}$$

Substituting these results into Eq. (10.25b) yields

$$P_b = Q\left(\sqrt{\frac{2E_p}{\mathcal{N}}}\right) \tag{10.29}$$

Also from Eq. (10.22b),

$$h(t) = 2p(T_b - t) \tag{10.30}$$

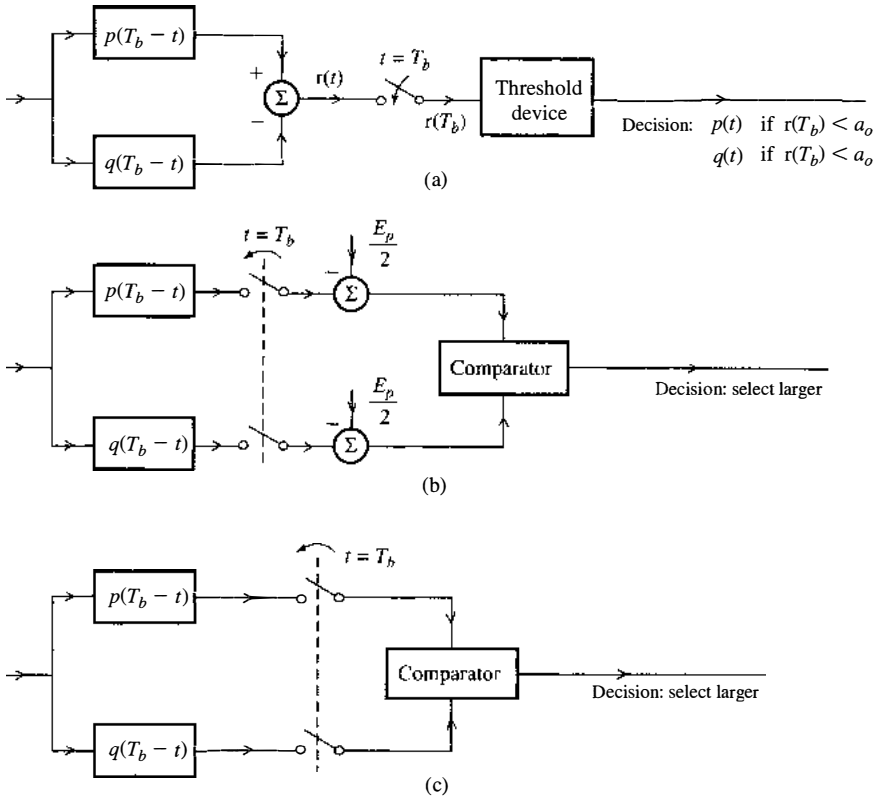


Figure 10.5 Realization of the optimum binary threshold detector.

Recall that the multiplication of $h(t)$ by any constant amplifies both the signal and the noise by the same factor, and hence does not affect the system performance. For convenience, we shall multiply $h(t)$ by 0.5 to obtain

$$h(t) = p(T_b - t) \tag{10.31}$$

From Eq. (10.27), the threshold a_o is

$$a_o = 0 \tag{10.32}$$

Therefore, for the polar case, the receiver in Fig. 10.5a reduces to that shown in Fig. 10.6a with threshold 0. This filter is equivalent to that in Fig. 10.3.

The error probability can be expressed in terms of a more basic parameter E_b, the energy per bit:

$$E_b = \text{ energy per bit}$$

In the polar case, $E_p = E_q$ and the bit energy E_b is

$$\begin{aligned} E_b &= E_p P(\mathrm{m} = \mathbf{1}) + E_q P(\mathrm{m} = \mathbf{0}) \\ &= E_p P(\mathrm{m} = \mathbf{1}) + E_p[1 - P(\mathrm{m} = \mathbf{1})] \\ &= E_p \end{aligned}$$

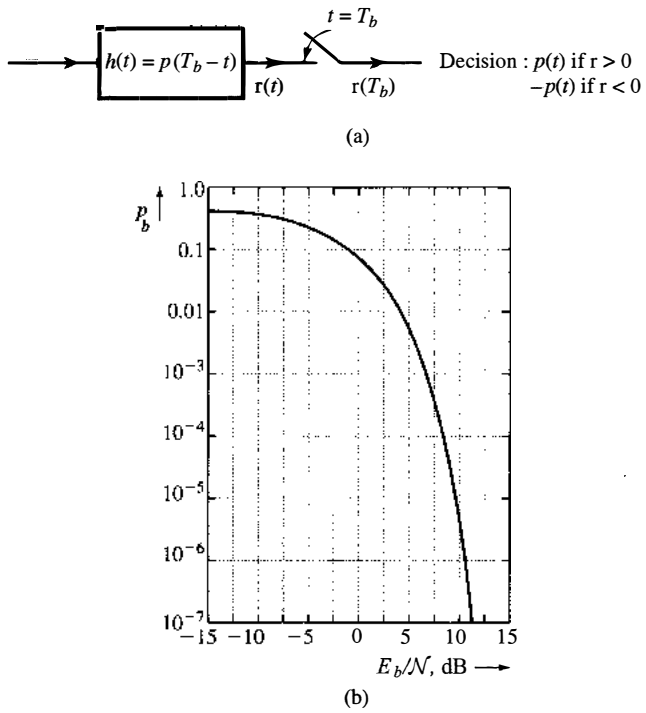


Figure 10.6 (a) Optimum threshold detector and (b) its error probability for polar signaling.

and from Eq. (10.29),

$$P_b = Q\left(\sqrt{\frac{2E_b}{\mathcal{N}}}\right) \tag{10.33}$$

The parameter $E_b/\mathcal{N}$ is the normalized energy per bit, which will be seen in future discussions as a fundamental parameter serving as a figure of merit in digital communication.* Because the signal power is equal to E_b times the bit rate, a given E_b is equivalent to a given signal power (for a given bit rate). Hence, when we compare systems, for a given value of E_b, we are comparing them for a given signal power.

Figure 10.6b plots P_b as a function of $E_b/\mathcal{N}$ (in decibels). Equation (10.33) indicates that, for optimum threshold detection, the polar system performance depends not on the pulse shape, but on the pulse energy.

On-Off Signaling

In the case of on-off signaling, $q(t) = 0$, and the receiver of Fig. 10.5a can remove the lower branch filter of $q(T_b - t)$. Based on Eq. (10.27), the optimum threshold for on-off signaling receiver is

$$a_o = E_p/2$$

* If the transmission rate is R_b pulses per second, the signal power S_i is $S_i = E_b R_b$, and $E_b/\mathcal{N} = S_i/\mathcal{N}R_b$. Observe that $S_i/\mathcal{N}R_b$ is similar to the parameter γ (signal-to-noise ratio $S_i/\mathcal{N}B$) used in analog systems.

Additionally, substituting $q(t) = 0$ into Eqs. (10.24) and (10.25) yields

$$E_q = 0, \qquad E_{pq} = 0, \qquad \text{and} \qquad P_b = Q\left(\sqrt{\frac{E_p}{2\mathcal{N}}}\right) \tag{10.34}$$

If both symbols m = **0** and m = **1** have equal probability 0.5, then the average bit energy is given by

$$E_b = \frac{E_p + E_q}{2} = \frac{E_p}{2}$$

Therefore, the BER can be written as

$$P_b = Q\left(\sqrt{\frac{E_b}{\mathcal{N}}}\right) \tag{10.35}$$

A comparison of Eqs. (10.35) and (10.33) shows that on-off signaling requires *exactly* twice as much energy per bit (3 dB more power) to achieve the same performance (i.e., the same P_b) as polar signaling.

Orthogonal Signaling

In orthogonal signaling, $p(t)$ and $q(t)$ are selected to be orthogonal over the interval $(0, T_b)$. This gives

$$E_{pq} = \int_0^{T_b} p(t)q(t)\,dt = 0 \tag{10.36}$$

On-off signaling is in fact a special case of orthogonal signaling. Two additional examples of binary orthogonal pulses are shown in Fig. 10.7. From Eq. (10.25),

$$P_b = Q\left(\sqrt{\frac{E_p + E_q}{2\mathcal{N}}}\right) \tag{10.37}$$

Assuming **1** and **0** to be equiprobable,

$$E_b = \frac{E_p + E_q}{2}$$

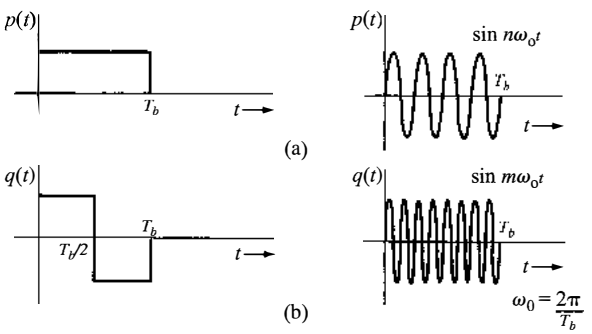


Figure 10.7 Examples of orthogonal signals.

and

$$P_b = Q\left(\sqrt{\frac{E_b}{\mathcal{N}}}\right) \tag{10.38}$$

This shows that the performance of any orthogonal binary signaling is inferior to that of polar signaling by 3 dB. This naturally includes on-off signaling.

10.3 COHERENT RECEIVERS FOR DIGITAL CARRIER MODULATIONS

We introduced amplitude shift keying (ASK), frequency shift keying (FSK), and phase shift keying (PSK) in Section 7.9. Figure 10.8 uses a rectangular baseband pulse to show the three binary schemes. The baseband pulse may be specifically shaped (e.g., a raised cosine) to eliminate intersymbol interference and to stay within a finite bandwidth.

BPSK

In particular, the binary PSK (BPSK) modulation transmits binary symbols via

$$\begin{aligned} \mathbf{1}: &\quad \sqrt{2}p'(t)\cos\omega_c t \\ \mathbf{0}: &\quad -\sqrt{2}p'(t)\cos\omega_c t \end{aligned}$$

Here $p'(t)$ denotes the baseband pulse shape. When $p(t) = \sqrt{2}p'(t)\cos\omega_c t$, this has exactly the same signaling form as the baseband polar signaling. Thus, the optimum binary receiver also takes the form of Fig. 10.5a. As a result, for equally likely binary data, the optimum threshold $a_o = 0$ and the minimum probability of detection error is identically

$$P_b = Q\left(\sqrt{\frac{2E_b}{\mathcal{N}}}\right) = Q\left(\sqrt{\frac{2E_p}{\mathcal{N}}}\right) \tag{10.39}$$

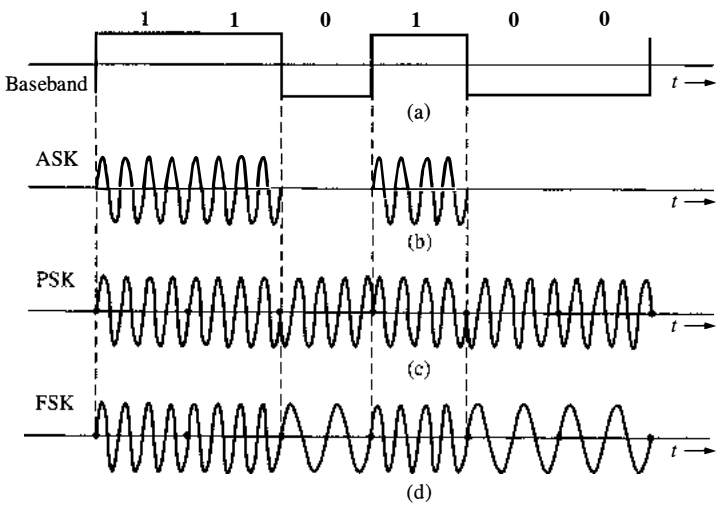


Figure 10.8 Digital modulated waveforms.

where the pulse energy is simply

$$\begin{aligned} E_p &= \int_0^{T_b} p^2(t)\,dt \\ &= 2\int_0^{T_b} [p'(t)]^2 \cos^2 \omega_c t\,dt \\ &= \int_0^{T_b} [p'(t)]^2\,dt \\ &= E_{p'} \end{aligned}$$

This result requires a carrier frequency sufficiently high such that $f_c T_b >> 1$.

Binary ASK

Similarly, for binary ASK, the transmission is

$$\begin{aligned} \mathbf{1}: &\quad \sqrt{2}p'(t)\cos \omega_c t \\ \mathbf{0}: &\quad 0 \end{aligned}$$

This coincides with the on-off signaling analyzed earlier such that the optimum threshold should be $a_o = E_p/2$ and the minimum BER for binary ASK is

$$P_b = Q\left(\sqrt{\frac{E_b}{N}}\right) \tag{10.40}$$

where

$$E_b = \frac{E_p}{2} = \frac{E_{p'}}{2}$$

Comparison of Eq. (10.39) and Eq. (10.40) shows that for the same performance, the pulse energy in ASK must be twice that in PSK. Hence, ASK requires 3 dB more power than PSK. Thus, in optimum (coherent) detection, PSK is always preferable to ASK. For this reason, ASK is of no practical importance in optimum detection. But ASK can be useful in noncoherent systems (e.g., optical communications). Envelope detection, for example, can be applied to ASK. In PSK, the information lies in the phase, and, hence, it cannot be detected noncoherently.

The baseband pulses $p(t)$ used in carrier systems should be shaped to minimize the ISI. The bandwidth of the PSK or ASK signal is twice that of the corresponding baseband signal because of modulation.*

Bandpass Matched Filter as a Coherent Receiver

For both PSK and ASK, the optimum matched filter receiver of Fig. 10.5a can be implemented. As shown in Fig. 10.9a, the received RF pulse can be detected by a filter matched to the RF pulse $p(t)$ followed by a sampler before a threshold detector.

On the other hand, the same matched filter receiver may also be modified into Fig. 10.9b without changing the signal samples for decision. The alternative implementation first demodulates the incoming RF signal coherently by multiplying it with $\sqrt{2}\cos \omega_c t$. The product is

* We can also use QAM (quadrature multiplexing) to double bandwidth efficiency.

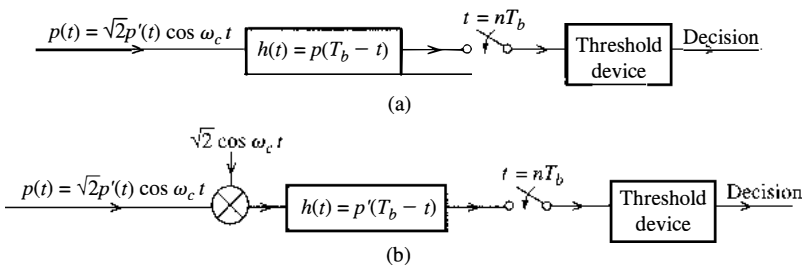


Figure 10.9 Coherent detection of digital modulated signals.

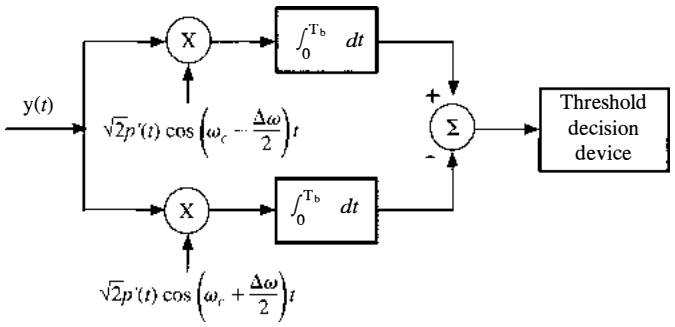


Figure 10.10 Optimum coherent detection of binary FSK signals.

the baseband pulse* $p'(t)$ plus a baseband noise with PSD $\mathcal{N}/2$ (see Example 9.13), and this is applied to a filter matched to the baseband pulse $p'(t)$. The two receiver schemes are equivalent. They can also be implemented as correlation receivers.

Frequency Shift Keying

In FSK, RF binary signals are transmitted as

$$\begin{aligned} \mathbf{0}: &\quad \sqrt{2}p'(t)\cos[\omega_c - (\Delta\omega/2)]t \\ \mathbf{1}: &\quad \sqrt{2}p'(t)\cos[\omega_c + (\Delta\omega/2)]t \end{aligned}$$

Such a waveform may be considered to be two interleaved ASK waves. Hence, the PSD will consist of two PSDs, centered at $[f_c - (\Delta f/2)]$ and $[f_c + (\Delta f/2)]$. For a large $\Delta f/f_c$, the PSD will consist of two nonoverlapping PSDs. For a small $\Delta f/f_c$, the two spectra merge, and the bandwidth decreases. But in no case is the bandwidth less than that of ASK or PSK.

The optimum correlation receiver for binary FSK is given in Fig. 10.10. Because the pulses have equal energy, when the symbols are equally likely, the optimum threshold $a_o = 0$.

Consider the rather common case of rectangular $p'(t) = A$, that is, no pulse shaping in FSK.

$$q(t) = \sqrt{2}A\cos\left(\omega_c - \frac{\Delta\omega}{2}\right)t$$

$$p(t) = \sqrt{2}A\cos\left(\omega_c + \frac{\Delta\omega}{2}\right)t$$

* There is also a spectrum of $p'(t)$ centered at $2\omega_c$, which is eventually eliminated by the filter matched to $p'(t)$.

To compute P_b from Eq. (10.25b), we need E_{pq},

$$\begin{aligned} E_{pq} &= \int_0^{T_b} p(t)q(t)\,dt \\ &= 2A^2 \int_0^{T_b} \cos\left(\omega_c - \frac{\Delta\omega}{2}\right)t \cos\left(\omega_c + \frac{\Delta\omega}{2}\right)t\,dt \\ &= A^2\left[\int_0^{T_b} \cos(\Delta\omega)t\,dt + \int_0^{T_b} \cos 2\omega_c t\,dt\right] \\ &= A^2 T_b\left[\frac{\sin(\Delta\omega)T_b}{(\Delta\omega)T_b} + \frac{\sin 2\omega_c T_b}{2\omega_c T_b}\right] \end{aligned}$$

In practice $\omega_c T_b \gg 1$, and the second term on the right-hand side can be ignored. Therefore,

$$E_{pq} = A^2 T_b \operatorname{sinc}(\Delta\omega T_b)$$

Similarly,

$$E_b = E_p = E_q = \int_0^{T_b} [p(t)]^2\,dt = A^2 T_b$$

The BER analysis of Eq. (10.25b) for equiprobable binary symbols **1** and **0** becomes

$$P_b = Q\left(\sqrt{\frac{E_b - E_b \operatorname{sinc}(\Delta\omega T_b)}{\mathcal{N}}}\right)$$

It is therefore clear that to minimize P_b, we should select $\Delta\omega$ for the binary FSK such that sinc $(\Delta\omega T_b)$ is minimum. Figure 10.11a shows sinc $(\Delta\omega T_b)$ as a function of $(\Delta\omega T_b)$. The minimum value of E_{pq} is $-0.217A^2T_b$ at $\Delta\omega \cdot T_b = 1.43\pi$ or when

$$\Delta f = \frac{\Delta\omega}{2\pi} = \frac{0.715}{T_b} = 0.715R_b$$

This leads to the minimum binary FSK BER

$$P_b = Q\left(\sqrt{\frac{1.217E_b}{\mathcal{N}}}\right) \tag{10.41a}$$

When $E_{pq} = 0$, we have the case of orthogonal signaling. From Fig. 10.11a, it is clear that $E_{pq} = 0$ for $\Delta f = n/2T_b$, where n is any integer. Although it appears that binary FSK can use any integer n when selecting Δf, larger Δf means wider separation between signaling frequencies $\omega_c - (\Delta\omega/2)$ and $\omega_c + (\Delta\omega/2)$, and consequently larger transmission bandwidth. To minimize the bandwidth, Δf should be as small as possible. Based on Fig. 10.11a, the minimum value of Δf that can be used for orthogonal signaling is $1/2T_b$. FSK using this value of Δf is known as **minimum shift keying (MSK)**.

Minimum Shift Keying

In MSK, not only are the two frequencies selected to be separated by $1/2T_b$, but we should also take care to preserve phase continuity when switching between $f \pm \Delta f$ at the transmitter.

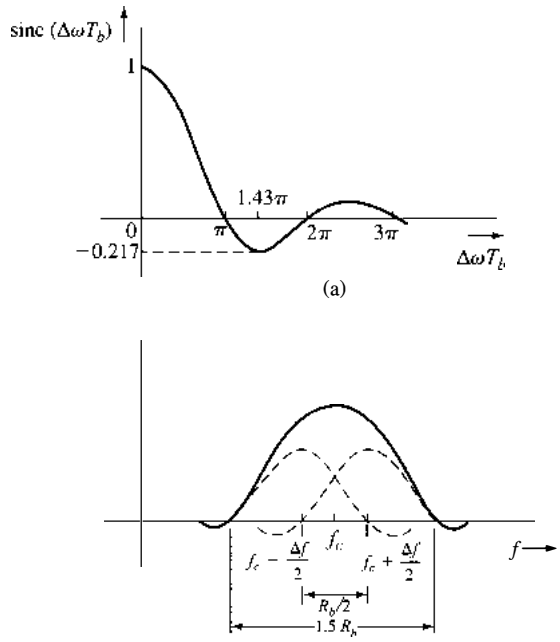


Figure 10.11 (a) The minimum of the sinc function and (b) the MSK spectrum.

This is because abrupt phase changes at the bit transition instants when we are switching frequencies would significantly increase the signal bandwidth. FSK schemes maintaining phase continuity are known as continuous phase FSK (CPFSK), of which MSK is one special case. These schemes have rapid spectral roll-off and better spectral efficiency.

To maintain phase continuity in CPFSK (or MSK), the phase at every bit transition is made to depend on the past data sequence. Consider, for example, the data sequence **1001**... starting at $t = 0$. The first pulse corresponding to the first bit **1** is $\cos[\omega_c + (\Delta\omega/2)]t$ over the interval 0 to T_b seconds. At $t = T_b$, this pulse ends with a phase $[\omega_c + (\Delta\omega/2)]T_b$. The next pulse, corresponding to the second data bit **0**, is $\cos[\omega_c - (\Delta\omega/2)]t$. To maintain phase continuity at the transition instant, this pulse is given additional phase $(\omega_c + \Delta\omega)T_b$. We achieve this continuity at each transition instant kT_b.

MSK being an orthogonal scheme, its error probability is given by

$$P_b = Q\left(\sqrt{\frac{E_b}{\mathcal{N}}}\right) \tag{10.41b}$$

Although this performance appears inferior to that of the optimum case in Eq. (10.41a), closer examination tells a different story. Indeed, this result is true **only if MSK is coherently detected as ordinary FSK** using an observation interval of T_b. However, recall that MSK is CPFSK, where the phase of each pulse is dependent on the past data sequence. Hence, better performance may be obtained by observing the received waveform **over a period longer than** T_b**.** Indeed, it can be shown that if an MSK signal is detected over an observation interval of $2T_b$, then the performance of MSK is identical to that of optimum PSK, that is,

$$P_b = Q\left(\sqrt{\frac{2E_b}{\mathcal{N}}}\right) \tag{10.41c}$$

MSK also has other useful properties. It has self-synchronization capabilities and its bandwidth is only $1.5R_b$, as shown in Fig. 10.11b. This is only 50% higher than for duobinary signaling. Moreover, the MSK spectrum decays much more rapidly as $1/f^4$, in contrast to the PSK (or bipolar) spectrum, which decays only as $1/f^2$ [see Eqs. (7.15) and (7.22)]. Because of these properties, MSK has received a great deal of practical attention. For more discussions, see Refs. 1 and 2.

10.4 SIGNAL SPACE ANALYSIS OF OPTIMUM DETECTION

Thus far, our discussions on digital receiver optimization have been limited to the simple case of linear threshold detection for binary transmissions under Gaussian channel noise. Such receivers are constrained by their linear structure. To determine the truly optimum receivers, we need to answer the question: Given an M-ary transmission with channel noise $\mathrm{n}(t)$ and channel output

$$\mathrm{y}(t) = p_i(t) + \mathrm{n}(t) \qquad 0 \leq t \leq T_o \qquad i = 1, \ldots, M$$

what receiver is *optimum* that can lead to minimum error probability?

To answer this question, we shall analyze the problem of digital signal detection from a more fundamental point of view. Recognize that the channel output is a random process $\mathrm{y}(t)$, $0 \leq t \leq T_o$. Thus, the receiver must make a decision by transforming $\mathrm{y}(t)$ into a finite-dimensional decision space. Such an analysis is greatly facilitated by a geometrical representation of signals and noises.

A Word about Notation: Let us clarify the notations used here to avoid confusion. As before, we use roman type to denote an RV or a random process [e.g., x or $\mathrm{x}(t)$]. A particular value assumed by the RV in a certain trial is denoted by italic type. Thus, x represents the value assumed by x. Similarly, $x(t)$ represents a particular sample function of the random process $\mathrm{x}(t)$. For random vectors, we follow the same convention: a random vector is denoted by roman boldface type, and a particular value assumed by the vector in a certain trial is represented by boldface italic type. Thus, $\mathbf{r}$ denotes a random vector, but $\boldsymbol{r}$ is a particular value of $\mathbf{r}$.

10.4.1 Geometrical Signal Space

We now formally show that a signal in an M-ary transmission system is in reality an n-dimensional vector and can be represented by a point in an n-dimensional hyperspace ($n \leq M$). The foundations for such a viewpoint were first laid during the introduction of the signal space in Sec. 2.6.

To begin, an ordered n-tuple $(x_1, x_2, \ldots, x_n)$ is an n-dimensional vector $\boldsymbol{x}$. The n-dimensional (signal) vector space is spanned by n unit vectors $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_n$

$$\begin{aligned} \boldsymbol{\varphi}_1 &= (1, 0, 0, \ldots, 0) \\ \boldsymbol{\varphi}_2 &= (0, 1, 0, \ldots, 0) \\ &\cdots \\ \boldsymbol{\varphi}_n &= (0, 0, 0, \ldots, 1) \end{aligned} \tag{10.42}$$

Any vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ can be expressed as a linear combination of n unit vectors,

$$\boldsymbol{x} = x_1\boldsymbol{\varphi}_1 + x_2\boldsymbol{\varphi}_2 + \cdots + x_n\boldsymbol{\varphi}_n \tag{10.43a}$$

$$= \sum_{k=1}^{n} x_k\boldsymbol{\varphi}_k \tag{10.43b}$$

This vector space is characterized by the definitions of the inner product between two vectors

$$<\boldsymbol{x}, \boldsymbol{y}> = \sum_{k=1}^{n} x_k y_k \tag{10.44}$$

and the vector norm

$$||\boldsymbol{x}||^2 = <\boldsymbol{x}, \boldsymbol{x}> = \sum_{k=1}^{n} x_k^2 \tag{10.45}$$

The norm $||\boldsymbol{x}||$ is the **length** of a vector. Vectors **x** and $\boldsymbol{y}$ are said to be **orthogonal** if their inner product

$$<\boldsymbol{x}, \boldsymbol{y}> = 0 \tag{10.46}$$

A set of n-dimensional vectors is said to be independent if none of the vectors in the set can be represented as a linear combination of the remaining vectors in that set. Thus, if $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m$ is an independent set, then the equality

$$a_1\boldsymbol{y}_1 + a_2\boldsymbol{y}_2 + \cdots + a_m\boldsymbol{y}_m = 0 \tag{10.47}$$

would require that $a_i = 0, i = 1, \ldots, m$. A subset of vectors in a given n-dimensional space can have dimensionality less than n. For example, in a three-dimensional space, all vectors lying in one plane can be specified by two dimensions, and all vectors lying along a line can be specified by one dimension.

An n-dimensional space can have at most n independent vectors. If a space has a maximum of n independent vectors, then every vector $\boldsymbol{x}$ in this space can be expressed as a linear combination of these n independent vectors. Thus, any vector in this space can be specified by n-tuples. For this reason, a set of n independent vectors in an n-dimensional space can be viewed as its **basis vectors**.

The members of a set of basis vectors form coordinate axes, and they are not unique. The n unit vectors in Eq. (10.42) are independent and can serve as basis vectors. These vectors have an additional property in that they are (mutually) **orthogonal** and have **normalized** length, that is,

$$<\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_k> = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases} \tag{10.48}$$

Such a set is an **orthonormal** set of vectors. They capture an orthogonal vector space. Any vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ can be represented as

$$\boldsymbol{x} = x_1\boldsymbol{\varphi}_1 + x_2\boldsymbol{\varphi}_2 + \cdots + x_n\boldsymbol{\varphi}_n$$

where x_i is the projection of $\boldsymbol{x}$ on the basis vector $\boldsymbol{\varphi}_k$ and is the kth coordinate. By using Eq. (10.48), the kth coordinate can be obtained from

$$<\boldsymbol{x}, \boldsymbol{\varphi}_k> = x_k \qquad k = 1, 2, \ldots, n \tag{10.49}$$

Since any vector in the n-dimensional space can be represented by this set of n basis vectors, this set forms a **complete** orthonormal (CON) set.

10.4.2 Signal Space and Basis Signals

The concepts of vector space and basis vectors can be generalized to characterize continuous time signals defined over a time interval Θ. As described in Sec. 2.6, a set of orthonormal signals $\{\varphi_i(t)\}$ can be defined for $t \in \Theta$ if

$$\int_{t\in\Theta} \varphi_j(t)\varphi_k(t)\,dt = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases} \tag{10.50}$$

If $\{\varphi_i(t)\}$ form a complete set of orthonormal basis functions of a signal space defined over Θ, then every signal $x(t)$ in *this* signal space can be expressed as

$$x(t) = \sum_k x_k\varphi_k(t) \quad t \in \Theta \tag{10.51}$$

where the signal component in the direction of $\varphi_k(t)$ is*

$$x_k = \int_{t\in\Theta} x(t)\varphi_k(t)\,dt \tag{10.52}$$

One such example is for $\Theta = (-\infty, \infty)$. Based on sampling theorem, all low-pass signals with bandwidth B Hz can be represented by

$$x(t) = \sum_k x_k \underbrace{\sqrt{2B}\,\text{sinc}\,(2\pi Bt - k\pi)}_{\varphi_k(t)} \tag{10.53a}$$

---

* If $\{\varphi_k(t)\}$ is complex, orthogonality implies

$$\int_{t\in\Theta} \varphi_j(t)\varphi_k^*(t)\,dt = 0$$

and Eq. (10.52) becomes

$$x_k = \int_{t\in\Theta} x(t)\varphi_k^*(t)\,dt$$

with

$$x_k = \int_{-\infty}^{\infty} x(t)\sqrt{2B}\ \text{sinc}\ (2\pi Bt - k\pi)\, dt = \frac{1}{\sqrt{2B}} x\left(\frac{k}{2B}\right) \tag{10.53b}$$

Just as there are an infinite number of possible sets of basis vectors for a vector space, there are an infinite number of possible sets of basis signals for a given signal space. For a band-limited signal space, $\{\sqrt{2B} \cdot \text{sinc}\ (2\pi Bt - k\pi)\}$ is one possible set of basis signals.

Note that $x\ (k/2B)$ are the Nyquist rate samples of the original band-limited signal. Since a band-limited signal cannot be time-limited, the total number of Nyquist samples needed will be infinite. Samples at large k, however, can be ignored, because their contribution is negligible. A rigorous development of this result, as well as an estimation of the error in ignoring higher dimensions, can be found in Landau and Pollak.[3]

Scalar Product and Signal Energy

In a certain signal space, let $x(t)$ and $y(t)$ be two signals. If $\{\varphi_k(t)\}$ are the orthonormal basis signals, then

$$x(t) = \sum_i x_i \varphi_i(t)$$

$$y(t) = \sum_j y_j \varphi_j(t)$$

Hence,

$$<x(t),\ y(t)> = \int_{t\in\Theta} x(t)y(t)\, dt = \int_{t\in\Theta} \left[\sum_i x_i\varphi_i(t)\right]\left[\sum_j y_j\varphi_j(t)\right] dt$$

Because the basis signals are orthonormal, we have

$$\int_{t\in\Theta} x(t)y(t)\, dt = \sum_k x_k y_k \tag{10.54a}$$

The right-hand side of Eq. (10.54a), however, is by the inner product of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. Therefore, we again arrive at Parseval's theorem,

$$<x(t),\ y(t)> = \int_{t\in\Theta} x(t)y(t)\, dt = \sum_k x_k y_k = <\boldsymbol{x}, \boldsymbol{y}> \tag{10.54b}$$

The signal energy for a signal $x(t)$ is a special case. The energy E_x is given by

$$\begin{aligned} E_x &= \int_{t\in\Theta} x^2(t)\, dt \\ &= <\boldsymbol{x}, \boldsymbol{x}> = ||\boldsymbol{x}||^2 \end{aligned} \tag{10.55}$$

Hence, the signal energy is equal to the square of the length of the corresponding vector.

Example 10.1 A signal space consists of four signals $s_1(t)$, $s_2(t)$, $s_3(t)$, and $s_4(t)$, as shown in Fig. 10.12. Determine a suitable set of basis vectors and the dimensionality of the signals. Represent these signals geometrically in the vector space.

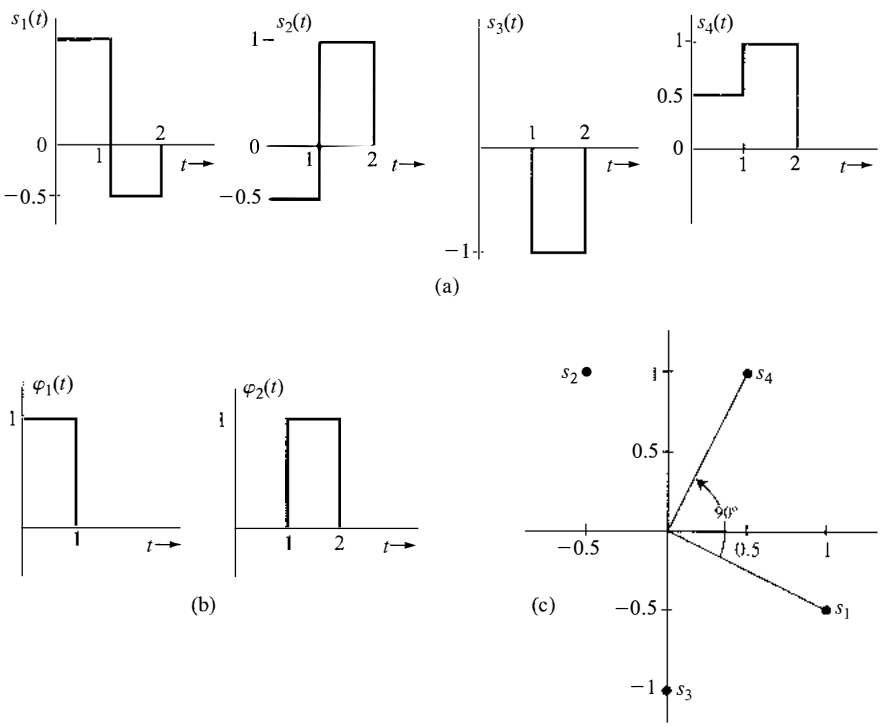


Figure 10.12 Signals and their representation in signal space.

The two rectangular pulses $\varphi_1(t)$ and $\varphi_2(t)$ in Fig. 10.12b are suitable as a basis signal set. In terms of this set, the vectors $\boldsymbol{s}_1$, $\boldsymbol{s}_2$, $\boldsymbol{s}_3$, and $\boldsymbol{s}_4$ corresponding to signals $s_1(t)$, $s_2(t)$, $s_3(t)$, and $s_4(t)$ are $\boldsymbol{s}_1 = (1, -0.5)$, $\boldsymbol{s}_2 = (-0.5, 1)$, $\boldsymbol{s}_3 = (0, -1)$, and $\boldsymbol{s}_4 = (0.5, 1)$. These points are plotted in Fig. 10.12c. Observe that the inner product between $\boldsymbol{s}_1$ and $\boldsymbol{s}_4$ is

$$<\boldsymbol{s}_1, \boldsymbol{s}_4> = 0.5 - 0.5 = 0$$

Hence, $\boldsymbol{s}_1$ and $\boldsymbol{s}_4$ are orthogonal. This result may be verified from the fact that

$$\int_{-\infty}^{\infty} s_1(t)s_4(t)\,dt = 0$$

Note that each point in the signal space in Fig. 10.12c corresponds to some waveform.

Determining an Orthonormal Basis Set

If there are a finite number of signals $x_i(t)$ in a given signal set of interest, then the orthonormal signal basis can either be selected heuristically or systematically. A heuristic approach requires a good understanding of the relationships among the different signals *as well as* a certain amount

of luck. On the other hand, **Gram-Schmidt orthogonalization** is a systematic approach to extract the basis signals from the known signal set. The details of this approach are given in Appendix C.

10.5 VECTOR DECOMPOSITION OF WHITE NOISE RANDOM PROCESSES

In digital communications, the message signal is always one of the M possible waveforms. It is therefore not difficult to represent all M waveforms via a set of CON basis functions. The real *challenge*, in fact, lies in the vector decomposition of the random noise $\mathrm{n}(t)$ at the receiver. A deterministic signal can be represented by one vector, a point in a signal space. Is it possible to represent a random process as a vector of random variables? If the answer is positive, then the detection problem can be significantly simplified.

Consider a complete orthonormal (CON) set of basis functions $\{\varphi_k(t)\}$ for a signal space defined over $[0, T_o]$. Then any deterministic signal $s(t)$ in this signal space will satisfy the following condition:

$$\int_0^{T_o} |s(t) - \sum_k s_k \varphi_k(t)|^2 dt = 0 \tag{10.56a}$$

This implies that for $t \in [0, T_o]$, we have the equality*

$$s(t) = \sum_k s_k \varphi_k(t)$$

However, for random processes defined over $[0, T_o]$, this statement is generally **not true**. Certain modifications are necessary.

10.5.1 Determining Basis Functions for a Random Process

First of all, a general random process $\mathrm{x}(t)$ cannot strictly satisfy Eq. (10.56a). Instead, a proper convergence requirement is in the mean square sense, that is,

$$E\left\{\int_0^{T_o} \left|\mathrm{x}(t) - \sum_k \mathrm{x}_k \varphi_k(t)\right|^2 dt\right\} = 0 \tag{10.56b}$$

This equality can be denoted as

$$\mathrm{x}(t) \overset{\text{m.s.}}{=} \sum_k \mathrm{x}_k \varphi_k(t) \tag{10.56c}$$

If $\mathrm{x}(t)$ and $\mathrm{y}(t)$ are equal in the mean square sense, then physically the difference between these two random processes have zero energy. As far as we are concerned in communications, signals (or signal differences) with zero energy have no physical effect and can be viewed as 0.

---

* Strictly speaking, this equality is true not for the entire interval $[0, T_o]$. The set of points for which equality does not hold is a measure zero set.

For a set of deterministic signals, the basis signals can be derived via the **Gram-Schmidt orthogonalization procedure.** However, Gram-Schmidt is invalid for random processes. Indeed, a random process $x(t)$ is an ensemble of signals. Thus, the basis signals $\{\varphi_k(t)\}$ must also depend on the characteristics of the random process.

The full and rigorous description of the decomposition of a random process can be found in some classic references.[4] Here, it suffices to state that the orthonormal basis functions must be solutions of the following integral equation

$$\lambda_i \cdot \varphi_i(t) = \int_0^{T_o} R_x(t, t_1) \cdot \varphi_i(t_1)\, dt_1 \qquad 0 \le t \le T_o \tag{10.57}$$

The solution Eq. (10.57) is known as the *Karhunen-Löeve* expansion. The auto-correlation function $R_x(t, t_1)$ is known as its kernel function. Indeed, Eq. (10.57) is reminiscent of the linear algebra equation with respect to eigenvalue λ and eigenvector $\boldsymbol{\phi}$:

$$\lambda \boldsymbol{\phi} = \boldsymbol{R}_x \boldsymbol{\phi}$$

in which $\boldsymbol{\phi}$ is a column vector and $\boldsymbol{R}_x$ is a positive semidefinite matrix; λ_i are known as the eigenvalues, whereas the basis functions $\varphi_i(t)$ are the corresponding eigenfunctions.

The *Karhunen-Löeve* expansion clearly establishes that the basis functions of a random process x(t) depend on its autocorrelation function $R_x(t, t_1)$. We cannot arbitrarily select a CON function set. In fact, solving the *Karhunen-Löeve* expansion can be a nontrivial task.

10.5.2 Geometrical Representation of White Noise Processes

For a stationary white noise process $x(t)$, the autocorrelation function is luckily

$$R_x(t, t_1) = \frac{\mathcal{N}}{2}\delta(t - t_1)$$

For this special *kernel*, the integral equation Eq. (10.57) is reduced to a simple form of

$$\lambda_i \cdot \varphi_i(t) = \int_0^{T_o} \frac{\mathcal{N}}{2}\delta(t - t_1) \cdot \varphi_i(t_1)\, dt_1 = \frac{\mathcal{N}}{2}\varphi_i(t) \qquad t \in (0, T_o) \tag{10.58}$$

This result implies that **any** CON set of basis functions can be used to represent **stationary white noise** processes. Additionally, the eigenvalues are identically $\lambda_i = \mathcal{N}/2$.

This particular result is of utmost importance to us. In most digital communication applications, we focus on the optimum receiver design and performance analysis under **white noise** channels. In the case of M-ary transmissions, we have an orthonormal set of basis functions $\{\varphi_k(t)\}$ to represent the M waveforms $\{s_i(t)\}$, such that

$$s_i(t) = \sum_k s_{i,k}\varphi_k(t) \qquad i = 1, \ldots, M \tag{10.59a}$$

Based on Eq. (10.58), these basis functions are **also** suitable for the representation of the white channel noise $n_w(t)$ such that

$$n_w(t) \overset{\text{m.s.}}{=} \sum_k n_k\varphi_k(t) \qquad 0 \le t \le T_o \tag{10.59b}$$

Consequently, when the transmitter sends $s_i(t)$, the received signal can be decomposed into

$$\begin{aligned} \mathrm{y}(t) &= s_i(t) + \mathrm{n}_w(t) \\ &\overset{\text{m.s.}}{=} \sum_k s_{i,k}\varphi_k(t) + \sum_k \mathrm{n}_k\varphi_k(t) \\ &\overset{\text{m.s.}}{=} \sum_k y_k\varphi_k(t) \end{aligned} \tag{10.59c}$$

by defining

$$\mathrm{y}_k = \int_0^{T_o} \mathrm{y}(t)\,\varphi_k^*(t)\,dt = s_{i,k} + \mathrm{n}_k \qquad \text{if } s_i(t) \text{ is sent} \tag{10.59d}$$

As a result, when the channel noise is white, the received channel output signal can be effectively represented by a sequence of random variables $\{\mathrm{y}_k\}$ of Eq. (10.59d). In other words, the optimum receiver for white noise channels can be derived from the information contained in

$$(\mathrm{y}_1, \mathrm{y}_2, \ldots, \mathrm{y}_k, \ldots).$$

We note that white noise $\mathrm{x}(t)$ consists of an ensemble of sample functions. The coefficients

$$\mathrm{x}_k = \int_0^{T_o} \mathrm{x}(t)\,\varphi_k^*(t)\,dt \qquad k = 1, 2, \ldots$$

in the decomposition of Eq. (10.59b) will be different for each sample function. Consequently, the coefficients are RVs. Each sample function will have a specific vector $(x_1, x_2, \ldots, x_n)$ and will map into one point in the signal space. This means that the ensemble of sample functions for the random process $\mathrm{x}(t)$ will map into an ensemble of points in the signal space, as shown in Fig. 10.13. Although this figure shows only a three-dimensional graph (because it is not possible to show a higher dimensional one), it is sufficient to indicate the idea.

For each trial of the experiment, the outcome (the sample function) is a certain point $\boldsymbol{x}$. The ensemble of points in the signal space appears as a dust ball, with the density of points directly proportional to the probability of observing $\mathbf{x}$ in that region. If we denote the joint PDF of $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n$ by $p_{\mathbf{x}}(\boldsymbol{x})$, then

$$p_{\mathbf{x}}(\boldsymbol{x}) = p_{\mathrm{x}_1\mathrm{x}_2\cdots\mathrm{x}_n}(x_1, x_2, \ldots, x_n) \tag{10.60}$$

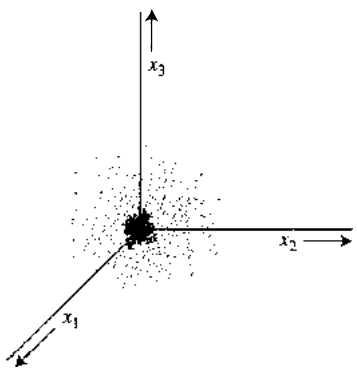


Figure 10.13 Geometrical representation of a Gaussian random process.

Thus, $p_{\mathbf{x}}(\boldsymbol{x})$ has a certain value at each point in the signal space, and $p_{\mathbf{x}}(\boldsymbol{x})$ represents the relative probability (dust density) of observing $\mathbf{x} = \boldsymbol{x}$.

10.5.3 White Gaussian Noise

If the channel noise $\mathrm{n}_w(t)$ is white and Gaussian, then from the discussions in Section 8.6, the expansion coefficients

$$\mathrm{n}_k = \int_0^{T_o} \mathrm{n}_w(t)\, \varphi_k(t)\, dt \tag{10.61}$$

are also Gaussian. Indeed, $(\mathrm{n}_1, \mathrm{n}_2, \ldots, \mathrm{n}_k, \ldots)$ are jointly Gaussian.

Here, we shall provide some fundamentals on Gaussian random variables. First, we define a column vector of n random variables as

$$\mathbf{x} = \begin{bmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \\ \vdots \\ \mathrm{x}_n \end{bmatrix}$$

Note that $\boldsymbol{x}^T$ denotes the transpose of $\boldsymbol{x}$. and $\bar{\mathbf{x}}$ denotes the mean of $\mathbf{x}$. Random variables (RVs) $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n$ are said to be jointly Gaussian if their joint PDF is given by

$$p_{\mathrm{x}_1\mathrm{x}_2\ldots\mathrm{x}_n}(x_1, x_2, \ldots, x_n) = \frac{1}{(2\pi)^{n/2}\sqrt{\det(K_{\mathrm{x}})}} \exp\left[-\frac{1}{2}(\boldsymbol{x} - \bar{\mathbf{x}})^T K_{\mathrm{x}}^{-1}(\boldsymbol{x} - \bar{\mathbf{x}})\right] \tag{10.62}$$

where K_{x} is the $n \times n$ covariance matrix

$$K_{\mathrm{x}} = \overline{(\mathbf{x} - \bar{\mathbf{x}}) \cdot (\mathbf{x} - \bar{\mathbf{x}})^T} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix} \tag{10.63a}$$

and the covariance of x_i and x_j is

$$\sigma_{ij} = \overline{(\mathrm{x}_i - \bar{\mathrm{x}}_i)(\mathrm{x}_j - \bar{\mathrm{x}}_j)} \tag{10.63b}$$

Here, we use conventional notations $\det(K_{\mathrm{x}})$ and K_{x}^{-1} to denote the determinant and the inverse of matrix K_{x}, respectively.

Gaussian variables are important not only because they are frequently observed, but also because they have certain properties that simplify many mathematical operations that are otherwise impossible or very difficult. We summarize these properties as follows:

P-1: The Gaussian density is completely specified by only the first- and second-order statistics $\bar{\mathbf{x}}$ and K_{x}. This follows from Eq. (10.62).

P-2: If n jointly Gaussian variables $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n$ are uncorrelated, then they are independent.

If the n variables are uncorrelated, $\sigma_{ij} = 0\,(i \neq j)$, and K_{x} reduces to a diagonal matrix. Thus, Eq. (10.62) becomes

$$p_{\mathrm{x}_1\mathrm{x}_2\cdots\mathrm{x}_n}(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left[\frac{-(x_i - \overline{\mathrm{x}_i})^2}{2\sigma_i^2}\right] \tag{10.64a}$$

$$= p_{\mathrm{x}_1}(x_1)p_{\mathrm{x}_2}(x_2)\ldots p_{\mathrm{x}_n}(x_n) \tag{10.64b}$$

As we observed earlier, independent variables are always uncorrelated, but uncorrelated variables are not necessarily independent. For the case of jointly Gaussian RVs, however, uncorrelatedness implies independence.

P-3: When $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n$ are jointly Gaussian, all the marginal densities, such as $p_{\mathrm{x}_i}(x_i)$, and all the conditional densities, such as $p_{\mathrm{x}_i\mathrm{x}_j|\mathrm{x}_k\mathrm{x}_l\cdots\mathrm{x}_p}(x_i, x_j|x_k, x_l, \ldots, x_p)$, are Gaussian. This property can be readily verified (Prob. 8.2-9).

P-4: Linear combinations of jointly Gaussian variables are also jointly Gaussian. Thus, if we form m variables $\mathrm{y}_1, \mathrm{y}_2, \ldots, \mathrm{y}_m$ $(m \leq n)$ obtained from

$$\mathrm{y}_i = \sum_{k=1}^{n} a_{ik}\mathrm{x}_k \tag{10.65}$$

then $\mathrm{y}_1, \mathrm{y}_2, \ldots, \mathrm{y}_m$ are also jointly Gaussian variables.

10.5.4 Properties of Gaussian Random Process

A random process $\mathrm{x}(t)$ is said to be Gaussian if the RVs $\mathrm{x}(t_1), \mathrm{x}(t_2), \ldots, \mathrm{x}(t_n)$ are jointly Gaussian [Eq. (10.62)] for every n and for every set $(t_1, t_2, \ldots, t_n)$. Hence, the joint PDF of RVs $\mathrm{x}(t_1), \mathrm{x}(t_2), \ldots, \mathrm{x}(t_n)$ of a Gaussian random process is given by Eq. (10.62) in which the mean and the covariance matrix K_{x} are specified by

$$\overline{\mathrm{x}(t_i)} \quad \text{and} \quad \sigma_{ij} = R_{\mathrm{x}}(t_i, t_j) - \overline{\mathrm{x}(t_i)} \cdot \overline{\mathrm{x}(t_j)} \tag{10.66}$$

This shows that a Gaussian random process is completely specified by its autocorrelation function $R_{\mathrm{x}}(t_i, t_j)$ and its mean value $\overline{\mathrm{x}(t)}$.

As discussed in Chapter 9, if the **Gaussian random process** satisfies two additional conditions:

$$R_{\mathrm{x}}(t_i, t_j) = R_{\mathrm{x}}(t_i - t_j) \tag{10.67a}$$

and

$$\overline{\mathrm{x}(t)} = \text{constant for all } t \tag{10.67b}$$

then it is a wide-sense stationary process. Moreover, Eqs. (10.67) also mean that the joint PDF of the Gaussian RVs $\mathrm{x}(t_1), \mathrm{x}(t_2), \ldots, \mathrm{x}(t_n)$ is also invariant to a shift of time origin. Hence,

we can conclude that *a wide-sense stationary Gaussian random process is also strict-sense stationary.*

Another significant property of the Gaussian process is that the response of a linear system to a Gaussian process is also a Gaussian process. This arises from property P-4 of the Gaussian RVs. Let $\mathrm{x}(t)$ be a Gaussian process applied to the input of a linear system whose unit impulse response is $h(t)$. If $\mathrm{y}(t)$ is the output (response) process, then

$$\begin{aligned}\mathrm{y}(t) &= \int_{-\infty}^{\infty} \mathrm{x}(t-\tau)h(\tau)\,d\tau \\ &= \lim_{\Delta\tau\to 0} \sum_{k=-\infty}^{\infty} \mathrm{x}(t-k\Delta\tau)h(k\Delta\tau)\,\Delta\tau\end{aligned}$$

is a weighted sum of Gaussian RVs. Because $\mathrm{x}(t)$ is a Gaussian process, all the variables $\mathrm{x}(t-k\Delta\tau)$ are jointly Gaussian (by definition). Hence, the variables $\mathrm{y}(t_1)$, $\mathrm{y}(t_2)$, ..., $\mathrm{y}(t_n)$ for all n and every set $(t_1, t_2, \ldots, t_n)$ are linear combinations of variables that are jointly Gaussian. Therefore, the variables $\mathrm{y}(t_1)$, $\mathrm{y}(t_2)$, ..., $\mathrm{y}(t_n)$ must be jointly Gaussian, according to the earlier discussion. It follows that the process $\mathrm{y}(t)$ is a Gaussian process.

To summarize, the Gaussian random process has the following properties:

1. A Gaussian random process is completely specified by its autocorrelation function and mean value.
2. If a Gaussian random process is wide-sense stationary, then it is stationary in the strict sense.
3. The response of a linear system to a Gaussian random process is also a Gaussian random process.

Consider a white noise process $\mathrm{n}_w(t)$ with PSD $\mathcal{N}/2$. Then any complete set of orthonormal basis signals $\varphi_1(t)$, $\varphi_2(t)$, ... can decompose $\mathrm{n}_w(t)$ into

$$\begin{aligned}\mathrm{n}_w(t) &= \mathrm{n}_1\varphi_1(t) + \mathrm{n}_2\varphi_2(t) + \cdots \\ &= \sum_k \mathrm{n}_k\varphi_k(t)\end{aligned}$$

White noise has infinite bandwidth. Consequently, the dimensionality of the signal space is infinity.

We shall now show that RVs n_1, n_2, ... are independent, with variance $\mathcal{N}/2$ each. First, we have

$$\begin{aligned}\overline{\mathrm{n}_j\mathrm{n}_k} &= \overline{\int_0^{T_o} \mathrm{n}_w(\alpha)\varphi_j(\alpha)\,d\alpha \int_0^{T_o} \mathrm{n}_w(\beta)\varphi_k(\beta)\,d\beta} \\ &= \int_0^{T_o}\int_0^{T_o} \overline{\mathrm{n}_w(\alpha)\mathrm{n}_w(\beta)}\varphi_j(\alpha)\varphi_k(\beta)\,d\alpha\,d\beta \\ &= \int_0^{T_o}\int_0^{T_o} R_{\mathrm{n}_w}(\beta-\alpha)\varphi_j(\alpha)\varphi_k(\beta)\,d\alpha\,d\beta\end{aligned}$$

Because $R_{n_w}(\tau) = (\mathcal{N}/2)\,\delta(\tau)$, then

$$\begin{aligned}\overline{n_j n_k} &= \int_0^{T_o}\int_0^{T_o} \frac{\mathcal{N}}{2}\delta(\beta-\alpha)\varphi_j(\alpha)\varphi_k(\beta)\,d\alpha\,d\beta \\ &= \frac{\mathcal{N}}{2}\int_0^{T_o}\varphi_j(\alpha)\varphi_k(\alpha)\,d\alpha \\ &= \begin{cases} 0 & j\neq k \\ \dfrac{\mathcal{N}}{2} & j=k \end{cases}\end{aligned} \tag{10.68}$$

Hence, n_j and n_k are uncorrelated Gaussian RVs, each with variance $\mathcal{N}/2$. Since they are Gaussian, uncorrelatedness implies independence. This proves the result.

For the time being, assume that we are considering an N-dimensional case. The joint PDF of independent joint Gaussian RVs $n_1, n_2, \ldots, n_N$, each with zero mean and variance $\mathcal{N}/2$, is [see Eq. (10.64)]

$$p_{\mathbf{n}}(\boldsymbol{n}) = \prod_{j=1}^{N}\frac{1}{\sqrt{2\pi\mathcal{N}/2}}e^{-n_j^2/2(\mathcal{N}/2)}$$

$$= \frac{1}{(\pi\mathcal{N})^{N/2}}e^{-(n_1^2+n_2^2+\cdots+n_N^2)/\mathcal{N}} \tag{10.69a}$$

$$= \frac{1}{(\pi\mathcal{N})^{N/2}}e^{-\|\boldsymbol{n}\|^2/\mathcal{N}} \tag{10.69b}$$

This shows that the PDF $p_{\mathbf{n}}(\boldsymbol{n})$ depends only on the norm $\|\boldsymbol{n}\|$, which is the sampled length of the noise vector $\mathbf{n}$ in the hyperspace, and is therefore spherically symmetrical if plotted in the N-dimensional hyperspace.

10.6 OPTIMUM RECEIVER FOR WHITE GAUSSIAN NOISE CHANNELS

10.6.1 Geometric Representations

We shall now consider, from a more fundamental point of view, the problem of M-ary communication in the presence of additive white Gaussian noise (AWGN). Such a channel is known as the AWGN channel. Unlike the linear receivers previously studied in Secs. 10.1 to 10.3, no constraint is placed on the optimum structure. We shall answer the fundamental question: What receiver will yield the minimum error probability?

The comprehension of the signal detection problem is greatly facilitated by geometrical representation of signals. In a signal space, we can represent a signal by a fixed point (or a vector). A random process can be represented by a random point (or a random vector). The region in which the random point may lie will be shown shaded, with the shading intensity proportional to the probability of observing the signal in that region. In the M-ary scheme, we use M symbols, or messages, $m_1, m_2, \ldots, m_M$. Each of these symbols is represented by a specified waveform. Let the corresponding waveforms be $s_1(t), s_2(t), \ldots, s_M(t)$. Thus, the symbol (or message) m_k is sent by transmitting the waveform $s_k(t)$. These waveforms are

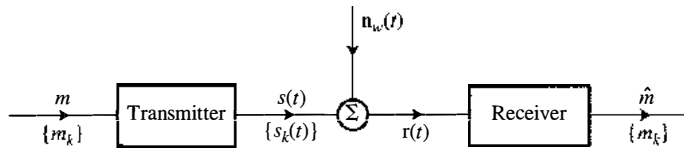


Figure 10.14 *M*-ary communication system.

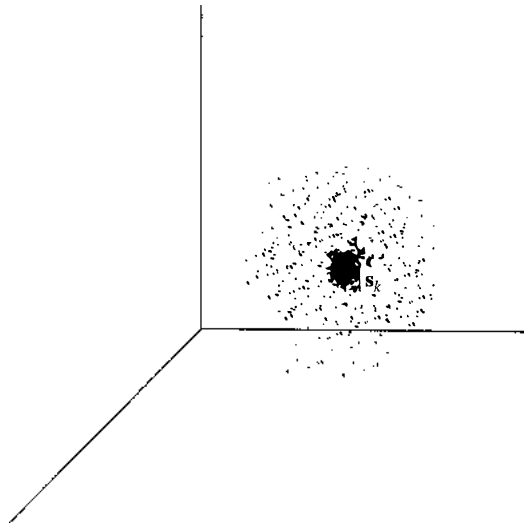


Figure 10.15 Effect of Gaussian channel noise on the received signal.

corrupted by AWGN $n_w(t)$ (Fig. 10.14) with PSD

$$S_{n_w}(\omega) = \frac{\mathcal{N}}{2}$$

At the receiver, the received signal $r(t)$ consists of one of the M message waveforms $s_k(t)$ plus the channel noise,

$$r(t) = s_k(t) + n_w(t) \tag{10.70a}$$

Because the noise $n_w(t)$ is white, we can use the same basis functions to decompose both $s_k(t)$ and $n_w(t)$. Thus, we can represent $r(t)$ in a signal space by denoting $\mathbf{r}$, $\boldsymbol{s}_k$, and $\mathbf{n}_w$ as the vectors representing signals $r(t)$, $s_k(t)$, and $n_w(t)$, respectively, Then it is evident that

$$\mathbf{r} = \boldsymbol{s}_k + \mathbf{n}_w \tag{10.70b}$$

The signal vector $\boldsymbol{s}_k$ is a fixed vector, because the waveform $s_k(t)$ is nonrandom, whereas the noise vector $\mathbf{n}_w$ is random. Hence, the vector $\mathbf{r}$ is also random. Because $n_w(t)$ is a Gaussian white noise, the probability distribution of $\mathbf{n}_w$ has spherical symmetry in the signal space (as shown in the last section). Hence, the distribution of $\mathbf{r}$ is a spherical distribution centered at a fixed point $\boldsymbol{s}_k$, as shown in Fig. 10.15. Whenever the message m_k is transmitted, the probability of observing the received signal $r(t)$ in a given scatter region is indicated by the intensity of the shading in Fig. 10.15. Actually, because the noise is white, the space has an infinite number of dimensions. For simplicity, however, we have shown the space to be three-dimensional. This will suffice to indicate our line of reasoning. We can draw similar scatter regions for various points $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_M$. Figure 10.16a shows the scatter regions for two messages m_j and m_k

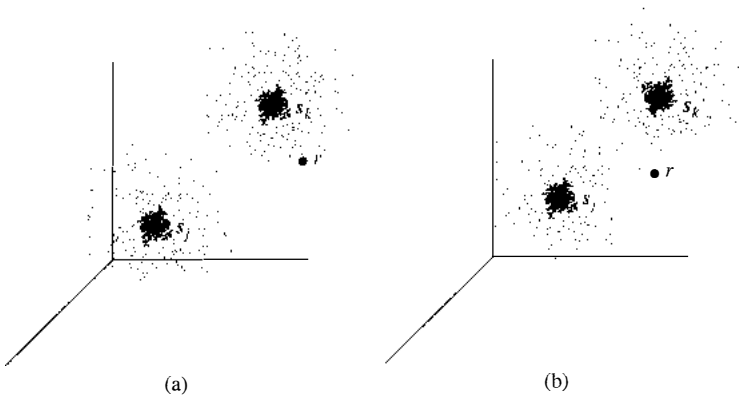


Figure 10.16 Binary communication in the presence of noise.

when s_j and s_k are widely separated in signal space. In this case, there is virtually no overlap between the two scattered regions. If either m_j or m_k is transmitted, the received signal will lie in one of the two scatter regions. From the position of the received signal, one can decide with a very small probability of error whether m_j or m_k was transmitted. In Fig. 10.16a, the received signal $\mathbf{r}$ is much closer to s_k than to s_j. It is therefore more likely that m_k was transmitted. Note that theoretically each scatter extends to infinity, although the probability of observing the received signal diminishes rapidly as a point is scattered away from the center. Hence, there will always be some overlap between the two scatter sets, resulting in a nonzero error probability. Thus, even though the received $\mathbf{r}$ is much closer to s_k in Fig. 10.16a, it may still be generated by s_j plus channel noise.

Figure 10.16b illustrates the case of stronger noise. In this case, there is a considerable overlap between the two scattered regions. Because the received signal $\boldsymbol{r}$ is closer to s_j than to s_k, it is more likely that m_j was transmitted. But in this case there is also a considerable probability that m_k may have been transmitted. Hence in this situation, there will be a much higher probability of error in any decision scheme.

The optimum receiver must decide, from a knowledge of $\boldsymbol{r}$, which message has been transmitted. The signal space must be divided into M nonoverlapping, or disjoint, decision regions $R_1, R_2, \ldots, R_M$, corresponding to the M messages $m_1, m_2, \ldots, m_M$. If $\boldsymbol{r}$ falls in the region R_k, the decision is m_k. The problem of designing the receiver then reduces to choosing the boundaries of these **decision regions** $R_1, R_2, \ldots, R_M$ to minimize the probability of error in decision making.

To recapitulate: A transmitter sends a sequence of messages from a set of M messages $m_1, m_2, \ldots, m_M$. These messages are represented by finite energy waveforms $s_1(t), s_2(t), \ldots, s_M(t)$. One waveform is transmitted every $T_o = T_M$ seconds. We assume that the receiver is time-synchronized with the transmitter. The waveforms are corrupted during transmissions by an AWGN of PSD $\mathcal{N}/2$. Knowing the received waveform, the receiver must decide which waveform was transmitted. The merit criterion of the receiver is the minimum probability of error in making this decision.

10.6.2 Dimensionality of the Detection Signal Space

Let us now discuss the dimensionality of the signal space in our detection problem. If there was no noise, we would be dealing with only M waveforms $s_1(t), s_2(t), \ldots, s_M(t)$. In this case a signal space of, at most, M dimensions would suffice. This is because the dimensionality

of a signal space is always equal to or less than the number of independent signals in the space (Sec. 10.4). For the sake of generality, we shall assume the space to have N dimensions ($N \leq M$). Let $\varphi_1(t)$, $\varphi_2(t)$, ..., $\varphi_N(t)$ be the orthonormal basis set for this space. Such a set can be constructed by using the Gram-Schmidt procedure discussed in Appendix C. We can then represent the signal waveform $s_k(t)$ as

$$s_j(t) = s_{j,1}\varphi_1(t) + s_{j,2}\varphi_2(t) + \cdots + s_{j,N}\varphi_N(t) \tag{10.71a}$$

$$= \sum_{k=1}^{N} s_{j,k}\varphi_k(t) \quad j = 1, 2, \ldots, M \tag{10.71b}$$

where

$$s_{j,k} = \int_{T_M} s_j(t)\varphi_k(t)\, dt \tag{10.71c}$$

Now consider the white Gaussian channel noise $\mathrm{n}_w(t)$. This signal has an infinite bandwidth ($B = \infty$). It has an infinite number of dimensions and obviously cannot be *fully* represented in a finite N-dimensional signal space discussed earlier. We can, however, split $\mathrm{n}_w(t)$ into two components: (1) the portion of $\mathrm{n}_w(t)$ inside the N-dimensional signal space, and (2) the remaining component orthogonal to the N-dimensional signal space. Let us denote the two components by $\mathrm{n}(t)$ and $\mathrm{n}_0(t)$. Thus,

$$\mathrm{n}_w(t) = \mathrm{n}(t) + \mathrm{n}_0(t) \tag{10.72}$$

where

$$\mathrm{n}(t) = \sum_{k=1}^{N} \mathrm{n}_j\varphi_j(t) \tag{10.73a}$$

and

$$\mathrm{n}_0(t) = \sum_{k=N+1}^{\infty} \mathrm{n}_j\varphi_j(t) \tag{10.73b}$$

where

$$\mathrm{n}_j = \int_{T_M} \mathrm{n}(t)\varphi_j(t)\, dt \tag{10.73c}$$

Because $\mathrm{n}_0(t)$ is orthogonal to the N-dimensional space, it is orthogonal to every signal in that space. Hence,

$$\int_{T_M} \mathrm{n}_0(t)\varphi_j(t)\, dt = 0 \qquad j = 1, 2, \ldots, N$$

Therefore,

$$\mathrm{n}_j = \int_{T_M} [\mathrm{n}(t) + \mathrm{n}_0(t)]\varphi_j(t)\, dt$$

$$= \int_{T_M} \mathrm{n}_w(t)\varphi_j(t)\, dt \quad j = 1, 2, \ldots, N \tag{10.74}$$

From Eqs. (10.73a) and (10.74), it is evident that we can filter out the component $n_0(t)$ from $n_w(t)$. This can be seen from the fact that the received signal, $r(t)$, can be expressed as

$$\begin{aligned} r(t) &= s_k(t) + n_w(t) \\ &= s_k(t) + n(t) + n_0(t) \\ &= q(t) + n_0(t) \end{aligned} \tag{10.75}$$

where $q(t)$ is the projection of $r(t)$ on the N-dimensional space:

$$q(t) = s_k(t) + n(t) \tag{10.76}$$

We can obtain the projection $q(t)$ from $r(t)$ by observing that [see Eqs. (10.71b) and (10.73a)]

$$q(t) = \sum_{j=1}^{N} (s_{kj} + n_j)\varphi_j(t) \tag{10.77}$$

From Eqs. (10.71c), (10.74), and (10.77) it follows that if we feed the received signal $r(t)$ into the system shown in Fig. 10.17, the resultant outcome will be $q(t)$. Thus, the orthogonal noise component can be filtered out without disturbing the message signal.

The question here is: Would such filtering help in our decision making? We can easily show that it cannot hurt us. The noise $n_w(t)$ is independent of the signal waveform $s_k(t)$. Therefore, its component $n_0(t)$ is also independent of $s_k(t)$. Thus, $n_0(t)$ contains no information about the transmitted signal, and discarding such a component from the received signal $r(t)$ will not cause any loss of information regarding the signal waveform $s_k(t)$. This, however, is not enough. We must also make sure that the noise being discarded $[n_0(t)]$ is not in any way related to the remaining noise component $n(t)$. If $n_0(t)$ and $n(t)$ are related in any way, it will be possible to obtain some information about $n(t)$ from $n_0(t)$, thereby enabling us to detect that signal with less error probability. If the components $n_0(t)$ and $n(t)$ are independent random processes, the component $n_0(t)$ does not carry any information about $n(t)$ and can be discarded. Under these conditions, $n_0(t)$ is **irrelevant** to the decision making at the receiver.

The process $n(t)$ is represented by components $n_1, n_2, \ldots, n_N$ along $\varphi_1(t), \varphi_2(t), \ldots, \varphi_N(t)$, and $n_0(t)$ is represented by the remaining components (infinite number) along the remaining basis signals in the complete set, $\{\varphi_k(t)\}$. Because the channel noise is white Gaussian, from Eq. (10.68) we observe that all the components are independent. Hence,

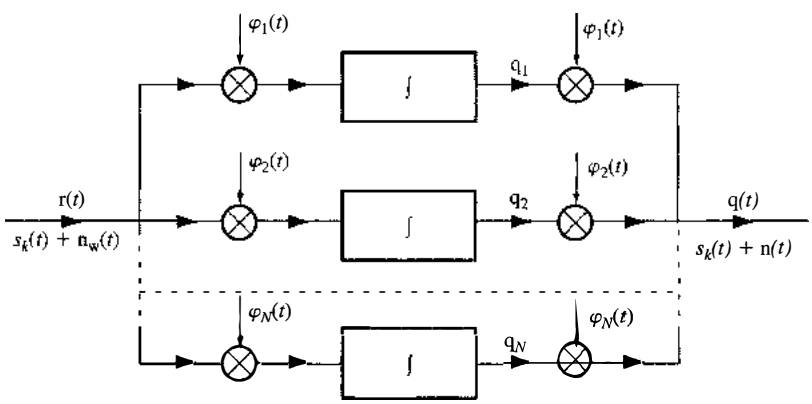


Figure 10.17 Eliminating the noise orthogonal to signal space.

the components representing $n_0(t)$ are independent of the components representing $n(t)$. Consequently, $n_0(t)$ is independent of $n(t)$ and contains only irrelevant data.

The received signal $r(t)$ is now reduced to the signal $q(t)$, which contains the desired signal waveform and the projection of the channel noise on the N-dimensional signal space. Thus, the signal $q(t)$ can be completely represented in the signal space. Let the vectors representing $n(t)$ and $q(t)$ be denoted by $\mathbf{n}$ and $\mathbf{q}$. Thus,

$$\mathbf{q} = \boldsymbol{s} + \mathbf{n}$$

where $\boldsymbol{s}$ may be any one of vectors $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_M$.

The random vector $\mathbf{n} = (n_1, n_2, \ldots, n_N)$ is represented by N independent Gaussian variables, each with zero mean and variance $\sigma_n^2 = \mathcal{N}/2$. The joint PDF of vector $\mathbf{n}$ in such a case has a spherical symmetry, as shown in Eq. (10.69b),

$$p_{\mathbf{n}}(\boldsymbol{n}) = \frac{1}{(\pi\mathcal{N})^{N/2}} e^{-|\boldsymbol{n}|^2/\mathcal{N}} \tag{10.78a}$$

Note that this is actually a compact notation for

$$p_{n_1,n_2,\ldots,n_N}(n_1, n_2, \ldots, n_N) = \frac{1}{(\pi\mathcal{N})^{N/2}} e^{-(n_1^2+n_2^2+\cdots+n_N^2)/\mathcal{N}} \tag{10.78b}$$

10.6.3 (Simplified) Signal Space and Decision Procedure

Our problem is now considerably simplified. The irrelevant noise component has been filtered out. The residual signal $q(t)$ can be represented in an N-dimensional signal space. We proceed to determine the M decision regions $R_1, R_2, \ldots, R_M$ in this space. The regions must be chosen to minimize the probability of error in making the decision.

Suppose the received vector $\mathbf{q} = \boldsymbol{q}$. Then if the receiver decides $\hat{m} = m_k$, the conditional probability of making the correct decision, given that $\mathbf{q} = \boldsymbol{q}$, is

$$P(C|\mathbf{q} = \boldsymbol{q}) = P(m_k|\mathbf{q} = \boldsymbol{q}) \tag{10.79}$$

where $P(C|\mathbf{q} = \boldsymbol{q})$ is the conditional probability of making the correct decision given $\mathbf{q} = \boldsymbol{q}$, and $P(m_k|\mathbf{q} = \boldsymbol{q})$ is the conditional probability that m_k was transmitted given $\mathbf{q} = \boldsymbol{q}$. The unconditional probability $P(C)$ is given by

$$P(C) = \int_{\boldsymbol{q}} P(C|\mathbf{q} = \boldsymbol{q}) p_{\mathbf{q}}(\boldsymbol{q})\, d\boldsymbol{q} \tag{10.80}$$

where the integration is performed over the entire region occupied by $\mathbf{q}$. Note that this is an N-fold integration with respect to the variables $q_1, q_2, \ldots, q_N$ over the signal waveform duration. Also, because $p_{\mathbf{q}}(\boldsymbol{q}) \geq 0$, this integral is maximum when $P(C|\mathbf{q} = \boldsymbol{q})$ is maximum. From Eq. (10.79) it now follows that if a decision $\hat{m} = m_k$ is made, the error probability is minimized if the probability

$$P(C) = \int_{\boldsymbol{q}} P(C|\mathbf{q} = \boldsymbol{q}) p_{\mathbf{q}}(\boldsymbol{q})\, d\boldsymbol{q}$$

is maximized. The probability $P(m_k|\mathbf{q} = \boldsymbol{q})$ is called the **a posteriori probability** of m_k. This is because it represents the probability that m_k was transmitted when $\boldsymbol{q}$ was being received.

The decision procedure to maximizing the probability of correct decision $P(C)$, thereby minimizing the probability of error, is now clear. Once we receive $\mathbf{q} = \boldsymbol{q}$, we evaluate all M a posteriori probability functions $\{P(m_j|\mathbf{q} = \boldsymbol{q})\}$. Then we make the decision in favor of that message for which the a posteriori probability is highest—that is, the receiver decides that $\hat{m} = m_k$ if

$$P(m_k|\mathbf{q} = \boldsymbol{q}) > P(m_j|\mathbf{q} = \boldsymbol{q}) \qquad \text{for all } j \neq k \tag{10.81}$$

Thus, the detector that minimizes the error probability is the **maximum a posteriori probability (MAP) detector.**

We can use Bayes' rule (Chapter 8) to determine the a posteriori probabilities. We have

$$P(m_k|\mathbf{q} = \boldsymbol{q}) = \frac{P(m_k)p_{\mathbf{q}}(\boldsymbol{q}|m_k)}{p_{\mathbf{q}}(\boldsymbol{q})} \tag{10.82}$$

Hence, the receiver decides $\hat{m} = m_k$ if the decision function

$$\frac{P(m_i)p_{\mathbf{q}}(\boldsymbol{q}|m_i)}{p_{\mathbf{q}}(\boldsymbol{q})} \qquad i = 1, 2, \ldots, M$$

is maximum for $i = k$.

Note that the denominator $p_{\mathbf{q}}(\boldsymbol{q})$ is common to all decision functions and is not effected by the decision. Hence, it may be ignored during the decision. Thus, the receiver sets $\hat{m} = m_k$ if the decision function

$$P(m_i)p_{\mathbf{q}}(\boldsymbol{q}|m_i) \qquad i = 1, 2, \ldots, M \tag{10.83}$$

is maximum for $i = k$. Thus, once $\boldsymbol{q}$ is obtained, we compute the decision function [Eq. (10.83)] for all messages $m_1, m_2, \ldots, m_M$ and decide that the message for which the function is maximum is the one most likely to have been sent.

We now turn our attention to finding the decision functions. The a priori probability $P(m_i)$ represents the probability that the message m_i will be transmitted. These probabilities must be known if the criterion discussed is to be used.* The term $p_{\mathbf{q}}(\boldsymbol{q}|m_i)$ represents the PDF of $\mathbf{q}$ when the transmitter sends $s(t) = s_i(t)$. Under this condition,

$$\mathbf{q} = \boldsymbol{s}_i + \mathbf{n}$$

and

$$\mathbf{n} = \mathbf{q} - \boldsymbol{s}_i$$

The point $\boldsymbol{s}_i$ is constant, and $\mathbf{n}$ is a random point. Obviously, $\mathbf{q}$ is a random point with the same distribution as $\mathbf{n}$ but centered at the points $\boldsymbol{s}_i$.

Alternatively, the probability density at $\mathbf{q} = \boldsymbol{q}$ (given $\mathrm{m} = m_i$) is the same as the probability $\mathbf{n} = \boldsymbol{q} - \boldsymbol{s}_i$. Hence [Eq. (10.78a)],

$$p_{\mathbf{q}}(\boldsymbol{q}|m_i) = p_{\mathbf{n}}(\boldsymbol{q} - \boldsymbol{s}_i) = \frac{1}{(\pi\mathcal{N})^{N/2}} e^{-\|\boldsymbol{q}-\boldsymbol{s}_i\|^2/\mathcal{N}} \tag{10.84}$$

* In case these probabilities are unknown, one must use other merit criteria, such as maximum likelihood or minimax, as will be discussed later.

The decision function in Eq. (10.83) now becomes

$$\frac{P(m_i)}{(\pi \mathcal{N})^{N/2}} e^{-||\boldsymbol{q}-\boldsymbol{s}_i||^2/\mathcal{N}} \tag{10.85}$$

Note that the decision function is always nonnegative for all values of i. Hence, comparing these functions is equivalent to comparing their logarithms, because the logarithm is a monotone function for the positive argument. Hence, for convenience, the decision function will be chosen as the logarithm of Eq. (10.85). In addition, the factor $(\pi \mathcal{N})^{N/2}$ is common for all i and can be left out. Hence, the decision function to maximize is

$$\ln P(m_i) - \frac{1}{\mathcal{N}} ||\boldsymbol{q} - \boldsymbol{s}_i||^2 \tag{10.86}$$

Note that $||\boldsymbol{q} - \boldsymbol{s}_i||^2$ is the square of the length of the vector $\boldsymbol{q} - \boldsymbol{s}_i$. Hence,

$$\begin{aligned} ||\boldsymbol{q} - \boldsymbol{s}_i||^2 &= <\boldsymbol{q} - \boldsymbol{s}_i, \boldsymbol{q} - \boldsymbol{s}_i> \\ &= ||\boldsymbol{q}||^2 + ||\boldsymbol{s}_i||^2 - 2<\boldsymbol{q}, \boldsymbol{s}_i> \end{aligned} \tag{10.87}$$

Hence, the decision function in Eq. (10.86) becomes (after multiplying throughout by $\mathcal{N}/2$)

$$\frac{\mathcal{N}}{2} \ln P(m_i) - \frac{1}{2} \left(||\boldsymbol{q}||^2 + ||\boldsymbol{s}_i||^2 - 2<\boldsymbol{q}, \boldsymbol{s}_i> \right) \tag{10.88}$$

Note that the term $||\boldsymbol{s}_i||^2$ is the square of the length of $\boldsymbol{s}_i$ and represents E_i, the energy of signal $s_i(t)$. The terms $\mathcal{N} \ln P(m_i)$ and E_i are constants in the decision function. Let

$$a_i = \tfrac{1}{2}[\mathcal{N} \ln P(m_i) - E_i] \tag{10.89}$$

Now the decision function in Eq. (10.88) becomes

$$a_i + <\boldsymbol{q}, \boldsymbol{s}_i> - \frac{||\boldsymbol{q}||^2}{2}$$

The term $||\boldsymbol{q}||^2/2$ is common to all M decision functions and can be omitted for the purpose of comparison. Thus, the new decision function b_i is

$$b_i = a_i + <\boldsymbol{q}, \boldsymbol{s}_i> \tag{10.90}$$

We compute this function b_i for $i = 1, 2, \ldots, N$, and the receiver decides that $\hat{m} = m_k$ if this function is the largest for $i = k$. If the signal $q(t)$ is applied at the input terminals of a system whose impulse response is $h(t)$, the output at $t = T_M$ is given by

$$\int_{-\infty}^{\infty} q(\tau) h(T_M - \tau)\, d\tau$$

If we choose a filter matched to $s_i(t)$, that is, $h(t) = s_i(T_M - t)$,

$$h(T_M - \tau) = s_i(\tau)$$

and based on Parseval's theorem, the output is

$$\int_{-\infty}^{\infty} q(\tau)s_i(\tau)\,d\tau = <\boldsymbol{q}, \boldsymbol{s}_i>$$

Hence, $<\boldsymbol{q}, \boldsymbol{s}_i>$ is the output at $t = T_M$ of a filter matched to $s_i(t)$ when $q(t)$ is applied to its input.

Actually, we do not have $q(t)$. The incoming signal $r(t)$ is given by

$$\begin{aligned} r(t) &= s_i(t) + \mathrm{n}_w(t) \\ &= \underbrace{s_i(t) + \mathrm{n}(t)}_{q(t)} + \underbrace{\mathrm{n}_0(t)}_{\text{irrelevant}} \end{aligned}$$

where $\mathrm{n}_0(t)$ is the (irrelevant) component of $\mathrm{n}_w(t)$ orthogonal to the N-dimensional signal space. Because $\mathrm{n}_0(t)$ is orthogonal to this space, it is orthogonal to every signal in this space. Hence, it is orthogonal to the signal $s_i(t)$, and

$$\int_{-\infty}^{\infty} \mathrm{n}_0(t)s_i(t)\,dt = 0$$

and

$$\begin{aligned} <\boldsymbol{q}, \boldsymbol{s}_i> &= \int_{-\infty}^{\infty} q(t)s_i(t)\,dt + \int_{-\infty}^{\infty} \mathrm{n}_0(t)s_i(t)\,dt \\ &= \int_{-\infty}^{\infty} [q(t) + \mathrm{n}_0(t)]s_i(t)\,dt \\ &= \int_{-\infty}^{\infty} r(t)s_i(t)\,dt \end{aligned} \tag{10.91}$$

Hence, it is immaterial whether we use $q(t)$ or $r(t)$ at the input. We thus apply the incoming signal $r(t)$ to a parallel bank of matched filters, and the output of the filters is sampled at $t = T_M$. Then a constant a_i is added to the ith filter output sample, and the resulting outputs are compared. The decision is made in favor of the signal for which this output is the largest. The receiver implementation for this decision procedure is shown in Fig. 10.18a. Section 10.1 has already established that a matched filter is equivalent to a correlator. One may therefore use correlators instead of matched filters. Such an arrangement is shown in Fig. 10.18b.

We have shown that in the presence of AWGN, the matched filter receiver is the optimum receiver when the merit criterion is minimum error probability. Note that the optimum system is found to be linear, although it was not constrained to be so. Therefore, for white Gaussian noise, the optimum receiver happens to be linear. The matched filter obtained in Secs. 10.1 and 10.2, as well as the decision procedure are identical to those derived here.

The optimum receiver can be implemented in another way. From Eq. (10.91), we have

$$<\boldsymbol{q}, \boldsymbol{s}_i> = <\boldsymbol{r}, \boldsymbol{s}_i>$$

From Eq. (10.44), we can rewrite this as

$$<\boldsymbol{q}, \boldsymbol{s}_i> = \sum_{j=1}^{N} r_j s_{ij}$$

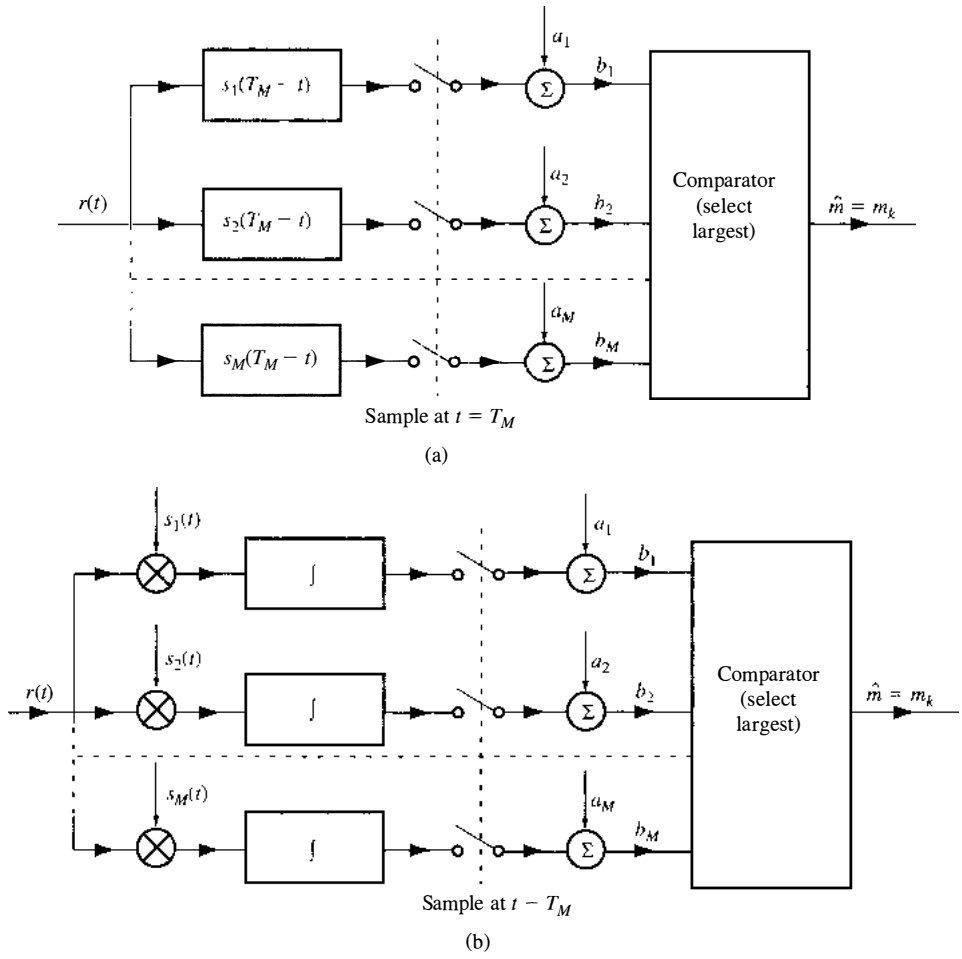
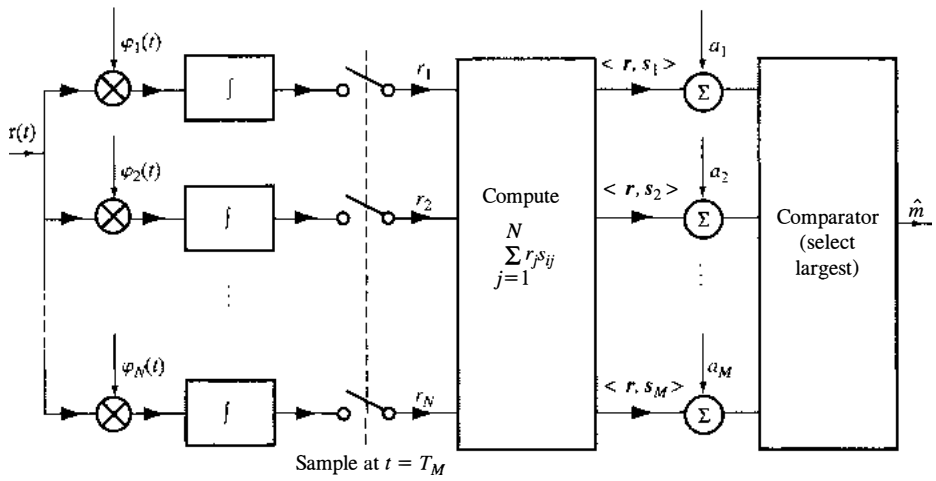


Figure 10.18
Optimum M-ary receiver:
(a) matched filter detector;
(b) correlation detector.

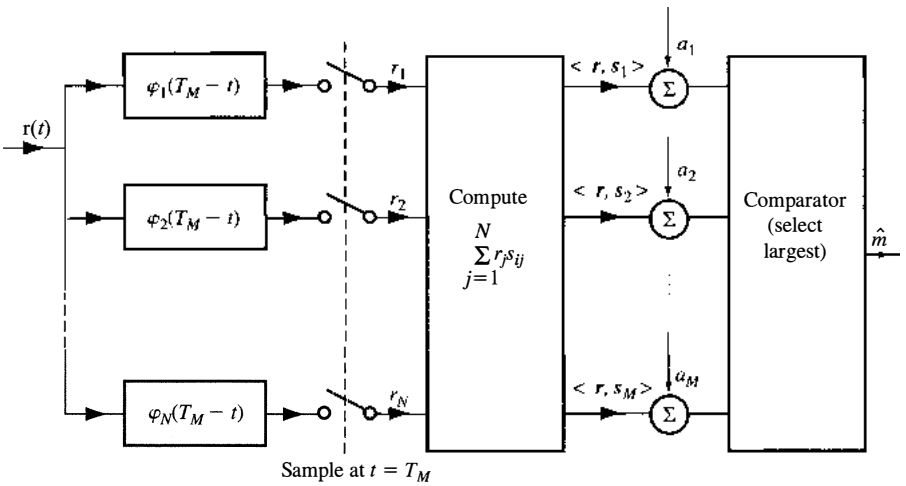
The term $<\boldsymbol{q}, \boldsymbol{s}_i>$ is computed according to this equation by first generating r_j and then computing the sum of $r_j s_{ij}$ (remember that the s_{ij} are known), as shown in Fig. 10.19a. The M correlator detectors in Fig. 10.18b can be replaced by N filters matched to $\varphi_1(t)$, $\varphi_2(t)$, ..., $\varphi_N(t)$, as shown in Fig. 10.19b. These types of optimum receiver (Figs. 10.18 and 10.19) perform identically. The choice will depend on the hardware cost. For example, if $N < M$ and signals $\{\varphi_j(t)\}$ are easier to generate than $\{s_j(t)\}$, then the design of Fig. 10.19 would be chosen.

10.6.4 Decision Regions and Error Probability

To compute the error probability of the optimum receiver, we must first determine decision regions in the signal space. As mentioned earlier, the signal space is divided into M nonoverlapping, or disjoint, decision regions R_1, R_2, ..., R_M, corresponding to M messages. If $\boldsymbol{q}$ falls in the region R_k, the decision is that m_k was transmitted. The decision regions are chosen to minimize the probability of error in the receiver. In light of this geometrical representation, we shall now try to interpret how the optimum receiver sets these decision regions.



(a)



(b)

Figure 10.19 Another form of optimum M-ary receiver: (a) correlator; (b) matched filter.

The decision function is given by Eq. (10.86). The optimum receiver sets $\hat{m} = m_k$ if the decision function

$$\mathcal{N} \ln P(m_i) - ||\boldsymbol{q} - \boldsymbol{s}_i||^2$$

is maximum for $i = k$. This equation defines the decision regions.

Geometric Interpretation in Signal Space

For simplicity, let us first consider the case of equiprobable messages, that is, $P(m_i) = 1/M$ for all i. In this case, the first term in the decision function is the same for all i and, hence, can be dropped. Thus, the receiver decides that $\hat{m} = m_k$ if the term $-||\boldsymbol{q} - \boldsymbol{s}_i||^2$ is largest (numerically the smallest) for $i = k$. Alternatively, this may be stated as follows: the receiver decides that $\hat{m} = m_k$ if the decision function $||\boldsymbol{q} - \boldsymbol{s}_i||^2$ is minimum for $i = k$. Note that $||\boldsymbol{q} - \boldsymbol{s}_i||$ is the distance of point $\boldsymbol{q}$ from point $\boldsymbol{s}_i$. Thus, the decision procedure in this case has a

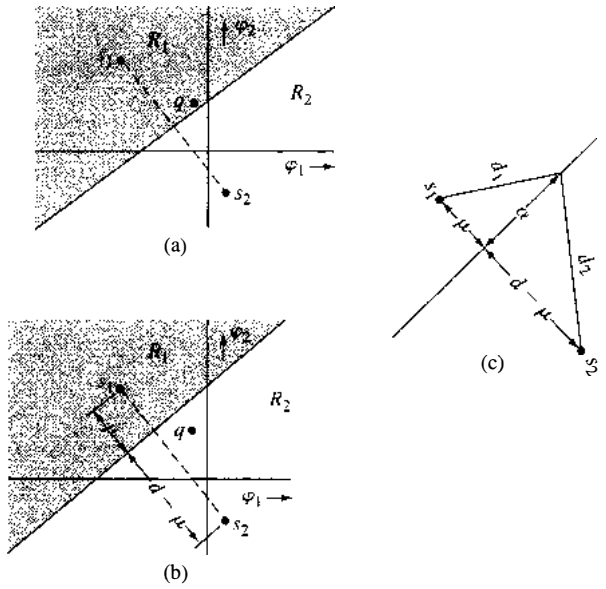


Figure 10.20 Determining optimum decision regions in a binary case.

simple interpretation in geometrical space. The decision is made in favor of that signal which is closest to $\boldsymbol{q}$, the projection of $\boldsymbol{r}$ [the component of $r(t)$] in the signal space.

This result is expected on qualitative grounds for Gaussian noise, because the Gaussian noise has a spherical symmetry. If, however, the messages are not equiprobable, we cannot go too far on purely qualitative grounds. Nevertheless, we can draw certain broad conclusions. If a particular message m_i is more likely than the others, one will be safer in deciding more often in favor of m_i than other messages. Hence, in such a case the decision regions will be biased, or weighted, in favor of m_i. This is shown by the appearance of the term $\ln P(m_i)$ in the decision function. To better understand this point, let us consider a two-dimensional signal space and two signals $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, as shown in Fig. 10.20a. In this figure, the decision regions R_1 and R_2 are shown for equiprobable messages; $P(m_1) = P(m_2) = 0.5$. The boundary of the decision region is the perpendicular bisector of the line joining points $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$. Note that any point on the boundary is equidistant from $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$. If $\boldsymbol{q}$ happens to fall on the boundary, we just "flip a coin" and decide whether to select m_1 or m_2. Figure 10.20b shows the case of two messages that are not equiprobable. To delineate the boundary of the decision regions, we use Eq. (10.86). The decision is m_1 if

$$||\boldsymbol{q} - \boldsymbol{s}_1||^2 - \mathcal{N} \ln P(m_1) < ||\boldsymbol{q} - \boldsymbol{s}_2||^2 - \mathcal{N} \ln P(m_2)$$

Otherwise, the decision is m_2.

Note that $||\boldsymbol{q} - \boldsymbol{s}_1||$ and $||\boldsymbol{q} - \boldsymbol{s}_2||$ represent distances d_1 and d_2, the distance of $\boldsymbol{q}$ from $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, respectively. Thus, the decision is m_1 if

$$d_1^2 - d_2^2 < \mathcal{N} \ln \frac{P(m_1)}{P(m_2)}$$

The right-hand side of this inequality is a constant c:

$$c = \mathcal{N} \ln \frac{P(m_1)}{P(m_2)}$$

Thus, the decision rule is

$$\text{Decision } (\boldsymbol{q}) = \begin{cases} m_1 & \text{if } d_1^2 - d_2^2 < c \\ m_2 & \text{if } d_1^2 - d_2^2 > c \\ \text{randomly } m_1 \text{ or } m_2 & \text{if } d_1^2 - d_2^2 = c \end{cases}$$

The boundary of the decision regions is given by $d_1^2 - d_2^2 = c$. We now show that such a boundary is given by a straight line perpendicular to line $\boldsymbol{s}_1$—$\boldsymbol{s}_2$ and passing through $\boldsymbol{s}_1$—$\boldsymbol{s}_2$ at a distance μ from $\boldsymbol{s}_1$, where

$$\mu = \frac{c + d^2}{2d} = \frac{\mathcal{N}}{2d} \ln \left[\frac{P(m_1)}{P(m_2)} \right] + \frac{d}{2} \tag{10.92}$$

where d is the distance between $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$. To prove this, we redraw the pertinent part of Fig. 10.20b as Fig. 10.20c, from which it is evident that

$$d_1^2 = \alpha^2 + \mu^2$$
$$d_2^2 = \alpha^2 + (d - \mu)^2$$

Hence,

$$d_1^2 - d_2^2 = 2d\mu - d^2 = c$$

Therefore,

$$\mu = \frac{c + d^2}{2d}$$

This is the desired result. Thus, along the decision boundary $d_1^2 - d_2^2$ is constant and equal to c. The boundaries of the decision regions for $M > 2$ may be determined via similar argument. The decision regions for the case of three equiprobable two-dimensional signals are shown in Fig. 10.21. The boundaries of the decision regions are perpendicular bisectors of the lines joining the original transmitted signals. If the signals are not equiprobable, then the boundaries will be shifted away from the signals with larger probabilities of occurrence.

For signals in N-dimensional space, the decision regions will be N-dimensional hypercones. If there are M messages $m_1, m_2, \ldots, m_M$ with decision regions $R_1, R_2, \ldots, R_M$, respectively, then $P(C|m_i)$, the probability of a correct decision when m_i is transmitted, is given by

$$P(C|m_i) = P(\boldsymbol{q} \text{ lies in } R_i) \tag{10.93}$$

and $P(C)$, the probability of a correct decision, is given by

$$P(C) = \sum_{i=1}^{M} P(m_i) P(C|m_i) \tag{10.94}$$

and P_{eM}, the probability of error, is given by

$$P_{eM} = 1 - P(C) \tag{10.95}$$

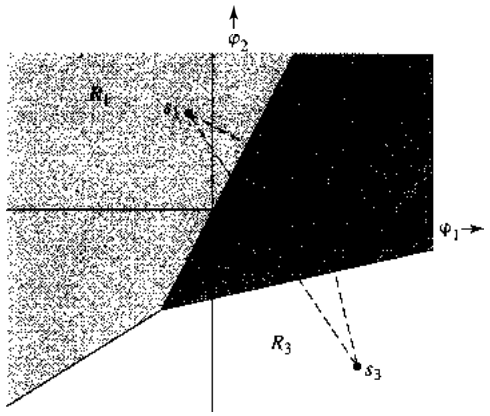


Figure 10.21 Determining optimum decision regions.

Example 10.2 Binary data is transmitted by using polar signaling over an AWGN channel with noise PSD $\mathcal{N}/2$. The two signals used are

$$s_1(t) = -s_2(t) = \sqrt{E}\varphi(t) \tag{10.96}$$

The symbol probabilities $P(m_1)$ and $P(m_2)$ are unequal. Design the optimum receiver and determine the corresponding error probability.

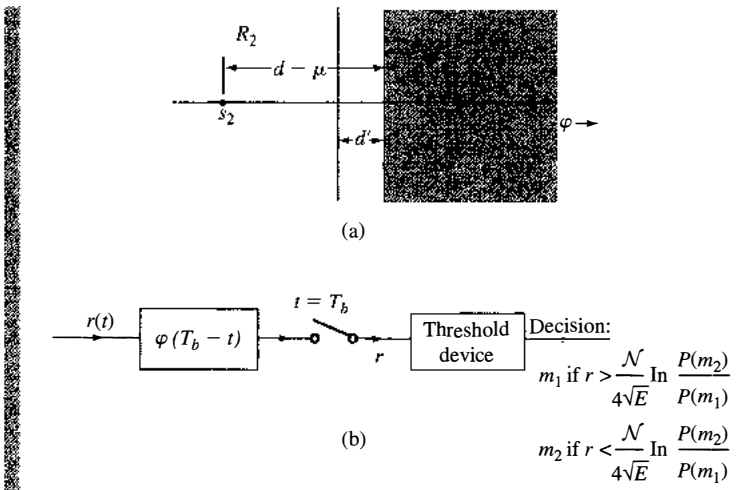


Figure 10.22 Decision regions for the binary case in this example.

The two signals are represented graphically in Fig. 10.22a. If the energy of each signal is E, the distance of each signal from the origin is $\sqrt{E}$. The distance d between the two signals is

$$d = 2\sqrt{E}$$

The decision regions R_1 and R_2 are shown in Fig. 10.22a. The distance μ is given by Eq. (10.92). Also, the conditional probability of correct decision

$$\begin{aligned} P(C|\mathrm{m} = m_1) &= P(\text{noise vector originating at } \boldsymbol{s}_1 \text{ remains in } R_1) \\ &= P(n > -\mu) \\ &= 1 - Q\left(\frac{\mu}{\sigma_n}\right) \\ &= 1 - Q\left(\frac{\mu}{\sqrt{\mathcal{N}/2}}\right) \end{aligned}$$

Similarly,

$$P(C|\mathrm{m} = m_2) = 1 - Q\left(\frac{d-\mu}{\sqrt{\mathcal{N}/2}}\right)$$

and the probability of correct decision is

$$\begin{aligned} P(C) &= P(m_1)\left[1 - Q\left(\frac{\mu}{\sqrt{\mathcal{N}/2}}\right)\right] + P(m_2)\left[1 - Q\left(\frac{d-\mu}{\sqrt{\mathcal{N}/2}}\right)\right] \\ &= 1 - P(m_1)Q\left(\frac{\mu}{\sqrt{\mathcal{N}/2}}\right) - P(m_2)Q\left(\frac{d-\mu}{\sqrt{\mathcal{N}/2}}\right) \end{aligned}$$

and

$$P_e = 1 - P(C) = P(m_1)Q\left(\frac{\mu}{\sqrt{\mathcal{N}/2}}\right) + P(m_2)Q\left(\frac{d-\mu}{\sqrt{\mathcal{N}/2}}\right) \tag{10.97a}$$

where

$$d = 2\sqrt{E} \tag{10.97b}$$

and

$$\mu = \frac{\mathcal{N}}{4\sqrt{E}} \ln \frac{P(m_1)}{P(m_2)} + \sqrt{E} \tag{10.97c}$$

When $P(m_1) = P(m_2) = 0.5$, $\mu = \sqrt{E} = d/2$, and Eq. (10.97a) reduces to

$$P_e = Q\left(\sqrt{\frac{2E}{\mathcal{N}}}\right) \tag{10.97d}$$

In this problem, because $N = 1$ and $M = 2$, the receiver in Fig. 10.19 is preferable to that in Fig. 10.18. For this case the receiver of the form in Fig. 10.19b reduces to that shown in Fig. 10.22b. The decision threshold d' as seen from Fig. 10.22a is

$$d' = \sqrt{E} - \mu = \frac{\mathcal{N}}{4\sqrt{E}} \ln \frac{P(m_2)}{P(m_1)}$$

Note that d' is the decision threshold. Thus, in Fig. 10.22b, if the receiver output $r > d'$, the decision is m_1. Otherwise the decision is m_2.

When $P(m_1) = P(m_2) = 0.5$, the decision threshold is zero. This is precisely the result derived in Sec. 10.1 for polar signaling.

10.6.5 Multiamplitude Signaling (PAM)

We now consider the M-ary generalization of the binary polar signaling, often known as pulse amplitude modulation (PAM). In the binary case, we transmit two symbols, consisting of the pulses $p(t)$ and $-p(t)$, where $p(t)$ may be either a baseband pulse or a carrier modulated by a baseband pulse. In the multiamplitude (PAM) case, the M symbols are transmitted by M pulses $\pm p(t)$, $\pm 3p(t)$, $\pm 5p(t)$, $\ldots$, $\pm(M-1)p(t)$. Thus, to transmit R_M M-ary digits per second, we are required to transmit R_M pulses per second of the form $kp(t)$. Pulses are transmitted every T_M seconds, so that $T_M = 1/R_M$. If E_p is the energy of pulse $p(t)$, then assuming that pulses $\pm p(t)$, $\pm 3p(t)$, $\pm 5p(t)$, $\ldots$, $\pm(M-1)p(t)$ are equally likely, the average pulse energy E_{pM} is given by

$$\begin{aligned} E_{pM} &= \frac{2}{M}[E_p + 9E_p + 25E_p + \cdots + (M-1)^2 E_p] \\ &= \frac{2E_p}{M}\sum_{k=0}^{\frac{M-2}{2}}(2k+1)^2 \\ &= \frac{M^2-1}{3}E_p \qquad (10.98a) \\ &\simeq \frac{M^2}{3}E_p \qquad M \gg 1 \qquad (10.98b) \end{aligned}$$

Recall that an M-ary symbol carries an information of $\log_2 M$ bits. Hence, the bit energy E_b is

$$E_b = \frac{E_{pM}}{\log_2 M} = \frac{M^2-1}{3\log_2 M}E_p \qquad (10.98c)$$

Because the transmission bandwidth is independent of the pulse amplitude, the M-ary bandwidth is the same as in the binary case for the given rate of pulses, yet it carries more information. This means that for a given information rate, the PAM bandwidth is less than that of the binary case by a factor of $\log_2 M$.

To calculate the error probability, we observe that because we are dealing with the same basic pulse $p(t)$, the optimum M-ary receiver is a filter matched to $p(t)$. When the input pulse is $kp(t)$, the output at the sampling instant will be

$$\mathrm{r}(T_M) = kA_p + \mathrm{n}_o(T_M)$$

Note that $A_p = E_p$, the energy of $p(t)$, and that σ_{n}^2, the variance of $\mathrm{n}_o(t)$, is $\mathcal{N}E_p/2$. Thus, the optimum receiver for the multiamplitude M-ary signaling case is identical to that of the polar

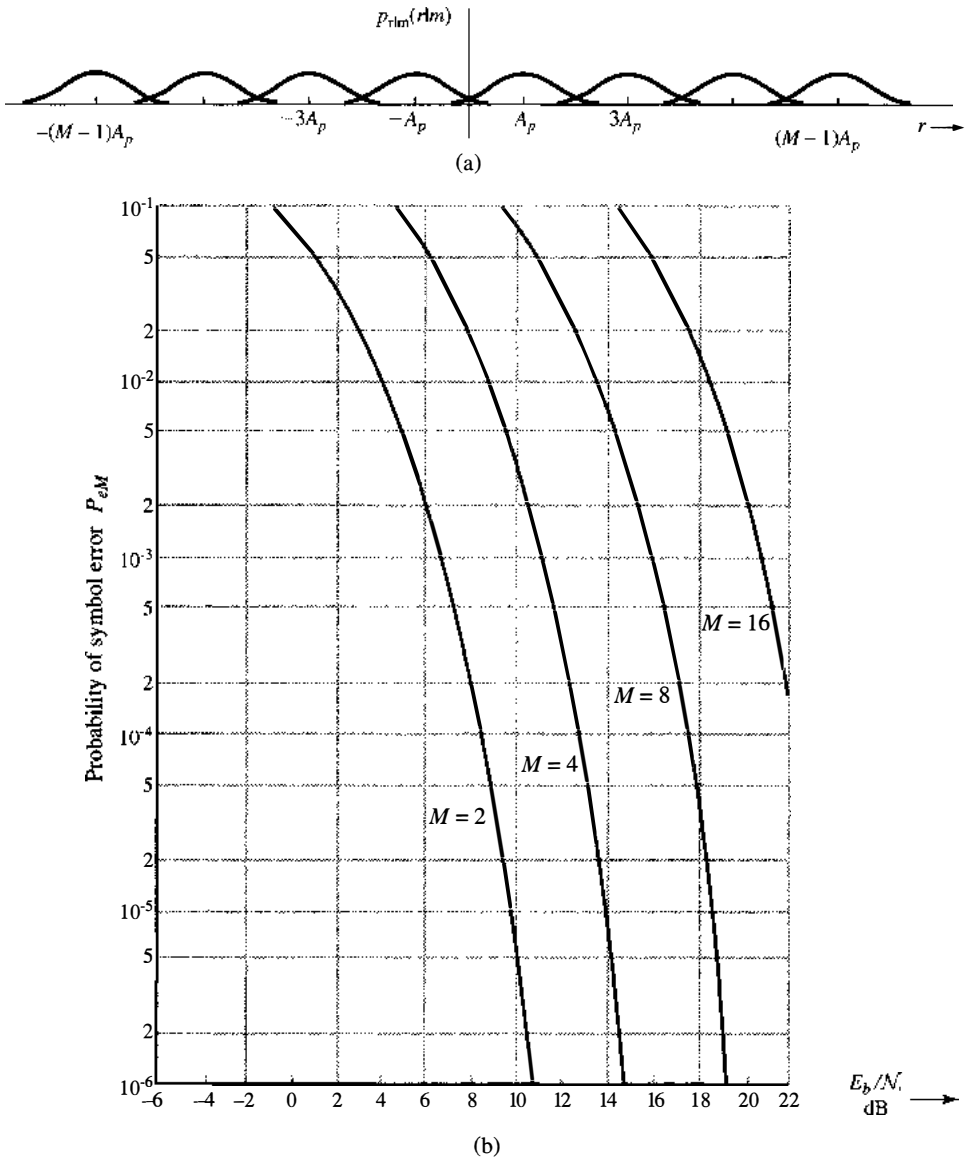


Figure 10.23
(a) Conditional PDFs in PAM. (b) Error probability in PAM.

binary case (see Fig. 10.3 or 10.6a). The sampler has M possible outputs

$$\pm kA_p + \mathrm{n}_o(T_M) \qquad k = 1, 3, 5, \ldots, M-1$$

that we wish to detect. The conditional PDFs $p(r|m_i)$ are Gaussian with mean $\pm kA_p$ and variance σ_{n}^2, as shown in Fig. 10.23a. Let P_{eM} be the error probability detecting a symbol and $P(\epsilon|m)$ be the error probability given that the symbol m is transmitted.

To calculate P_{eM}, we observe that the case of the two extreme symbols [represented by $\pm(M-1)p(t)$] is similar to the binary case because they have to guard against only one neighbor. As for the remaining symbols, they must guard against neighbors on both sides, and, hence, $P(\epsilon|m)$ in this case is twice that of the extreme symbol. From Fig. 10.23a it is evident that $P(\epsilon|m_i)$ is $Q(A_p/\sigma_{\mathrm{n}})$ for the two extreme signals and is $2Q(A_p/\sigma_{\mathrm{n}})$ for the remaining

$(M-2)$ symbols. Hence,

$$P_{eM} = \sum_{i=1}^{M} P(m_i)P(\epsilon|m_i) \tag{10.99a}$$

$$= \frac{1}{M}\sum_{i=1}^{M} P(\epsilon|m_i)$$

$$= \frac{1}{M}\left[Q\left(\frac{A_p}{\sigma_n}\right) + Q\left(\frac{A_p}{\sigma_n}\right) + (M-2)2Q\left(\frac{A_p}{\sigma_n}\right)\right]$$

$$= \frac{2(M-1)}{M}Q\left(\frac{A_p}{\sigma_n}\right) \tag{10.99b}$$

For a matched filter receiver, $(A_p/\sigma_n)^2 = 2E_p/\mathcal{N}$, and

$$P_{eM} = 2\left(\frac{M-1}{M}\right)Q\left(\sqrt{\frac{2E_p}{\mathcal{N}}}\right) \tag{10.99c}$$

$$= 2\left(\frac{M-1}{M}\right)Q\left[\sqrt{\frac{6\log_2 M}{M^2-1}\left(\frac{E_b}{\mathcal{N}}\right)}\right] \tag{10.99d}$$

Bit Error Rate (BER)

It is somewhat unfair to compare M-ary signaling on the basis of P_{eM}, the error probability of an M-ary symbol, which conveys the information of $k = \log_2 M$ bits. Because not all bits are wrong when an M-ary symbol is wrong, this weighs unfairly against larger M. For a fair comparison, we should compare various schemes in terms of their probability of bit error P_b, rather than P_{eM}, the probability of symbol error (symbol error rate). We now show that for multiamplitude signaling $P_b \approx P_{eM}/\log_2 M$.

Because the type of errors that predominate are those in which a symbol is mistaken for its immediate neighbors (see Fig. 10.23a), it would be logical to assign neighboring M-ary symbols, binary code words that differ in the least possible digits. The Gray code* is suitable for this purpose because adjacent binary combinations in this code differ only by one digit. Hence, an error in one M-ary symbol detection most likely will cause only one error in a group of $\log_2 M$ binary digits transmitted by the M-ary symbol. Hence, the bit error rate $P_b = P_{eM}/\log_2 M$. Figure 10.23b shows P_{eM} as a function of $E_b/\mathcal{N}$ for several values of M. Note that the relationship $P_b = P_{eM}/\log_2 M$, valid for PAM, is not necessarily valid for other schemes to the specific code structure. One must recompute the relationship between P_b and P_{eM} for each specific scheme.

---

* The Gray code can be constructed as follows. Construct an n-digit natural binary code (NBC) corresponding to 2^n decimal numbers. If $b_1 b_2 \ldots b_n$ is a code word in this code, then the corresponding Gray code word $g_1 g_2 \ldots g_n$ is obtained by the rule

$$g_1 = b_1$$

$$g_k = b_k \oplus b_{k-1} \qquad k \geq 2$$

Thus for $n = 3$, the binary code **000, 001, 010, 011, 100, 101, 110, 111** is transformed into the Gray code **000, 001, 011, 010, 110, 111, 101, 100**.

Trade-off between Power and Bandwidth

To maintain a given information rate, the pulse transmission rate in the M-ary case is reduced by the factor $k = \log_2 M$. This means the bandwidth of the M-ary case is reduced by the same factor $k = \log_2 M$. But to maintain the same P_{eM}, Eqs. (10.99) show that the power transmitted per bit (which is proportional to E_b) increases roughly as

$$M^2/\log_2 M = 2^{2k}/k$$

On the other hand, if we maintain a given bandwidth, the information rate in the M-ary case is increased by the factor $k = \log_2 M$. The transmitted power is equal to E_b times the bit rate. Hence, an increased data rate also necessitates increased power by the factor

$$(M^2/\log_2 M)(\log_2 M) = 2^{2k}$$

Thus, the power increases exponentially with the increase in information rate by a factor of k. In high-powered radio systems, such a power increase may not be tolerable. Multiamplitude systems are attractive when bandwidth is very costly. Thus we can see how to trade power for bandwidth. Because the voice channels of a telephone network have a fixed bandwidth, multiamplitude (or multiphase, or a combination of both) signaling is a more attractive method of increasing the information rate. This is how voiceband computer modems achieve high data rate.

All the results derived here apply to baseband as well as modulated digital systems with coherent detection. For noncoherent detection, similar relationships exist between the binary and M-ary systems.*

10.6.6 M-ary QAM Analysis

In M-ary QAM, the transmitted signal is represented by

$$s_i(t) = a_i \underbrace{\sqrt{\frac{2}{T_s}} \cos \omega_c t}_{\varphi_1(t)} + b_i \underbrace{\sqrt{\frac{2}{T_s}} \sin \omega_c t}_{\varphi_2(t)} \tag{10.100}$$

where

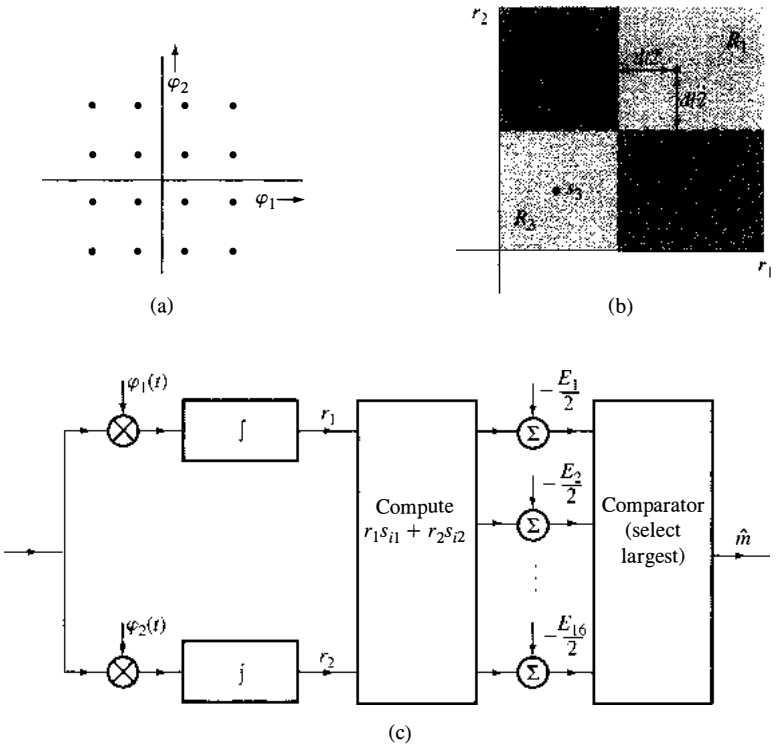
$$a_i = \pm\frac{d}{2}, \pm\frac{3d}{2}, \ldots \pm \frac{(\sqrt{M}-1)d}{2}$$

$$b_i = \pm\frac{d}{2}, \pm\frac{3d}{2}, \ldots \pm \frac{(\sqrt{M}-1)d}{2}$$

It is easy to observe that the QAM signal space is two-dimensional with basis functions $\varphi_1(t)$ and $\varphi_2(t)$. Instead of determining the optimum receiver and its error probability for an arbitrary QAM constellation, we illustrate the basic approach by analyzing the 16-point QAM configuration shown in Fig. 10.24a. We assume all signals to be equiprobable in an AWGN channel.

* For the noncoherent case, the baseband pulses must be of the same polarity; for example, $0, p(t), 2p(t), \ldots, (M-1)p(t)$.

Figure 10.24
16-ary QAM.



Let us first calculate the error probability. The first quadrant of the signal space is reproduced in Fig. 10.24b. Because all the signals are equiprobable, the decision region boundaries will be perpendicular bisectors joining various signals, as shown in Fig. 10.24b.

From Fig. 10.24b it follows that

$$\begin{aligned} P(C|m_1) &= P(\text{noise vector originating at } \boldsymbol{s}_1 \text{ lies within } R_1) \\ &= P\left(\mathrm{n}_1 > -\frac{d}{2},\ \mathrm{n}_2 > -\frac{d}{2}\right) \\ &= P\left(\mathrm{n}_1 > -\frac{d}{2}\right) P\left(\mathrm{n}_2 > -\frac{d}{2}\right) \\ &= \left[1 - Q\left(\frac{d/2}{\sigma_{\mathrm{n}}}\right)\right]^2 \\ &= \left[1 - Q\left(\frac{d}{\sqrt{2N}}\right)\right]^2 \end{aligned}$$

For convenience, let us define

$$p = 1 - Q\left(\frac{d}{\sqrt{2N}}\right) \tag{10.101}$$

Hence,

$$P(C|m_1) = p^2$$

Using similar arguments, we have

$$P(C|m_2) = P(C|m_4) = \left[1 - Q\left(\frac{d}{\sqrt{2N}}\right)\right]\left[1 - 2Q\left(\frac{d}{\sqrt{2N}}\right)\right]$$
$$= p(2p - 1)$$

and

$$P(C|m_3) = (2p - 1)^2$$

Because of the symmetry of the signals in all four quadrants, we get similar probabilities for the four signals in each quadrant. Hence, the probability of correct decision is

$$\begin{aligned} P(C) &= \sum_{i=1}^{16} P(C|m_i)P(m_i) \\ &= \frac{1}{16}\sum_{i=1}^{16} P(C|m_i) \\ &= \frac{1}{16}[4p^2 + 4p(2p-1) + 4p(2p-1) + 4(2p-1)^2] \\ &= \frac{1}{4}[9p^2 - 6p + 1] \\ &= \left(\frac{3p-1}{2}\right)^2 \end{aligned} \tag{10.102}$$

and

$$P_{eM} = 1 - P(C) = \frac{9}{4}\left(p + \frac{1}{3}\right)(1 - p)$$

In practice, $P_{eM} \to 0$ if SNR is high and, hence, $P(C) \to 1$. This means $p \simeq 1$ and $p + \frac{1}{3} \simeq 1\frac{1}{3}$ [see Eq. (10.102)], and

$$P_{eM} \simeq 3(1 - p) = 3Q\left(\frac{d}{\sqrt{2N}}\right) \tag{10.103}$$

To express this in terms of the received power S_i, we determine $\bar{E}$, the average energy of the signal set in Fig. 10.24. Because E_k, the energy of s_k, is the square of the distance of s_k from the origin,

$$E_1 = \left(\frac{3d}{2}\right)^2 + \left(\frac{3d}{2}\right)^2 = \frac{9}{2}d^2$$
$$E_2 = \left(\frac{3d}{2}\right)^2 + \left(\frac{d}{2}\right)^2 = \frac{5}{2}d^2$$

Similarly,

$$E_3 = \frac{d^2}{2} \qquad \text{and} \qquad E_4 = \frac{5}{2}d^2$$

Hence,

$$\bar{E} = \frac{1}{4}\left[\frac{9}{2}d^2 + \frac{5}{2}d^2 + \frac{d^2}{2} + \frac{5}{2}d^2\right] = \frac{5}{2}d^2$$

and $d^2 = 0.4\bar{E}$. Moreover, for $M = 16$, each symbol carries the information of $\log_2 16 = 4$ bits. Hence, the energy per bit E_b is

$$E_b = \frac{\bar{E}}{4}$$

and

$$\frac{E_b}{\mathcal{N}} = \frac{\bar{E}}{4\mathcal{N}} = \frac{5d^2}{8\mathcal{N}}$$

Hence, for large $E_b/\mathcal{N}$

$$\begin{aligned} P_{eM} &= 3Q\left(\frac{d}{\sqrt{2\mathcal{N}}}\right) \\ &= 3Q\left(\sqrt{\frac{4}{5}\frac{E_b}{\mathcal{N}}}\right) \end{aligned} \tag{10.104}$$

A comparison of this with binary PSK [Eq. 10.33)] shows that 16-point QAM requires almost 2.5 times as much power as does binary PSK; but the rate of transmission is increased by a factor of $\log_2 M = 4$. This comparison does not take into account the fact that P_b, the BER, is somewhat smaller than P_{eM}.

In terms of receiver implementation, because $N = 2$ and $M = 16$, the receiver in Fig. 10.19 is preferable. Such a receiver is shown in Fig. 10.24c. Note that because all signals are equiprobable,

$$a_i = -\frac{E_i}{2}$$

PSK is a special case of QAM with all signal points lying on a circle. Hence, the same analytical approach applies. However, the analysis may be more convenient if a polar coordinate is selected. We use the following example to illustrate the two different approaches.

Example 10.3 MPSK

Determine the error probability of the optimum receiver for equiprobable MPSK signals, each with energy E.

Figure 10.25a shows the MPSK signal configuration for $M = 8$. Because all the signals are equiprobable, the decision regions are conical, as shown. The message m_1 is transmitted by a signal $s_1(t)$ represented by the vector $\mathbf{s}_1 = (s_1, 0)$. If the projection in the signal space of the received signal $\mathbf{r}$ is $\mathbf{q} = (q_1, q_2)$, and the noise is $\mathbf{n} = (n_1, n_2)$, then

$$\mathbf{q} = (s_1 + \mathrm{n}_1,\ \mathrm{n}_2) = (\underbrace{\sqrt{E} + \mathrm{n}_1}_{q_1},\ \underbrace{\mathrm{n}_2}_{q_2})$$

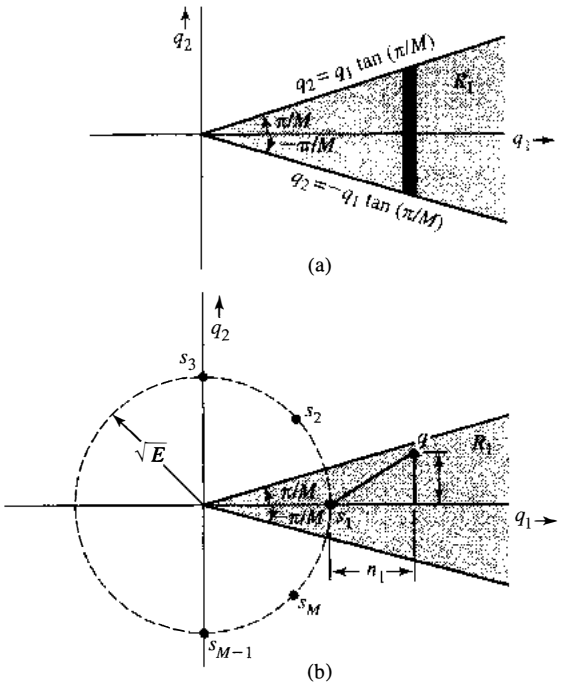


Figure 10.25 MPSK signals.

Also,

$$P(C|m_1) = P(\mathbf{q} \text{ lies in } R_1)$$

This is simply the volume under the conical region of the joint PDF of q_1 and q_2. Because n_1 and n_2 are independent Gaussian RVs with variance $\mathcal{N}/2$, q_1 and q_2 are independent Gaussian variables with means $\sqrt{E}$ and 0, respectively, and each with variance $\mathcal{N}/2$. Hence,

$$p_{q_1 q_2}(q_1, q_2) = \left[\frac{1}{\sqrt{\pi \mathcal{N}}} e^{-(q_1 - \sqrt{E})^2/\mathcal{N}}\right]\left[\frac{1}{\sqrt{\pi \mathcal{N}}} e^{-q_2^2/\mathcal{N}}\right]$$

and

$$P(C|m_1) = \frac{1}{\pi \mathcal{N}} \int_{R_1} e^{-[(q_1 - \sqrt{E})^2 + q_2^2]/\mathcal{N}} \, dq_1 \, dq_2 \tag{10.105a}$$

$$= \frac{1}{\pi \mathcal{N}} \int_{q_1} \left(\int_{q_2} e^{-q_2^2/\mathcal{N}} \, dq_2 \right) e^{-(q_1 - \sqrt{E})^2/\mathcal{N}} \, dq_1 \tag{10.105b}$$

To integrate over R_1, we first integrate over the solid vertical strip in Fig. 10.25b. Along the border of R_1,

$$q_2 = \pm \left(\tan \frac{\pi}{M}\right) q_1$$

Hence,

$$P(C|m_1) = \frac{1}{\pi\mathcal{N}} \int_0^\infty \left(\int_{-q_1 \tan(\pi/M)}^{q_1 \tan(\pi/M)} e^{-q_2^2/\mathcal{N}}\, dq_2 \right) e^{-\left(q_1 - \sqrt{E}\right)^2/\mathcal{N}}\, dq_1$$

$$= \frac{1}{\sqrt{\pi\mathcal{N}}} \int_0^\infty \left[1 - 2Q\left(\frac{q_1 \tan(\pi/M)}{\sqrt{\mathcal{N}/2}} \right) \right] e^{-\left(q_1 - \sqrt{E}\right)^2/\mathcal{N}}\, dq_1$$

Changing the variable to $x = \sqrt{2/\mathcal{N}}\, q_1$, we get

$$P(C|m_1) = \frac{1}{\sqrt{2\pi}} \int_0^\infty \left[1 - 2Q\left(x \tan \frac{\pi}{M} \right) \right] e^{-\left(x - \sqrt{2E/\mathcal{N}}\right)^2/2}\, dx \tag{10.106a}$$

Using the fact that E_b, the energy per bit, is $E/\log_2 M$, we have

$$P(C|m_1) = \frac{1}{\sqrt{2\pi}} \int_0^\infty \left[1 - 2Q\left(x \tan \frac{\pi}{M} \right) \right] e^{-\left[x - \sqrt{(2\log_2 M)(E_b/\mathcal{N})}\right]^2/2} dx \tag{10.106b}$$

The integration can also be performed in cylindrical coordinates using the transformation $q_1 = \rho\sqrt{\mathcal{N}/2} \cos\theta$ and $q_2 = \rho\sqrt{\mathcal{N}/2} \sin\theta$. The limits on ρ are $(0, \infty)$ and those on θ are $-\pi/M$ to π/M. Hence,

$$P(C|m_1) = \frac{1}{2\pi} \int_{-\pi/M}^{\pi/M} d\theta \int_0^\infty \rho e^{-(\rho^2 - 2\rho\sqrt{2E/\mathcal{N}}\cos\theta + 2E/\mathcal{N})/2}\, d\rho \tag{10.107a}$$

$$= \frac{1}{2\pi} \int_{-\pi/M}^{\pi/M} d\theta \int_0^\infty \rho e^{-[\rho^2 - 2\rho\sqrt{(2\log_2 M)(E_b/\mathcal{N})}\cos\theta + (2\log_2 M)(E_b/\mathcal{N})]/2} d\rho \tag{10.107b}$$

Because of the symmetry of the signal configuration, $P(C|m_i)$ is the same for all i. Hence,

$$P(C) = P(C|m_1)$$

and

$$P_{eM} = 1 - P(C|m_1)$$

On the other hand, because $s_i(t) = \sqrt{2E/T_M} \cos(\omega_o t + \theta_i)$, where $\omega_o = 2\pi/T_M$, $\theta_i = 2\pi i/M$, the optimum receiver turns out to be just a phase detector similar to that shown in Fig. 10.24 (Prob. 10.6-10). Based on this observation, an alternative expression of P_{eM} can also be found. Since $p_\Theta(\theta)$ of the phase Θ of a sinusoid plus a bandpass Gaussian noise is found in Eq. (9.86d),

$$P_{eM} = 1 - \int_{-\pi/M}^{\pi/M} p_\Theta(\theta)\, d\theta$$

The PDF $p_\Theta(\theta)$ in Eq. (9.86d) involves A (the sinusoid amplitude) and σ_n^2 (the noise variance). Assuming matched filtering and white noise [see Eq. (10.11a)],

$$\frac{A^2}{\sigma_n^2} = \frac{2E_p}{\mathcal{N}} = \frac{2E_b \log_2 M}{\mathcal{N}}$$

Hence,

$$P_{eM} = 1 - \frac{1}{2\pi}\int_{-\pi/M}^{\pi/M} e^{-(E_b \log_2 M/\mathcal{N})}\left\{1 + \sqrt{\frac{4\pi E_b \log_2 M}{\mathcal{N}}}\cos\theta\, e^{(E_b \cos^2\theta \log_2 M/\mathcal{N})} \times \left[1 - Q\left(\sqrt{\frac{2E_b \log_2 M}{\mathcal{N}}}\cos\theta\right)\right]\right\} d\theta \tag{10.108}$$

Figure 10.26 shows the plot of P_{eM} as a function of $E_b/\mathcal{N}$. For $E_b/\mathcal{N} \gg 1$ (weak noise) and $M \gg 2$, Eq. (10.108) can be approximated by[7]

$$P_{eM} \simeq 2Q\left(\sqrt{\frac{2E_b \log_2 M}{\mathcal{N}}}\sin\frac{\pi}{M}\right) \tag{10.109a}$$

$$\simeq 2Q\left(\sqrt{\frac{2\pi^2 E_b \log_2 M}{M^2 \mathcal{N}}}\right) \tag{10.109b}$$

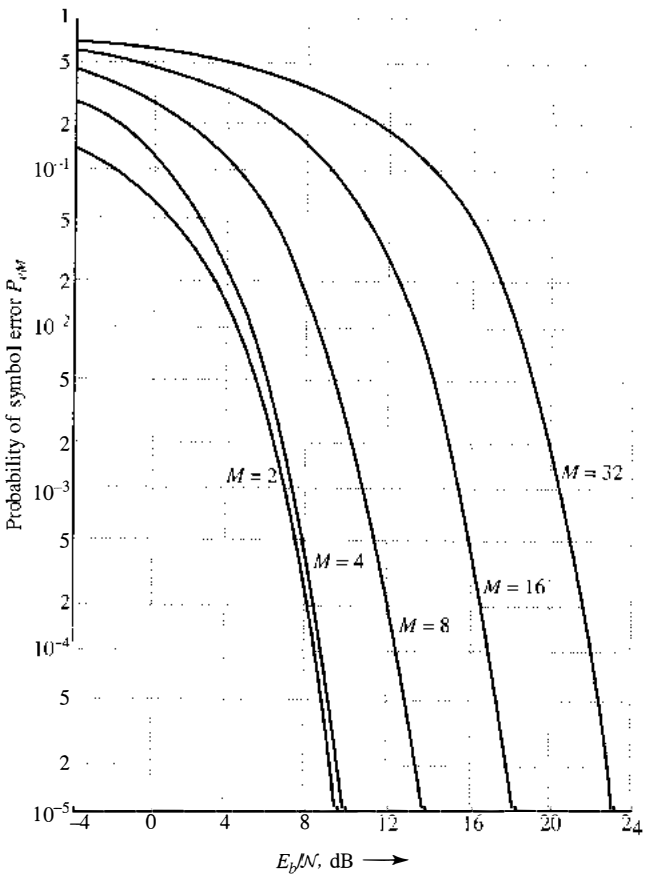


Figure 10.26 Error probability of MPSK.

10.7 GENERAL EXPRESSION FOR ERROR PROBABILITY OF OPTIMUM RECEIVERS

Thus far we have considered rather simple schemes in which the decision regions can be found easily. The method of computing error probabilities from knowledge of decision regions has also been discussed. When the number of signal space dimensions grows, it becomes harder to visualize the decision regions graphically, and as a result the method loses its power. We now develop an analytical expression for computing error probability for a general M-ary scheme.

From the structure of the optimum receiver in Fig. 10.18, we observe that if m_1 is transmitted, then the correct decision will be made only if

$$\mathrm{b}_1 > \mathrm{b}_2, \mathrm{b}_3, \ldots, \mathrm{b}_M$$

In other words,

$$P(C|m_1) = \text{probability } (\mathrm{b}_1 > \mathrm{b}_2, \mathrm{b}_3, \ldots, \mathrm{b}_M|m_1) \tag{10.110}$$

If m_1 is transmitted, then (Fig. 10.18)

$$\mathrm{b}_k = \int_0^{T_M} [s_1(t) + \mathrm{n}(t)]s_k(t)\,dt + a_k \tag{10.111}$$

Let

$$\rho_{ij} = \int_0^{T_M} s_i(t)s_j(t)\,dt \qquad i, j = 1, 2, \ldots, M \tag{10.112}$$

where the ρ_{ij} are known as **cross-correlations**. Thus (if m_1 is transmitted),

$$\mathrm{b}_k = \rho_{1k} + \int_0^{T_M} \mathrm{n}(t)s_k(t)\,dt + a_k \tag{10.113a}$$

$$= \rho_{1k} + a_k + \sum_{j=1}^{N} s_{kj}\mathrm{n}_j \tag{10.113b}$$

where n_j is the component of $\mathrm{n}(t)$ along $\varphi_j(t)$. Note that $\rho_{1k} + a_k$ is a constant, and variables n_j ($j = 1, 2, \ldots, N$) are independent jointly Gaussian variables, each with zero mean and a variance of $\mathcal{N}/2$. Thus, variables b_k are a linear combination of jointly Gaussian variables. It follows that the variables $\mathrm{b}_1, \mathrm{b}_2, \ldots, \mathrm{b}_M$ are also jointly Gaussian. The probability of making a correct decision when m_1 is transmitted can be computed from Eq. (10.110). Note that b_1 can lie anywhere in the range $(-\infty, \infty)$. More precisely, if $p(b_1, b_2, \ldots, b_M|m_1)$ is the joint conditional PDF of $\mathrm{b}_1, \mathrm{b}_2, \ldots, \mathrm{b}_M$, then Eq. (10.110) can be expressed as

$$P(C|m_1) = \int_{-\infty}^{\infty}\int_{-\infty}^{b_1}\cdots\int_{-\infty}^{b_1} p(b_1, b_2, \ldots, b_M|m_1)\,db_1, db_2, \ldots, db_M \tag{10.114a}$$

where the limits of integration of b_1 are $(-\infty, \infty)$, and for the remaining variables the limits are $(-\infty, b_1)$. Thus,

$$P(C|m_1) = \int_{-\infty}^{\infty} db_1 \int_{-\infty}^{b_1} db_2 \cdots \int_{-\infty}^{b_1} p(b_1, b_2, \ldots, b_M|m_1)\, db_M \tag{10.114b}$$

Similarly, $P(C|m_2), \ldots, P(C|m_M)$ can be computed, and

$$P(C) = \sum_{j=1}^{M} P(C|m_j)P(m_j)$$

and

$$P_{eM} = 1 - P(C)$$

Example 10.4 Orthogonal Signal Set

In this set all M equal-energy signals $s_1(t), s_2(t), \ldots, s_M(t)$ are mutually orthogonal. As an example, a signal set for $M = 3$ is shown in Fig. 10.27.

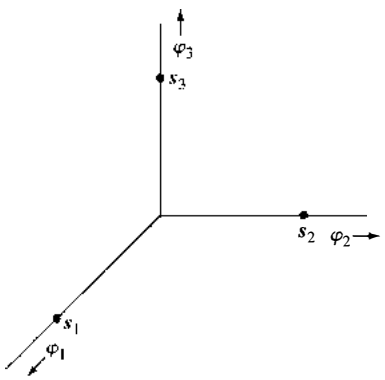


Figure 10.27 Orthogonal signals.

The orthogonal set $\{s_k(t)\}$ is characterized by

$$\langle s_j, s_k \rangle = \begin{cases} 0 & j \neq k \\ E & j = k \end{cases} \tag{10.115}$$

Hence,

$$\rho_{ij} = \langle s_i, s_j \rangle = \begin{cases} 0 & i \neq j \\ E & i = j \end{cases} \tag{10.116}$$

Further, we shall assume all signals to be equiprobable. This yields

$$a_k = \frac{1}{2}\left[\mathcal{N}\ln\left(\frac{1}{M}\right) - E_k\right]$$
$$= -\frac{1}{2}(\mathcal{N}\ln M + E)$$

where $E_k = E$ is the energy of each signal. Note that a_k is the same for every signal. Because the constants a_k enter the expression only for the sake of comparison (Fig. 10.19b), when they are the same, they can be ignored (by setting $a_k = 0$). Also for an orthogonal set,

$$s_k(t) = \sqrt{E}\,\varphi_k(t) \tag{10.117}$$

Therefore,

$$s_{kj} = \begin{cases} \sqrt{E} & k = j \\ 0 & k \neq j \end{cases} \tag{10.118}$$

Hence, from Eqs. (10.113b), (10.116), and (10.118), we have (when m_1 is transmitted)

$$\mathrm{b}_k = \begin{cases} E + \sqrt{E}\,\mathrm{n}_1 & k = 1 \\ \sqrt{E}\,\mathrm{n}_k & k = 2, 3, \ldots, M \end{cases} \tag{10.119}$$

Note that $\mathrm{n}_1, \mathrm{n}_2, \ldots, \mathrm{n}_M$ are independent Gaussian variables, each with zero mean and variance $\mathcal{N}/2$. Variables b_k that are of the form $(\alpha \mathrm{n}_k + \beta)$ are also independent Gaussian variables. Equation (10.119) shows that the variable b_1 has the mean E and variance $(\sqrt{E})^2(\mathcal{N}/2) = \mathcal{N}E/2$. Hence,

$$p_{\mathrm{b}_1}(b_1) = \frac{1}{\sqrt{\pi\mathcal{N}E}} e^{-(b_1 - E)^2/\mathcal{N}E}$$
$$p_{\mathrm{b}_k}(b_k) = \frac{1}{\sqrt{\pi\mathcal{N}E}} e^{-b_k^2/\mathcal{N}E} \qquad k = 2, 3, \ldots, M$$

Because $\mathrm{b}_1, \mathrm{b}_2, \ldots, b_M$ are independent, the joint probability density is the product of the individual densities:

$$p(b_1, b_2, \ldots, b_M | m_1) = \frac{1}{\sqrt{\pi\mathcal{N}E}} e^{-(b_1 - E)^2/\mathcal{N}E} \prod_{k=2}^{M} \left(\frac{1}{\sqrt{\pi\mathcal{N}E}} e^{-b_k^2/\mathcal{N}E}\right)$$

and

$$P(C|m_1) = \frac{1}{\sqrt{\pi\mathcal{N}E}} \int_{-\infty}^{\infty} db_1 [e^{-(b_1 - E)^2/\mathcal{N}E}] \times \prod_{k=2}^{M} \left(\int_{-\infty}^{b_1} \frac{1}{\sqrt{\pi\mathcal{N}E}} e^{-b_k^2/\mathcal{N}E}\, db_k\right)$$
$$= \frac{1}{\sqrt{\pi\mathcal{N}E}} \int_{-\infty}^{\infty} db_1 [e^{-(b_1 - E)^2/\mathcal{N}E}] \times \left(\int_{-\infty}^{b_1} \frac{1}{\sqrt{\pi\mathcal{N}E}} e^{-x^2/\mathcal{N}E}\, dx\right)^{M-1}$$
$$= \frac{1}{\sqrt{\pi\mathcal{N}E}} \int_{-\infty}^{\infty} \left[1 - Q\left(\frac{b_1}{\sqrt{\mathcal{N}E/2}}\right)\right]^{M-1} \times e^{-(b_1 - E)^2/\mathcal{N}E}\, db_1 \tag{10.120a}$$

Changing the variable so that $b_1/\sqrt{\mathcal{N}E/2} = y$, and recognizing that $E/\mathcal{N} = (\log_2 M)\, E_b/\mathcal{N}$, we obtain

$$P(C|m_1) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\left(y-\sqrt{2E/\mathcal{N}}\right)^2/2} [1 - Q(y)]^{M-1}\, dy \tag{10.120b}$$

$$= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\left[y-\sqrt{(2\log_2 M)E_b/\mathcal{N}}\right]^2/2} [1 - Q(y)]^{M-1}\, dy \tag{10.120c}$$

Note that this signal set is geometrically symmetrical; that is, every signal has the same relationship with other signals in the set. As a result,

$$P(C|m_1) = P(C|m_2) = \cdots = P(C|m_M)$$

Hence,

$$P(C) = P(C|m_1)$$

and

$$P_{eM} = 1 - P(C)$$

$$= 1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\left[y-\sqrt{(2\log_2 M)E_b/\mathcal{N}}\right]^2/2} [1 - Q(y)]^{M-1}\, dy \tag{10.120d}$$

In Fig. 10.28 the result of P_{eM} vs. $E_b/\mathcal{N}$ is computed and plotted. This plot shows an interesting behavior for the case of $M = \infty$. As M increases, the performance improves but at the expense of larger bandwidth. Hence, this is a typical case of trading bandwidth for performance.

Multitone Signaling (MFSK)

In the case of multitone signaling, M symbols are transmitted by M orthogonal pulses of frequencies $\omega_1, \omega_2, \ldots, \omega_M$, each of duration T_M. Thus, the M transmitted pulses are of the form

$$\sqrt{2}p'(t)\cos \omega_k t \qquad \omega_k = \frac{2\pi(N+k)}{T_M}$$

The receiver (Fig. 10.29) is a simple extension of the binary receiver. The incoming pulse is multiplied by the corresponding references $\sqrt{2}\cos \omega_i t$ $(i = 1, 2, \ldots, M)$. The filter $H(f)$ is matched to the baseband pulse $p'(t)$ such that

$$h(t) = p'(T_M - t)$$

The same result is obtained if in the ith bank, instead of using a multiplier and $H(f)$, we use a filter matched to the RF pulse $p'(t)\cos \omega_i t$. The M bank outputs sampled at $t = T_M$ are $b_1, b_2, \ldots, b_M$.

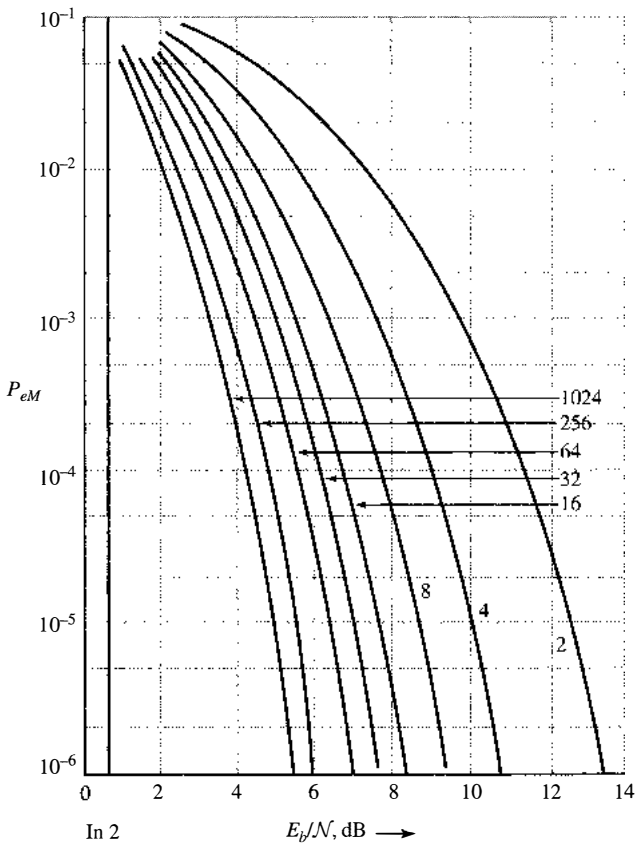


Figure 10.28 Error probability of orthogonal signaling and coherent MFSK.

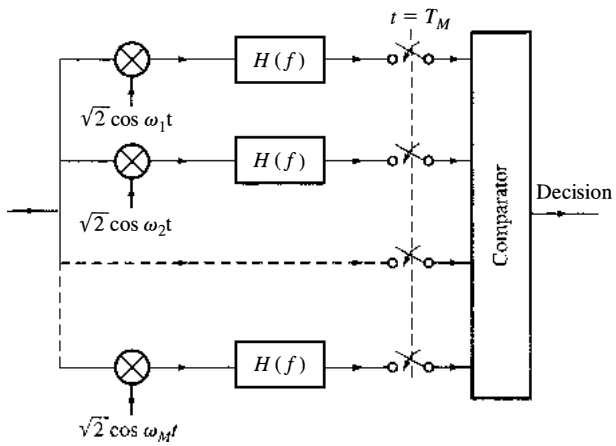


Figure 10.29 Coherent MFSK receiver.

Because the M signal pulses are orthogonal, the analysis from Example 10.4 is directly applicable with error probability

$$P_{eM} = 1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\left(y - \sqrt{2E_b \log_2 M/\mathcal{N}}\right)^2/2} [1 - Q(y)]^{M-1}\, dy \tag{10.121}$$

The M-ary results were shown in Fig. 10.28.

The integral appearing on the right-hand side of Eq. (10.121) is computed and plotted in Fig. 10.28 (P_{eM} vs. $E_b/\mathcal{N}$). This plot shows an interesting behavior for the case of $M = \infty$. By properly taking the limit of P_{eM} in Eq. (10.121) as $M \to \infty$, it can be shown that[5]

$$\lim_{M\to\infty} P_{eM} = \begin{cases} 1 & E_b/\mathcal{N} < \log_e 2 \\ 0 & E_b/\mathcal{N} \geq \log_e 2 \end{cases}$$

Because the signal power $S_i = E_b R_b$, where R_b is the bit rate, it follows that for error-free communication,

$$\frac{E_b}{\mathcal{N}} \geq \log_e 2 = \frac{1}{1.44} \quad \text{or} \quad \frac{S_i}{\mathcal{N} R_b} \geq \frac{1}{1.44}$$

Hence,

$$R_b \leq 1.44 \frac{S_i}{\mathcal{N}} \text{ bit/s} \tag{10.122}$$

This shows that M-ary orthogonal signaling can transmit error-free data at a rate of up to $1.44\, S_i/\mathcal{N}$ bit/s as $M \to \infty$ (see Fig. 10.28).

Bit Error Rate (BER) of Orthogonal Signaling

For PAM and MPSK, we have shown that, by applying the Gray code, $P_b = P_{eM}/\log_2 M$. This result is not valid for MFSK because the errors that predominate in PAM and MPSK, are those in which a symbol is mistaken for its immediate neighbor. We can use the Gray code to assign the adjacent symbols codes that differ in just one digit. In MFSK, on the other hand, a symbol is equally likely to be mistaken for any of the remaining $M - 1$ symbols. Hence, $P(\epsilon)$, the probability of mistaking one particular M-ary symbol for another, is equally likely,

$$P(\epsilon) = \frac{P_{eM}}{M-1} = \frac{P_{eM}}{2^k - 1}$$

If an M-ary symbol differs by 1 bit from N_1 number of symbols, and differs by 2 bits from N_2 number of symbols, and so on, then $\overline{N}_\epsilon$, the average number of bits in error in reception of an M-ary symbol, is

$$\begin{aligned} \overline{N}_\epsilon &= \sum_{n=1}^{k} n N_n P(\epsilon) \\ &= \sum_{n=1}^{k} n N_n \frac{P_{eM}}{2^k - 1} \\ &= \frac{P_{eM}}{2^k - 1} \sum_{n=1}^{k} n \binom{k}{n} \\ &= k\, 2^{k-1} \frac{P_{eM}}{2^k - 1} \end{aligned}$$

This is an average number of bits in error in a sequence of k bits (one M-ary symbol). Consequently, the BER, P_b, is this figure divided by k,

$$P_b = \frac{2^{k-1}}{2^k - 1} P_{eM} \approx \frac{P_{eM}}{2} \qquad k \gg 1$$

From this discussion, one very interesting fact emerges: whenever the optimum receiver is used, the error probability does not depend on specific signal waveforms; it depends only on their geometrical configuration in the signal space.

Bandwidth and Power Trade-offs of M-ary Orthogonal Signals

As illustrated by Landau and Pollak,[3] the dimensionality of a signal is $2BT_M + 1$, where T_M is the signal duration and B is its essential bandwidth. It follows that for an N-dimensional signal space ($N \leq M$), the bandwidth is $B = (N-1)/2T_M$. Thus, reducing the dimensionality N reduces the bandwidth.

We can verify that N-dimensional signals can be transmitted over $(N-1)/2T_M$ Hz by constructing a specific signal set. Let us choose the following orthonormal signals:

$$\begin{aligned}
\varphi_0(t) &= \frac{1}{\sqrt{T_M}} \\
\varphi_1(t) &= \sqrt{\frac{2}{T_M}} \sin \omega_o t \\
\varphi_2(t) &= \sqrt{\frac{2}{T_M}} \cos \omega_o t \qquad \omega_o = \frac{2\pi}{T_M} \\
\varphi_3(t) &= \sqrt{\frac{2}{T_M}} \sin 2\omega_o t \qquad 0 \leq t \leq T_M \\
\varphi_4(t) &= \sqrt{\frac{2}{T_M}} \cos 2\omega_o t \\
&\vdots \\
\varphi_{k-1}(t) &= \sqrt{\frac{2}{T_M}} \sin\left(\frac{k}{2}\omega_o t\right) \\
\varphi_k(t) &= \sqrt{\frac{2}{T_M}} \cos\left(\frac{k}{2}\omega_o t\right)
\end{aligned} \tag{10.123}$$

These $k+1$ orthogonal pulses have a total bandwidth of $(k/2)(\omega_o/2\pi) = k/2T_M$ Hz. Hence, when $k+1 = N$, the bandwidth* is $(N-1)/2T_M$. Thus, $N = 2T_M B + 1$.

To attain a given error probability, there is a trade-off between the average energy of the signal set and its bandwidth. If we reduce the signal space dimensionality, the transmission bandwidth is reduced. But the distances among signals are now smaller, because of the reduced dimensionality. This will increase P_{eM}. Hence, to maintain a given low P_{eM}, we must now move the signals farther apart; that is, we must increase energy. Thus, the cost of reduced bandwidth is paid in terms of increased energy. The trade-off between SNR and bandwidth can also be described from the perspective of information theory (Sec. 13.6).

M-ary signaling provides us with additional means of exchanging, or trading, the transmission rate, transmission bandwidth, and transmitted power. It provides us flexibility in designing a proper communication system. Thus, for a given rate of transmission, we can trade the transmission bandwidth for transmitted power. We can also increase the information rate by a

* Here we are ignoring the band spreading at the edge. This spread is about $1/T_M$ Hz. The actual bandwidth exceeds $(N-1)/2T_M$ by this amount.

factor of k ($k = \log_2 M$) by paying a suitable price in terms of the transmission bandwidth or the transmitted power. Figure 10.28 showed that in multitone signaling the transmitted power decreases with M. However, the transmission bandwidth increases linearly with M, or exponentially with the rate increase factor k ($M = 2^k$). Thus, multitone signaling is radically different from multiamplitude or multiphase signaling. In the latter, the bandwidth is independent of M, but the transmitted power increases as $M^2/\log_2 M = 2^{2k}/k$; that is, the power increases exponentially with the information rate increase factor k. Thus, in the multitone case, the bandwidth increases exponentially with k, and in the multiamplitude or multiphase case, the power increases exponentially with k.

The practical implication is that we should use multiamplitude or multiphase signaling if the bandwidth is at a premium (as in telephone lines) and multitone signaling when power is at a premium (as in space communication). A compromise exists between these two extremes. Let us investigate the possibility of increasing the information rate by a factor k simply by increasing the number of binary pulses transmitted by a factor k. In this case, the transmitted power increases linearly with k. Also because the bandwidth is proportional to the pulse rate, the transmission bandwidth increases linearly with k. Thus, in this case, we can increase the information rate by a factor of k by increasing both the transmission bandwidth and the transmitted power linearly with k, thus avoiding the phantom of the exponential increase that was required in the M-ary system. But here we must increase both the bandwidth and the power, whereas formerly the increase in information rate can be achieved by increasing either the bandwidth or the power. We have thus a great flexibility in trading various parameters and thus in our ability to match our resources to our requirements.

Example 10.5 We are required to transmit 2.08×10^6 binary digits per second with $P_b \leq 10^{-6}$. Three possible schemes are considered:

(**a**) Binary
(**b**) 16-ary ASK
(**c**) 16-ary PSK

The channel noise PSD is $S_n(\omega) = 10^{-8}$. Determine the transmission bandwidth and the signal power required at the receiver input in each case.

(a) Binary: We shall consider polar signaling (the most efficient scheme),

$$P_b = P_e = 10^{-6} = Q\left(\sqrt{\frac{2E_b}{\mathcal{N}}}\right)$$

This yields $E_b/\mathcal{N} = 11.35$. The signal power $S_i = E_b R_b$. Hence,

$$S_i = 11.35\mathcal{N}R_b = 11.35(2 \times 10^{-8})(2.08 \times 10^6) = 0.47 \text{ W}$$

Assuming raised-cosine baseband pulses of roll-off factor 1, the bandwidth B_T is

$$B_T = R_b = 2.08 \text{ MHz}$$

(b) 16-ary ASK: Because each 16-ary symbol carries the information equivalent of $\log_2 16 = 4$ binary digits, we need transmit only $R_M = (2.08 \times 10^6)/4 = 0.52 \times 10^6$ 16-ary pulses per second. This requires a bandwidth B_T of 520 kHz for baseband pulses

and 1.04 MHz for modulated pulses (assuming raised-cosine pulses). Also,

$$P_b = 10^{-6} = \frac{P_{eM}}{\log_2 16}$$

Therefore,

$$P_{eM} = 4 \times 10^{-6} = 2\left(\frac{M-1}{M}\right) Q\left[\sqrt{\frac{6E_b \log_2 16}{\mathcal{N}(M^2-1)}}\right]$$

For $M = 16$, this yields $E_b = 0.499 \times 10^{-5}$. If the M-ary pulse rate is R_M, then

$$\begin{aligned} S_i &= E_{pM} R_M = E_b \log_2 M \cdot R_M \\ &= 0.499 \times 10^{-5} \times 4 \times (0.52 \times 10^6) = 9.34\,\text{W} \end{aligned}$$

(c) 16-ary PSK: We need transmit only $R_M = 0.52 \times 10^6$ pulses per second. For baseband pulses, this will require a bandwidth of 520 kHz. But PSK is a modulated signal, and the required bandwidth is $2(0.52 \times 10^6) = 1.04$ MHz. Also,

$$P_{eM} = 4P_b = 4 \times 10^{-6} \simeq 2Q\left[\sqrt{\frac{2\pi^2 E_b \log_2 16}{256\mathcal{N}}}\right]$$

This yields $E_b = 137.8 \times 10^{-8}$ and

$$\begin{aligned} S_i &= E_b \log_2 16 R_M \\ &= (137.8 \times 10^{-8}) \times 4 \times (0.52 \times 10^6) = 2.86\,\text{W} \end{aligned}$$

10.8 EQUIVALENT SIGNAL SETS

The computation of error probabilities is greatly facilitated by the translation and rotation of coordinate axes. We now show that such operations are permissible.

Consider a signal set with its corresponding decision regions, as shown in Fig. 10.30a. The conditional probability $P(C|m_1)$ is the probability that the noise vector drawn from s_1 lies within R_1. Note that this probability does not depend on the origin of the coordinate system. We may translate the coordinate system any way we wish. This is equivalent to translating the signal set and the corresponding decision regions. Thus, the $P(C|m_i)$ for the translated system shown in Fig. 10.30b is identical to that of the system in Fig. 10.30a.

In the case of Gaussian noise, we make another important observation. The rotation of the coordinate system does not affect the error probability because the noise-vector probability density has spherical symmetry. To show this, we shall consider Fig. 10.30c, which shows the signal set in Fig. 10.30a translated and rotated. Note that a rotation of the coordinate system is equivalent to a rotation of the signal set in the opposite sense. Here for convenience we rotate the signal set instead of the coordinate system. It can be seen that the probability that the noise vector $\mathbf{n}$ drawn from s_1 lies in R_1 is the same in Fig. 10.30a and c, since this probability

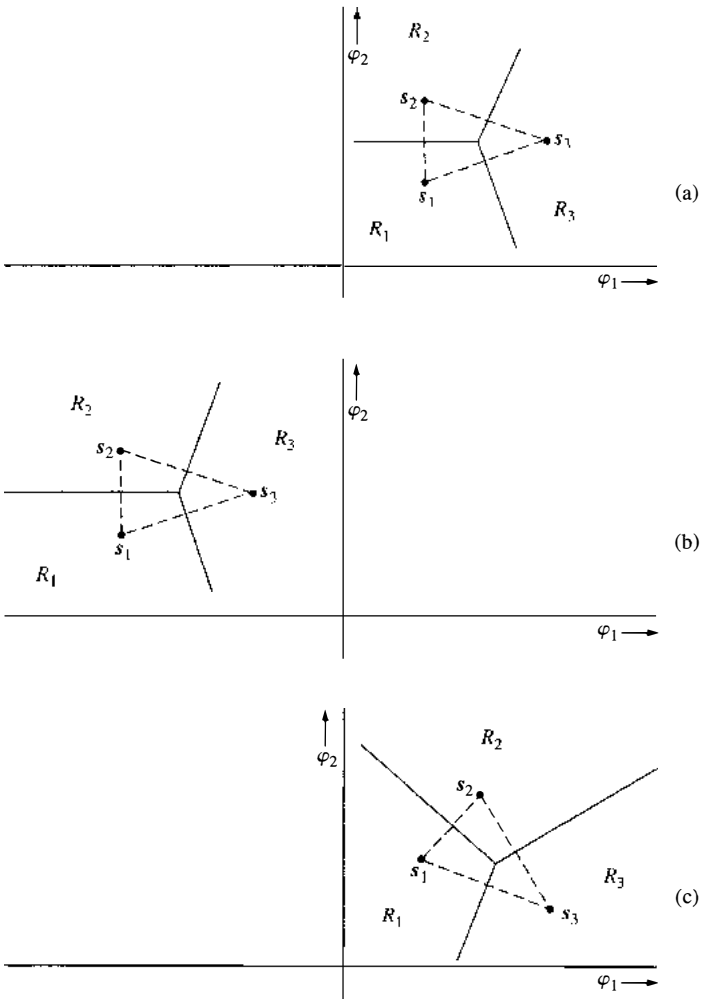


Figure 10.30 Translation and rotation of coordinate axes.

is given by the integral of the noise probability density $p_{\mathbf{n}}(\boldsymbol{n})$ over the region R_1. Because $p_{\mathbf{n}}(\boldsymbol{n})$ has a spherical symmetry for Gaussian noise, the probability will remain unaffected by a rotation of the region R_1. Clearly, for additive Gaussian channel noise, translation and rotation of the coordinate system (or translation and rotation of the signal set) do not affect the error probability. Note that when we rotate or translate a set of signals, the resulting set represents an entirely different set of signals. Yet the error probabilities of the two sets are identical. Such sets are called **equivalent sets.**

The following example demonstrates the utility of translation and rotation of a signal set in the computation of error probability.

Example 10.6 A quaternary PSK (QPSK) signal set is shown in Fig. 10.31a:

$$\boldsymbol{s}_1 = -\boldsymbol{s}_2 = \sqrt{E}\,\boldsymbol{\varphi}_1$$

$$\boldsymbol{s}_3 = -\boldsymbol{s}_4 = \sqrt{E}\,\boldsymbol{\varphi}_2$$

Assuming all symbols to be equiprobable, determine P_{eM} for an AWGN channel with noise PSD $\mathcal{N}/2$.

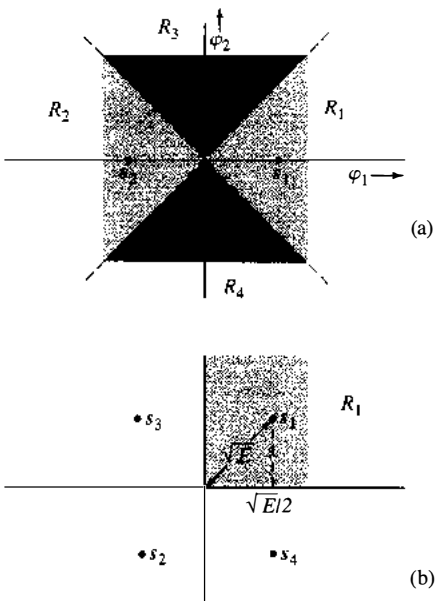


Figure 10.31 Analysis of QPSK.

This problem has already been solved in Example 10.4 for a general value of M. Here we shall solve it for $M = 4$ to demonstrate the power of the rotation of axes.

Because all the symbols are equiprobable, the decision region boundaries will be perpendicular bisectors of lines joining various signal points (Fig. 10.31a). Now

$$P(C|m_1) = P(\text{noise vector originating at } \boldsymbol{s}_1 \text{ remains in } R_1) \tag{10.124}$$

This can be found by integrating the joint PDF of components n_1 and n_2 (originating at $\boldsymbol{s}_1$) over the region R_1. This double integral can be found by using suitable limits, as in Eq. (10.106). The problem is greatly simplified, however, if we rotate the signal set by 45°, as shown in Fig. 10.31b. The decision regions are rectangular, and if n_1 and n_2 are noise components along $\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_2$, then Eq. (10.124) can be expressed as

$$\begin{aligned} P(C|m_1) &= P\left(n_1 > -\sqrt{\frac{E}{2}},\ n_2 > -\sqrt{\frac{E}{2}}\right) \\ &= P\left(n_1 > -\sqrt{\frac{E}{2}}\right) P\left(n_2 > -\sqrt{\frac{E}{2}}\right) \\ &= \left[1 - Q\left(\sqrt{\frac{E}{2\sigma_n^2}}\right)\right]^2 \end{aligned}$$

$$= \left[1 - Q\left(\sqrt{\frac{E}{\mathcal{N}}}\right)\right]^2 \tag{10.125a}$$

$$= \left[1 - Q\left(\sqrt{\frac{2E_b}{\mathcal{N}}}\right)\right]^2 \tag{10.125b}$$

10.8.1 Minimum Energy Signal Set

As noted earlier, an infinite number of possible equivalent signal sets exist. Because signal energy depends on its distance from the origin, however, equivalent sets do not necessarily have the same average energy. Thus, among the infinite possible equivalent signal sets, the one in which the signals are closest to the origin, has the minimum average signal energy (or transmitted power).

Let $m_1, m_2, \ldots, m_M$ be M messages with waveforms $s_1(t), s_2(t), \ldots, s_M(t)$, represented, respectively, by points $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_M$ in the signal space. The mean energy of these signals is $\overline{E}$, given by

$$\overline{E} = \sum_{i=1}^{M} P(m_i)\, \|\boldsymbol{s}_i\|^2$$

Translation of this signal set is equivalent to subtracting some vector $\boldsymbol{a}$ from each signal. We now use this simple operation to yield a minimum mean energy set. We basically wish to find the vector $\boldsymbol{a}$ such that the new mean energy

$$\overline{E}' = \sum_{i=1}^{M} P(m_i)||\boldsymbol{s}_i - \boldsymbol{a}||^2 \tag{10.126}$$

is minimum. We can show that $\boldsymbol{a}$ must be the center of gravity of M points located at $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_M$ with masses $P(m_1), P(m_2), \ldots, P(m_M)$, respectively,

$$\boldsymbol{a} = \sum_{i=1}^{M} P(m_i)\boldsymbol{s}_i = \overline{\boldsymbol{s}_i} \tag{10.127}$$

To prove this, suppose the mean energy is minimum for some translation $\boldsymbol{b}$. Then

$$\begin{aligned}
\overline{E}' &= \sum_{i=1}^{M} P(m_i)||\boldsymbol{s}_i - \boldsymbol{b}||^2 \\
&= \sum_{i=1}^{M} P(m_i)||(\boldsymbol{s}_i - \boldsymbol{a}) + (\boldsymbol{a} - \boldsymbol{b})||^2 \\
&= \sum_{i=1}^{M} P(m_i)||\boldsymbol{s}_i - \boldsymbol{a}||^2 + 2<(\boldsymbol{a} - \boldsymbol{b}), \sum_{i=1}^{M} P(m_i)(\boldsymbol{s}_i - \boldsymbol{a})> + \sum_{i=1}^{M} P(m_i)||\boldsymbol{a} - \boldsymbol{b}||^2
\end{aligned}$$

Observe that the second term in the foregoing expression vanishes according to Eq. (10.127) because

$$\sum_{i=1}^{M} P(m_i)(\boldsymbol{s}_i - \boldsymbol{a}) = \sum_{i=1}^{M} P(m_i)\boldsymbol{s}_i - \boldsymbol{a}\sum_{i=1}^{M} P(m_i)$$
$$= \boldsymbol{a} - \boldsymbol{a} \cdot 1 = 0$$

Hence,

$$\overline{E}' = \sum_{i=1}^{M} P(m_i)\,\|\boldsymbol{s}_i - \boldsymbol{a}\|^2 + \sum_{i=1}^{M} P(m_i)\,\|\boldsymbol{a} - \boldsymbol{b}\|^2$$

This is minimum when $\boldsymbol{b} = \boldsymbol{a}$. Note that the rotation of the coordinates does not change the energy, and, hence, there is no need to rotate the signal set to minimize the energy after the translation.

Example 10.7 For the binary orthogonal signal set of Fig. 10.32a, determine the minimum energy equivalent signal set.

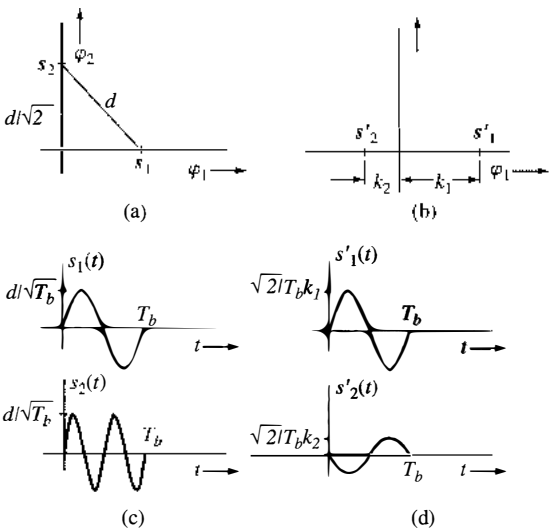


Figure 10.32 Equivalent signal sets.

The minimum energy set for this case is shown in Fig. 10.32b. The origin lies at the center of gravity of the signals. We have also rotated the signals for convenience. The distances k_1 and k_2 must be such that

$$k_1 + k_2 = d$$

and

$$k_1 P(m_1) = k_2 P(m_2)$$

Solution of these two equations yields

$$k_1 = P(m_2)d$$

and

$$k_2 = P(m_1)d$$

Both signal sets (Fig. 10.32a and b) have the same error probability, but the latter has a smaller mean energy. If $\bar{E}$ and $\overline{E'}$ are the respective mean energies of the two sets, then

$$\bar{E} = P(m_1)\frac{d^2}{2} + P(m_2)\frac{d^2}{2} = \frac{d^2}{2}$$

and

$$\begin{aligned}\overline{E'} &= P(m_1)k_1^2 + P(m_2)k_2^2 \\ &= P(m_1)P^2(m_2)d^2 + P(m_2)P^2(m_1)d^2 \\ &= P(m_1)P(m_2)d^2\end{aligned}$$

Note that for $P(m_1) + P(m_2) = 1$, the product $P(m_1)P(m_2)$ is maximum when $P(m_1) = P(m_2) = 1/2$, in which case

$$P(m_1)P(m_2) = \frac{1}{4}$$

and consequently

$$\overline{E'} \leq \frac{d^2}{4}$$

Therefore,

$$\overline{E'} \leq \frac{\bar{E}}{2}$$

and for the case of equiprobable signals,

$$\overline{E'} = \frac{\bar{E}}{2}$$

In this case,

$$k_1 = k_2 = \frac{d}{2}$$

$$\bar{E} = \frac{d^2}{2} \qquad \text{and} \qquad \overline{E'} = \frac{d^2}{4}$$

The signals in Fig. 10.32b are called **antipodal signals** when $k_1 = k_2$. The error probability of the signal set in Fig. 10.32a (and 10.32b) is equal to that in Fig. 10.22a and can be found from Eq. (10.97a).

As a concrete example, let us choose the basis signals as sinusoids of frequency $\omega_o = 2\pi/T_M$:

$$\varphi_1(t) = \sqrt{\frac{2}{T_M}} \sin \omega_o t \qquad 0 \le t < T_M$$

$$\varphi_2(t) = \sqrt{\frac{2}{T_M}} \sin 2\omega_o t$$

Hence,

$$s_1(t) = \frac{d}{\sqrt{2}}\varphi_1(t) = \frac{d}{\sqrt{T_M}} \sin \omega_o t \qquad 0 \le t < T_M$$

$$s_2(t) = \frac{d}{\sqrt{2}}\varphi_2(t) = \frac{d}{\sqrt{T_M}} \sin 2\omega_o t$$

The signals $s_1(t)$ and $s_2(t)$ are shown in Fig. 10.32c, and the geometrical representation is shown in Fig. 10.32a. Both signals are located at a distance $d/\sqrt{2}$ from the origin, and the distance between the signals is d.

The minimum energy signals $s_1'(t)$ and $s_2'(t)$ for this set are given by

$$s_1'(t) = \sqrt{\frac{2}{T_M}} P(m_2) d \sin \omega_o t \qquad 0 \le t < T_M$$

$$s_2'(t) = -\sqrt{\frac{2}{T_M}} P(m_1) d \sin \omega_o t$$

These signals are sketched in Fig. 10.32d.

10.8.2 Simplex Signal Set

A minimum energy equivalent set of an equiprobable orthogonal set is called a **simplex**, or **transorthogonal**, **signal set**. A simplex set can be derived as an equivalent set from the orthogonal set in Eq. (10.115).

To obtain the minimum energy set, the origin should be shifted to the center of gravity of the signal set. For the two-dimensional case (Fig. 10.33a), the simplex set is shown in Fig. 10.33c, and for the three-dimensional case (Fig. 10.33b), the simplex set is shown in Fig. 10.33d. Note that the dimensionality of the simplex signal set is less than that of the orthogonal set by 1. This is true in general for any value of M. It can be shown that the simplex signal set is the optimum (minimum error probability) for the case of equiprobable signals embedded in white Gaussian noise when energy is constrained.[4, 8]

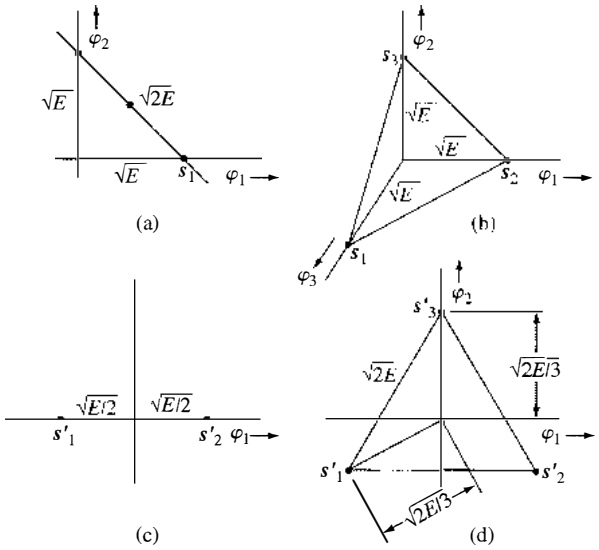


Figure 10.33
Simplex signals.

We can calculate the mean energy of the simplex set by noting that it is obtained by translating the orthogonal set by a vector $\boldsymbol{a}$, given in Eq. (10.127),

$$\boldsymbol{a} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{s}_i$$

For orthogonal signals,

$$\boldsymbol{s}_i = \sqrt{E}\,\boldsymbol{\varphi}_i$$

Therefore,

$$\boldsymbol{a} = \frac{\sqrt{E}}{M}\sum_{i=1}^{M} \boldsymbol{\varphi}_i$$

where E is the energy of each signal in the orthogonal set and $\boldsymbol{\varphi}_i$ is the unit vector along the ith coordinate axis. The signals in the simplex set are given by

$$\begin{aligned} \boldsymbol{s}'_k &= \boldsymbol{s}_k - \boldsymbol{a} \\ &= \sqrt{E}\boldsymbol{\varphi}_k - \frac{\sqrt{E}}{M}\sum_{i=1}^{M} \boldsymbol{\varphi}_i \end{aligned} \tag{10.128}$$

The energy E' of signal $\boldsymbol{s}'_k$ is given by $|\boldsymbol{s}'_k|^2$,

$$E' = \langle \boldsymbol{s}'_k, \boldsymbol{s}'_k \rangle \tag{10.129}$$

Substituting Eq. (10.128) into Eq. (10.129) and observing that the set φ_i is orthonormal, we have

$$\begin{aligned} E' &= E - \frac{E}{M} \\ &= E\left(1 - \frac{1}{M}\right) \end{aligned} \tag{10.130}$$

Hence, for the same performance (error probability), the mean energy of the simplex signal set is $1 - 1/M$ times that of the orthogonal signal set. For $M \gg 1$, the difference is not significant. For this reason and because of the simplicity in generating, orthogonal signals, rather than simplex signals are used in practice whenever M exceeds 4 or 5.

In Sec. 13.6, we shall show that in the limit as $M \to \infty$, the orthogonal (as well as the simplex) signals attain the upper bound of performance predicted by Shannon's theorem.

10.9 NONWHITE (COLORED) CHANNEL NOISE

Thus far we have restricted our analysis exclusively to white Gaussian channel noise. Our analysis can be extended to nonwhite, or colored, Gaussian channel noise. To proceed, the Karhunen-Löeve expansion of Eq. (10.57) must be solved for the colored noise with autocorrelation function $R_x(t, t_1)$. This general solution, however, can be quite complex to implement.[4]

Fortunately, for a large class of colored Gaussian noises, the power spectral density $S_n(f)$ is nonzero within the message signal bandwidth B. This property provides an effective alternative. We use a noise-whitening filter $H(f)$ at the input of the receiver, where

$$H(f) = \frac{1}{\sqrt{S_n(f)}} e^{-j2\pi f t_d}$$

The delay t_d is introduced to ensure that the whitening filter is causal (realizable).

Consider a signal set $\{s_i(t)\}$ and a channel noise $n(t)$ that is not white [$S_n(f)$ is not constant]. At the input of the receiver, we use a noise-whitening filter $H(f)$ that transforms the colored noise into white noise (Fig. 10.34). But it also alters the signal set $\{s_i(t)\}$ to $\{s_i'(t)\}$, where

$$s_i'(t) = s_i(t) * h(t)$$

We now have a new signal set $\{s_i'(t)\}$ mixed with white Gaussian noise, for which the optimum receiver and the corresponding error probability can be determined by the method discussed earlier.

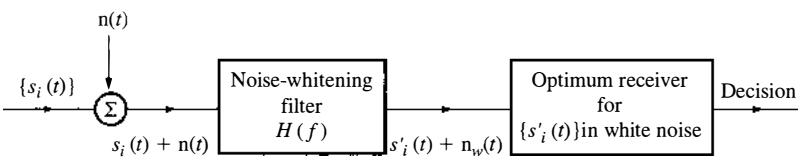


Figure 10.34 Optimum M-ary receiver for nonwhite channel noise.

10.10 OTHER USEFUL PERFORMANCE CRITERIA

The optimum receiver uses the decision strategy that makes the best possible use of the observed data and any a priori information available. The strategy will also depend on the weights assigned to various types of error. In this chapter we have thus far assumed that all errors have equal weight (or equal cost). This assumption is not justified in all cases, and we may therefore have to alter the decision rule.

Generalized Bayes Receiver

If we are given a priori probabilities and the cost functions of errors of various types, the receiver that minimizes the average cost of decision is called the **Bayes receiver**, and the decision rule is **Bayes' decision rule**. Note that the receiver that has been discussed so far is the Bayes receiver under the condition that all errors have equal cost (equal weight). To generalize this rule, let

$$C_{kj} = \text{cost of deciding that } \hat{m} = m_k \text{ when } m_j \text{ was transmitted} \tag{10.131}$$

and, as usual,

$$P(m_i|\boldsymbol{q}) = \text{conditional probability that } m_i \text{ was transmitted when } \boldsymbol{q} \text{ is received}$$

If $\boldsymbol{q}$ is received, then the probability that m_j was transmitted is $P(m_j|\boldsymbol{q})$ for all $j = 1, 2, \ldots, M$. Hence, the average cost of the decision $\hat{m} = m_k$ is β_k, given by

$$\begin{aligned}\beta_k &= C_{k1}P(m_1|\boldsymbol{q}) + C_{k2}P(m_2|\boldsymbol{q}) + \cdots + C_{kM}(m_M|\boldsymbol{q}) \\ &= \sum_{j=1}^{M} C_{kj}P(m_j|\boldsymbol{q})\end{aligned} \tag{10.132}$$

Thus, if $\boldsymbol{q}$ is received, the optimum receiver decides that $\hat{m} = m_k$ if

$$\beta_k < \beta_i \qquad \text{for all } i \neq k$$

or

$$\sum_{j=1}^{M} C_{kj}P(m_j|\boldsymbol{q}) < \sum_{j=1}^{M} C_{ij}P(m_j|\boldsymbol{q}) \qquad \text{for all } i \neq k \tag{10.133}$$

Use of Bayes' mixed rule in Eq. (10.133) yields

$$\sum_{j=1}^{M} C_{kj}P(m_j)p_{\mathbf{q}}(\boldsymbol{q}|m_j) < \sum_{j=1}^{M} C_{ij}P(m_j)p_{\mathbf{q}}(\boldsymbol{q}|m_j) \qquad \text{for all } i \neq k \tag{10.134}$$

Note that C_{kk} is the cost of setting $\hat{m} = m_k$ when m_k is transmitted. This cost is generally zero. If we assign equal weight to all other errors, then

$$C_{kj} = \begin{cases} 0 & k = j \\ 1 & k \neq j \end{cases} \tag{10.135}$$

and the decision rule in Eq. (10.134) reduces to the rule in Eq. (10.83), as expected. The generalized Bayes receiver for $M = 2$, assuming $C_{11} = C_{22} = 0$, sets $\hat{m} = m_1$ if

$$C_{12}P(m_2)p_{\mathbf{q}}(\boldsymbol{q}|m_2) < C_{21}P(m_1)p_{\mathbf{q}}(\boldsymbol{q}|m_1)$$

Otherwise, the receiver decides that $\hat{m} = m_2$.

Maximum Likelihood Receiver

The strategy used in the Bayes receiver discussed in the preceding subsection is general, except that it can be implemented only when the a priori probabilities $P(m_1)$, $P(m_2)$, ..., $P(m_M)$ are known. Frequently this information is not available. Under these conditions various possibilities exist, depending on the assumptions made. When, for example, there is no reason to expect any one signal to be more likely than any other, we may assign equal probabilities to all the messages:

$$P(m_1) = P(m_2) = \cdots = P(m_M) = \frac{1}{M}$$

Bayes' rule [Eq. (10.83)] in this case becomes: set $\hat{m} = m_k$ if

$$p_{\mathbf{q}}(\boldsymbol{q}|m_k) > p_{\mathbf{q}}(\boldsymbol{q}|m_i) \qquad \text{for all } i \neq k \tag{10.136}$$

Observe that $p_{\mathbf{q}}(\boldsymbol{q}|m_k)$ represents the probability of observing $\boldsymbol{q}$ when m_k is transmitted. Thus, the receiver chooses that signal which, when transmitted, will maximize the likelihood (probability) of observing the received $\boldsymbol{q}$. Hence, this receiver is called the **maximum likelihood receiver**. Note that the maximum likelihood receiver is a Bayes receiver for the cost of Eq. (10.135) under the condition that the a priori message probabilities are equal. In terms of geometrical concepts, the maximum likelihood receiver decides in favor of that signal which is closest to the received data $\boldsymbol{q}$. The practical implementation of the maximum likelihood receiver is the same as that of the Bayes receiver (Figs. 10.18 and 10.19) under the condition that all a priori probabilities are equal to $1/M$.

If the signal set is geometrically symmetrical, and if all a priori probabilities are equal (maximum likelihood receiver), then the decision regions for various signals are congruent. In this case, because of symmetry, the conditional probability of a correct decision is the same no matter which signal is transmitted, that is,

$$P(C|m_i) = \text{constant} \qquad \text{for all } i$$

Because

$$P(C) = \sum_{i=1}^{M} P(m_i)P(C|m_i)$$

in this case

$$P(C) = P(C|m_i) \tag{10.137}$$

Thus, the error probability of the maximum likelihood receiver is independent of the actual source statistics $P(m_i)$ for the case of symmetrical signal sets. It should, however, be realized

that if the actual source statistics were known beforehand, one could use Bayes' decision rule to design a better receiver.

It is apparent that if the source statistics are not known, the maximum likelihood receiver proves very attractive for a symmetrical signal set. In such a receiver one can specify the error probability independently of the actual source statistics.

Minimax Receiver

Designing a receiver with a certain decision rule completely specifies the conditional probabilities $P(C|m_i)$. The probability of error is given by

$$\begin{aligned} P_{eM} &= 1 - P(C) \\ &= 1 - \sum_{i=1}^{M} P(m_i)P(C|m_i) \end{aligned}$$

Thus, in general, for a given receiver (with some specified decision rule) the error probability depends on the source statistics $P(m_i)$. The error probability is the largest for some source statistics. The error probability in the worst possible case is $[P_{eM}]_{\max}$ and represents the upper bound on the error probability of the given receiver. This upper bound $[P_{eM}]_{\max}$ serves as an indication of the quality of the receiver. Each receiver (with a certain decision rule) will have a certain $[P_{eM}]_{\max}$. The receiver that has the smallest upper bound on the error probability, that is, the minimum $[P_{eM}]_{\max}$, is called the **minimax receiver**.

We shall illustrate the minimax concept for a binary receiver with on-off signaling. The conditional PDFs of the receiving-filter output sample r at $t = T_b$ are $p(r|\mathbf{1})$ and $p(r|\mathbf{0})$. These are the PDFs of r for the "on" and the "off" pulse (i.e., no pulse), respectively. Figure 10.35a shows these PDFs with a certain threshold a. If we receive $r \geq a$, we choose the hypothesis "signal present" (**1**), and the shaded area to the right of a is the probability of **false alarm** (deciding "signal present" when in fact the signal is not present). If $r < a$, we choose the hypothesis "signal absent" (**0**), and the shaded area to the left of a is the probability of **false dismissal** (deciding "signal absent" when in fact the signal is present). It is obvious that the larger the threshold a, the larger the false dismissal error and the smaller the false alarm error (Fig. 10.35b).

We shall now find the minimax condition for this receiver. For the minimax receiver, we consider all possible receivers (all possible values of a in this case) and find the maximum

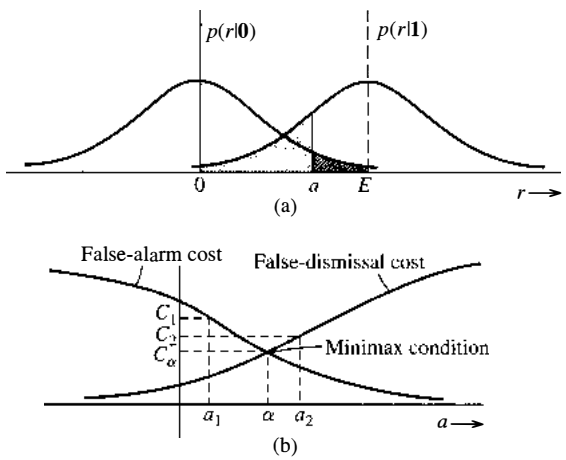


Figure 10.35 Explanation of minimax concept.

error probability (or cost) that occurs under the worst possible a priori probability distribution. Let us choose $a = a_1$, as shown in Fig. 10.35b. In this case the worst possible case occurs when $P(\mathbf{0}) = 1$ and $P(\mathbf{1}) = 0$, that is, when the signal $s(t)$ is always absent. The type of error in this case is false alarm. These errors have a cost C_1. On the other hand, if we choose $a = a_2$, the worst possible case occurs when $P(\mathbf{0}) = 0$ and $P(\mathbf{1}) = 1$, that is, when the signal is always present, causing only the false-dismissal type of errors. These errors have a cost C_2. It is evident that for the setting $a = \alpha$, the costs of false alarm and false dismissal are equal, namely, C_α. Hence, for all possible source statistics the cost is C_α. Because $C_\alpha < C_1$ and C_2, this cost is the *minimum* of the maximum possible cost (because the worst cases are considered) that accrues for all values of a. Hence, $a = \alpha$ represents the minimax setting.

It follows from this discussion that the minimax receiver is rather conservative. It is designed under the pessimistic assumption that the worst possible source statistics exist. The maximum likelihood receiver, on the other hand, is designed on the assumption that all messages are equally likely. It can, however, be shown that for a symmetrical signal set, the maximum likelihood receiver is in fact the minimax receiver. This can be proved by observing that for a symmetrical set, the probability of error of a maximum likelihood receiver (equal a priori probabilities) is independent of the source statistics [Eq. (10.137)]. Hence, for a symmetrical set, the error probability $P_{eM} = \alpha$ of a maximum likelihood receiver is also equal to its $[P_{eM}]_{\max}$. We now show that no other receiver exists whose $[P_{eM}]_{\max}$ is less than the α of a maximum likelihood receiver for a symmetrical signal set. This is seen from the fact that for equiprobable messages, the maximum likelihood receiver is optimum by definition. All other receivers must have $P_{eM} > \alpha$ for equiprobable messages. Hence, $[P_{eM}]_{\max}$ for these receivers can never be less than α. This proves that the maximum likelihood receiver is indeed the minimax receiver for a symmetrical signal set.

10.11 NONCOHERENT DETECTION

If the phase θ in the received RF pulse $\sqrt{2}p'(t)\cos(\omega_c t + \theta)$ is unknown, we can no longer use coherent detection techniques. Instead, we must rely on noncoherent techniques, such as envelope detection. It can be shown[9, 10] that when the phase θ of the received pulse is random and uniformly distributed over $(0, 2\pi)$, the optimum detector is a filter matched to the RF pulse $\sqrt{2}p'(t)\cos\omega_c t$ followed by an envelope detector, a sampler (to sample at $t = T_b$), and a comparator to make the decision (Fig. 10.36).

Amplitude Shift Keying

The noncoherent detector for ASK is shown in Fig. 10.36. The filter $H(f)$ is a filter matched to the RF pulse, ignoring the phase. This means the filter output amplitude A_p will not necessarily be maximum at the sampling instant. But the envelope will be close to maximum at the sampling

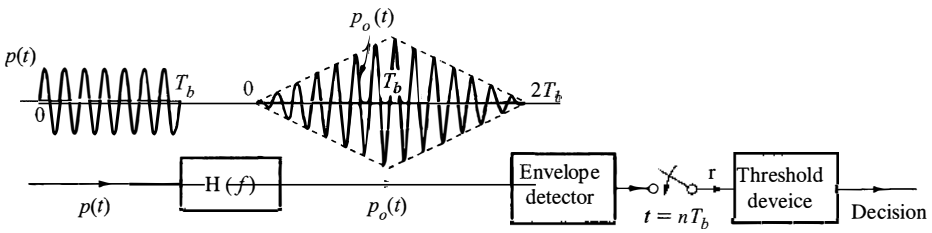


Figure 10.36 Noncoherent detection of digital modulated signals for ASK.

instant (Fig. 10.36). The matched filter output is now detected by an envelope detector. The envelope is sampled at $t = T_b$ for making the decision.

When a **1** is transmitted, the output of the envelope detector at $t = T_b$ is an envelope of a sine wave of amplitude A_p in a Gaussian noise of variance σ_n^2. In this case, the envelope r has a Ricean density, given by [Eq. (9.86a)]

$$p_r(r|m = 1) = \frac{r}{\sigma_n^2} e^{-(r^2+A_p^2)/2\sigma_n^2} I_0\left(\frac{rA_p}{\sigma_n^2}\right) \tag{10.138a}$$

Also, when $A_p \gg \sigma_n$ (small-noise case) from Eq. (9.86c), we have

$$p_r(r|m = 1) \simeq \sqrt{\frac{r}{2\pi A_p \sigma_n^2}}\, e^{-(r-A_p)^2/2\sigma_n^2} \tag{10.138b}$$

$$\simeq \frac{1}{\sigma_n\sqrt{2\pi}} e^{-(r-A_p)^2/2\sigma_n^2} \tag{10.138c}$$

Observe that for small noise, the PDF of r is practically Gaussian, with mean A_p and variance σ_n^2. When **0** is transmitted, the output of the envelope detector is an envelope of a Gaussian noise of variance σ_n^2. The envelope in this case has a Rayleigh density, given by [Eq. (9.81)]

$$p_r(r|m = 0) = \frac{r}{\sigma_n^2} e^{-r^2/2\sigma_n^2}$$

Both $p_r(r|m = 1)$ and $p_r(r|m = 0)$ are shown in Fig. 10.37. Using the argument used earlier (see Fig. 10.4), the optimum threshold is found to be the point where the two densities intersect. Hence, the optimum threshold a_o is

$$\frac{a_o}{\sigma_n^2} e^{-(a_o^2+A_p^2)/2\sigma_n^2} I_0\left(\frac{A_p a_o}{\sigma_n^2}\right) = \frac{a_o}{\sigma_n^2} e^{-a_o^2/2\sigma_n^2}$$

or

$$e^{-A_p^2/2\sigma_n^2} I_0\left(\frac{A_p a_o}{\sigma_n^2}\right) = 1$$

This equation is satisfied to a close approximation for

$$a_o = \frac{A_p}{2}\sqrt{1 + \frac{8\sigma_n^2}{A_p^2}}$$

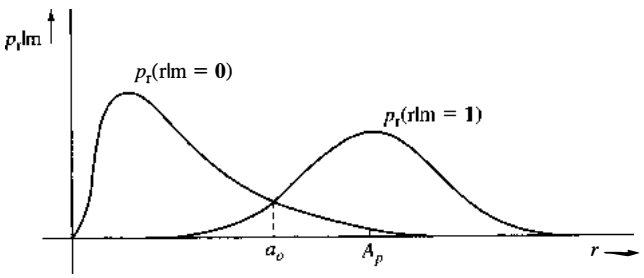


Figure 10.37 Conditional PDFs in the noncoherent defection of ASK signals.

Because the matched filter is used, $A_p = E_p$ and $\sigma_n^2 = \mathcal{N}E_p/2$. Moreover, for ASK there are, on the average, only $R_b/2$ nonzero pulses per second. Thus, $E_b = E_p/2$. Hence,

$$\left(\frac{A_p}{\sigma_n}\right)^2 = \frac{2E_p}{\mathcal{N}} = 4\frac{E_b}{\mathcal{N}}$$

and

$$a_o = E_b\sqrt{1 + \frac{2}{E_b/\mathcal{N}}} \tag{10.139a}$$

Observe that the optimum threshold is not constant but depends on $E_b/\mathcal{N}$. This is a serious drawback in a fading channel. For a strong signal, $E_b/\mathcal{N} \gg 1$,

$$a_o \simeq E_b = \frac{A_p}{2} \tag{10.139b}$$

and

$$\begin{aligned} P(\epsilon|\mathrm{m} = \mathbf{0}) &= \int_{A_p/2}^{\infty} p_r(r|\mathrm{m} = \mathbf{0})\, dr \\ &= \int_{A_p/2}^{\infty} \frac{r}{\sigma_n^2} e^{-r^2/2\sigma_n^2}\, dr \\ &= e^{-A_p^2/8\sigma_n^2} \\ &= e^{-\frac{1}{2}E_b/\mathcal{N}} \end{aligned} \tag{10.140}$$

Also,

$$P(\epsilon|\mathrm{m} = \mathbf{1}) = \int_{-\infty}^{A_p/2} p_r(r|\mathrm{m} = \mathbf{1})\, dr$$

Evaluation of this integral is somewhat cumbersome.[4] For a strong signal (that is, for $E_b/\mathcal{N} \gg 1$), the Ricean PDF can be approximated by the Gaussian PDF [Eq. (9.86c)], and

$$\begin{aligned} P(\epsilon|\mathrm{m} = \mathbf{1}) &\approx \frac{1}{\sigma_n\sqrt{2\pi}} \int_{-\infty}^{A_p/2} e^{-(r-A_p)^2/2\sigma_n^2}\, dr \\ &= Q\left(\frac{A_p}{2\sigma_n}\right) \\ &= Q\left(\sqrt{\frac{E_b}{\mathcal{N}}}\right) \end{aligned} \tag{10.141}$$

As a result,

$$P_b = P_m(\mathbf{0})(\epsilon|\mathrm{m} = \mathbf{0}) + P_m(\mathbf{1})P(\epsilon|\mathrm{m} = \mathbf{1})$$

Assuming $P_m(\mathbf{1}) = P_m(\mathbf{0}) = 0.5$,

$$P_b = \frac{1}{2}\left[e^{-\frac{1}{2}E_b/\mathcal{N}} + Q\left(\sqrt{\frac{E_b}{\mathcal{N}}}\right)\right] \tag{10.142a}$$

Using the $Q(\cdot)$ approximation in Eq. (8.38a),

$$P_b \simeq \frac{1}{2}\left(1 + \frac{1}{\sqrt{2\pi E_b/\mathcal{N}}}\right)e^{-\frac{1}{2}E_b/\mathcal{N}} \qquad E_b/\mathcal{N} \gg 1 \tag{10.142b}$$

$$\simeq \frac{1}{2}e^{-\frac{1}{2}E_b/\mathcal{N}} \tag{10.142c}$$

Note that in an optimum receiver, for $E_b/\mathcal{N} \gg 1$, $P(\epsilon|\mathrm{m} = \mathbf{1})$ is much smaller than $P(\epsilon|\mathrm{m} = \mathbf{0})$. For example, at $E_b/\mathcal{N} = 10$, $P(\epsilon|\mathrm{m} = \mathbf{0}) \simeq 8.7P(\epsilon|\mathrm{m} = \mathbf{1})$. Hence, mistaking **0** for **1** is the type of error that predominates. The timing information in noncoherent detection is extracted from the envelope of the received signal by methods discussed in Sec. 7.5.2.

For a coherent detector,

$$P_b = Q\left(\sqrt{\frac{E_b}{\mathcal{N}}}\right)$$

$$\simeq \frac{1}{\sqrt{2\pi E_b/\mathcal{N}}}e^{-\frac{1}{2}E_b/\mathcal{N}} \qquad E_b/\mathcal{N} \gg 1 \tag{10.143}$$

This appears similar to Eq. (10.142c) (the noncoherent case). Thus for a large $E_b/\mathcal{N}$, the performances of the coherent detector and the envelope detector are similar (Fig. 10.38).

Frequency Shift Keying

A noncoherent receiver for FSK is shown in Fig. 10.39. The filters $H_0(f)$ and $H_1(f)$ are matched to the two RF pulses corresponding to **0** and **1**, respectively. The outputs of the

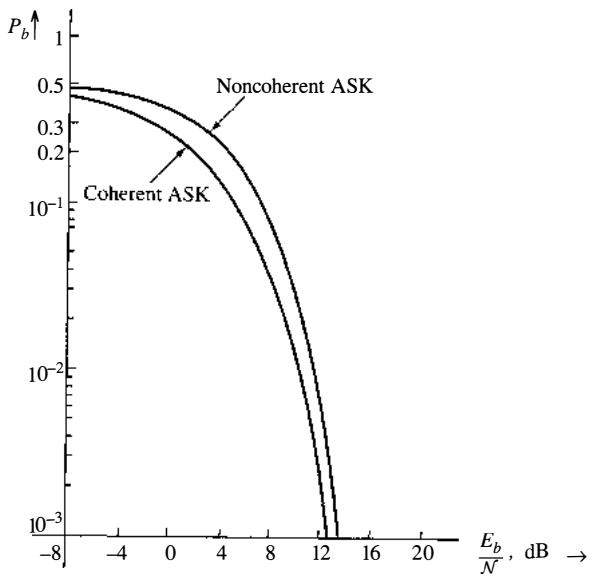


Figure 10.38 Error probability of noncoherent ASK detection.

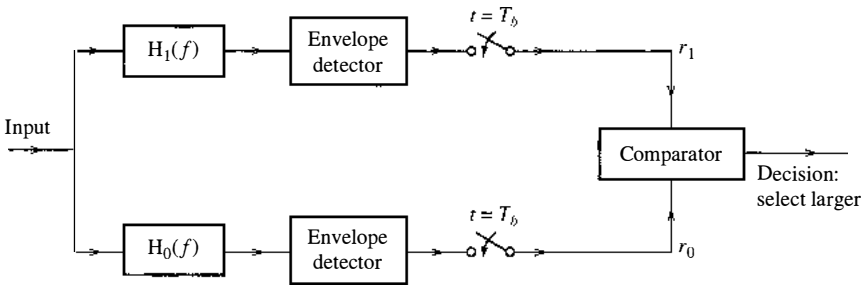


Figure 10.39 Noncoherent detection of binary FSK.

envelope detectors at $t = T_b$ are r_0 and r_1, respectively. The noise components of outputs of filters $H_0(f)$ and $H_1(f)$ are the Gaussian RVs n_0 and n_1, respectively, with $\sigma_{n_0} = \sigma_{n_1} = \sigma_n$.

If **1** is transmitted ($m = 1$), then at the sampling instant, the envelope r_1 has the Ricean PDF*

$$p_{r_1}(r_1) = \frac{r_1}{\sigma_n^2} e^{-(r_1^2+A_p^2)/2\sigma_n^2} I_0\left(\frac{r_1 A_p}{\sigma_n^2}\right)$$

and r_0 is the noise envelope with Rayleigh density

$$P_{r_0}(r_0) = \frac{r_0}{\sigma_n^2} e^{-r_0^2/2\sigma_n^2}$$

The decision is $m = 1$ if $r_1 > r_0$ and $m = 0$ if $r_1 < r_0$. Hence, when binary **1** is transmitted, an error is made if $r_0 > r_1$,

$$P(\epsilon|m = 1) = P(r_0 > r_1)$$

The event $r_0 > r_1$ is the same as the joint event "r_1 has any positive value† and r_0 has a value greater than r_1." This is simply the joint event $(0 < r_1 < \infty, r_0 > r_1)$. Hence,

$$\begin{aligned} P(\epsilon|m = 1) &= P(0 < r_1 < \infty, r_0 > r_1) \\ &= \int_0^\infty \int_{r_1}^\infty p_{r_1 r_0}(r_1, r_0)\, dr_1\, dr_0 \end{aligned}$$

Because r_1 and r_0 are independent, $p_{r_1 r_0} = p_{r_1} p_{r_0}$. Hence,

$$\begin{aligned} P(\epsilon|m = 1) &= \int_0^\infty \frac{r_1}{\sigma_n^2} e^{-(r_1^2+A_p^2)/2\sigma_n^2} I_0\left(\frac{r_1 A_p}{\sigma_n^2}\right) \int_{r_1}^\infty \frac{r_0}{\sigma_n^2} e^{-r_0^2/2\sigma_n^2}\, dr_1\, dr_0 \\ &= \int_0^\infty \frac{r_1}{\sigma_n^2} e^{-(2r_1^2+A_p^2)/2\sigma_n^2} I_0\left(\frac{r_1 A_p}{\sigma_n^2}\right) dr_1 \end{aligned}$$

Letting $x = \sqrt{2}\, r_1$ and $\alpha = A_p/\sqrt{2}$, we have

$$P(\epsilon|m = 1) = \tfrac{1}{2} e^{-A_p^2/4\sigma_n^2} \int_0^\infty \frac{x}{\sigma_n^2} e^{-(x^2+\alpha^2)/2\sigma_n^2} I_0\left(\frac{x\alpha}{\sigma_n^2}\right) dx$$

* An orthogonal FSK is assumed. This ensures that r_0 and r_1 have Rayleigh and Rice densities, respectively, when **1** is transmitted.

† r_1 is the envelope detector and can take only positive values.

Observe that the integrand is a Ricean density, and, hence, its integral is unity. Therefore,

$$P(\epsilon|\mathrm{m} = \mathbf{1}) = \tfrac{1}{2}\, e^{-A_p^2/4\sigma_\mathrm{n}^2} \tag{10.144a}$$

Note that for a matched filter,

$$\rho_{\max}^2 = \frac{A_p^2}{\sigma_\mathrm{n}^2} = \frac{2E_p}{\mathcal{N}}$$

For FSK, $E_b = E_p$, and Eq. (10.144a) becomes

$$P(\epsilon|\mathrm{m} = \mathbf{1}) = \tfrac{1}{2}\, e^{-\frac{1}{2}E_b/\mathcal{N}} \tag{10.144b}$$

Similarly,

$$P(\epsilon|\mathrm{m} = \mathbf{0}) = \tfrac{1}{2}\, e^{-\frac{1}{2}E_b/\mathcal{N}} \tag{10.144c}$$

and

$$P_b = \tfrac{1}{2}\, e^{-\frac{1}{2}E_b/\mathcal{N}} \tag{10.145}$$

This behavior is similar to that of noncoherent ASK [Eq. (10.142c)]. Again we observe that for $E_b/\mathcal{N} \gg 1$, the performance of coherent and noncoherent FSK are essentially similar.

From the practical point of view, FSK is to be preferred over ASK because FSK has a fixed optimum threshold, whereas the optimum threshold of ASK depends on $E_b/\mathcal{N}$ (the signal level). Hence, ASK is particularly susceptible to signal fading. Because the decision of FSK involves a comparison between r_0 and r_1, both variables will be affected equally by signal fading. Hence, channel fading does not degrade the noncoherent FSK performance as it does the noncoherent ASK. This is the outstanding advantage of noncoherent FSK over noncoherent ASK. In addition, unlike noncoherent ASK, probabilities $P(\epsilon|\mathrm{m} = \mathbf{1})$ and $P(\epsilon|\mathrm{m} = \mathbf{0})$ are equal in noncoherent FSK. The price paid by FSK for such an advantage is its larger bandwidth requirement.

Noncoherent MFSK

From the practical point of view, phase coherence of M frequencies is difficult to maintain. Hence in practice, coherent MFSK is rarely used. Noncoherent MFSK is much more common. The receiver for noncoherent MFSK is similar to that for binary noncoherent FSK (Fig. 10.39), but with M banks corresponding to M frequencies, in which filter $H_i(f)$ is matched to the RF pulse $p(t)\cos\omega_i t$. The analysis is straightforward. If $\mathrm{m} = \mathbf{1}$ is transmitted, then r_1 is the envelope of a sinusoid of amplitude A_p plus bandpass Gaussian noise, and r_j ($j = 2, 3, \ldots, M$) is the envelope of the bandpass Gaussian noise. Hence, r_1 has Ricean density, and $\mathrm{r}_2, \mathrm{r}_3, \ldots, \mathrm{r}_M$ have Rayleigh density. From the same arguments used in the coherent case, we have

$$\begin{aligned} P_{\mathrm{CM}} &= P(C|\mathrm{m} = \mathbf{1}) = P(0 \le \mathrm{r}_1 < \infty,\ \mathrm{n}_2 < \mathrm{r}_1,\ \mathrm{n}_3 < \mathrm{r}_1, \ldots,\ \mathrm{n}_M < \mathrm{r}_1) \\ &= \int_0^\infty \frac{\mathrm{r}_1}{\sigma_\mathrm{n}^2} I_0\left(\frac{r_1 A_p}{\sigma_\mathrm{n}^2}\right) e^{-(r_1^2+A_p^2)/2\sigma_\mathrm{n}^2} \left(\int_0^{r_1} \frac{x}{\sigma_\mathrm{n}^2}\, e^{-x^2/2\sigma_\mathrm{n}^2}\, dx\right)^{M-1} dr_1 \\ &= \int_0^\infty \frac{\mathrm{r}_1}{\sigma_\mathrm{n}^2} I_0\left(\frac{r_1 A_p}{\sigma_\mathrm{n}^2}\right) e^{-(r_1^2+A_p^2)/2\sigma_\mathrm{n}^2} \left(1 - e^{-r_1^2/2\sigma_\mathrm{n}^2}\right)^{M-1} dr_1 \end{aligned}$$

Substituting $r_1^2/2\sigma_n^2 = x$ and $(A_p/\sigma_n)^2 = 2E_p/\mathcal{N} = 2E_b \log M/\mathcal{N}$, we obtain

$$P_{\text{CM}} = e^{-(E_b \log_2 M/\mathcal{N})} \int_0^\infty e^{-x}(1 - e^{-x})^{M-1} I_0\left(2\sqrt{\frac{xE_b \log_2 M}{\mathcal{N}}}\right) dx \tag{10.146a}$$

Using the binomial theorem to expand $(1 - e^x)^{M-1}$, we obtain

$$(1 - e^x)^{M-1} = \sum_{m=0}^{M-1} \binom{M-1}{m} (-1)^m e^{-mx}$$

Substitution of this equality into Eq. (10.146a) and recognizing that

$$\int_0^\infty y e^{-ay^2} I_0(by)\, dy = \frac{1}{2a} e^{-b^2/4a}$$

we obtain (after interchanging the order of summation and integration)

$$P_{\text{CM}} = \sum_{m=0}^{M-1} \binom{M-1}{m} \frac{(-1)^m}{m+1} e^{-mE_b \log_2 M/\mathcal{N}(m+1)} \tag{10.146b}$$

and

$$P_{eM} = 1 - P_{\text{CM}} = \sum_{m=1}^{M-1} \binom{M-1}{m} \frac{(-1)^{m+1}}{m+1} e^{-mE_b \log_2 M/\mathcal{N}(m+1)} \tag{10.146c}$$

The error probability P_{eM} is shown in Fig. 10.40 as a function of $E_b/\mathcal{N}$. It can be seen that the performance of noncoherent MFSK is only slightly inferior to that of coherent MFSK, particularly for large M.

Differentially Coherent PSK

Just as it is impossible to demodulate a DSB-SC signal with an envelope detector, it is also impossible to demodulate PSK (which is really DSB-SC) noncoherently. We can, however, demodulate PSK without the synchronous, or coherent, local carrier by using what is known as differential PSK (DPSK).

The optimum receiver is shown in Fig. 10.41. This receiver is very much like a correlation detector (Fig. 10.3), which is equivalent to a matched filter detector. In a correlation detector, we multiply pulse $p(t)$ by a locally generated pulse $p(t)$. In the case of DPSK, we take advantage of the fact that the two RF pulses used in transmission are identical except for the sign (or phase). In the detector in Fig. 10.41, we multiply the incoming pulse by the preceding pulse. Hence, the preceding pulse serves as a substitute for the locally generated pulse. The only difference is that the preceding pulse is noisy because of channel noise, and this tends to degrade the performance in comparison to coherent PSK. When the output r is positive, the present pulse is identical to the previous one, and when r is negative, the present pulse is the negative of the previous pulse. Hence, from the knowledge of the first reference digit, it is possible to detect all the received digits. Detection is facilitated by using so-called *differential encoding*, identical to what was discussed in Sec. 7.3.6 for duobinary signaling.

To derive the DPSK error probability, we observe that DPSK by means of differential coding is essentially an orthogonal signaling scheme. A binary **1** is transmitted by a sequence

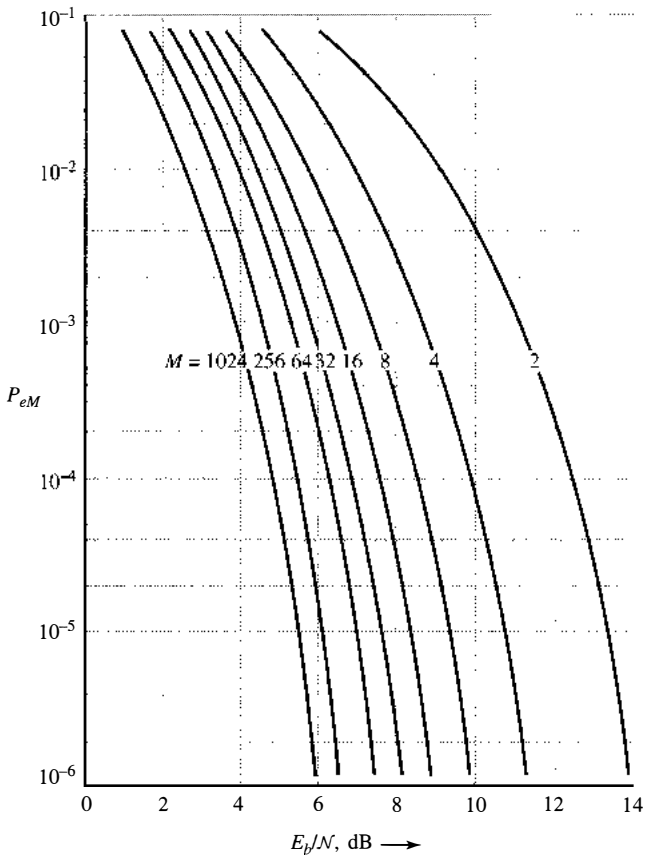


Figure 10.40 Error probability of noncoherent MFSK.

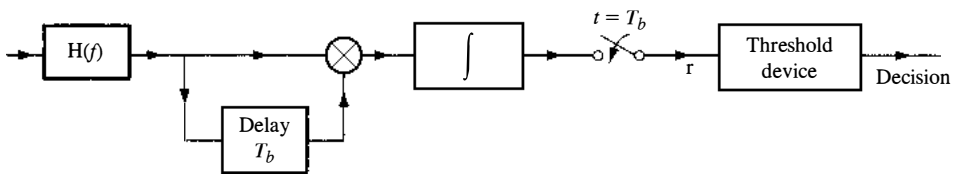


Figure 10.41 Differential PSK detection.

of two pulses (p, p) or $(-p, -p)$ over $2T_b$ seconds (no transition). Similarly, a binary **0** is transmitted by a sequence of two pulses $(p, -p)$ or $(-p, p)$ over $2T_b$ seconds (transition). Either of the pulse sequences used for binary **1** is orthogonal to either of the pulse sequences used for binary **0**. Because no local carrier is generated for demodulation, the detection is noncoherent, with an effective pulse energy equal to $2E_p$ (twice the energy of pulse p). The actual energy transmitted per digit is only E_p, however, the same as in noncoherent FSK. Consequently, the performance of DPSK is 3 dB superior to that of noncoherent FSK. Hence from Eq. (10.145), we can write P_b for DPSK as

$$P_b = \frac{1}{2}e^{-E_b/\mathcal{N}} \tag{10.147}$$

This error probability (Fig. 10.42) is superior to that of noncoherent FSK by 3 dB and is essentially similar to coherent PSK for $E_b/\mathcal{N} \gg 1$ [Eq. (10.39)]. This is as expected, because

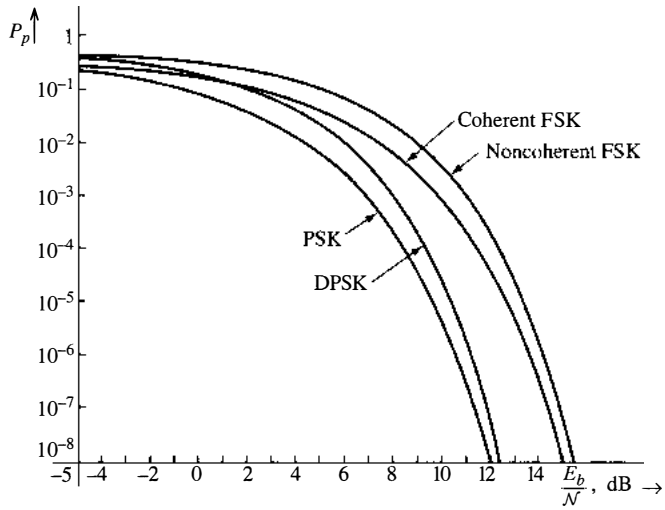


Figure 10.42 Error probability of PSK, DPSK, and coherent and noncoherent FSK.

we saw earlier that DPSK appears similar to PSK for large SNR. Rigorous derivation of Eq. (10.147) can be found in the literature.[7]

10.12 MATLAB EXERCISES

In this group of computer exercises, we give readers an opportunity to test the implementation and the performance of basic digital communication systems.

COMPUTER EXERCISE 10.1: BINARY POLAR SIGNALING WITH DIFFERENT PULSES

In the first exercise, we validate the performance analysis of the binary polar signaling presented in Section 10.1. Optimum (matched filter) detection are always used at the receiver. In the program `Ex10_1.m`, three different pulses are used for polar signaling:

- Rectangular pulse $p(t) = u(t) - u(t - T)$.
- Half-sine pulse $p(t) = \sin(\pi t/T)[u(t) - u(t - T)]$.
- Root-raised cosine pulse with roll-off factor $r = 0.5$ (or bandwidth $0.75/T$) and truncated to duration of $6T$.

```
% Matlab Program <Ex10_1.m>
% This Matlab exercise <Ex10_1.m> performs simulation of
% binary baseband polar transmission in AWGN channel.
% The program generates polar baseband signals using 3 different
% pulse shapes (root-raised cosine (r=0.5), rectangular, half-sine)
% and estimate the bit error rate (BER) at different Eb/N for display
clear;clf;
L=1000000;              % Total data symbols in experiment is 1 million
% To display the pulse shape, we oversample the signal
% by factor of f_ovsamp=8
f_ovsamp=8;             % Oversampling factor vs data rate
delay_rc=3;
% Generating root-raised cosine pulseshape (rolloff factor = 0.5)
```

```
prcos=rcosflt([ 1 ], 1, f_ovsamp, 'sqrt', 0.5, delay_rc);
prcos=prcos(1:end-f_ovsamp+1);
prcos=prcos/norm(prcos);
pcmatch=prcos(end:-1:1);
% Generating a rectangular pulse shape
prect=ones(1,f_ovsamp);
prect=prect/norm(prect);
prmatch=prect(end:-1:1);
% Generating a half-sine pulse shape
psine=sin([0:f_ovsamp-1]*pi/f_ovsamp);
psine=psine/norm(psine);
psmatch=psine(end:-1:1);
% Generating random signal data for polar signaling
s_data=2*round(rand(L,1))-1;
% upsample to match the 'fictitious oversampling rate'
% which is f_ovsamp/T   (T=1 is the symbol duration)
s_up=upsample(s_data,f_ovsamp);

% Identify the decision delays due to pulse shaping
% and matched filters
delayrc=2*delay_rc*f_ovsamp;
delayrt=f_ovsamp-1;
delaysn=f_ovsamp-1;
% Generate polar signaling of different pulse-shaping
xrcos=conv(s_up,prcos);
xrect=conv(s_up,prect);
xsine=conv(s_up,psine);
t=(1:200)/f_ovsamp;
subplot(311)
figwave1=plot(t,xrcos(delayrc/2:delayrc/2+199));
title('(a) Root-raised cosine pulse.');
set(figwave1,'Linewidth',2);
subplot(312)
figwave2=plot(t,xrect(delayrt:delayrt+199));
title('(b) Rectangular pulse.')
set(figwave2,'Linewidth',2);
subplot(313)
figwave3=plot(t,xsine(delaysn:delaysn+199));
title('(c) Half-sine pulse.')
xlabel('Number of data symbol periods')
set(figwave3,'Linewidth',2);
% Find the signal length
Lrcos=length(xrcos);Lrect=length(xrect);Lsine=length(xsine);
BER=[];
noiseq=randn(Lrcos,1);
% Generating the channel noise (AWGN)
for i=1:10,
    Eb2N(i)=i;                          %(Eb/N in dB)
    Eb2N_num=10^(Eb2N(i)/10);           % Eb/N in numeral
    Var_n=1/(2*Eb2N_num);               %1/SNR is the noise variance
    signois=sqrt(Var_n);                % standard deviation
    awgnois=signois*noiseq;             % AWGN
    % Add noise to signals at the channel output
    yrcos=xrcos+awgnois;
```

```
    yrect=xrect+awgnois(1:Lrect);
    ysine=xsine+awgnois(1:Lsine);

    % Apply matched filters first
    z1=conv(yrcos,pcmatch);clear awgnois, yrcos;
    z2=conv(yrect,prmatch);clear yrect;
    z3=conv(ysine,psmatch);clear ysine;

    % Sampling the received signal and acquire samples
    z1=z1(delayrc+1:f_ovsamp:end);
    z2=z2(delayrt+1:f_ovsamp:end);
    z3=z3(delaysn+1:f_ovsamp:end);
    % Decision based on the sign of the samples
    dec1=sign(z1(1:L));dec2=sign(z2(1:L));dec3=sign(z3(1:L));
    % Now compare against the original data to compute BER for
    % the three pulses
    BER=[BER;sum(abs(s_data-dec1))/(2*L)...
        sum(abs(s_data-dec2))/(2*L) ...
        sum(abs(s_data-dec3))/(2*L)];
    Q(i)=0.5*erfc(sqrt(Eb2N_num));      %Compute the Analytical BER
end
figure(2)
subplot(111)
figber=semilogy(Eb2N,Q,'k-',Eb2N,BER(:,1),'b-*',...
    Eb2N,BER(:,2),'r-o',Eb2N,BER(:,3),'m-v');
legend('Analytical', 'Root-raised cosine','Rectangular','Half-sine')
xlabel('E_b/N (dB)');ylabel('BER')
set(figber,'Linewidth',2);
figure(3)
% Spectrum comparison
[Psd1,f]=pwelch(xrcos,[],[],[],'twosided',f_ovsamp);
[Psd2,f]=pwelch(xrect,[],[],[],'twosided',f_ovsamp);
[Psd3,f]=pwelch(xsine,[],[],[],'twosided',f_ovsamp);
figpsd1=semilogy(f-f_ovsamp/2,fftshift(Psd1));
ylabel('Power spectral density');
xlabel('frequency in unit of {1/T}');
tt1=title('(a) PSD using root-raised cosine pulse (rolloff factor r=0.5)')
set(tt1,'FontSize',11);
figure(4)
figpsd2=semilogy(f-f_ovsamp/2,fftshift(Psd2));
ylabel('Power spectral density');
xlabel('frequency in unit of {1/T}');;
tt2=title('(b) PSD using rectangular NRZ pulse');
set(tt2,'FontSize',11);
figure(5)
figpsd3=semilogy(f-f_ovsamp/2,fftshift(Psd3));
ylabel('Power spectral density');
xlabel('frequency in unit of {1/T}');
tt3=title('(c) PSD using half-sine pulse');
set(tt3,'FontSize',11);
```

This program first shows the polar modulated binary signals in a snapshot given by Fig. 10.43. The 3 different waveforms are the direct results of their different pulse shapes. Nevertheless, their bit error

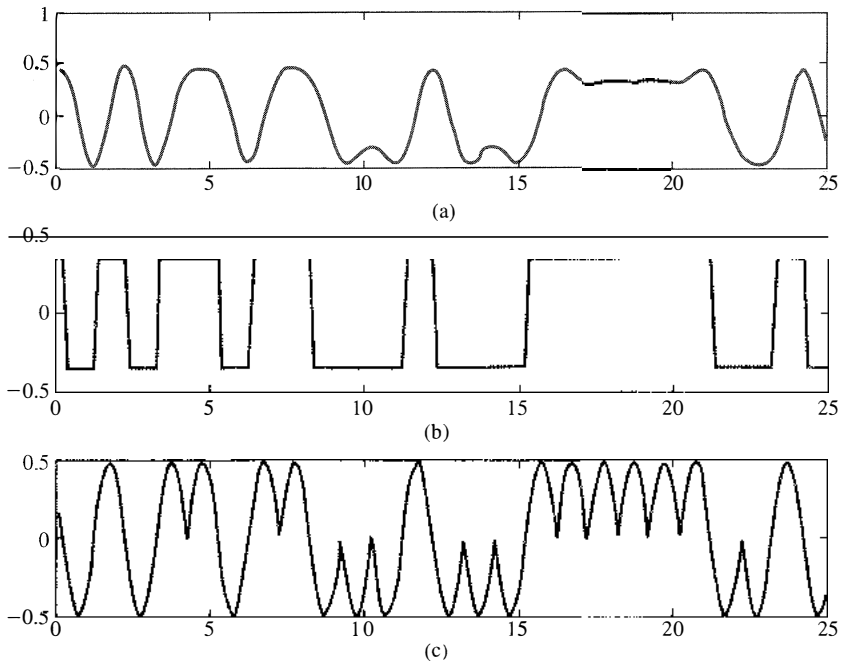


Figure 10.43 Snapshot of the modulated signals from three difference pulse shapes: (a) root-raised cosine pulses, of rolloff factor=0.5; (b) rectangular pulse; (c) half-sine pulse pulse.

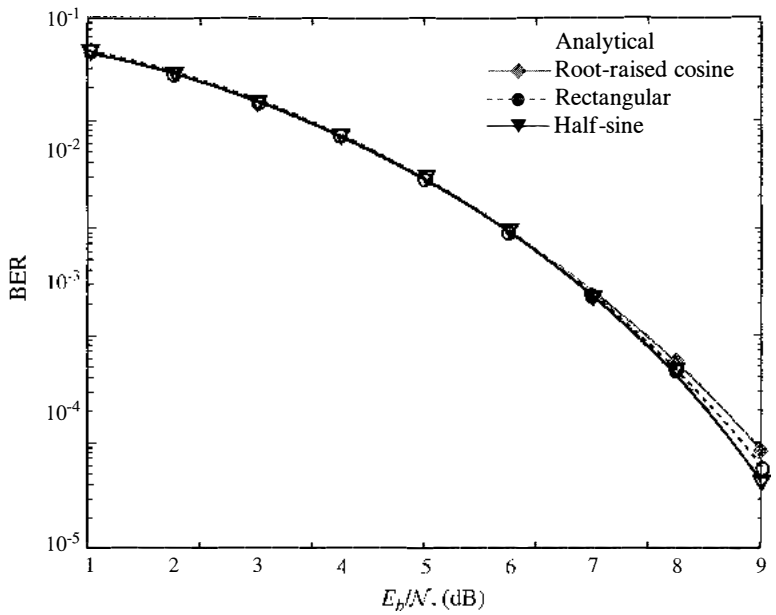


Figure 10.44 BER of optimum (matched filter) detection of polar signaling using three difference pulse shapes: (a) root-raised cosine pulse of roll-off factor 0.5; (b) rectangular pulse; (c) half-sine pulse.

rate (BER) performances are identical, as shown in Fig. 10.44. This confirms the results from Sec. 10.1 that the polar signal performance is independent of the pulse shape.

The program also provides the power spectral density (PSD) for binary polar signaling using the three different modulated signals. From Fig. 10.45, we can see that that the root-raised cosine pulse clearly requires the least bandwidth. The half-sine signaling exhibits larger main lobe but smaller overall bandwidth. The sharp-edged rectangular pulse is the least bandwidth efficient. Thus, despite registering

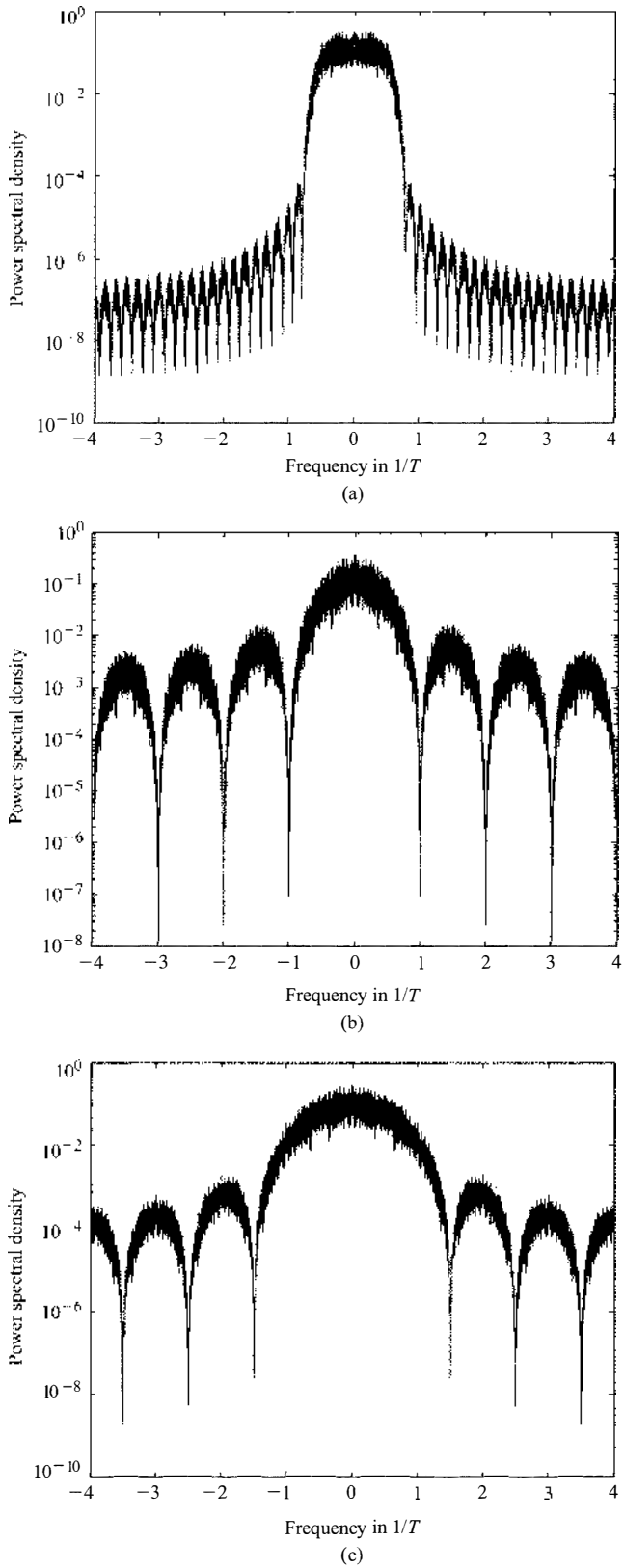


Figure 10.45 Power spectral density of the binary polar transmission using three difference pulse shapes: (a) root-raised cosine pulse of roll-off factor 0.5; (b) rectangular NRZ pulse; (c) half-sine pulse pulse.

the same BER from simulation, the three different polar modulations require drastically different amount of channel bandwidth.

COMPUTER EXERCISE 10.2: ON-OFF BINARY SIGNALING

Next, we present an exercise that implements and tests the on-off signaling as well as a more generic orthogonal type of signaling. Recall that on-off signaling is a special form of orthogonal binary signaling. MATLAB program `Ex10_2.m` will measure the receiver BER of both signaling schemes.

```
% MATLAB PROGRAM <Ex10_2.m>
% This Matlab exercise <Ex10_2.m> generate
% on/off baseband signals using root-raised cosine
% pulseshape (rolloff factor = 0.5) and orthogonal baseband
% signal before estimating the bit error rate (BER) at different
% Eb/N ratio for display and comparison
clear;clf
L=1000000;          % Total data symbols in experiment is 1 million
% To display the pulse shape, we oversample the signal
% by factor of f_ovsamp=8
f_ovsamp=16;        % Oversampling factor vs data rate
delay_rc=3;
% Generating root-raised cosine pulseshape (rolloff factor = 0.5)
prcos=rcosflt([ 1 ], 1, f_ovsamp, 'sqrt', 0.5, delay_rc);
prcos=prcos(1:end-f_ovsamp+1);
prcos=prcos/norm(prcos);
pcmatch=prcos(end:-1:1);
% Generating a rectangular pulse shape
psinh=sin([0:f_ovsamp-1]*pi/f_ovsamp);
psinh=psinh/norm(psinh);
phmatch=psinh(end:-1:1);
% Generating a half-sine pulse shape
psine=sin([0:f_ovsamp-1]*2*pi/f_ovsamp);
psine=psine/norm(psine);
psmatch=psine(end:-1:1);
% Generating random signal data for polar signaling
s_data=round(rand(L,1));
% upsample to match the 'fictitious oversampling rate'
% which is f_ovsamp/T   (T=1 is the symbol duration)
s_up=upsample(s_data,f_ovsamp);
s_cp=upsample(1-s_data,f_ovsamp);

% Identify the decision delays due to pulse shaping
% and matched filters
delayrc=2*delay_rc*f_ovsamp;
delayrt=f_ovsamp-1;
% Generate polar signaling of different pulse-shaping
xrcos=conv(s_up,prcos);
xorth=conv(s_up,psinh)+conv(s_cp,psine);
t=(1:200)/f_ovsamp;
figure(1)
subplot(211)
figwave1=plot(t,xrcos(delayrc/2:delayrc/2+199));
title('(a) On/off root-raised cosine pulse.');
```

```
set(figwave1,'Linewidth',2);
subplot(212)
figwave2=plot(t,xorth(delayrt:delayrt+199));
title('(b) Orthogonal modulation.')
set(figwave2,'Linewidth',2);
% Find the signal length
Lrcos=length(xrcos);Lrect=length(xorth);
BER=[];
noiseq=randn(Lrcos,1);
% Generating the channel noise (AWGN)
for i=1:12,
    Eb2N(i)=i;                          %(Eb/N in dB)
    Eb2N_num=10^(Eb2N(i)/10);       % Eb/N in numeral
    Var_n=1/(2*Eb2N_num);        %1/SNR is the noise variance
    signois=sqrt(Var_n);            % standard deviation
    awgnois=signois*noiseq;         % AWGN
    % Add noise to signals at the channel output
    yrcos=xrcos+awgnois/sqrt(2);
    yorth=xorth+awgnois(1:Lrect);


    % Apply matched filters first
    z1=conv(yrcos,pcmatch);clear awgnois, yrcos;
    z2=conv(yorth,phmatch);
    z3=conv(yorth,psmatch);clear yorth;

    % Sampling the received signal and acquire samples
    z1=z1(delayrc+1:f_ovsamp:end);
    z2=z2(delayrt+1:f_ovsamp:end-f_ovsamp+1);
    z3=z3(delayrt+1:f_ovsamp:end-f_ovsamp+1);
    % Decision based on the sign of the samples
    dec1=round((sign(z1(1:L)-0.5)+1)*.5);dec2=round((sign(z2-z3)+1)*.5);
    % Now compare against the original data to compute BER for
    % the three pulses
    BER=[BER;sum(abs(s_data-dec1))/L sum(abs(s_data-dec2))/L];
    Q(i)=0.5*erfc(sqrt(Eb2N_num/2));  % Compute the Analytical BER
end
figure(2)
subplot(111)
figber=semilogy(Eb2N,Q,'k-',Eb2N,BER(:,1),'b-*',Eb2N,BER(:,2),'r-o');
fleg=legend('Analytical', 'Root-raised cosine on/off','Orthogonal
signaling');
fx=xlabel('E_b/N (dB)');fy=ylabel('BER');
set(figber,'Linewidth',2);set(fleg,'FontSize',11);
set(fx,'FontSize',11);
set(fy,'FontSize',11);
% We can plot the individual pulses used for the binary orthogonal
% signaling
figure(3)
subplot(111);
pulse=plot((0:f_ovsamp)/f_ovsamp,[psinh 0],'k-',...
    (0:f_ovsamp)/f_ovsamp,[psine 0],'k-o');
pleg=legend('Half-sine pulse', 'Sine pulse');
ptitle=title('Binary orthogonal signals');
```

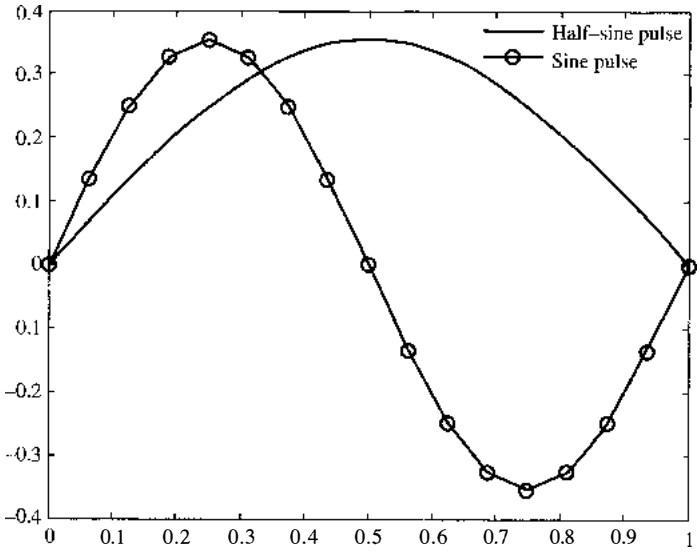


Figure 10.46 Waveforms of the two pulses used in orthogonal binary signaling: solid curver, half-sine pulse; curve with circles, sine pulse.

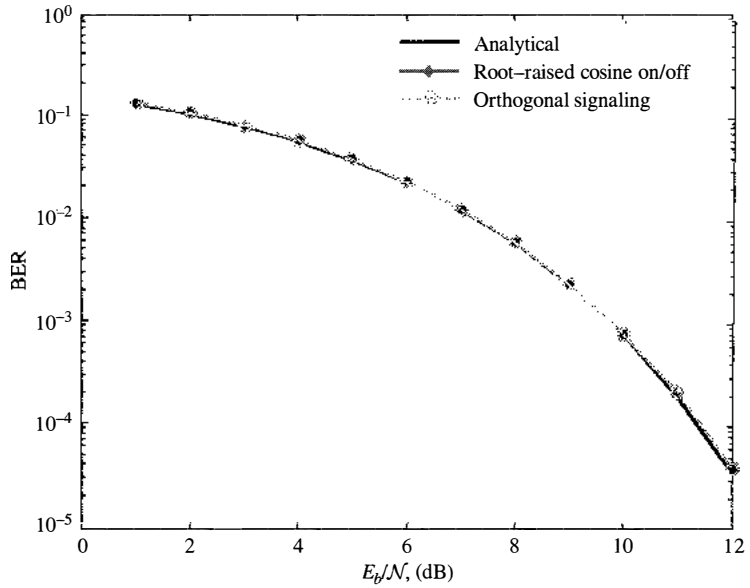


Figure 10.47 Measured BER results in comparison with analytical BER.

```
set(pulse,'Linewidth',2);
set(pleg,'Fontsize',10);
set(ptitle,'FontSize',11);
```

For the on-off signaling, we will continue to use the root-raised cosine pulse from Computer Exercise 10.1. For a more generic orthogonal signaling, we use two pulse shapes of length T. Figure 10.46 shows these orthogonal pulses. Finally, Fig. 10.47 displays the measured BER for both signaling schemes against the BER obtained from analysis. It is not surprising that both measured results match the analytical BER very well.

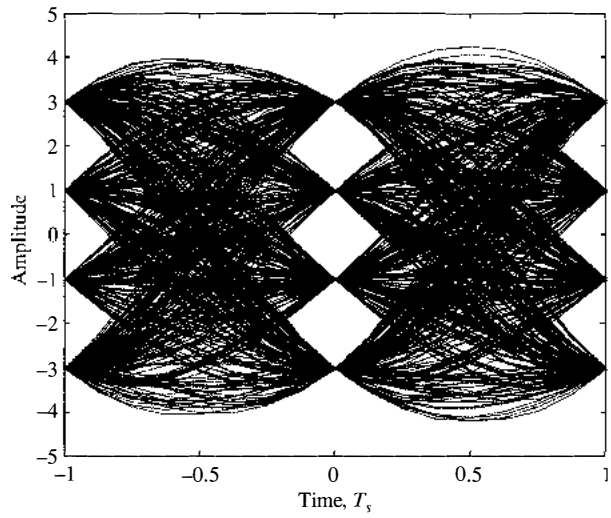


Figure 10.48 Eye diagram of the real (in-phase) component of the 16-QAM transmission at the receiver matched filter output.

COMPUTER EXERCISE 10.3: 16-QAM MODULATION

In this exercise, we will consider a more complex QAM constellation for transmission. The M-ary QAM was analyzed in Sec. 10.6.6. In MATLAB program `Ex10_3.m`, we control the transmission bandwidth by applying the root-raised cosine pulse with roll-off factor of 0.5 as the baseband pulse shape. For each symbol period T, eight uniform samples are used to approximate and emulate the continuous time signals. Figure 10.48 illustrates the open eye diagram of the in-phase (real) part of the matched filter output prior to being sampled. Very little ISI is observed at the point of sampling, validating the use of the root-raised cosine pulse shape in conjunction with the matched filter detector for ISI-free transmission.

```
% Matlab Program <Ex10_3.m>
% This Matlab exercise <Ex10_3.m> performs simulation of
% QAM-16 baseband polar transmission in AWGN channel.
% Root-raised cosine pulse of rolloff factor = 0.5 is used
% Matched filter receiver is designed to detect the symbols
% The program estimates the symbol error rate (BER) at different Eb/N
clear;clf;
L=1000000;              % Total data symbols in experiment is 1 million
% To display the pulse shape, we oversample the signal
% by factor of f_ovsamp=8
f_ovsamp=8;             % Oversampling factor vs data rate
delay_rc=4;
% Generating root-raised cosine pulseshape (rolloff factor = 0.5)
prcos=rcosflt([ 1 ], 1, f_ovsamp, 'sqrt', 0.5, delay_rc);
prcos=prcos(1:end-f_ovsamp+1);
prcos=prcos/norm(prcos);
pcmatch=prcos(end:-1:1);

% Generating random signal data for polar signaling
s_data=4*round(rand(L,1))+2*round(rand(L,1))-3+...
    +j*(4*round(rand(L,1))+2*round(rand(L,1))-3);
% upsample to match the
% 'oversampling rate'
```

```
% which is f_ovsamp/T   (T=1 is the symbol duration)
s_up=upsample(s_data,f_ovsamp);

% Identify the decision delays due to pulse shaping
% and matched filters
delayrc=2*delay_rc*f_ovsamp;
% Generate QAM-16 signaling with pulse-shaping
xrcos=conv(s_up,prcos);

% Find the signal length
Lrcos=length(xrcos);
SER=[];
noiseq=randn(Lrcos,1)+j*randn(Lrcos,1);
Es=10;                  % symbol energy
% Generating the channel noise (AWGN)
for i=1:9,
    Eb2N(i)=i*2;                          %(Eb/N in dB)
    Eb2N_num=10^(Eb2N(i)/10);           % Eb/N in numeral
    Var_n=Es/(2*Eb2N_num);               %1/SNR is the noise variance
    signois=sqrt(Var_n/2);                % standard deviation
    awgnois=signois*noiseq;             % AWGN
    % Add noise to signals at the channel output
    yrcos=xrcos+awgnois;

    % Apply matched filters first
    z1=conv(yrcos,pcmatch);clear awgnois, yrcos;

    % Sampling the received signal and acquire samples
    z1=z1(delayrc+1:f_ovsamp:end);

    % Decision based on the sign of the samples
    dec1=sign(real(z1(1:L)))+sign(real(z1(1:L))-2)+...
        sign(real(z1(1:L))+2)+...
        j*(sign(imag(z1(1:L)))+sign(imag(z1(1:L))-2)+...
        sign(imag(z1(1:L))+2));
    % Now compare against the original data to compute BER for
    % the three pulses
    %BER=[BER;sum(abs(s_data-dec1))/(2*L)]
    SER=[SER;sum(s_data~=dec1)/L];
    Q(i)=3*0.5*erfc(sqrt((2*Eb2N_num/5)/2));
%Compute the Analytical BER
end

figure(1)
subplot(111)
figber=semilogy(Eb2N,Q,'k-',Eb2N,SER,'b-*');
axis([2 18 .99e-5 1]);
legend('Analytical', 'Root-raised cosine');
xlabel('E_b/N (dB)');ylabel('Symbol error probability');
set(figber,'Linewidth',2);
% Constellation plot
figure(2)
subplot(111)
plot(real(z1(1:min(L,4000))),imag(z1(1:min(L,4000))),'.');
```

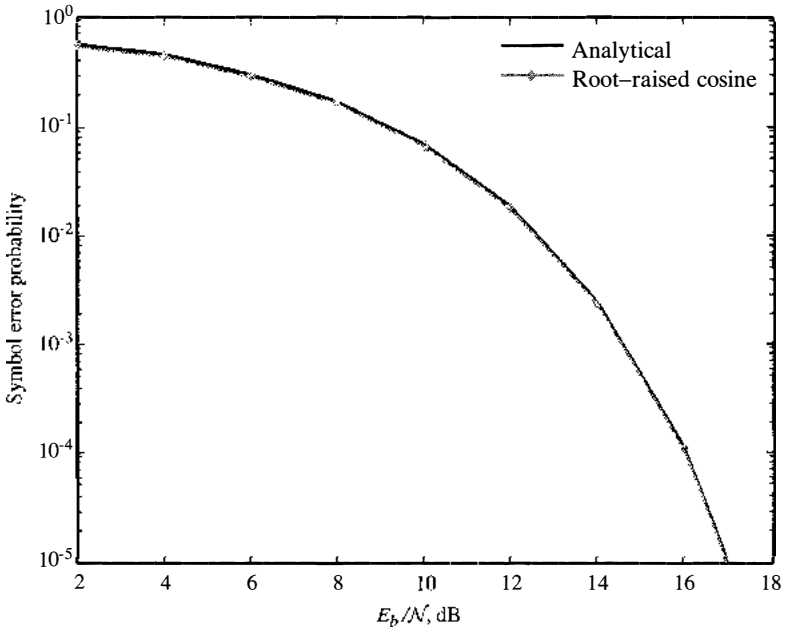


Figure 10.49 Symbol error probability of 16-QAM using root-raised cosine pulse in comparison with the analytical result.

```
axis('square')
xlabel('Real part of matched filter output samples')
ylabel('Imaginary part of matched filter output samples')
```

Because the signal uses 16-QAM constellations, instead of measuring the BER, we will measure the symbol error rate (SER) at the receiver. Figure 10.49 illustrates that the measured SER matches the analytical result from Sec. 10.6 very closely.

The success of the optimum QAM receiver can also be shown by observing the real part and the imaginary part of the samples taken at the matched filter output. By using a dot to represent each measured sample, we create what is known as a **"scatter plot,"** which clearly demonstrates the reliability of the decision that follows. If the dots in the scatter plot are closely clustered around the original constellation point, then the decision is mostly likely going to be reliable. Conversely, large number of decision errors can occur. Figure 10.50 illustrates the scatter plot from the measurement taken at the receiver when $E_b/\mathcal{N} = 18$ dB. The close clustering of the measured sample points is a strong indication that the resulting SER will be very low.

COMPUTER EXERCISE 10.4: NONCOHERENT FSK DETECTION

To test the results of a noncoherent binary FSK receiver, we provide MATLAB program `Ex10_4.m`, which assumes the orthogonality of the two frequencies used in FSK. As expected, the measured BER results in Figure 10.51 matches the analytical BER results very well.

```
% MATLAB PROGRAM <Ex10_4.m>
% This program provides simulation for noncoherent detection of
% orthogonal signaling including BFSK.  Noncoherent MFSK detection
% only needs to compare the magnitude of each frequency bin.
L=100000;          %Number of data symbols in the simulation
```

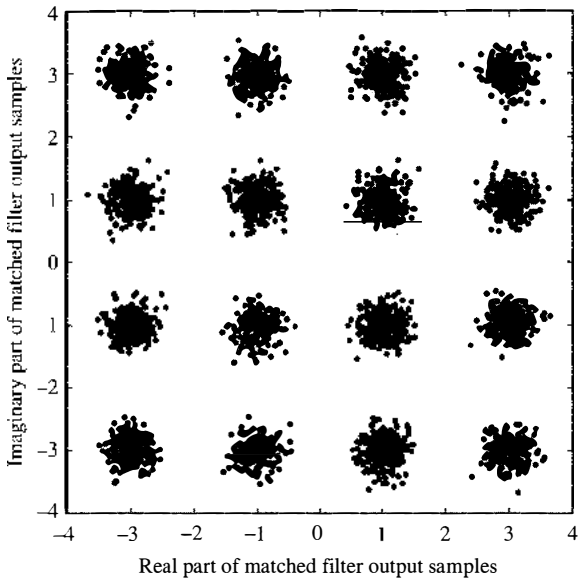


Figure 10.50 Scatter plot of the matched filter output for the 16-QAM signaling with root-raised cosine pulse when $E_b/\mathcal{N} = 18$ dB.

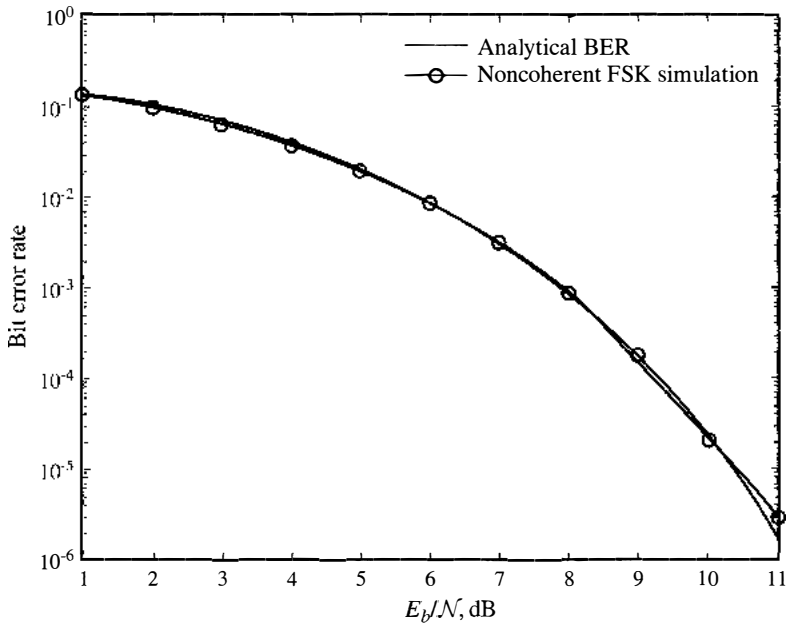


Figure 10.51 BER from noncoherent detection of binary FSK.

```
s_data=round(rand(L,1));
%  Generating random phases on the two frequencies
xbase1=[exp(j*2*pi*rand) 0];
xbase0=[0 exp(j*2*pi*rand)];
%  Modulating two orthogonal frequencies
xmodsig=s_data*xbase1+(1-s_data)*xbase0;
%  Generating noise sequences for both frequency channels
noisei=randn(L,2);
```

```
noiseq=randn(L,2);
BER=[];
BER_az=[];
% Generating the channel noise (AWGN)
for i=1:12,
    Eb2N(i)=i;                          %(Eb/N in dB)
    Eb2N_num=10^(Eb2N(i)/10);       % Eb/N in numeral
    Var_n=1/(2*Eb2N_num);        %1/SNR is the noise variance
    signois=sqrt(Var_n);            % standard deviation
    awgnois=signois*(noisei+j*noiseq);      % AWGN complex channels
    % Add noise to signals at the channel output
    ychout=xmodsig+awgnois;
    % Non-coherent detection
    ydim1=abs(ychout(:,1));
    ydim2=abs(ychout(:,2));
    dec=(ydim1>ydim2);
    % Compute BER from simulation
    BER=[BER; sum(dec~=s_data)/L];
    % Compare against analytical BER.
    BER_az=[BER_az; 0.5*exp(-Eb2N_num/2)];
end
figber=semilogy(Eb2N,BER_az,'k-',Eb2N,BER,'k-o');
set(figber,'Linewidth',2);
legend('Analytical BER', 'Noncoherent FSK simulation');
fx=xlabel('E_b/N (dB)');
fy=ylabel('Bit error rate');
set(fx,'FontSize',11); set(fy,'Fontsize',11);
```

COMPUTER EXERCISE 10.5: NONCOHERENT DETECTION OF BINARY DIFFERENTIAL PSK

To test the results of a binary differential phase shift keying system, we present MATLAB program `Ex10_5.m`. As in previous cases, the measured BER results in Figure 10.52 matches the analytical BER results very well.

```
% MATLAB PROGRAM <Ex10_5.m>
% This program provides simulation for differential detection of
% binary DPSK.  Differential detetion only needs to compare the
% successive phases of the signal samples at the receiver
%
clear;clf
L=1000000;         %Number of data symbols in the simulation
s_data=round(rand(L,1));
%  Generating initial random phase
initphase=[2*rand];
% differential modulation
s_denc=mod(cumsum([0;s_data]),2);
% define the phase divisible by pi
xphase=initphase+s_denc;
clear s_denc;
% modulate the phase of the signal
xmodsig=exp(j*pi*xphase); clear xphase;
```

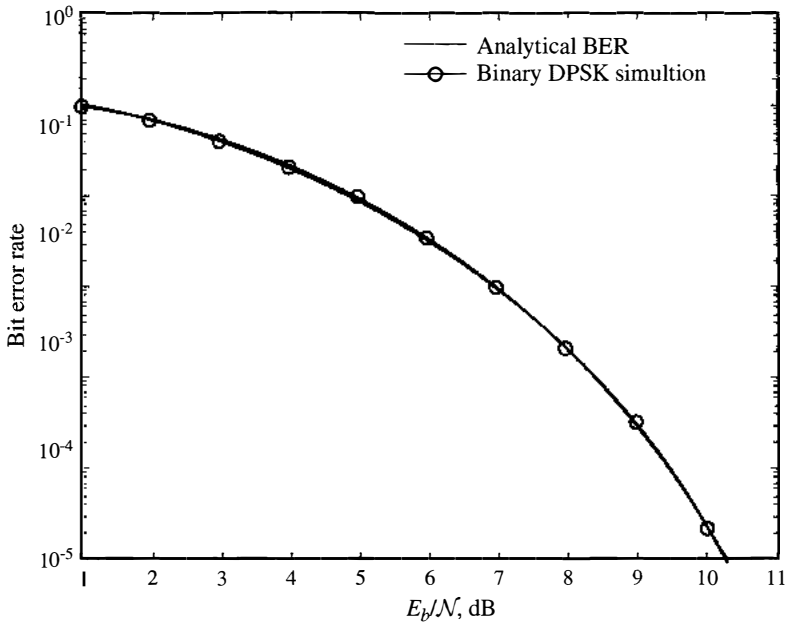


Figure 10.52 Analytical BER results from noncoherent detection of binary DPSK simulation (round points).

```
Lx=length(xmodsig);
%  Generating noise sequence
noiseq=randn(Lx,2);
BER=[];
BER_az=[];
% Generating the channel noise (AWGN)
for i=1:11,
    Eb2N(i)=i;                          %(Eb/N in dB)
    Eb2N_num=10^(Eb2N(i)/10);        % Eb/N in numeral
    Var_n=1/(2*Eb2N_num);         %1/SNR is the noise variance
    signois=sqrt(Var_n);             % standard deviation
    awgnois=signois*(noiseq*[1;j]);          % AWGN complex channels
    % Add noise to signals at the channel output
    ychout=xmodsig+awgnois;
    % Non-coherent detection
    yphase=angle(ychout);        %find the channel output phase
    clear ychout;
    ydfdec=diff(yphase)/pi;      %calculate phase difference
    clear yphase;
    dec=(abs(ydfdec)>0.5);       %make hard decisions
    clear ydfdec;
    % Compute BER from simulation
    BER=[BER; sum(dec~=s_data)/L];
    % Compare against analytical BER.
    BER_az=[BER_az; 0.5*exp(-Eb2N_num)];
end
% now plot the results
figber=semilogy(Eb2N,BER_az,'k-',Eb2N,BER,'k-o');
axis([1 11 .99e-5 1]);
set(figber,'Linewidth',2);
```

```
legend('Analytical BER', 'Binary DPSK simulation');
fx=xlabel('E_b/N (dB)');
fy=ylabel('Bit error rate');
set(fx,'FontSize',11); set(fy,'Fontsize',11);
```

REFERENCES

1. S. Pasupathy, "Minimum Shift Keying: A Spectrally Efficient Modulation," *IEEE Commun. Soc. Mag.*, vol. 17, pp. 14–22, July 1979.
2. J. J. Spilker, *Digital Communications by Satellite*, Prentice-Hall, Englewood Cliffs, NJ, 1977.
3. H. J. Landau and H. O. Pollak, "Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty, III: The Dimensions of Space of Essentially Time- and Band-Limited Signals," *Bell Syst. Tech. J.*, vol. 41, pp. 1295–1336, July 1962.
4. H. L. Van Trees, *Detection, Estimation, and Modulation Theory*, vols. I, II, and III, Wiley, New York, 1968–1971.
5. A. J. Viterbi, *Principles of Coherent Communication*, McGraw-Hill, New York, 1966.
6. H. J. Landau and D. Slepian, "On the Optimality of the Regular Simplex Code," *Bell Syst. Tech. J.*, vol. 45, pp. 1247–1272, Oct. 1966.
7. S. G. Wilson, *Digital Modulation and Coding*, Prentice Hall, Upper Saddle River, NJ, 1996.
8. A. V. Balakrishnan, "Contribution to the Sphere-Packing Problem of Communication Theory," *J. Math. Anal. Appl.*, vol. 3, pp. 485–506, Dec. 1961.
9. E. Arthurs and H. Dym, "On Optimum Detection of Digital Signals in the Presence of White Gaussian Noise—A Geometric Interpretation and a Study of Three Basic Data Transmission Systems," *IRE Trans. Commun. Syst.*, vol. CS-10, pp. 336–372, Dec. 1962.
10. B. P. Lathi, *An Introduction to Random Signals and Communication Theory*, International Textbook Co., Scranton, PA, 1968.

PROBLEMS

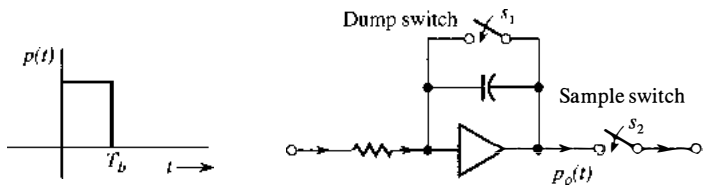
10.1-1 The so-called integrate-and-dump filter is shown in Fig. P10.1-1. The feedback amplifier is an ideal integrator. The switch s_1 closes momentarily and then opens at the instant $t = T_b$, thus dumping all the charge on C and causing the output to go to zero. The switch s_2 samples the output immediately before the dumping action.

(a) Sketch the output $p_o(t)$ when a square pulse $p(t)$ is applied to the input of this filter.

(b) Sketch the output $p_o(t)$ of the filter matched to the square pulse $p(t)$.

(c) Show that the performance of the integrate-and-dump filter is identical to that of the matched filter; that is, show that ρ in both cases is identical.

Figure P.10.1-1

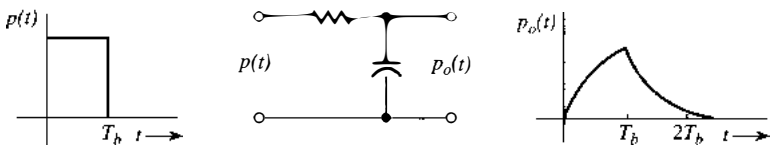


10.1-2 An alternative to the optimum filter is a suboptimum filter, where we assume a particular filter form and adjust its parameters to maximize ρ. Such filters are inferior to the optimum filter but may be simpler to design.

For a rectangular pulse $p(t)$ of height A and width T_b at the input (Fig. P10.1-2), determine $\rho_{\max}$ if, instead of the matched filter, a one-stage RC filter with $H(\omega) = 1/(1 + j\omega Rc)$ is used. Assume a white Gaussian noise of PSD $\mathcal{N}/2$. Show that the optimum performance is achieved when $1/RC = 1.26/T_b$.

Hint: Set $d\rho^2/dx = 0$ $(x = T_b/RC)$.

Figure P.10.1-2



(a) (b) (c)

10.2-1 In coherent detection of a binary PPM, a half-width pulse $p_0(t)$ of is transmitted with different delays for binary digit "0" and "1" over $0 \leq t \leq T_b$. Note that

$$p_0(t) = u(t) - u(t - T_b/2)$$

The binary PPM transmission is to simply transmit

$$\begin{array}{ll} p_0(t), & \text{if "0" is sent} \\ p_0(t - T_b/2), & \text{if "1" is sent} \end{array}.$$

The channel noise is AWGN with spectrum level of $\mathcal{N}/2$.

(a) Determine the optimum receiver architecture for this binary system. Sketch the optimum receiver filter response in the time domain.

(b) If $P[\text{"0"}] = 0.4$ and $P[\text{"1"}] = 0.6$, find the optimum threshold and the resulting receiver bit error rate.

(c) The receiver was misinformed and believes that $P[\text{"0"}] = 0.5 = P[\text{"1"}]$. It hence designed a receiver based on this information. Find the true probability of error when, in fact, the actual prior probabilities are $P[\text{"0"}] = 0.4$ and $P[\text{"1"}] = 0.6$. Compare this result with the result in part **(b)**.

10.2-2 In the coherent detection of binary chirp modulations, the transmission over $0 \leq t \leq T_b$ is

$$\begin{array}{ll} A\cos(\alpha_0 t^2 + \theta_0), & \text{if "0" is sent} \\ A\cos(\alpha_1 t^2 + \theta_1), & \text{if "1" is sent} \end{array}.$$

The channel noise is AWGN with spectrum $\mathcal{N}/2$. The binary digits are equally likely.

(a) Design the optimum receiver.

(b) Find the probability of bit error for the optimum receiver in part **(a)**.

10.2-3 In coherent schemes, a small pilot is added for synchronization. Because the pilot does not carry information, it causes degradation in P_b. Consider a coherent PSK that uses the following two pulses of duration T_b:

$$p(t) = A\sqrt{1 - m^2}\cos\omega_c t + A\,m\sin\omega_c t$$

$$q(t) = -A\sqrt{1 - m^2}\cos\omega_c t + A\,m\sin\omega_c t$$

where $A\,m \sin \omega_c t$ is the pilot. Show that when the channel noise is white Gaussian,

$$P_b = Q\left[\sqrt{\frac{2E_b(1-m^2)}{\mathcal{N}}}\right]$$

Hint: Use Eq. (10.25b).

10.2-4 For polar binary communication systems, each error in the decision has some cost. Suppose that when m = **1** is transmitted and we read it as m = **0** at the receiver, a quantitative penalty, or cost, C_{10} is assigned to such an error, and, similarly, a cost C_{01} is assigned when m = **0** is transmitted and we read it as m = **1**. For the polar case where $P_{\mathrm{m}}(\mathbf{0}) = P_{\mathrm{m}}(\mathbf{1}) = 0.5$, show that for white Gaussian channel noise, the optimum threshold that minimizes the overall cost is not 0 but is a_o, given by

$$a_o = \frac{\mathcal{N}}{4} \ln \frac{C_{01}}{C_{10}}$$

Hint: See Hint for Prob. 8.2-11.

10.2-5 For a polar binary system with unequal message probabilities, show that the optimum decision threshold a_o is given by

$$a_o = \frac{\mathcal{N}}{4} \ln \frac{P_{\mathrm{m}}(\mathbf{0})C_{01}}{P_{\mathrm{m}}(\mathbf{1})C_{10}}$$

where C_{01} and C_{10} are the cost of the errors as explained in Prob. 10.2-4, and $P_{\mathrm{m}}(\mathbf{0})$ and $P_{\mathrm{m}}(\mathbf{1})$ are the probabilities of transmitting **0** and **1**, respectively.

Hint: See Hint for Prob. 8.2-11.

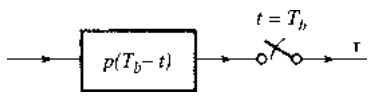
10.2-6 For 4-ary communication, messages are chosen from any one of four message symbols, $m_1 =$ 00, $m_2 = 01$, $m_3 = 10$, and $m_4 = 11$, which are transmitted by pulses $\pm p(t)$, 0, and $\pm 3p(t)$, respectively. A filter matched to $p(t)$ is used at the receiver. Denote the energy of $p(t)$ as E_p. The channel noise is AWGN with spectrum $\mathcal{N}/2$.

(a) If r is the matched filter output at t_m, plot $p_{\mathrm{r}}(r|m_i)$ (00, 01, 10, and 11) for the four message symbols, assuming that all message symbols are equally likely.

(b) To minimize the probability of detection error in part **(a)**, determine the optimum decision thresholds and the corresponding error probability P_e as a function of the average symbol energy to noise ratio.

10.2-7 Binary data is transmitted by using a pulse $p(t)$ for **0** and a pulse $\gamma p(t)$ for **1**. Let $\gamma > 1$. Show that the optimum receiver for this case consists of a filter matched to $p(t)$ plus a detection threshold as shown in Fig. P10.2-7. Determine the error probability P_b of this receiver as a function of $E_b/\mathcal{N}$ if **0** and **1** are equiprobable.

Figure P.10.2-7



10.2-8 In a binary transmission, a raised-cosine roll-off pulse $p(t)$ with roll-off factor 0.2 is used for baseband polar transmission. The ideal low-pass channel has a bandwidth of $f_0 = 5000$ Hz.

(a) If the channel noise is AWGN with spectrum $\mathcal{N}/2$, find the optimum receiver filter and sketch its frequency response.

(b) If the channel noise is Gaussian with spectrum

$$S_n(f) = 0.5\mathcal{N}\frac{1}{1 + (f/f_0)^2}$$

find the optimum receiver filter and sketch its frequency response.

10.3-1 In an FSK system, RF binary signals are transmitted as

$$\begin{aligned} \mathbf{0}: &\quad \sqrt{2}\sin(\pi t/T_b)\cos[\omega_c - (\Delta\omega/2)]t \qquad 0 \le t \le T_b \\ \mathbf{1}: &\quad \sqrt{2}\sin(\pi t/T_b)\cos[\omega_c + (\Delta\omega/2)]t \qquad 0 \le t \le T_b \end{aligned}$$

The channel noise is AWGN. Let the binary inputs be equally likely.

(a) Derive the optimum coherent receiver and the optimum threshold.

(b) Find the minimum probability of bit error.

(c) Is the possible to find the optimum $\Delta\omega$ to minimize the probability of bit error?

10.4-1 Consider four signals in the time interval $(0, T)$:

$$\begin{aligned} p_0(t) &= u(t) - u(t - T) \\ p_1(t) &= \sin(2\pi t/T)[u(t) - u(t - T)] \\ p_2(t) &= \sin(\pi t/T)[u(t) - u(t - T)] \\ p_3(t) &= \cos(\pi t/T)[u(t) - u(t - T)] \end{aligned}$$

Apply the Gram-Schmidt procedure and find a set of orthonormal basis signals for this signal space. What is the dimension of this signal space?

10.4-2 The basis signals of a three-dimensional signal space are given by $\varphi_1(t) = p(t)$, $\varphi_2(t) = p(t - T_o)$, and $\varphi_3(t) = p(t - 2T_o)$, where

$$p(t) = \sqrt{\frac{2}{T_o}}\sin\left(\frac{\pi t}{T_o}\right)[u(t) - u(t - T_o)]$$

(a) Sketch the waveforms of the signals represented by $(1, 1, 1), (-2, 0, 1), (1/3, 2, -\frac{1}{2})$, and $(-\frac{1}{2}, -1, 2)$ in this space.

(b) Find the energy of each signal in part **(a)**.

10.4-3 Repeat Prob. 10.4-2 if

$$\varphi_1(t) = \frac{1}{\sqrt{T_o}}$$

$$\varphi_2(t) = \sqrt{\frac{2}{T_o}}\cos\frac{\pi}{T_o}t \quad \varphi_3(t) = \sqrt{\frac{2}{T_o}}\cos\frac{2\pi}{T_o}t \qquad 0 \le t \le T_o$$

10.4-4 For the three basis signals given in Prob. 10.4-3, assume that a signal is written as

$$x(t) = 1 + 2\sin^3\left(\frac{\pi t}{T_o}\right)$$

(a) Use the three basis signals in terms of minimum error energy to find the best approximation of $x(t)$. What is the minimum approximation error energy?

(b) By adding another basis signal

$$\varphi_4(t) = \sqrt{\frac{2}{T_o}}\sin\frac{\pi}{T_o}t \qquad 0 \le t \le T_o$$

find the reduction of minimum approximation error energy.

10.4-5 Assume that $p(t)$ is as in Prob. 10.4-2 and

$$\varphi_k(t) = p[t - (k-1)T_o] \qquad k = 1,2,3,4,5$$

(a) Sketch the signals represented by $(-1, 2, 3, 1, 4), (2, 1, -4, -4, 2), (3, -2, 3, 4, 1)$, and $(-2, 4, 2, 2, 0)$ in this space.

(b) Find the energy of each signal.

(c) Find the angle between all pairs of the signals.

Hint: Recall that the inner product between vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is related to the angle θ between the two vectors via $< \boldsymbol{a}, \boldsymbol{b} >= ||\boldsymbol{a}|| \cdot ||\boldsymbol{b}|| \cos(\theta)$.

10.5-1 Assume that $p(t)$ is as in Prob. 10.4-2 and

$$s_k(t) = p[t - (k-1)T_o] \qquad k = 1,2,3,4,5$$

When $s_k(t)$ is transmitted, the received signal under noise $\mathrm{n}_w(t)$ is

$$\mathrm{y}(t) = s_k(t) + \mathrm{n}_w(t) \qquad 0 \le t \le 5T_o$$

Given a noise $n_w(t)$ that is white Gaussian with spectrum $\mathcal{N}/2$, complete the following.

(a) Define a set of basis functions for $y(t)$ such that

$$E\{|\mathrm{y}(t) - \sum \mathrm{y}_i\varphi_i(t)|^2\} = 0$$

(b) Characterize the random variable y_i when $s_k(t)$ is transmitted.

(c) Determine the joint probability density function of random variable $\{\mathrm{y}_1, \ldots, \mathrm{y}_5\}$ when $s_k(t)$ is transmitted.

10.5-2 For a certain stationary Gaussian random process $\mathrm{x}(t)$, it is given that $R_{\mathrm{X}}(\tau) = e^{-\tau^2}$. Determine the joint PDF of RVs $\mathrm{x}(t)$, $\mathrm{x}(t + 0.5)$, $\mathrm{x}(t + 1)$, and $\mathrm{x}(t + 2)$.

10.5-3 A Gaussian noise is characterized by its mean and its autocorrelation function. A stationary Gaussian noise $\mathrm{x}(t)$ has zero mean and autocorrelation function $R_{\mathrm{X}}(\tau)$.

(a) If $\mathrm{x}(t)$ is the input to a linear time-invariant system with impulse response $h(t)$, determine the mean and the autocorrelation function of the linear system output $\mathrm{y}(t)$.

(b) If $x(t)$ is the input to a linear time-varying system whose output is

$$y(t) = \int_{-\infty}^{\infty} h(t, \tau)x(\tau)\, d\tau$$

show what kind of output process this generates, and determine the mean and the autocorrelation function of the linear system output $y(t)$.

10.5-4 Determine the output PSD of the linear system in part **(a)** of Prob. 10.5-3.

10.5-5 Determine the output PSD of the linear system in part **(b)** of Prob. 10.5-3.

10.6-1 Consider the preprocessing of Fig. 10.17. The channel noise $n_w(t)$ is white Gaussian.

(a) Find the signal energy of $r(t)$ and $q(t)$ over the finite time interval $[0, T_M]$.

(b) Prove that although $r(t)$ and $q(t)$ are not equal, both contain all the useful signal content.

(c) Show that the joint probability density function of $(q_1, q_2, \ldots, q_N)$, under the condition that $s_k(t)$ is transmitted, can be written as

$$p_{\mathbf{q}}(\boldsymbol{q}) = \frac{1}{(\pi\mathcal{N})^{N/2}} \exp\left(-||\boldsymbol{q} - \boldsymbol{s}_k||^2/\mathcal{N}\right).$$

10.6-2 Consider an additive white noise channel. After signal projection, the received $N \times 1$ signal vector is given by

$$\mathbf{q} = \boldsymbol{s}_i + \mathbf{n}$$

when message m_i is transmitted. The noise vector $\mathbf{n}$ has joint probability density function

$$\prod_{i=1}^{N} \frac{1}{\tau} \exp\left(\frac{-|n_i|}{(2\tau)}\right)$$

(a) Find the (MAP) detector that can minimize the probability of detection error.

(b) Follow the derivations of optimum detector for AWGN noise to derive the optimum receiver structure for this non-Gaussian white noise channel.

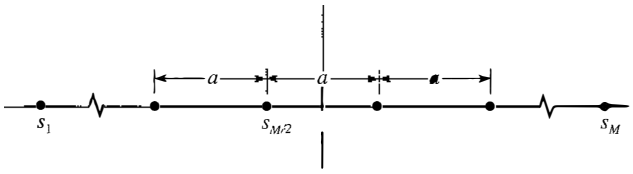
(c) Show how the decision regions are different between Gaussian and non-Gaussian noises in a two-dimensional ($N = 2$) signal space.

10.6-3 A binary source emits data at a rate of 400,000 bit/s. Multiamplitude shift keying (PAM) with $M = 2$, 16, and 32 is considered. In each case, determine the signal power required at the receiver input and the minimum transmission bandwidth required if $S_n(\omega) = 10^{-8}$ and the bit error rate P_b, is required to be less than 10^{-6}.

10.6-4 Repeat Prob. 10.6-3 for M-ary PSK.

10.6-5 A source emits M equiprobable messages, which are assigned signals s_1, s_2, ..., s_M, as shown in Fig. P10.6-5. Determine the optimum receiver and the corresponding error probability P_{eM} for an AWGN channel as a function of $E_b/\mathcal{N}$.

Figure P.10.6-5

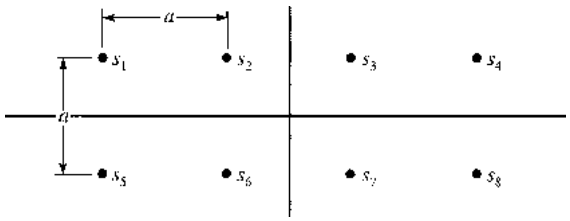


10.6-6 A source emits eight equiprobable messages, which are assigned QAM signals $s_1, s_2, \ldots, s_8$, as shown in Fig. Pl 0.6-6.

(a) Find the optimum receiver for an AWGN channel.

(b) Determine the decision regions and the error probability P_{eM} of the optimum receiver as a function of E_b.

Figure P.10.6-6



10.6-7 Prove that for $E_b/\mathcal{N} \gg 1$ and $M \gg 2$, the error probability approximation of Eq. (10.109b) for MPSK holds.

10.6-8 Use the approximation of Eq. (10.109b) for 16 PSK to compare the symbol error probabilities of 16-QAM and 16-PSK. Show approximately how many decibels of $E_b/\mathcal{N}$ (SNR) loss 16-PSK incurs versus 16-QAM (by ignoring the constant difference in front of the $Q-$ function).

10.6-9 Compare the symbol error probabilities of 16-PAM, 16-PSK, and 16-QAM. Sketch them as functions of $E_b/\mathcal{N}$.

10.6-10 Show that for MPSK, the optimum receiver of the form in Fig. 10.19a is equivalent to a phase comparator. Assume all messages equiprobable and an AWGN channel.

10.6-11 A ternary signaling has three signals for transmission:

$$m_o : 0, \qquad m_1 : 2p(t), \qquad m_2 : -2p(t).$$

(a) If $P(m_o) = P(m_1) = P(m_2) = 1/3$, determine the optimum decision regions and P_{eM} of the optimum receiver as a function of $\overline{E}$. Assume an AWGN channel.

(b) Find P_{eM} as a function of $\overline{E}/\mathcal{N}$.

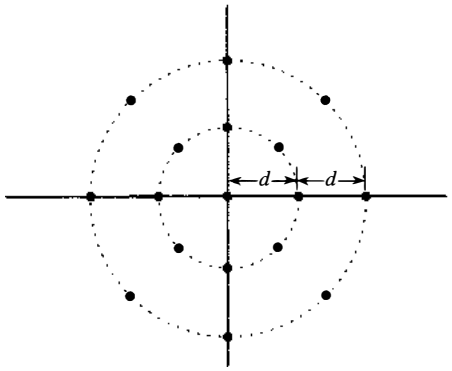
(c) Repeat parts **(a)** and **(b)** if $P(m_o) = 1/2$ and $P(m_1) = P(m_2) = 0.25$,

10.6-12 A 16-ary signal configuration is shown in Fig. P10.6-12. Write the expression (do not evaluate various integrals) for the P_{eM} of the optimum receiver, assuming all symbols to be equiprobable. Assume an AWGN channel.

10.6-13 A five-signal configuration in a two-dimensional space is shown in Fig. P10.6-13.

(a) Choose the $\varphi_1(t) = \sqrt{2/T_o}\cos\omega_c t$ and $\varphi_2(t) = \sqrt{2/T_o}\sin\omega_c t$ and sketch the waveforms of the five signals.

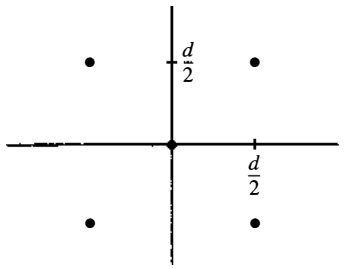
Figure P.10.6-12



(b) In the signal space, sketch the optimum decision regions, assuming an AWGN channel.

(c) Determine the error probability P_{eM} as a function of $\overline{E}$ of the optimum receiver.

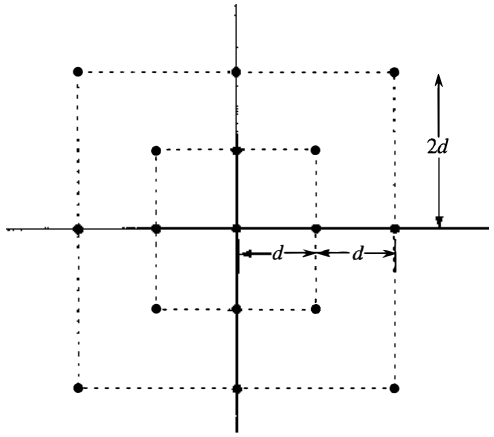
Figure P.10.6-13



10.6-14 A 16-point QAM signal configuration is shown in Fig. P10.6-14. Assuming that all symbols are equiprobable, determine the error probability P_{eM} as a function of E_b of the optimum receiver for an AWGN channel.

Compare the performance of this scheme with the result of rectangular 16-point QAM in Sec. 10.6.

Figure P.10.6-14



10.7-1 The vertices of an N-dimensional hypercube are a set of 2^N signals

$$s_k(t) = \frac{d}{2} \sum_{j=1}^{N} a_{kj} \varphi_j(t)$$

where $\{\varphi_1(t), \varphi_2(t), \ldots, \varphi_N(t)\}$ is a set of N orthonormal signals, and a_{kj} is either 1 or -1. Note that all the N signals are at a distance of $\sqrt{N}d/2$ from the origin and form the vertices of the N-dimensional cube.

(a) Sketch the signal configuration in the signal space for $N = 1$, 2, and 3.

(b) For each configuration in part **(a)**, sketch one possible set of waveforms.

(c) If all the 2^N symbols are equiprobable, find the optimum receiver and determine the error probability P_{eM} of the optimum receiver as a function of E_b assuming an AWGN channel.

10.7-2 An orthogonal signal set is given by

$$s_k(t) = \sqrt{E}\, \varphi_k(t) \qquad k = 1, 2, \ldots, N$$

A biorthogonal signal set is formed from the orthogonal set by augmenting it with the negative of each signal. Thus, we add to the orthogonal set another set

$$s_{-k}(t) = -\sqrt{E}\, \varphi_k(t)$$

This gives $2N$ signals in an N-dimensional space. Assuming all signals to be equiprobable and an AWGN channel, obtain the error probability of the optimum receiver. How does the bandwidth of the biorthogonal set compare with that of the orthogonal set?

10.8-1 **(a)** What is the minimum energy equivalent signal set of a binary on-off signal set?

(b) What is the minimum energy equivalent signal set of a binary FSK signal set?

(c) Using geometrical signal space concepts, explain why the binary on-off and the binary orthogonal sets have identical error probabilities and why the binary polar energy requirements are 3 dB lower than those of the on-off or the orthogonal set.

10.8-2 A source emits four equiprobable messages m_1, m_2, m_3, and m_4, encoded by signals $s_1(t)$, $s_2(t)$, $s_3(t)$, and $s_4(t)$, respectively, where

$$\left.\begin{aligned} s_1(t) &= 20\sqrt{2} \sin \tfrac{2\pi}{T_M} t \\ s_2(t) &= 0 \\ s_3(t) &= 10\sqrt{2} \cos \tfrac{2\pi}{T_M} t \\ s_4(t) &= -10\sqrt{2} \cos \tfrac{2\pi}{T_M} t \end{aligned}\right\} \qquad T_M = \frac{1}{20}$$

Each of these signal durations is $0 \le t \le T_M$ and is zero outside this interval. The signals are transmitted over AWGN channels.

(a) Represent these signals in a signal space.

(b) Determine the decision regions.

(c) Obtain an equivalent minimum energy signal set.

(d) Determine the optimum receiver.

10.8-3 A quaternary signaling scheme uses four waveforms,

$$s_1(t) = 4\,\varphi_1(t)$$
$$s_2(t) = 2\varphi_1(t) + 2\,\varphi_2(t)$$
$$s_3(t) = -2\,\varphi_1(t) - 2\varphi_2(t)$$
$$s_4(t) = -4\,\varphi_2(t)$$

where $\varphi_1(t)$ and $\varphi_2(t)$ are orthonormal basis signals. All the signals are equiprobable, and the channel noise is white Gaussian with PSD $S_n(\omega) = 10^{-4}$.

(a) Represent these signals in the signal space, and determine the optimum decision regions.

(b) Compute the error probability of the optimum receiver.

(c) Find the minimum energy equivalent signal set.

(d) Determine the amount of average energy reduction of the minimum energy equivalent signal set is transmitted.

10.8-4 An $M = 4$ orthogonal signaling system uses $\sqrt{E}\cdot\varphi_1(t)$, $-\sqrt{E}\cdot\varphi_2(t)$, $\sqrt{E}\cdot\varphi_3(t)$, and $\sqrt{E}\cdot\varphi_4(t)$ in its transmission.

(a) Find the minimum energy equivalent signal set.

(b) Sketch the minimum energy equivalent signal set in three-dimensional space.

(c) Determine the amount of average energy reduction by using the minimum energy equivalent signal set.

10.8-5 A ternary signaling scheme ($M = 3$) uses the three waveforms:

$$s_1(t) = [u(t) - u(t - T_0/3)]$$
$$s_2(t) = u(t) - u(t - T_0)$$
$$s_3(t) = -[u(t - 2T/3) - u(t - T_0)]$$

The transmission rate is $1/T_0 = 200$ kilosymbols per second. All three messages are equiprobable, and the channel noise is white Gaussian with PSD $S_n(\omega) = 2 \times 10^{-6}$.

(a) Determine the decision regions of the optimum receiver.

(b) Determine the minimum energy signal set and sketch the waveforms.

(c) Compute the mean energies of the signal set and its minimum energy equivalent set, found in part **(b)**.

10.8-6 Repeat Prob. 10.8-5 if $P(m_1) = 0.5$, $P(m_2) = 0.25$, and $P(m_3) = 0.25$.

10.8-7 A binary signaling scheme uses the two waveforms

$$s_1(t) = \text{rect}\left(\frac{t - 0.001}{0.002}\right) \quad \text{and} \quad s_2(t) = -\Delta\left(\frac{t - 0.001}{0.002}\right)$$

(see Chapter 3 for the definitions of these signals). The signaling rate is 1000 pulses per second. Both signals are equally likely, and the channel noise is white Gaussian with PSD $S_n(\omega) = 2 \times 10^{-4}$.

(a) Determine the minimum energy equivalent signal set.

(b) Determine the error probability of the optimum receiver.

(c) Use a suitable orthogonal signal space to represent these signals as vectors.

Hint: Use Gram-Schmidt orthogonalization to determine the appropriate basis signals $\varphi_1(t)$ and $\varphi_2(t)$.

10.10-1 In a binary transmission with messages m_0 and m_1, the costs are defined as

$$C_{00} = C_{11} = 1 \qquad \text{and} \qquad C_{01} = C_{10} = 4$$

The two messages are equally likely. Determine the optimum Bayes receiver.

10.10-2 In a binary transmission with messages m_0 and m_1, the cost are defined as

$$C_{00} = C_{11} = 0 \qquad \text{and} \qquad C_{01} = C_{10} = C$$

The probability of m_0 is 1/3 and the probability of m_1 is 2/3.

(a) Determine the optimum Bayes receiver.

(b) Determine the minimum probability of error receiver.

(c) Determine the maximum likelihood receiver.

(d) Compare the probability of error between the two receivers in parts **(b)** and **(c)**.

10.11-1 Plot and compare the probabilities of error for the non coherent detection of binary ASK, binary FSK, and binary DPSK.

10.11-2 Derive the probability of symbol error for different representations of QPSK signaling.

11 SPREAD SPECTRUM COMMUNICATIONS

In traditional digital communication systems, the design of baseband pulse-shaping and modulation techniques aims to minimize the amount of bandwidth consumed by the modulated signal during transmission. This principal objective is clearly motivated by the desire to achieve good spectral efficiency and thus to conserve bandwidth resource. Nevertheless, a narrowband digital communication system exhibits two major weaknesses. First, its concentrated spectrum makes it an easy target for detection and interception by unintended users (e.g., battlefield enemies and unauthorized eavesdroppers). Second, its narrow band, having very little redundancy, is more susceptible to jamming, since even a partial band jamming can ruin the signal reception.

Spread spectrum technologies were initially developed for the military and intelligence communities to overcome the two aforementioned shortcomings against interception and jamming. The basic idea was to expand each user signal to occupy a much broader spectrum than necessary. For fixed transmission power, a broader spectrum means both lower signal power level and higher spectral redundancy. The low signal power level makes the communication signals difficult to detect and intercept, whereas high spectral redundancy makes the signals more resistant to partial band jamming, whether intentional or unintentional.

There are two dominant spread spectrum technologies: frequency hopping spread spectrum (FHSS) and direct sequence spread spectrum (DSSS). In this chapter, we provide detailed descriptions on both systems.

11.1 FREQUENCY HOPPING SPREAD SPECTRUM (FHSS) SYSTEMS

The concept of frequency hopping spread spectrum (FHSS) is in fact quite simple and easy to understand. Each user can still use its conventional modulation. The only difference is that now the carrier frequency can vary over regular intervals. When each user can vary its carrier frequency according to a predetermined, pseudorandom pattern, its evasive signal effectively occupies a broader spectrum and becomes harder to intercept and jam.

The implementation of an FHSS system is shown in Fig. 11.1. If we first ignore the two frequency converters, this system is no different from a simple digital communication system with an FSK modulator and a demodulator. The only difference in this FHSS system lies in

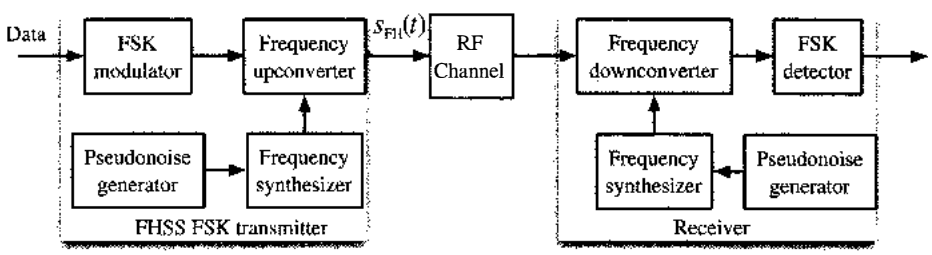


Figure 11.1 Frequency hopping spread spectrum system.

the *carrier* frequency hopping controlled at the transmitter by the pseudonoise (PN) generator. To track the hopping carrier frequency, the receiver must utilize the same PN generator in synchronization with the transmitter PN generator.

We note that most FHSS signals adopt binary or M-ary FSK modulations instead of the more efficient PAM, PSK, or QAM. The motivation for choosing FSK stems from its ability to utilize the less complex noncoherent detection. In contrast, coherent detection is generally needed for PAM, PSK, and QAM modulations. Due to the PN hopping pattern, coherent detection would require the receiver to maintain phase coherence with the transmitter at every one of the frequencies used in the hopping pattern. Such requirement would be difficult to satisfy during frequency hopping. On the other hand, FSK detection can be noncoherent without the need for carrier phase coherence and can be easily incorporated into FHSS systems.

The frequency upconverter, as discussed in Example 4.2 of Chapter 4, can be a mixer or a multiplier followed by a bandpass filter. Denote T_s as the symbol period. Then the M-ary FSK modulation signal can be written as

$$s_{\text{FSK}}(t) = A\cos(\omega_m t + \phi_m) \qquad mT_s \le t \le (m+1)T_s \tag{11.1a}$$

in which the M-ary FSK angular frequencies are specified by

$$\omega_m = \omega_c \pm \frac{1}{2}\Delta\omega,\ \omega_c \pm \frac{3}{2}\Delta\omega, \ldots,\ \omega_c \pm \frac{M-1}{2}\Delta\omega \tag{11.1b}$$

The frequency synthesizer output is constant for a period of T_c often known as a "chip." If we denote the frequency synthesizer output as ω_h in a given chip, then the FHSS signal is

$$s_{\text{FH}}(t) = A\cos[(\omega_h + \omega_m)t + \phi_m] \tag{11.2}$$

for the particular chip period T_c. The frequency hopping pattern is controlled by the PN generator and typically looks like Fig. 11.2. At the receiver, an identical PN generator enables the receiver to detect the FHSS signal within the correct frequency band (i.e., the band the signal has hopped to). If the original FSK signal only has bandwidth B_s Hz, then the FHSS signal will occupy a bandwidth L times larger

$$B_c = L \cdot B_s$$

This factor L is known as the spreading factor.

For symbol period T_s and chip period T_c, the corresponding symbol rate is $R_s = 1/T_s$ and the hopping rate is $R_c = 1/T_c$. There are two types of frequency hopping in FHSS. If $T_c \ge T_s$, then the FH is known as slow hopping. If $T_c < T_s$, it is known as fast FHSS, and there are multiple hops within each data symbol. In other words, under fast hopping, each data symbol

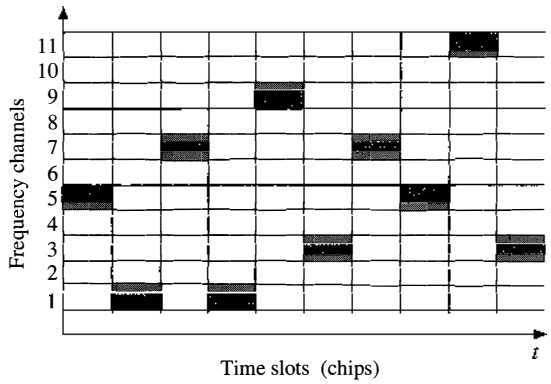


Figure 11.2 Typical Bluetooth frequency hopping pattern.

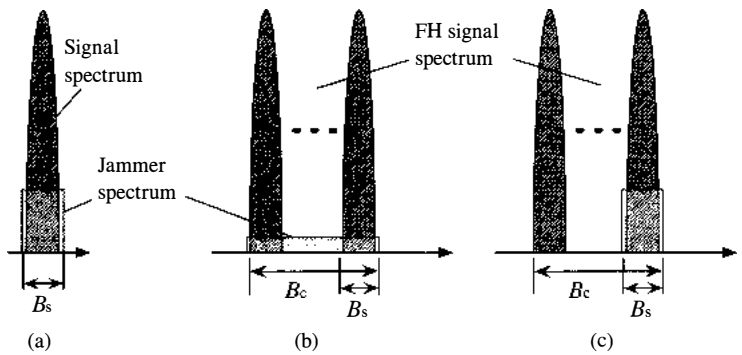


Figure 11.3 Effects of (a) narrowband jamming, (b) FHSS under broadband jamming, and (c) partial band jamming.

is spread across multiple frequency bands because of the fast hopping, and must be detected by detection over these frequency bands.

One major advantage of FHSS lies in its ability to combat jamming. Suppose a jamming source has a finite level of jamming power P_J. Against a narrowband signal with bandwidth B_s, the jamming source can transmit within B_s at all time, creating an interference PSD level of P_J/B_s. Hence, the signal-to-interference ratio (SIR) for the narrowband (NB) transmission is

$$\left(\frac{E_b}{I}\right)_{\text{NB}} = \frac{E_b}{P_J/B_s} = \frac{E_b}{P_J}B_s \tag{11.3a}$$

On the other hand, against FHSS signal with total bandwidth of B_c, the jamming source must divide its limited power and will generate a much lower level of interference PSD with average value P_J/B_c. As a result, at any given time, the signal bandwidth is still B_s and the SIR is

$$\left(\frac{E_b}{I}\right)_{\text{FH}} = \frac{E_b}{P_J/B_c} = \frac{E_b}{P_J}B_c = L \cdot \left(\frac{E_b}{I}\right)_{\text{NB}} \tag{11.3b}$$

Therefore, with a spreading factor of L, an FH signal is L times more resistant to a jamming signal with finite power than a narrowband transmission. Figure 11.3a and b illustrate the different effects of the finite jamming power on narrowband and FHSS signals.

On the other hand, the jammer may decide to concentrate all its power P_J in a narrow signal bandwidth against FHSS. This will achieve partial band jamming. If the frequency hopping is slow, such that $T_c = T_s$, then on average one out of every L user symbols will encounter

the strong interference, as shown in Fig. 11.3c, Consider BFSK. We can assume a very strong interference such that the bits transmitted in the jammed frequency band have the worst BER of 0.5. Then, after averaging of the L bands, the total BER of this partially jammed FHSS system will be

$$P_b = \frac{L-1}{L} \cdot \frac{1}{2} \exp\left(-\frac{E_b}{2\mathcal{N}}\right) + \frac{1}{L} \cdot \frac{1}{2} \geq \frac{1}{2L} \tag{11.4}$$

Thus, the partially jammed FHSS signal detection has rather high BER under slow hopping. By employing a strong enough forward error correction (FEC) codes, to be discussed in Chapter 14, such data errors can be corrected by the receiver.

Example 11.1 Consider the case of a fast hopping system in which $T_c \ll T_s$. There are L frequency bands for this FHSS system. Assume that a jamming source jams one of the L bands. Let the number of hops per T_s be less than L and no frequency is repeated in each T_s. Derive the BER performance of a fast hopping BFSK system under this partial jamming.

With fast hopping, each user symbol hops over

$$L_h \triangleq T_s/T_c \quad Lh \leq L$$

narrow bands. Hence on average, a user symbol will encounter partial jamming with a probability of L_h/L. When a BFSK symbol does not encounter partial jamming during hopping, its BER remains unchanged. If a BFSK symbol does encounter partial band jamming, we can approximate its BER performance by discarding the energy in the jammed band. In other words, we can approximate the BFSK symbol performance under jamming by letting its useful signal energy be

$$\frac{L_h - 1}{L_h} \cdot E_b$$

Thus, on average, the BFSK performance under fast hopping consists of statistical average of the two types of BFSK bits:

$$\begin{aligned} P_b &= \frac{1}{2} \exp\left(-\frac{E_b}{2\mathcal{N}}\right) \cdot \left(1 - \frac{L_h}{L}\right) + \frac{1}{2} \exp\left(-\frac{E_b}{2\mathcal{N}} \cdot \frac{L_h - 1}{L_h}\right) \cdot \frac{L_h}{L} \\ &= \frac{1}{2}\left(1 - \frac{T_s}{LT_c}\right) \exp\left(-\frac{E_b}{2\mathcal{N}}\right) + \frac{1}{2}\left(\frac{T_s}{LT_c}\right) \exp\left(-\frac{E_b}{2\mathcal{N}} \cdot \frac{T_s - T_c}{T_s}\right) \end{aligned}$$

In particular, when $L \gg 1$, fast hopping FHSS clearly achieves much better BER as

$$P_b \approx \frac{1}{2}\left(1 - \frac{T_s}{LT_c}\right) \exp\left(-\frac{E_b}{2\mathcal{N}}\right) + \frac{1}{2}\left(\frac{T_s}{LT_c}\right) \exp\left(-\frac{E_b}{2\mathcal{N}} \cdot 1\right) = \frac{1}{2} \exp\left(-\frac{E_b}{2\mathcal{N}}\right)$$

In other words, by using fast hopping, the BER performance of FHSS under partial jamming approaches the BER without jamming.

11.2 MULTIPLE FHSS USER SYSTEMS AND PERFORMANCE

Clearly, FHSS systems provide better security against potential enemy jammers or interceptors. Without full knowledge of the hopping pattern that has been established, adversaries cannot follow, eavesdrop on, or jam an FHSS user transmission. On the other hand, if an FHSS system has only one transmitter, then its use of the much larger bandwidth B_c would be too wasteful. To improve the frequency efficiency of FHSS systems, multiple users may be admitted over the same frequency band B_c with little performance loss.

As shown in Fig. 11.4, each of the M users is assigned a unique PN hopping code that controls its frequency hopping pattern in FHSS. The codes can be chosen so that the users never or rarely collide in the spectrum with one another. With multiple users accessing the same L bands, spectral efficiency can be made equal to the original FSK signal without any loss of FHSS security advantages. Thus, multiple user access becomes possible by assigning these distinct hopping (spreading) codes to different users, leading to code division multiple access (CDMA).

Generally, any overlapping of two or more user PN sequences would lead to signal collision in frequency bands where the PN sequence values happen to be identical during certain chips. Theoretically, well-designed hopping codes can prevent such user collisions. However, in practice, the lack of a common synchronization clock observable by all users means that each user exercises frequency hopping independently. Also, sometimes there are more than L active users gaining access to the FHSS system. Both cases lead to user collision. For slow and fast FHSS systems alike, such collision, would lead to significant increases in user detection errors.

Performance of FHSS with Multiple User Access

For any particular FHSS CDMA user, the collision problem would typically be limited to its partial band. In fact, the effect of such collisions is similar to the situation of partial band jamming, as analyzed next.

Recall that the performance analysis of FSK systems has been discussed in Chapter 10 (Sec. 10.7) under AWGN channels. It has been shown that the probability of symbol detection error for noncoherent M-ary FSK signals is

$$P_{eM} = 1 - P_{cM} = \sum_{m=1}^{M-1} \binom{M-1}{m} \frac{(-1)^{m-1}}{m+1} e^{-mE_b \log_2 M/N(m+1)} \tag{11.5}$$

For slow FHSS systems, each data symbol is transmitted using a fixed frequency carrier. Therefore, the detection error probability of slow FHSS system is identical to Eq. (11.5).

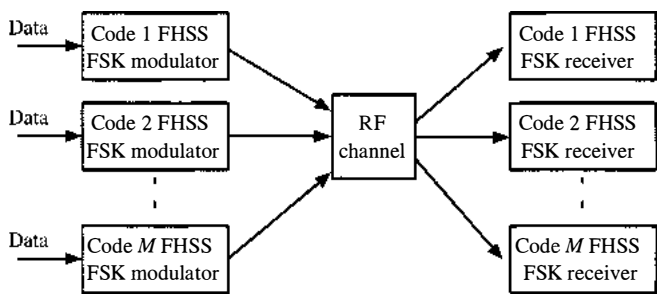


Figure 11.4 CDMA in FHSS in which each of the M users is assigned a unique PN code.

In particular, the BER of the binary FSK system is shown to be [see Eq. (10.145) in Sec. 10.11]

$$P_b = \frac{1}{2}e^{-E_b/2\mathcal{N}}$$

However, if two users transmit simultaneously in the same frequency band, a collision or a "hit" occurs. In this case we will assume that the probability of error is 0.5.* Thus the overall probability of bit error can be modeled as

$$P_b = \frac{1}{2}e^{-E_b/2\mathcal{N}}(1 - P_h) + \frac{1}{2}P_h \tag{11.6}$$

where P_h is the probability of a hit, which we must determine. Consider random hopping. If there are L frequency slots, there is a $1/L$ probability that a given interferer will be present in the desired user's slot. If there are $M - 1$ interferers or other users, the probability that at least one is present in the desired frequency slot is

$$P_h = 1 - \left(1 - \frac{1}{L}\right)^{M-1} \approx \frac{M-1}{L} \tag{11.7}$$

assuming L is large. Substituting this into Eq. (11.6) gives

$$P_b = \frac{1}{2}e^{-E_b/2\mathcal{N}}\left(1 - \frac{M-1}{L}\right) + \frac{1}{2}\frac{M-1}{L} \tag{11.8}$$

If $M = 1$, the probability of error reduces to the BER of BFSK. If $M \neq 1$, by letting $E_b/\mathcal{N}$ to approach infinity, we see that under random hopping,

$$\lim_{E_b/\mathcal{N}\to\infty} P_b = \frac{1}{2}\frac{M-1}{L} = \frac{1}{2}P_h \tag{11.9}$$

which illustrates the irreducible floor of the detected bit error rate due to multiple access interference (MAI). It is therefore important to design hopping patterns to reduce P_h with multiple users.

Asynchronous FHSS

The previous analysis assumes that all users hop their carrier frequencies in synchronization. This is known as *slotted frequency hopping*. Such kind of time slotting is easy to maintain if distances between all transmitter-receiver pairs are essentially the same. This may not be a realistic scenario for many FHSS systems. Even when synchronization can be achieved between individual user clocks, different transmission paths will not arrive synchronously due to the various propagation delays. A simple development for asynchronous performance can be shown following the approach of Geronoitis and Pursley,[1] which shows that the probability of a hit in the asynchronous case is

$$P_h = 1 - \left[1 - \frac{1}{L}\left(1 + \frac{1}{N_b}\right)\right]^{M-1} \tag{11.10}$$

* This is actually pessimistic, since studies have shown that this value can be lower.

where N_b is the number of bits per hop. Comparing Eqs. (11.7) and (11.10) we see that, for the asynchronous case, the probability of a hit is increased, as expected. By using Eq. (11.10) in Eq. (11.6), we obtain the probability of error for the asynchronous case as

$$P_b = \frac{1}{2}e^{-E_b/\mathcal{N}}\left[1 - \frac{1}{L}\left(1 + \frac{1}{N_b}\right)\right]^{M-1} + \frac{1}{2}\left\{1 - \left[1 - \frac{1}{L}\left(1 + \frac{1}{N_b}\right)\right]^{M-1}\right\} \quad (11.11)$$

As in the case of partial band jamming, the BER of the FHSS users decreases as the spreading factor increases. Additionally, by incorporating a sufficiently strong FEC at the transmitter code, the FHSS CDMA users can accommodate most of the collisions.

Example 11.2 Consider an AWGN channel with noise level $\mathcal{N} = 10^{-11}$. A user signal is a binary FSK modulation of data rate 16 kbit/s that occupies a bandwidth of 20 kHz. The received signal power is -20 dBm. An enemy has a jamming source that can jam either a narrowband or a broadband signal. The jamming power is finite such that the total received jamming signal power is at most -26 dBm. Use a spreading factor $L = 20$ to determine the approximate improvement of signal-to-noise ratio for the FHSS system under jamming.

Since $P_s = -20\,\text{dBm} = 10^{-5}$ W and $T_b = 1/16,000$, the energy per bit equals

$$E_b = P_s \cdot T_b = \frac{1}{1.6 \times 10^9}$$

On the other hand, the noise level is $\mathcal{N} = 10^{-11}$. Let the jamming signal have Gaussian distribution. The jamming power level equals $P_J = -26\,\text{dBm} = 4 \times 10^{-6}$ W.

When jamming occurs over the narrow band of 20 kHz, the power level of the interference is

$$J_n = \frac{P_J}{20{,}000\,\text{Hz}} = 2 \times 10^{-10}$$

Thus, the resulting signal-to-noise ratio is

$$\frac{E_b}{J_n + \mathcal{N}} = \frac{(1.6 \times 10^9)^{-1}}{2 \times 10^{-10} + 10^{-10}} \approx 4.74\ \text{dB}$$

If the jamming must cover the entire spread spectrum L times wider, then the power level of the interference becomes 20 times weaker:

$$J_n = \frac{P_J}{400{,}000\,\text{Hz}} = 1 \times 10^{-11}$$

Thus, the resulting signal-to-noise ratio in this case is

$$\frac{E_b}{J_n + \mathcal{N}} = \frac{(1.6 \times 10^9)^{-1}}{10^{-11} + 10^{-11}} \approx 14.95\,\text{dB}$$

The improvement of SNR is approximately 10 dB.

11.3 APPLICATIONS OF FHSS

FHSS has been adopted in several practical applications. The most notable ones among them are the wireless local area network (WLAN) standard for Wi-Fi, known as the IEEE 802.11[2] and the wireless personal area network (WPAN) standard of Bluetooth.

From IEEE 802.11 to Bluetooth

IEEE 802.11 was the first Wi-Fi standard initially released in 1997. With data rate limited to 2 Mbit/s, 802.11 only had very limited deployment before 1999, when the release and much broader adoption of IEEE 802.11a and 802.11b removed the FHSS option. Now virtually obsolete, IEEE 802.11 was miraculously revived in the highly successful commercial product sold as *Bluetooth*.[3] Bluetooth differs from Wi-Fi in that Wi-Fi systems are required to provide higher throughput and covers greater distances.[4] Wi-Fi can also be more costly and consumes more power.

Bluetooth, on the other hand, is an ultra-short-range communication system used in electronic products such as cellphones, computers, automobiles, modems, headsets, and appliances. Replacing line-of-sight infrared, Bluetooth can be used when two or more devices are in proximity to each other. It does not require high bandwidth. Because Bluetooth is basically the same as the IEEE 802.11 frequency hopping (FH) option, we only need to describe its details.

The protocol operates in the license-free industrial, scientific, and medical (ISM) band of 2.4 to 2.4835 GHz. To avoid interfering with other devices and networks in the ISM band, the Bluetooth protocol divides the band into 79 channels of 1 MHz bandwidth and executes (slow) frequency hopping at a rate of up to 1600 Hz. Two Bluetooth devices synchronize frequency hopping by communicating in a master-slave mode relationship. A network group of up to eight devices form a **piconet**, which has one master. A slave node of one piconet can be the master of another piconet. Relationships between master and slave nodes in piconets are shown in Fig. 11.5. A master Bluetooth device can communicate with up to seven active devices. At any time, The master device can bring into active status up to 255 further inactive, or parked, devices. One special feature of Bluetooth is its ability to implement adaptive frequency hopping (AFH). This adaptivity is built in to allow Bluetooth devices to avoid crowded frequencies in the hopping sequence.

The modulation of the (basic rate) Bluetooth signal is shown in Fig. 11.6. The binary signal is transmitted by means of Gaussian pulse shaping on the FSK modulation signal. As shown

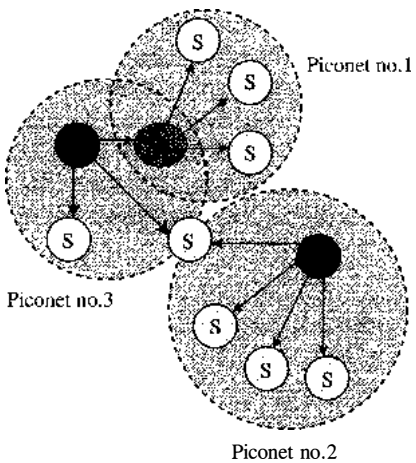


Figure 11.5 An area with the coverage of three piconets: **m**, master nodes; **s**, slave nodes; **s/m**, slave/master. A node can be both a master of one piconet (no. 1) and a slave of another (no. 3).

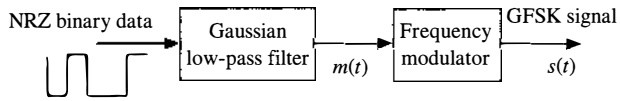


Figure 11.6 FHSS modulation in 802.11 and Bluetooth.

TABLE 11.1
Major Specifications of 802.11 FHSS and Bluetooth.

	802.11 FHSS	Bluetooth (basic rate)
Frequency band	ISM (2.4–2.4835 GHz)	
Duplex format	TDD	
Single-channel bandwidth	1 MHz	
Number of nonoverlapping channels	79	
BT_s product	0.5	
Minimum hopping distance	6	
Modulation	GFSK-2 and GFSK-4	GFSK-2
Data rate	1 Mbit/s and 2 Mbit/s	723.1 kbit/s
Hopping rate	2.5–160 Hz	1600 Hz

in Fig. 11.6, a simple binary FSK replaces the Gaussian low-pass filter with a direct path. The inclusion of the Gaussian low-pass filter generates what is known as the Gaussian FSK (or GFSK) signal. GFSK is a continuous phase FSK. It achieves better bandwidth efficiency by enforcing phase continuity. Better spectral efficiency is also achieved through partial response signaling (PRS) in GFSK. The Gaussian filter response stretches each bit over multiple symbol periods.

More specifically, the Gaussian LPF impulse response is ideally given by

$$h(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-t^2/2\sigma^2} \quad \sigma = \frac{\sqrt{\ln 2}}{2\pi B}$$

where B is the 3 dB bandwidth of the Gaussian low-pass filter. Because this response is noncausal, the practical implementation truncates the filter response to $4T_s$ seconds. This way, each bit of information is extended over a window 3 times broader than the bit duration T_s.

Note that the selection of B is determined by the symbol rate $1/T_s$. In 802.11 and Bluetooth, $B = 0.5/T_s$ is selected. The FM modulation index must be between 0.28 and 0.35. The GFSK symbol rate is always 1 MHz; binary FSK and four-level FSK can be implemented as GFSK-2 and GFSK-4, achieving data throughput of 1 and 2 Mbit/s, respectively. Table 11.1 summarizes the key parameters and differences in IEEE 802.11 and the Bluetooth (basic rate).

We note that our discussions on Bluetooth have focused on the (basic rate) versions 1.1 and 1.2. More recently, version 2 of Bluetooth has been released.[4] Version 2.0 implementations feature Bluetooth Enhanced Data Rate (EDR) and reach 2.1 Mbit/s. Technically, version 2.0 devices retain the FHSS feature but resort to the more efficient (differential) PSK modulations.

SINCGARS

SINCGARS stands for **s**ingle **c**hannel **g**round and **a**irborne **r**adio **s**ystem. It represents a family of VHF-FM combat radios used by the U.S. military. First produced by ITT in 1983, SINCGARS transmits voice with FM and data with binary CPFSK at 16 kbit/s, occupying a bandwidth of 25 kHz. There can be as many as 2320 channels within the operational band of 30 to 87.975 MHz.

To combat jamming, SINCGARS radios can implement frequency hopping at the rather slow rate of 100 Hz. Because the hopping rate is quite slow, SINCGARS is no longer effective against modern jamming devices. For this reason, SINCGARS is being replaced by the newer and more versatile JTRS (joint tactical radio system).

From Hollywood to CDMA

Like many good ideas, the concept of *frequency hopping* also had multiple claims of inventors. One such patent that gained little attention was awarded to Willem Broertjes of Amsterdam, Netherlands, in August 1932 (U.S. Patent no. 1,869,659).[5] However, the most intriguing patent on frequency hopping came from one of Hollywood's well-known actresses during World War II, Hedy Lamarr. In 1942 she and her coinventor George Antheil (an eccentric composer) were awarded U.S. patent no. 2,292,387 for their "Secret Communications System." The patent was designed to make radio-guided torpedoes harder to detect or to jam. Largely because of the Hollywood connection, Hedy Lamarr became a legendary figure in the wireless communication community, often credited as the *inventor* of CDMA, whereas other less glamorous figures such as Willem Broertjes have been largely forgotten.

Hedy Lamarr was a major movie star of her time.[6] Born Hedwig Eva Maria Kiesler in Vienna, Austria, she first gained fame in the 1933 Austrian film *Ecstasy* for some shots that were highly unconventional in those days. In 1937, escaping the Nazis and her first husband (a Nazi arms dealer), she went to London, where she met Louis Burt Mayer, cofounder and boss of the MGM studio. Mayer helped the Austrian actress's Hollywood career by giving her a movie contract and a new name—Hedy Lamarr. Lamarr starred with famous colleagues such as Clark Gable, Spencer Tracy, and Judy Garland, appearing in more than a dozen films during her film career.

Clearly gifted scientifically, Hedy Lamarr worked with George Antheil, a classical composer, to help the war effort. They originated an idea of a sophisticated antijamming device for use in radio-controlled torpedoes. In August 1942, under her married name at the time, Hedy Kiesler Markey, Hedy Lamarr was awarded U.S. Patent no. 2,292,387 (Fig. 11.7), together with George Antheil. They donated the patent as their contribution to the war effort. Drawing inspiration from the composer's piano, their invention of frequency hopping uses 88 frequencies, one for each note on a piano keyboard.

However, the invention would not be implemented during World War II. It was simply too difficult to pack vacuum tube electronics into a torpedo. The idea of frequency hopping, nevertheless, became reality 20 years later during the 1962 Cuban missile crisis, when the system was installed on ships sent to block communications to and from Cuba. Ironically, by then, the Lamarr-Antheil patent had expired. The idea of frequency hopping, or more broadly, the idea of spread spectrum, has since been extensively used in military and civilian communications, including cellular phones, wireless LAN, Bluetooth, and numerous other wireless communications systems.

Only in recent years has Hedy Lamarr started receiving a new kind of recognition as a celebrity inventor. In 1997 Hedy Lamarr and George Antheil received the Electronic Frontier Foundation (EFF) Pioneer Award. Furthermore, in August 1997, Lamarr was honored with the prized BULBIE Gnass Spirit of Achievement Bronze Award (the "Oscar" of inventing). If she had won an Academy Award for her film works in film, she would have been the only person to receive two entirely different "Oscar" awards! Still, inventors around the world are truly delighted to welcome a famous movie celebrity into their ranks.

Inventor *Hedy Kiesler Markey* died in 2000 at the age of 86.

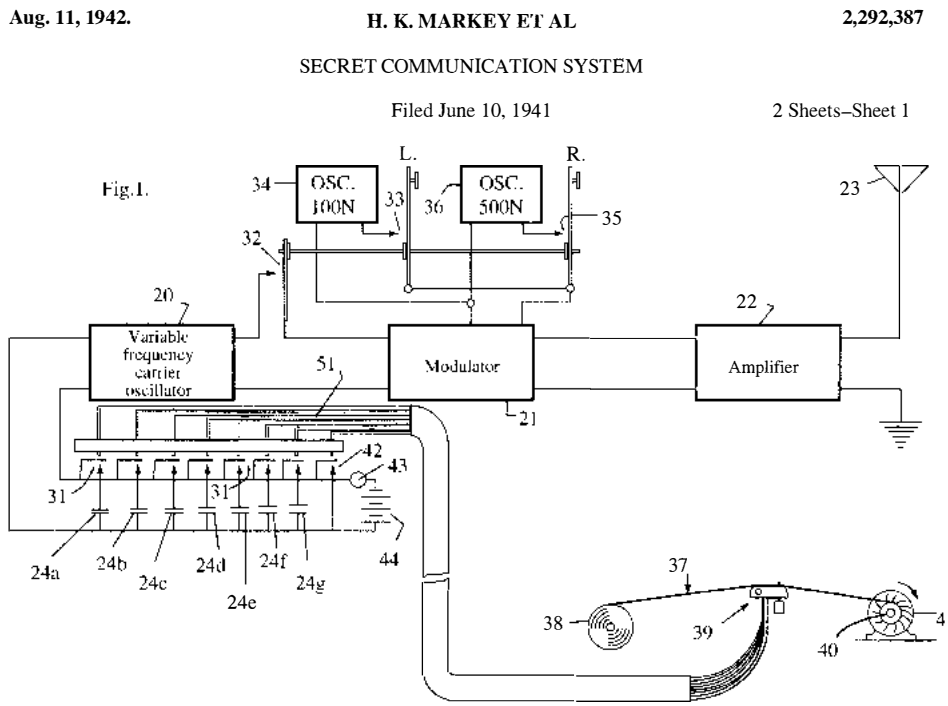


Figure 11.7 Figure 1 from the Lamarr-Antheil patent (From U.S. Patent and Trademark Office.)

11.4 DIRECT SEQUENCE SPREAD SPECTRUM

FHSS systems exhibit some important advantages, including low-complexity transceivers and resistance to jamming. However, the difficulty of carrier synchronization under frequency hopping means that only noncoherent demodulations for FSK and DPSK are actually practical. As shown in the analysis from Sec. 10.11, FSK and DPSK tend to have poorer BER performance (power efficiency) and poorer bandwidth efficiency compared with QAM systems, which require coherent detection. Furthermore, its susceptibility to collision makes FHSS a less effective technology for CDMA. As modern communication systems have demonstrated, direct sequence spread spectrum (DSSS) systems are much more efficient in bandwidth and power utilization.[7] Today, DSSS has become the dominant CDMA technology in advanced wireless communication systems. It is not an exaggeration to state that DSSS and CDMA are almost synonymous.

Optimum Detection of DSSS PSK

Direct sequence spread spectrum is a technology that is more suitable for integration with bandwidth-efficient linear modulations such as QAM/PSK. Although there are several different ways to view DSSS, its key operation of spectrum spreading is achieved by a PN sequence, also known as the PN code or PN chip. The PN sequence is mostly binary, consisting of **1**s and **0**s, which are represented by polar signaling of $+1$ and -1. To minimize interference and to facilitate chip synchronization, the PN sequence has some nice autocorrelation and cross-correlation properties.

Direct sequence spread spectrum (DSSS) expands the traditional narrowband signal by utilizing a spreading signal $c(t)$. As shown in Fig. 11.8, the original data signal is linearly

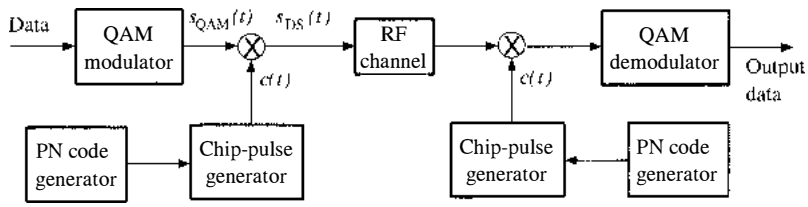


Figure 11.8
DSSS system.

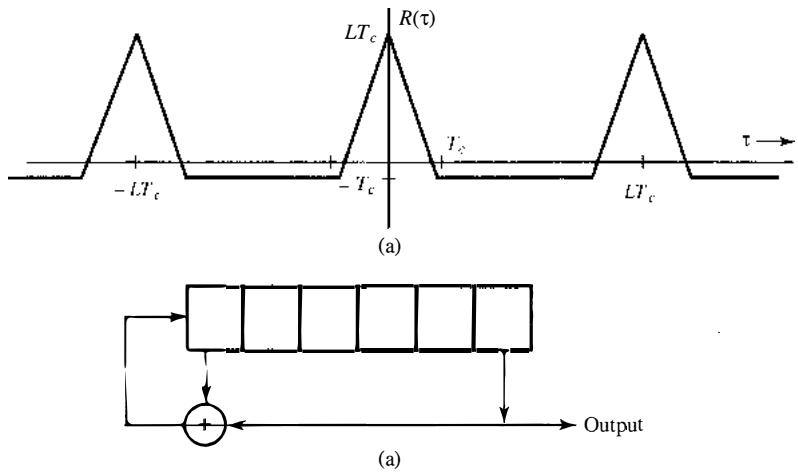


Figure 11.9
(a) PN sequence autocorrelation function.
(b) Six-stage generator of a maximum length PN sequence.

modulated into a QAM signal $s_{\text{QAM}}(t)$. Instead of transmitting this signal directly over its required bandwidth, DSSS modifies the QAM signal by multiplying the spreading chip signal $c(t)$ with the QAM narrowband signal. Although the signal carrier frequency remains unchanged at ω_c, the new signal after spreading becomes

$$s_{\text{DS}}(t) = s_{\text{QAM}}(t)c(t) \tag{11.12}$$

Hence, the transmitted signal $s_{\text{DS}}(t)$ is a product of two signals whose spread bandwidth is equal to the bandwidth sum of of the QAM signal $s_{\text{QAM}}(t)$ and the spreading signal $c(t)$.

PN Sequence Generation

A good PN sequence $c(t)$ is characterized by an autocorrelation that is similar to that of a white noise. This means that the autocorrelation function of a PN sequence should be high near $\tau = 0$ and low for all $\tau \neq 0$, as shown in Fig. 11.9a. Moreover, in CDMA applications several users share the same band using different PN sequences. Hence, it is necessary that the cross-correlation among different pairs of PN sequences be small to reduce mutual interference.

A PN code is periodic. A digital shift register circuit with output feedback can generate a sequence with long period and low susceptibility to structural identification by an outsider. The most widely known binary PN sequences are the **maximum length** shift register sequences (m-sequences). Such a sequence, which can be generated by an m-stage

shift register with suitable feedback connection, has a length $L = 2^m - 1$ bits, the maximum period for such a finite state machine. Figure 11.9b shows a shift register encoder for $m = 6$ and $L = 63$. For such "short" PN sequences, the autocorrelation function is nearly an impulse and is periodic

$$R_c(\tau) = \int_0^{T_s} c(t)c(t+\tau)\,d\tau = \begin{cases} LT_c & \tau = 0, \pm LT_c, \ldots \\ -T_c & \tau \neq 0, \pm LT_c, \ldots \end{cases} \tag{11.13}$$

As a matter of terminology, a DSSS spreading code is a *short code* if the PN sequence period equals the data symbol period T_s. A DSSS spreading code is a *long code* if the PN sequence period is a (typically large) multiple of the data symbol period.

Single-User DSSS Analysis

The simplest analysis of DSSS system can be based on Fig. 11.8. To achieve spread spectrum, the chip signal $c(t)$ typically varies much faster than the QAM symbols. As shown in Fig. 11.8, there are multiple chips of ± 1 within each symbol duration of T_s. Denote the spreading factor

$$L = T_s/T_c \qquad T_c = \text{chip period}$$

Then the spread signal spectrum is essentially L times broader than the original modulation spectrum

$$B_c = (L+1)B_s \approx L \cdot B_s$$

Note that the spreading signal $c(t) = \pm 1$ at any given instant. Given the polar nature of the binary chip signal, the receiver, under an AWGN channel, can easily "despread" the received signal

$$y(t) = s_{\text{DS}}(t) + \text{n}(t) = s_{\text{QAM}}(t)c(t) + \text{n}(t) \tag{11.14}$$

by multiplying the chip signal with the received signal

$$\begin{aligned} r(t) &= c(t)y(t) \\ &= s_{\text{QAM}}(t)c^2(t) + \text{n}(t)c(t) && (11.15) \\ &= s_{\text{QAM}}(t) + \underbrace{\text{n}(t)c(t)}_{\text{x}(t)} && (11.16) \end{aligned}$$

Thus, this multiplication allows the receiver to successfully "despread" the spread spectrum signal. The analysis of the DSSS receiver depends on the characteristics of the noise $\text{x}(t)$. Because $c(t)$ is deterministic, and $\text{n}(t)$ is Gaussian with zero mean, $\text{x}(t)$ remains Gaussian with zero mean. As a result, the receiver performance analysis requires finding only the PSD of $\text{x}(t)$.

To determine the power spectral density of the "despread" noise $\mathrm{x}(t) = \mathrm{n}(t)c(t)$, we can start from the definition of PSD (Sec. 9.3):

$$S_{\mathrm{x}}(f) = \lim_{T\to\infty} \overline{\left[\frac{|\mathrm{X}_T(f)|^2}{T}\right]}$$

$$= \lim_{T\to\infty} \overline{\left[\frac{1}{T}\int_{-T/2}^{T/2}\int_{-T/2}^{T/2} \mathrm{x}(t_1)\mathrm{x}(t_2)e^{-j2\pi f(t_2-t_1)}\,dt_1\,dt_2\right]} \tag{11.17a}$$

$$= \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{T/2}\int_{-T/2}^{T/2} \overline{\mathrm{x}(t_1)\mathrm{x}(t_2)}e^{-j2\pi f(t_2-t_1)}\,dt_1\,dt_2$$

$$= \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{T/2}\int_{-T/2}^{T/2} c(t_1)c(t_2)\overline{\mathrm{n}(t_1)\mathrm{n}(t_2)}e^{-j2\pi f(t_2-t_1)}\,dt_1\,dt_2$$

$$= \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{T/2}\int_{-T/2}^{T/2} c(t_1)c(t_2)R_{\mathrm{n}}(t_2-t_1)e^{-j2\pi f(t_2-t_1)}\,dt_1\,dt_2 \tag{11.17b}$$

Recall that

$$R_{\mathrm{n}}(t_2-t_1) = \int_{-\infty}^{\infty} S_n(\nu)e^{j2\pi\nu(t_2-t_1)}\,d\nu$$

We therefore have

$$S_{\mathrm{x}}(f) = \lim_{T\to\infty} \frac{1}{T}\int_{-\infty}^{\infty}\int_{-T/2}^{T/2}\int_{-T/2}^{T/2} c(t_1)c(t_2)S_n(\nu)e^{-j2\pi(f-\nu)(t_2-t_1)}\,dt_1\,dt_2\,d\nu \tag{11.18a}$$

$$= \int_{-\infty}^{\infty} S_{\mathrm{n}}(\nu)\lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2}\int_{-T/2}^{T/2} c(t_1)c(t_2)e^{-j2\pi(f-\nu)(t_2-t_1)}\,dt_1\,dt_2\,d\nu$$

$$= \int_{-\infty}^{\infty} S_{\mathrm{n}}(\nu)\lim_{T\to\infty}\frac{1}{T}\left|\int_{-T/2}^{T/2} c(t)e^{-j2\pi(f-\nu)t}\,dt\right|^2 d\nu$$

$$= \int_{-\infty}^{\infty} S_{\mathrm{n}}(\nu)\lim_{T\to\infty}\frac{|C_T(f-\nu)|^2}{T}\,d\nu$$

$$= \int_{-\infty}^{\infty} S_{\mathrm{n}}(\nu)S_c(f-\nu)\,d\nu \tag{11.18b}$$

The last equality comes from the definition of PSD for $c(t)$. Equation (11.18) illustrates the dependency of the detector noise PSD on the chip signal $c(t)$. As long as the PN sequence is almost orthogonal such that it satisfies Eq. (11.13), then

$$R_c(\tau) \approx LT_c \cdot \sum_i \delta(\tau - i\cdot LT_c) \tag{11.19a}$$

$$S_c(f) \approx LT_c \cdot \frac{1}{LT_c}\sum_k \delta(f - k/LT_c) = \sum_k \delta(f - k/LT_c) \tag{11.19b}$$

and

$$S_x(f) = \sum_k S_n(f - k/LT_c) \tag{11.20}$$

In other words, as long as the chip sequence is approximately orthogonal, the noise at the QAM detector remains a white Gaussian with zero mean. For practical reasons, the white noise $n(t)$ is filtered at the receiver to be band-limited to $1/2T_c$. As a result, the noise spectrum after the despreader still is

$$S_x(f) = \frac{\mathcal{N}}{2} \tag{11.21}$$

In other words, the spectral level also remains unchanged. Thus, the performance analysis carried out for coherent QAM and PSK detections in Chapter 10 can be applied directly.

In Sec. 10.6 we showed that for a channel with (white) noise of PSD $\mathcal{N}/2$, the error probability of optimum receiver for polar signaling is given by

$$P_b = Q\left(\sqrt{\frac{2E_b}{\mathcal{N}}}\right) \tag{11.22}$$

where E_b is the energy per bit (energy of one pulse). This result demonstrates that the error probability of an optimum receiver is unchanged regardless of whether or not we use DSSS. While this result appears to be somewhat surprising, in fact, it is quite consistent with the AWGN analysis. For *single user*, the only change in DSSS lies in the spreading of transmissions over a broader spectrum by effectively using a new pulse shape $c(t)$. Hence, the modulation remains QAM whereas the channel remains AWGN. Consequently, the coherent detection analysis of Sec. 10.6 is fully applicable to DSSS signals.

11.5 RESILIENT FEATURES OF DSSS

As in FHSS, DSSS systems provide better security against potential jamming or interception by spreading the overall signal energy over a bandwidth L times broader. First, its low power level is difficult for interceptors to detect. Furthermore, without the precise knowledge of the user spreading code [or $c(t)$], adversaries cannot despread and recover the baseband QAM signal effectively. In addition, partial band jamming signals interfere with only a portion of the signal energy. They do not block out the entire signal spectrum and are hence not effective against DSSS signals.

To analyze the effect of partial band jamming, consider an interference $i(t)$ that impinges on the receiver to yield

$$y(t) = s_{\text{QAM}}(t)c(t) + i(t)$$

Let the interference bandwidth be B_i. After despreading, the output signal plus interference becomes

$$y(t)c(t) = s_{\text{QAM}}(t) + i(t)c(t) \tag{11.23}$$

It is important to observe that the interference term has a new frequency response because of despreading by $c(t)$

$$i_a(t) = i(t)c(t) \Longleftrightarrow I(f) * C(f) \tag{11.24}$$

which has approximate bandwidth $B_c + B_i = LB_s + B_i$.

DSSS Analysis against Narrowband Jammers

If the interference has the same bandwidth as the QAM signal B_s, then the "despread" interference $i_a(t)$ will now have bandwidth equal to $(L + 1)B_s$. In other words, the narrowband interference $i(t)$ will in fact be **spread** L times larger by the "despreading" signal $c(t)$.

If the narrowband interference has total power P_i and bandwidth B_s, then the original interference spectral level before despreading is

$$S_i(f) = \frac{P_i}{B_s} \qquad f \in (f_c - 0.5B_s, f_c + 0.5B_s)$$

After despreading, the spectrum of the interference $i_a(t)$ becomes

$$S_{i_a}(f) = \frac{P_i}{(L+1)B_s} \qquad f \in [f_c - 0.5(L+1)B_s, \quad f_c + 0.5(L+1)B_s]$$

Because of the despreading operation, the narrowband interference is only $1/(L + 1)$ the original spectral strength. Note that the desired QAM signal still has its original bandwidth $(\omega_c - \pi B_s, \omega_c + \pi B_s)$. Hence, against narrowband interferences, despreading can reduce the signal-to-interference ratio (SIR) by a factor of

$$\frac{\dfrac{E_b}{P_i/B_s}}{\dfrac{E_b}{P_i/(L+1)B_s}} = L + 1 \tag{11.25}$$

This result illustrates that DSSS is very effective against narrowband (partial band) jamming signals. It effectively improves the SIR by the spreading factor. The "spreading" effect of the despreader on a narrowband interference signal is illustrated in Fig. 11.10.

The ability of DSSS to combat narrowband jamming also means that a narrowband communication signal can coexist with DSSS signals. The SIR analysis and Fig. 11.10 already established the resistance of DSSS signals to narrowband interferers. Conversely, if a narrowband signal must be demodulated in the presence of a DSSS signal, then the narrowband

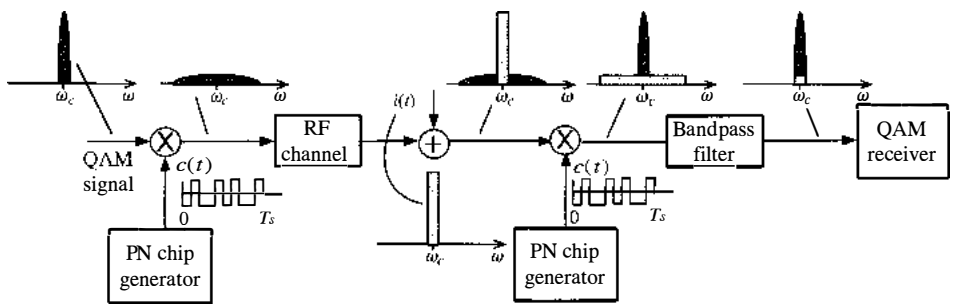


Figure 11.10 Narrowband interference mitigation by the DSSS despreader.

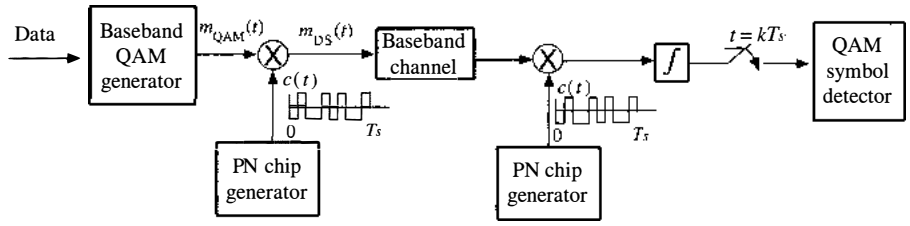


Figure 11.11 Equivalent baseband diagram of DSSS system.

signal can also be extracted with little interference from the DSSS signal by replacing the despreader with a narrow bandpass filter. In this case, the roles of signal and interference are in fact reversed.

DSSS Analysis against Broadband Jammers

In many cases, interferences come from broadband sources that are not generated from the DSSS spreading approach. Against such interferences, the despreading operation only mildly broadens and weakens the interference spectrum.

Let the interference be broadband with the same bandwidth LB_s as the spread signal. Based on Eq. (11.24), the interference after despreading would be $i_a(t)$, which has bandwidth of $2LB_s$. In other words, broadband interference $i(t)$ will in fact be expanded to a spectrum nearly twice as wide and half as strong in intensity. From this discussion, we can see that a DSSS signal is most effective against narrowband interferences and not as effective against broadband interferences.

11.6 CODE DIVISION MULTIPLE-ACCESS (CDMA) OF DSSS

The RF diagram of a DSSS system can be equivalently represented by the baseband diagram of Fig. 11.11, which provides a new perspective on the DSSS system that is amenable to analysis. Let the (complex-valued) QAM data symbol be

$$s_k = a_k + j\,b_k \qquad (k-1)T_s \le t < kT_s \tag{11.26}$$

Then it is clear from the PN chip sequence that the baseband signal after spreading is

$$s_k \cdot c(t) = (a_k + j\,b_k) \cdot c(t) \qquad (k-1)T_s \le t < kT_s \tag{11.27}$$

In other words, the symbol s_k is using

$$c(t) \qquad (k-1)T_s \le t < kT_s$$

as its pulse shape for transmission. Consequently, at the receiver, the optimum receiver would require $c(t)$ to be used as a correlator receiver (or, equivalently, a matched filter). As evident from the diagram of Fig. 11.11, the despreader serves precisely the function of the optimum matched filter (or correlator receiver). Such a receiver is known as a conventional single-user optimum receiver.

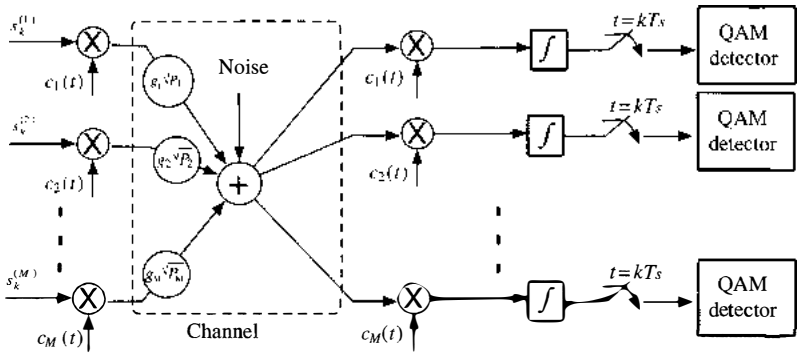


Figure 11.12 A code division multiple-access (CDMA) system based on DSSS.

We have shown that DSSS systems enjoy advantages against the threat of narrowband jamming and attempts at interception. However, if a DSSS system has only one signal to transmit, then its use of the larger bandwidth B_c would be too wasteful. Just as in FHSS, CDMA of DSSS can be achieved by letting multiple users, each given a distinct PN spreading signal $c_i(t)$, access the broad bandwidth of $L B_s$ simultaneously. Such a multiple-access system with M users based on CDMA is shown in Fig. 11.12. Each user can apply a single-user optimum receiver.

Because these CDMA users will be transmitting without time division or frequency division, multiple-access interference (MAI) exists at each of the receivers. To analyze a DSSS system with M multiple-access users, we compute the interference at the output of a given receiver caused by the remaining $M - 1$ users. It is simpler to focus on the time interval $[(k-1)T_s, kT_s]$ and the kth symbol of all M users. In Fig. 11.12, we have made the multiple assumptions for analytical simplicity. Here we state them explicitly:

- The ith user transmits one symbol $s_k^{(i)}$ over the interval $[(k-1)T_s, kT_s]$.
- There is no relative delay among M users, and each receiver receives the kth symbol of all M users within $[(k-1)T_s, kT_s]$.
- All user symbols have unit power; that is, $E\{|s_k^{(i)}|^2\} = 1$.
- The ith user's transmission power is P_i.
- The ith user channel has a scalar gain of g_i.
- The channel is AWGN with noise $\mathrm{n}(t)$.

The first two assumptions indicate that all M users are *synchronous*. While asynchronous CDMA systems are commonplace in practice, their analysis is a straightforward but nontrivial generalization of the synchronous case.*

Because all users share the same bandwidth, every receiver will have equal access to the same channel output signal

$$y(t) = \sum_{j=1}^{M} g_j \sqrt{P_j} s_k^{(j)} c_j(t) + \mathrm{n}(t) \tag{11.28a}$$

* In asynchronous CDMA analysis, the analysis window must be enlarged to translate it into a nearly equivalent synchronous CDMA case with many more equivalent users.[8, 9]

After application of the matched filter (despreading), the ith receiver output at the sampling instant $t = kT_s$ is

$$\begin{aligned} \mathrm{r}_k^{(i)} &= \int_{(k-1)T_s}^{kT_s} c_i(t)y(t)\,dt \\ &= \sum_{j=1}^{M} g_j\sqrt{P_j}s_k^{(j)} \int_{(k-1)T_s}^{kT_s} c_i(t)c_j(t)\,dt + \int_{(k-1)T_s}^{kT_s} c_i(t)\mathrm{n}(t)\,dt \\ &= \sum_{j=1}^{M} g_j\sqrt{P_j}R_{ij}(k)s_k^{(j)} + \mathrm{n}_i(k) \end{aligned} \tag{11.28b}$$

For notational convenience, we have defined the (time-varying) cross-correlation coefficient between two spreading codes as

$$R_{ij}(k) = \int_{(k-1)T_s}^{kT_s} c_i(t)c_j(t)\,dt \tag{11.28c}$$

and the ith receiver noise sample as

$$\mathrm{n}_i(k) = \int_{(k-1)T_s}^{kT_s} c_i(t)\mathrm{n}(t)\,dt \tag{11.28d}$$

It is important to note that the noise samples of Eq. (11.28d) are Gaussian with mean

$$\overline{\mathrm{n}_i(k)} = \int_{(k-1)T_s}^{kT_s} c_i(t)\overline{\mathrm{n}(t)}\,dt = 0$$

The cross-correlation between two noise samples can be found as

$$\begin{aligned} \overline{\mathrm{n}_i(k)\mathrm{n}_j(\ell)} &= \int_{(k-1)T_s}^{kT_s}\int_{(\ell-1)T_s}^{\ell T_s} c_i(t_1)c_j(t_2)\overline{\mathrm{n}(t_1)\mathrm{n}(t_2)}\,dt_1\,dt_2 \\ &= \int_{(k-1)T_s}^{kT_s}\int_{(\ell-1)T_s}^{\ell T_s} c_i(t_1)c_j(t_2)R_{\mathrm{n}}(t_2-t_1)\,dt_1\,dt_2 \\ &= \int_{(k-1)T_s}^{kT_s}\int_{(\ell-1)T_s}^{\ell T_s} c_i(t_1)c_j(t_2)\frac{\mathcal{N}}{2}\delta(t_2-t_1)\,dt_1\,dt_2 \\ &= \frac{\mathcal{N}}{2}\delta[k-\ell]\int_{(k-1)T_s}^{kT_s} c_i(t_1)c_j(t_1)\,dt_1 \qquad (11.29a) \\ &= \frac{\mathcal{N}}{2}R_{ij}(k)\delta[k-\ell] \qquad (11.29b) \end{aligned}$$

Equation (11.29) shows that the noise samples at the DSSS CDMA receiver are temporally white. This means that the Gaussian noise samples at different sampling time instants are independent of one another. Therefore, the optimum detection of $\{s_k^{(i)}\}$ can be based on the samples $\{\mathrm{r}_k^{(i)}\}$ at time $t = kT$.

For short-code CDMA, $\{c_i(t)\}$ are periodic and the period equals T_s. In other words, the PN spreading signals $\{c_i(t)\}$ are identical over each period $[(k-1)T_s, kT_s]$. Therefore, in

short code CDMA systems, the cross-correlation coefficient between two spreading codes is a constant

$$R_{i,j}(k) = R_{i,j} \tag{11.30}$$

Note that the decision variable of the ith receiver is

$$\mathrm{r}_k^{(i)} = g_i\sqrt{P_i}R_{i,i}(k)s_k^{(i)} + \underbrace{\sum_{j\neq i}^{M} g_j\sqrt{P_j}R_{i,j}(k)s_k^{(j)}}_{I_k^{(i)}} + \mathrm{n}_i(k) \tag{11.31}$$

The term $I_k^{(i)}$ is an additional term resulting from the multiple-access interference of the $M-1$ interfering signals. When the spreading codes are selected to satisfy the orthogonality condition

$$R_{i,j}(k) = 0 \qquad i \neq j$$

then the CDMA multiple-access interference is zero, and each CDMA user obtains performance identical to that of the single DSSS user or a single baseband QAM user.

There are various ways to generate orthogonal spreading codes. Walsh-Hadamard codes are the best-known orthogonal spreading codes. Given a code length of L identical to the spreading factor, there are a total of L orthogonal Walsh-Hadamard codes. A simple example of the Walsh-Hadamard code for $L = 8$ is given here. Each row in the matrix of Eq. (11.32) is a spreading code of length 8:

$$W_8 = \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \end{bmatrix} \tag{11.32}$$

At the next level, Walsh-Hadamard code has length 16, which can be obtained from W_8 via

$$W_{2^k} = \begin{bmatrix} W_{2^{k-1}} & W_{2^{k-1}} \\ W_{2^{k-1}} & -W_{2^{k-1}} \end{bmatrix}$$

In fact, starting from $W_1 = [1]$ with $k = 0$, this recursion can be used to generate length $L = 2^k$ Walsh-Hadamard codes.

Gaussian Approximation of Nonorthogonal MAI

In practical applications, many user spreading codes are not fully orthogonal. As a result, the effect of MAI on user detection performance may be serious. To analyze the effect of MAI on a single-user receiver, we need to study the MAI probability distribution. The exact probability analysis of I_k is difficult. An alternative is to use a good approximation. When M is large, one may invoke the central limit theorem to approximate the MAI as a Gaussian random variable.

Recall that the QAM symbols $s_k^{(j)}$ are independent with zero mean and unit variance, that is,

$$\overline{s_k^{(j)}} = 0$$
$$\overline{s_k^{(j)} s_k^{(i)*}} = 0 \qquad i \neq j$$
$$\overline{\left|s_k^{(j)}\right|^2} = 1$$

Hence, we can approximate the MAI as Gaussian with mean

$$\overline{I_k^{(i)}} = \sum_{j \neq i}^{M} g_j \sqrt{P_j} R_{i,j}(k) \overline{s_k^{(j)}} = 0 \tag{11.33}$$

and variance

$$\overline{\left|I_k^{(i)}\right|^2} = \sum_{j \neq i}^{M} |g_j|^2 P_j \left|R_{i,j}(k)\right|^2 \tag{11.34}$$

The effect of this MAI approximation is a strengthened channel noise. Effectively, the performance of detection based on decision variable $r_k^{(i)}$ is degraded by the additional Gaussian MAI. Based on single-user analysis, the new equivalent SNR is degraded and becomes

$$\frac{2E_b}{E_b\left(|g_i|^2 P_i \left|R_{i,i}(k)\right|^2\right)^{-1} \sum_{j \neq i}^{M} |g_j|^2 P_j \left|R_{i,j}(k)\right|^2 + \mathcal{N}}$$

For the special case of BPSK or polar signaling, the BER of the ith CDMA user is approximately

$$Q\left(\frac{2E_b}{E_b\left(|g_i|^2 P_i \left|R_{i,i}(k)\right|^2\right)^{-1} \sum_{j \neq i}^{M} |g_j|^2 P_j \left|R_{i,j}(k)\right|^2 + \mathcal{N}}\right) \tag{11.35}$$

Observe that when a single user is present ($M = 1$), Eq. (11.35) becomes the well-known polar BER result of

$$P_b = Q\left(\frac{2E_b}{\mathcal{N}}\right)$$

as expected. The same result is also true when all spreading codes are mutually orthogonal such that $R_{i,j}(k) = 0, i \neq j$.

In the extreme case of noise-free systems, when the signal-to-noise ratio is very high ($E_b/\mathcal{N} \to \infty$), we obtain

$$\lim_{E_b/\mathcal{N} \to \infty} P_b = Q\left(\frac{|g_i|^2 P_i \left|R_{i,j}(k)\right|^2}{\sum_{j \neq i}^{M} |g_j|^2 P_j \left|R_{i,j}(k)\right|^2}\right)$$

This shows the presence of an irreducible error floor for the MAI limited case. This noise floor vanishes when the spreading codes are mutually orthogonal such that $R_{i,j}(k) = 0$ if $i \neq j$.

The Near-Far Problem

The Gaussian approximation of the MAI has limitations when used to predict system performance. While the central limit theorem implies that $I_k^{(i)}$ will tend toward a Gaussian distribution near the center of its distribution, convergence may require very large number of CDMA users M. In a typical CDMA system, the user number M is only in the order of 64 to 128. When M is not sufficiently large, the Gaussian approximation of the MAI may be highly inaccurate, particularly in a near-far environment.

The so-called *near-far* environment describes the following scenario.

- The desired transmitter is much farther away from its receivers than some interfering transmitters.
- The spreading codes are not mutually orthogonal; that is, $R_{i,j}(k) \neq 0$ when $i \neq j$.

If we assume identical user transmission power in all cases, (i.e., $P_j = P_o$), in the near-far environment the desired signal channel gain g_i is much smaller than some interferers' channel gains. In other words, there may exist some user set $\mathcal{J}$ such that

$$g_i \ll g_j \qquad j \in \mathcal{J} \tag{11.36}$$

As a result, Eq. (11.31) becomes

$$\begin{aligned} \mathbf{r}_k^{(i)} &= \sqrt{P_o} g_i R_{i,i}(k) s_k^{(i)} + \sqrt{P_o} \sum_{j \in \mathcal{J}} g_j R_{i,j}(k) s_k^{(j)} + \left[\sqrt{P_o} \sum_{j \notin \mathcal{J}} g_j R_{i,j}(k) s_k^{(j)} + \mathrm{n}_i(k) \right] \\ &= \sqrt{P_o} g_i R_{i,i}(k) s_k^{(i)} + \sqrt{P_o} \sum_{j \in J} g_j R_{i,j}(k) s_k^{(j)} + \mathbf{n}_i'(k) \end{aligned} \tag{11.37}$$

where we have defined an equivalent noise term

$$\mathbf{n}_i'(k) = \sqrt{P_o} \sum_{j \notin \mathcal{J}} g_j R_{i,j}(k) s_k^{(j)} + \mathrm{n}_i(k) \tag{11.38}$$

that is approximately Gaussian.

In a near-far environment, it becomes likely that the smaller signal channel gain and the nonzero cross-correlation result in the domination of the (far) signal component

$$g_i R_{i,i}(k) s_k^{(i)}$$

by the strong (near) interference

$$\sum_{j \in \mathcal{J}} g_j R_{i,j}(k) s_k^{(j)}$$

The Gaussian approximation analysis of the BER in Eq. (11.35) no longer applies.

Example 11.3 Consider a CDMA system with two users ($M = 2$). Both signal transmission powers are 10 mW. The receiver for user 1 can receive signals from both user signals. To this receiver, the two signal channel gains are

$$g_1 = 10^{-4} \qquad g_2 = 10^{-1}$$

The spreading gain equals $L = 128$ such that

$$R_{1,1}(k) = 128 \qquad R_{1,2}(k) = -1$$

The sampled noise $\mathrm{n}_1(k)$ is Gaussian with zero mean and variance of 10^{-6}. Determine the BER for the desired user 1 signal.

The receiver decision variable at time k is

$$\begin{aligned}\mathrm{r}_k &= \sqrt{10^{-2}} \cdot 10^{-4} \cdot 128 \cdot s_k^{(1)} + \sqrt{10^{-2}} \cdot 10^{-1} \cdot (-1) \cdot s_k^{(2)} + \mathrm{n}_1(k) \\ &= 10^{-2}\left[0.128 s_k^{(1)} - s_k^{(2)} + 100\mathrm{n}_1(k)\right]\end{aligned}$$

For equally likely data symbols ± 1, the BER of user 1 is

$$\begin{aligned}P_b &= 0.5 \cdot P\left[\mathrm{r}_k > 0 | s_k^{(1)} = -1\right] + 0.5 \cdot P\left[\mathrm{r}_k < 0 | s_k^{(1)} = 1\right] \\ &= P\left[\mathrm{r}_k < 0 | s_k^{(1)} = 1\right] \\ &= P\left[0.128 - s_k^{(2)} + 100\mathrm{n}_1(k) < 0\right]\end{aligned}$$

Because of the equally likely data symbol $P\left[s_k^{(2)} = \pm 1\right] = 0.5$, we can utilize the total probability theorem to obtain

$$\begin{aligned}P_b &= 0.5P\left[0.128 - s_k^{(2)} + 100\mathrm{n}_1(k) < 0 | s_k^{(2)} = 1\right] \\ &\quad + 0.5P\left[0.128 - s_k^{(2)} + 100\mathrm{n}_1(k) < 0 | s_k^{(2)} = -1\right] \\ &= 0.5P\left[0.128 - 1 + 100\mathrm{n}_1(k) < 0\right] + 0.5P\left[0.128 + 1 + 100\mathrm{n}_1(k) < 0\right] \\ &= 0.5P\left[100\mathrm{n}_1(k) < 0.872\right] + 0.5P\left[100\mathrm{n}_1(k) < -1.128\right] \\ &= 0.5\left[1 - Q(8.72)\right] + 0.5Q(11.28) \\ &\approx 0.5\end{aligned}$$

Thus, the BER of the desired signal is essentially 0.5, which means that the desired user is totally dominated by the interference in this particular near-far environment.

Power Control in CDMA

Because the *near-far* problem is a direct result of difference in user signal powers at the receiver, one effective approach to overcome the *near-far* effect is to increase the power of the "far" users while decrease the power of the "near" users. This power balancing approach is known in CDMA as *power control*.

Power control assumes that all receivers are collocated. For example, cellular communications take place by connecting a number of mobile phones within each cell to a base station that serves the cell. All mobile phone transmissions within the cell are received and detected at the base station. The transmission from a mobile unit to the base station is known as the *uplink* or *reverse link*, as opposed to *downlink* or *forward link* when the base station transmits

to a mobile user. It is clear that the near-far effect does not occur during *downlink*. In fact, because multiple user transmissions can be perfectly synchronized, downlink CDMA can be easily made synchronous to maintain orthogonality. Also at each mobile receiver, all signal transmissions have equal channel gain because all originate from the same base station. Neither near-far condition can be satisfied. For this reason, CDMA mobile users in downlink do not require *power control* or other means to combat strong MAI.

When CDMA is used on the *uplink* to enable multiple mobile users to transmit their signals to the base station, the near-far problem will often occur. By adopting *power control*, the base station can send instructions to the mobile phones to increase or to decrease their transmission powers. The goal is for all user signals to arrive at the base station receivers with similar power levels despite their different channel gains. In other words, a constant value of $|g_i|^2 P_i$ is achieved because *power control* via receiver feedback provides instructions to the mobile transmitters.

One of the major second-generation cellular standards, cdmaOne (also known as IS-95), pioneered by Qualcomm, is a DSSS CDMA system. It applies *power control* to overcome the near-far problem at base station receivers.

Power control takes two forms: *open loop* and *closed loop*. Under open-loop power control, a mobile station adjusts its power based on the strength of the signal it receives from the base station. This presumes that a reciprocal relationship exists between forward and reverse links, an assumption that may not hold if the links operate in different frequency bands. As a result, closed-loop power control is often required because the base station can order the mobile station to change its transmitted power.

Near-Far Resistance

An important concept of near-far resistance was defined by S. Verdú.[10] The main objective is to determine whether a CDMA receiver can overcome the MAI by simply increasing the signal-to-noise ratio $E_b/\mathcal{N}$. A receiver is defined as *near-far resistant* if, for every user in the CDMA system, there exists a nonzero γ such that no matter how strong the interferences are, the probability of bit error $P_b^{(i)}$ as a function of $E_b/\mathcal{N}$ satisfies

$$\lim_{\mathcal{N}\to 0} \frac{P_b^{(i)}(E_b/\mathcal{N})}{Q(\sqrt{\gamma \cdot 2E_b/\mathcal{N}})} < +\infty$$

This means that a near-far resistant receiver should have no BER floor as $\mathcal{N} \to 0$. Our analysis of the conventional matched filter receiver based even on Gaussian approximation has demonstrated the lack of near-far resistance by the conventional single-user receiver. Although power control alleviates the near-far effect, it does not make the conventional receiver near-far resistant. To achieve near-far resistance, we will need to apply multiuser detection receivers to jointly detect all user symbols instead of approximating the sum of interferences as additional Gaussian noise.

11.7 MULTIUSER DETECTION (MUD)

Multiuser detecion (MUD) is an alternative to power control as a tool against near-far effect. Unlike power control, MUD can equalize the received signal power without feedback from the receivers to the transmitters. Instead, MUD is a centralized receiver that aims to jointly detect all user signals despite the difference of the received signal strength.

For MUD, the general assumption is that the receiver has access to all M signal samples of Eq. (11.31). In addition, the receiver has knowledge of the following information:

1. User signal strengths $g_i\sqrt{P_i}$.
2. Spreading sequence cross-correlation $R_{i,j}(k)$.
3. Statistics of the noise samples $\mathrm{n}_i(k)$.

To explain the different MUD receivers, it is more convenient to write Eq. (11.31) in vector form:

$$\begin{bmatrix} \mathrm{r}_k^{(1)} \\ \mathrm{r}_k^{(2)} \\ \vdots \\ \mathrm{r}_k^{(M)} \end{bmatrix} = \begin{bmatrix} R_{1,1}(k) & R_{1,2}(k) & \cdots & R_{1,M}(k) \\ R_{2,1}(k) & R_{2,2}(k) & \cdots & R_{1,M}(k) \\ \vdots & \vdots & \cdots & \vdots \\ R_{M,1}(k) & R_{M,2}(k) & \cdots & R_{M,M}(k) \end{bmatrix} \begin{bmatrix} g_1\sqrt{P_1} & & & \\ & g_2\sqrt{P_2} & & \\ & & \ddots & \\ & & & g_M\sqrt{P_M} \end{bmatrix} \begin{bmatrix} s_k^{(1)} \\ s_k^{(2)} \\ \vdots \\ s_k^{(M)} \end{bmatrix} + \begin{bmatrix} \mathrm{n}_1(k) \\ \mathrm{n}_2(k) \\ \vdots \\ \mathrm{n}_M(k) \end{bmatrix} \tag{11.39a}$$

We can define the vectors

$$\mathbf{r}_k = \begin{bmatrix} \mathrm{r}_k^{(1)} \\ \mathrm{r}_k^{(2)} \\ \vdots \\ \mathrm{r}_k^{(M)} \end{bmatrix} \qquad \boldsymbol{s}_k = \begin{bmatrix} s_k^{(1)} \\ s_k^{(2)} \\ \vdots \\ s_k^{(M)} \end{bmatrix} \qquad \mathbf{n}_k = \begin{bmatrix} \mathrm{n}_1(k) \\ \mathrm{n}_2(k) \\ \vdots \\ \mathrm{n}_M(k) \end{bmatrix} \tag{11.39b}$$

We can also define matrices

$$\boldsymbol{R}_k = \begin{bmatrix} R_{1,1}(k) & R_{1,2}(k) & \cdots & R_{1,M}(k) \\ R_{2,1}(k) & R_{2,2}(k) & \cdots & R_{1,M}(k) \\ \vdots & \vdots & \cdots & \vdots \\ R_{M,1}(k) & R_{M,2}(k) & \cdots & R_{M,M}(k) \end{bmatrix} \tag{11.39c}$$

$$\boldsymbol{D} = \begin{bmatrix} g_1\sqrt{P_1} & & & \\ & g_2\sqrt{P_2} & & \\ & & \ddots & \\ & & & g_M\sqrt{P_M} \end{bmatrix} \tag{11.39d}$$

Then the M output signal samples available for MUD can be written as

$$\mathbf{r}_k = \boldsymbol{R}_k \cdot \boldsymbol{D} \cdot \boldsymbol{s}_k + \mathbf{n}_k \tag{11.39e}$$

Notice that the noise vector $\mathbf{n}_k$ is Gaussian with zero mean and covariance matrix [Eq. (11.29)]

$$\overline{\mathbf{n}_k \left(\mathbf{n}_k^*\right)^T} = \frac{\mathcal{N}}{2}\boldsymbol{R}_k \tag{11.40}$$

The goal of MUD receivers is to determine the unknown user data vector $\boldsymbol{s}_k$ based on the received signal vector value $\mathbf{r}_k = \boldsymbol{r}_k$. Based on the system model of Eq. (11.39), different joint MUD receivers can be derived according different criteria.

To simplify our notation in MUD discussions, we denote $\boldsymbol{A}^*$ as the conjugate of matrix $\boldsymbol{A}$ and $\boldsymbol{A}^T$ as the transpose of matrix $\boldsymbol{A}$. Moreover, we denote the conjugate transpose of matrix $\boldsymbol{A}$ as

$$\boldsymbol{A}^H = \left(\boldsymbol{A}^*\right)^T$$

The conjugate transpose of a matrix is also known as its Hermitian.

Optimum MUD: Maximum Likelihood Receiver

The optimum MUD based on the signal model of Eq. (11.39) is the maximum likelihood detector (MLD) under the assumption of equally likely input symbols. As discussed in Sec. 10.6, the optimum receiver with minimum probability of symbol error is the MAP receiver

$$\boldsymbol{s}_k = \arg \max_{\boldsymbol{s}_k} p\left(\boldsymbol{s}_k | \boldsymbol{r}_k\right) \tag{11.41a}$$

If all possible values of $\boldsymbol{s}_k$ are equally likely, then the MAP detector reduces to the *maximum likelihood* detector (or MLD)

$$\boldsymbol{s}_k = \arg \max_{\boldsymbol{s}_k} p\left(\boldsymbol{r}_k | \boldsymbol{s}_k\right) \tag{11.41b}$$

Because the noise vector $\mathbf{n}_k$ is jointly Gaussian with zero mean and covariance matrix $0.5\mathcal{N}\boldsymbol{R}_k$, we have

$$p\left(\boldsymbol{r}_k | \boldsymbol{s}_k\right) = (\pi\mathcal{N})^{-M} \left[\det\left(\boldsymbol{R}_k\right)\right]^{-1} \exp\left[-\frac{1}{\mathcal{N}} \left(\boldsymbol{r}_k - \boldsymbol{R}_k \boldsymbol{D} \boldsymbol{s}_k\right)^T \boldsymbol{R}_k^{-1} \left(\boldsymbol{r}_k - \boldsymbol{R}_k \boldsymbol{D} \boldsymbol{s}_k\right)^*\right] \tag{11.42}$$

The MLD receiver can be implemented as

$$\begin{aligned} \max_{\boldsymbol{s}_k} p\left(\boldsymbol{r}_k | \boldsymbol{s}_k\right) &\Longleftrightarrow \min_{\boldsymbol{s}_k} \left(\boldsymbol{r}_k - \boldsymbol{R}_k \boldsymbol{D} \boldsymbol{s}_k\right)^T \boldsymbol{R}_k^{-1} \left(\boldsymbol{r}_k - \boldsymbol{R}_k \boldsymbol{D} \boldsymbol{s}_k\right)^* \\ &\Longleftrightarrow \min_{\boldsymbol{s}_k} \left\| \boldsymbol{R}_k^{-1/2} \left(\boldsymbol{r}_k - \boldsymbol{R}_k \boldsymbol{D} \boldsymbol{s}_k\right) \right\|^2 \end{aligned} \tag{11.43}$$

The maximum likelihood MUD receiver is illustrated in Fig. 11.13.

Thus, the maximum likelihood MUD receiver must calculate and compare the values of

$$\left\| \boldsymbol{R}_k^{-1/2} \left(\boldsymbol{r}_k - \boldsymbol{R}_k \boldsymbol{D} \boldsymbol{s}_k\right) \right\|^2$$

for all possible choices of the unknown user symbol vector $\mathbf{s}_k$. If each user uses 16-QAM to modulate its data, the complexity of this optimum MUD receiver requires 16^M evaluations of Eq. (11.43). It is evident that the optimum maximum likelihood MUD has a rather high complexity. Indeed, the computational complexity increases exponentially with the number of CDMA users.[10] This is the price paid for this *optimum* and near-far resistant CDMA receiver.[10]

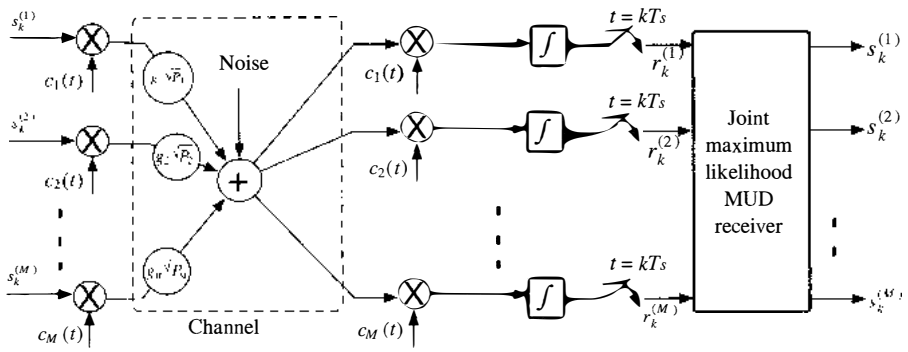


Figure 11.13 Maximum likelihood multiuser detection (MUD) receiver.

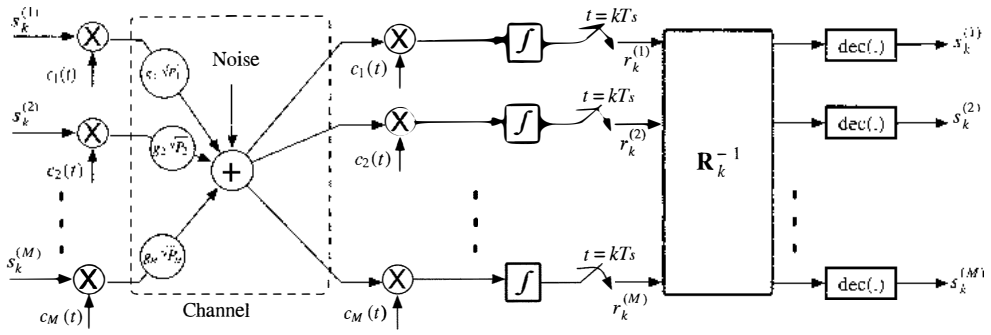


Figure 11.14 Decorrelator MUD receiver.

Decorrelator Receiver

The high complexity of the maximum likelihood MUD receiver reduces its attractiveness in practical applications. To bring down the computational cost, several low-complexity and suboptimum MUD receivers have been proposed. The decorrelator MUD is a linear method that simply uses matrix multiplication to remove the MAI among different users. Based on Eq. (11.39), the MAI among different users is caused by the nondiagonal correlation matrix $\boldsymbol{R}_k$. Thus, the MAI effect can be removed by premultiplying $\boldsymbol{r}_k$ with the pseudoinverse of $\boldsymbol{R}_k$ to "decorrelate" the user signals.

$$\boldsymbol{R}_k^{-1} \cdot \mathbf{r}_k = \boldsymbol{D}\boldsymbol{s}_k + \boldsymbol{R}_k^{-1} \cdot \mathbf{n}_k \tag{11.44}$$

This decorrelating operation leaves only the noise term $\boldsymbol{R}_k^{-1}\,\mathbf{n}_k$ that can affect the user signal. A QAM hard-decision device can be applied to detect the user symbols

$$\hat{\boldsymbol{s}}_k = \text{dec}\left(\boldsymbol{R}_k^{-1}\,\mathbf{r}_k\right) \tag{11.45}$$

Figure 11.14 is the block diagram of a decorrelator MUD receiver. Since the major operation of a decorrelating MUD receiver lies in the matrix multiplication of $\boldsymbol{R}_k^{-1}$, the computational complexity increases only in the order of $O(M^2)$. The decorrelator receiver is near-far resistant, as detailed by Lupas and Verdú.[11]

Minimum Mean Square Error (MSE) Receiver

The drawback of the decorrelator MUD receiver lies in the noise transformation by $\boldsymbol{R}_k^{-1}\,\mathbf{n}_k$. In fact, when the correlation matrix $\boldsymbol{R}_k$ is ill conditioned, the noise transformation has the negative effect of noise amplification. To mitigate this risk, a different and more robust MUD[12, 13] is to minimize the mean square error by applying a good linear MUD receiver by finding the

optimum matrix $\boldsymbol{G}_k$:

$$\min_{\boldsymbol{G}} E\{||s_k - \boldsymbol{G}\,\mathbf{r}_k||^2\} \tag{11.46}$$

This $\boldsymbol{G}$ still represents a linear detector. Once $\boldsymbol{G}$ has been determined, the MUD receiver simply takes a hard decision on the linearly transformed signal, that is,

$$\hat{s}_k = \text{dec}\,(\boldsymbol{G}\,\mathbf{r}_k) \tag{11.47}$$

The optimum matrix $\boldsymbol{G}$ can be determined by applying the principle of orthogonality [Eq. (8.84), Sec. 8.5]. The principle of orthogonality requires that the error vector

$$s_k - \boldsymbol{G}\,\mathbf{r}_k$$

be orthogonal to the received signal vector $\mathbf{r}_k$. In other words,

$$\overline{(s_k - \boldsymbol{G}\,\mathbf{r}_k)\,\mathbf{r}_k^H} = 0 \tag{11.48}$$

Thus, the optimum receiver matrix $\boldsymbol{G}$ can be found as

$$\boldsymbol{G} = \overline{s_k \mathbf{r}_k^H}\left[\overline{\mathbf{r}_k\,\mathbf{r}_k^H}\right]^{-1} \tag{11.49}$$

Because the noise vector $\mathbf{n}_k$ and the signal vector s_k are independent,

$$\overline{s_k \mathbf{n}_k^H} = \mathbf{0}_{M\times M}$$

is their cross-correlation.

In addition, we have earlier established equalities

$$\overline{s_k s_k^H} = \boldsymbol{I}_{M\times M} \qquad \overline{\mathbf{n}_k \mathbf{n}_k^H} = \frac{\mathcal{N}}{2}\boldsymbol{R}_k$$

where we use $\boldsymbol{I}_{M\times M}$ to denote the $M \times M$ identity matrix. Hence, we have

$$\overline{\boldsymbol{r}_k \boldsymbol{r}_k^H} = \boldsymbol{R}_k \boldsymbol{D}\boldsymbol{D}^H \boldsymbol{R}_k^H + \frac{\mathcal{N}}{2}\boldsymbol{R}_k \tag{11.50a}$$

$$\overline{s_k \boldsymbol{r}_k^H} = \boldsymbol{D}^H \boldsymbol{R}_k^H \tag{11.50b}$$

The optimum linear receiver matrix is therefore

$$\boldsymbol{G}_k = \boldsymbol{D}^H \boldsymbol{R}_k^H \left(\boldsymbol{R}_k \boldsymbol{D}\boldsymbol{D}^H \boldsymbol{R}_k^H + \frac{\mathcal{N}}{2}\boldsymbol{R}_k\right)^{-1} \tag{11.51}$$

It is clear that when the channel noise is zero (i.e., $\mathcal{N} = 0$), then the optimum matrix given by Eq. (11.51) degenerates into

$$\boldsymbol{G}_k = \boldsymbol{D}^H \boldsymbol{R}_k^H \left(\boldsymbol{R}_k \boldsymbol{D}\boldsymbol{D}^H \boldsymbol{R}_k^H\right)^{-1} = (\boldsymbol{R}_k \boldsymbol{D})^{-1}$$

which is essentially the decorrelator receiver.

The MMSE linear MUD receiver is shown in Fig. 11.15. Similar to the decorrelator receiver, the major computational requirement comes from the matrix multiplication of $\boldsymbol{G}_k$. The MMSE linear receiver is also near-far resistant.[11]

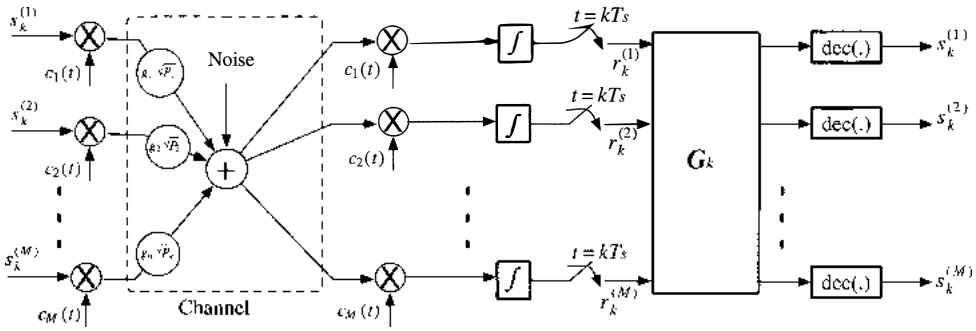


Figure 11.15 Minimum mean square error MUD receiver.

Decision Feedback Receiver

We note that both the decorrelator and the MMSE MUD receivers apply linear matrix processing. Hence, they are known as linear receivers with low complexity. On the other hand, the optimum MUD receiver is nonlinear but requires much higher complexity. There is also a very popular suboptimum receiver that is nonlinear. This method is based on the concept of successive interference cancellation, known as the **decision feedback** MUD receiver.[14, 15]

The main *motivation* behind the decision feedback MUD receiver lies in the fact that in a near-far environment, **not all** users suffer equally. In a near-far environment, the stronger signals are actually winners, whereas the weaker signals are losers. In fact, when a particular user has a strength $\sqrt{P_\ell} g_\ell$ that is stronger than those of all other users, its conventional matched filter receiver can in fact deliver better performance than is possible in an environment of equal strength. Hence, it would make sense to rank the received users in the order of their individual strength measured by $\{P_i g_i^2\}$. The strongest user QAM symbols can then be detected first, using only the conventional matched filter receivers designed for single users. Once the strongest user symbols is known, its interference effects on the remaining user signals can be canceled. By canceling the strongest user symbol from the received signal vectors, there are only $M-1$ unknown user symbols for detection. Among them, the next strongest user signal can be detected more accurately after the strongest interference has been removed. Hence, its effect can also subsequently be canceled from received signals, to benefit the $M-2$ remaining user symbols, and so on. Finally, the weakest user signal will be detected last, after all the MAI has been canceled.

Clearly, the decision feedback MUD receiver relies on the successive interference cancellation of stronger user interferences for the benefit of weaker user signal detection. For this reason, the decision feedback MUD receiver is also known as the successive interference cancellation (SIC) receiver. The block diagram of the decision feedback MUD receiver appears in Fig. 11.16. Based on Eq. (11.31), the following steps summarize the SIC receiver:

Decision Feedback MUD

Step 1. Rank all user signal strengths $\{P_i g_i^2\}$. Without loss of generality, we assume that

$$P_1 g_1^2 > P_2 g_2^2 > \cdots > P_{M-1} g_{M-1}^2 > P_M g_M^2$$

Let

$$y_1^{(i)} = r_k^{(i)}$$

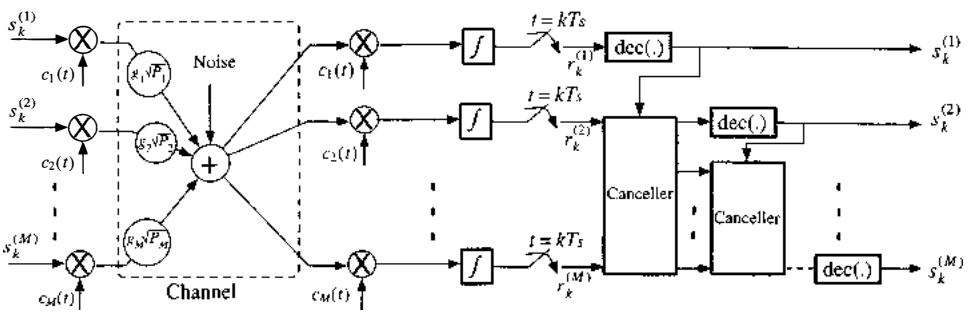


Figure 11.16 Decision feedback MUD receiver based on successive interference cancellation (assuming that all M users are ranked in the order of descending gains).

and

$$\ell = 1$$

Step 2. Detect the ℓth (strongest) user symbol via

$$\hat{s}_k^{(\ell)} = \text{dec}\left(y_\ell^{(\ell)}\right)$$

Step 3. Cancel the first (strongest) user interference from the received signals

$$y_{\ell+1}^{(i)} = y_\ell^{(i)} - g_\ell \sqrt{\boldsymbol{P}_\ell} R_{i,\ell}(k) \hat{s}_k^{(\ell)} \qquad i = \ell + 1, \ldots, M$$

Step 4. Let $\ell = \ell + 1$ and repeat step 2 until $\ell = M$.

A decision feedback MUD receiver requires very little computation, since the interference cancellation step requires only $O(M^2)$ complexity. It is a very sensible and low-complexity receiver. Given correct symbol detection, strong interference cancellation from received weak signals completely eliminates the near-far problem. The key **drawback** or weakness of the decision feedback receiver lies in the effect of error propagation. Error propagation takes place when, in step 2, a user symbol $s_k^{(\ell)}$ is detected incorrectly. As a result, this erroneous symbol used in the interference cancellation of step 3 may in fact strengthen the MAI. This leads to the probability of more decision errors of the subsequent user symbol, $s_k^{(\ell+1)}$, which in turn can cause more decision errors. Analysis on the effect of error propagation can be found in Refs. 14 and 15.

11.8 MODERN PRACTICAL DSSS CDMA SYSTEMS

Since the 1990s, many important commercial applications have emerged for spread spectrum, including cellular telephones, personal communications, and position location. Here we discuss several popular applications of CDMA technology to illustrate the benefits of spread spectrum.

11.8.1 CDMA in Cellular Phone Networks

Cellular Networks

The cellular network divides a service area into smaller geographical **cells** (Fig. 11.17). Each cell has a **base station** tower to connect with mobile users it serves. All base stations are wired

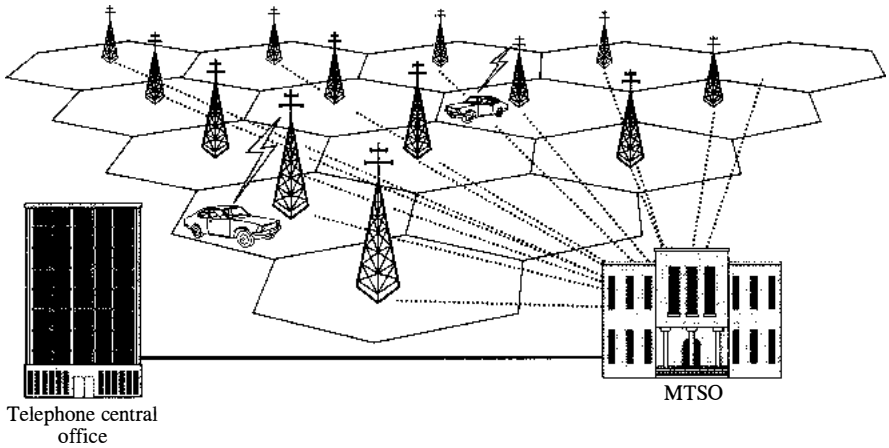


Figure 11.17 Cellular telephone system.

to the **mobile telephone switching office** (**MTSO**), which in turn is wired to the telephone central office. A caller communicates via radio channel to its base station, which sends the signal to the MTSO. The MTSO connects to the receiver either via the land-based telephone system or via another base station. As the caller moves from one cell to another, a *handoff* process takes place. During handoff, the MTSO automatically switches the user to an available channel in the new cell while the call is in progress. The handoff is so rapid that users usually do not notice it.

The true ingenuity of the cellular network lies in its ability to reuse the same frequency band in multiple cells. Without cells, high-powered transmitters can be used to cover an entire city. But this would allow a frequency channel to be used only by one user in the city at any moment. This posed serious limitations on the number of channels and simultaneous users. The limitation is overcome in the cellular scheme by reusing the same frequencies in all the cells except those immediately adjacent. This is possible because the transmitted powers are kept small enough to prevent the signals from one cell from reaching beyond the immediately adjacent cells. We can accommodate any number of users by increasing the number of cells as we reduce the cell size and the power levels correspondingly.

The 1G (first-generation) analog cellular schemes use audio signal to modulate an FM signal with transmission bandwidth 30 kHz. This wideband FM signal results in a good SNR but is highly inefficient in bandwidth usage and frequency reuse. The 2G (second-generation) cellular systems are all digital. Among them, the **GSM** and **cdmaOne** are two of the most widely deployed cellular systems. GSM adopts a TDMA technology through which eight users share a 200 kHz channel. The competing technology of cdmaOne (known earlier as IS-95) is a DSSS system.

Why CDMA in Cellular Systems

Although spread spectrum is inherently well suited against narrowband interferences and affords a number of advantages in the areas of networking and handoff, the key characteristic underlying the broad application of CDMA for wireless cellular systems is the potential for improved spectral utilization. The capacity for improvement has two key sources. First, the use of CDMA allows improved frequency reuse. Narrowband systems cannot use the same transmission frequency in adjacent cells because of the potential for interference. CDMA has inherent resistance to interference. Although users using different spreading codes from adjacent cells will contribute to the total interference level, their contribution will be significantly less than the interference from the same cell users. This leads to a much improved frequency

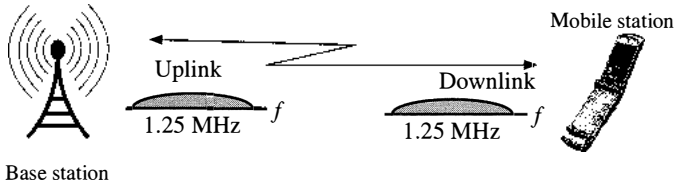


Figure 11.18 RF bandwidth requirements for IS-95 uplink and downlink.

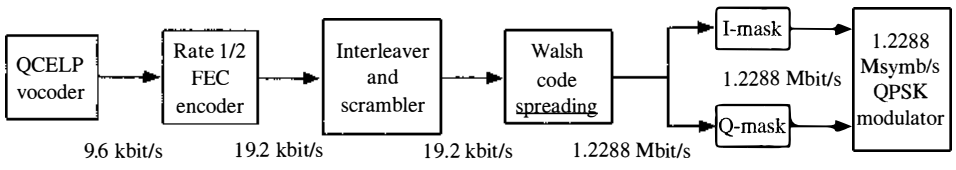


Figure 11.19 Forward link modulation and Walsh code spreading of cdmaOne (IS-95).

reuse efficiency. In addition, CDMA provides better overall capacity when the data traffic load is dynamic. This is because users in a lightly loaded CDMA system would have a lower interference level and better performance, whereas TDMA users with fixed channel bandwidth do not enjoy such benefit.

CDMA Cellular System: cdmaOne (IS-95)

The first commercially successful CDMA system in cellular applications was developed by the Electronic Industries Association (EIA) as interim standard-95 (IS-95). Now under the official name of **cdmaOne**, it employs DSSS by adopting 1.2288-Mchip/s spreading sequences on both uplink and downlink. The uplink and downlink transmissions both occupy 1.25 MHz of RF bandwidth, as illustrated in Fig. 11.18.

The QCELP (Qualcomm code-excited linear prediction) vocoder is used for voice encoding. Since the voice coder exploits gaps and pauses in speech, the data rate is variable from 1.2 to 9.6 kbit/s. To keep the symbol rate constant, whenever the bit rate falls below the peak bit rate of 9.6 kbit/s, repetition code is used to fill the gaps. For example, if the output of the voice coder (and subsequently the convolutional coder) falls to 2.4 kbit/s, the output is repeated three more times before it reaches the interleaver. The transmitter of cdmaOne takes advantage of this repetition time by reducing the output power during three out of the four identical symbols by at least 20 dB. In this way, the multiple-access interference is diminished. This "voice activity gating" reduces MAI and increases overall system capacity.

The modulation of cdmaOne uses QPSK on the downlink, and the uplink uses a variant of QPSK known as the offset QPSK (or OQPSK). There are other important differences between the forward and reverse links. Figure 11.19 outlines the basic operations of spreading and modulation on the forward link. After a rate 1/2 convolutional error correction code, the voice data becomes 19.2 kbit/s. Interleaving then shuffles the data to alleviate burst error effects, and long-code scrambling provides some nominal privacy protection. The data rate remains at 19.2 kbit/s before being spread by a length 64 Walsh-Hadamard short-code to result in a sequence of rate 1.2288 Mbit/s. Because forward link uses synchronous transmissions, in the absence of channel distortions, there can be as many as 64 orthogonal data channels, each using a distinct Walsh-Hadamard code. Both the in-phase (I) and the quadrature (Q) components of the QPSK modulations carry the same data over the 1.25 MHz bandwidth, although different masking codes are applied to I and Q.

The performance of the reverse link is of greater concern for two reasons. First, as discussed earlier, the reverse link is subject to near-far effects. Second, since all transmissions on the forward link originate at the same base station, it uses the orthogonal Walsh-Hadamard

spreading codes to generate synchronous signals with zero cross-correlation. Reverse-link does not enjoy this luxury. For this reason, more powerful error correction (rate 1/3) is employed on the reverse link. Still, like the forward link, the raw QCELP vocoder bit rate is 9.6 kbit/s, which is eventually spread to 1.2288 Mchip/s over a 1.25 MHz bandwidth.

As mentioned earlier, the near-far problem needs to be addressed when spread spectrum is utilized in mobile communications. To combat this problem, IS-95 uses power control. On the forward link there is a subchannel for power control purposes. Every 1.25 ms, the base station receiver estimates the signal strength of the mobile unit. If it is too high, the base transmits a **1** on the subchannel. If it is too low, it transmits a **0**. In this way, the mobile station adjusts its power based on the 800 bit/s power control signal to reduce interference to other users.

3G Cellular Services[16–19]

In the new millennium, wireless service providers are shifting their voice-centric 2G cellular systems to the next-generation (3G) wireless systems, which are capable of supporting high-speed data transmission and internet connection. For this reason, the International Mobile Telecommunications-2000 standard (IMT-2000) is the global standard for third-generation wireless communications. IMT-2000 provides a framework for worldwide wireless access of fixed and mobile wireless access systems. The goal is to provide wireless cellular coverage up to 144 kbit/s for high-speed mobile, 384 kbit/s for pedestrian, and 2.048 Mbit/s for indoor users. Among the 3G standards, there are three major wireless technologies based on CDMA DSSS, namely, the two competing versions of wideband CDMA from the 3rd Generation Partnership Project (3GPP) and the 3rd Generation Partnership Project 2 (3GPP2), plus the TD-SCDMA from the 3GPP for China.

Because 3G cellular systems continue to use the existing cellular band, a high data rate for one user means a reduction of service for other active CDMA users within the same cell. Otherwise, given the limited bandwidth, it is impossible to serve the same number of active users as in **cdmaOne** while supporting data rate as high as 2.048 Mbit/s. Thus, the data rate to and from the mobile unit must be variable according to the data traffic intensity within the cell. Since most data traffic patterns (including internet usage) tend to be bursty, variable rate data service offered by 3G cellular is suitable for such applications.

Unlike FDMA and TDMA, CDMA provides a perfect environment for variable data rate and requires very simple modifications. While FDMA and TDMA would require grouping multiple frequency bands or time slots dynamically to support variable rate, CDMA needs to change only the spreading gain. In other words, at higher data rates, a CDMA transmitter can use a lower spreading factor. In this mode, its MAI to other users is high, and fewer such users can be accommodated. At lower data rates, the transmitter uses a larger spreading factor, thus allowing more users to transmit.

In 3GPP2's CDMA2000 standard, there are two ratio transmission modes: 1xRTT utilizing one 1.25 MHz band and 3xRTT that aggregates three 1.25 MHz bands. On 1xRTT forward link, the maximum data rate is 307.2 kbit/s with a spreading gain of 4. Thus, the chip rate is still 1.2288 Mchip/s. A more recent 3GPP2 release is called CDMA 2000 1x EV-DO revision A, where EV-DO stands for "evolution data-optimized." It can support a peak data rate of 3.1 Mbit/s on the forward link of 1.25 MHz bandwidth. It does so by applying adaptive coding and adaptive modulations, including QPSK, 8-PSK, and 16-QAM. At the peak rate, the spreading gain is 1 (i.e., no spreading).

At the same time, the WCDMA by 3GPP applies similar ideas. Unlike CDMA2000, WCDMA has a standard bandwidth of 5 MHz. When spreading is used, the chip rate is 4.096 Mchip/s. On downlink, the variable spreading factor of 3GPP WCDMA ranges from 512 to 4. With QPSK modulation, this provides a variable data rate from 16 kbit/s to 2.048 Mchip/s.

Similar to CDMA2000, 3GPP WCDMA also has a counterpart to EV-DO known as *high-speed packet access* (HSPA). On downlink, the recent HSPA release (Release 6) achieves the peak rate of 14.4 Mbit/s. However, existing deployments can support a peak rate of only 7.2 Mbit/s. Still, at this high rate, most data users would be quite satisfied, with the exception perhaps of high-definition TV viewers.

Power Control vs. MUD

It is interesting to note that despite intense academic research interest in multiuser CDMA receivers (in the 1980s and 1990s), all cellular CDMA systems described here rely on power control to combat the near-far problem. The reason lies in the fact that power control is quite simple to implement and has proven to be very effective. On the other hand, MUD receivers require more computational complexity. To be effective, MUD receivers also require too much channel and signal information about all active users. Moreover, MUD receivers alone cannot completely overcome the disparity of performance in a near-far environment.

11.8.2 CDMA in the Global Positioning System (GPS)

What Is GPS?

The Global Positioning System (GPS) is the only fully functional global satellite navigation system. Utilizing a constellation of at least 24 satellites in medium Earth orbit to transmit precise RF signals, the system enables a GPS receiver to determine its location, speed, and direction.

A GPS receiver calculates its position based on its distances to three or more GPS satellites. Measuring the time delay between transmission and reception of each GPS microwave signal gives the distance to each satellite, since the signal travels at a known speed. The signals also carry information about the satellites' location. By determining the position of, and distance to, at least three satellites, the receiver can compute its position using triangularization. Receivers typically do not have perfectly accurate clocks and therefore track one or more additional satellites to correct the receiver's clock error.

Each GPS satellite continuously broadcasts its (navigation) message via BPSK at the rate of 50 bit/s. This message is transmitted by means of two CDMA spreading codes: one for the coarse/acquisition (C/A) mode and one for the precise (P) mode (encrypted for military use). The C/A spreading code is a PN sequence with period of 1023 chips sent at 1.023 Mchip/s. The spreading gain is $L = 20,460$. Most commercial users access only the C/A mode.*

Originally developed for the military, GPS is now finding many uses in civilian life such as marine, aviation, and automotive navigation, as well as surveying and geological studies. GPS allows a person to determine the time and the person's precise location (latitude, longitude, and altitude) anywhere on earth with an accuracy of inches. The person can also find the velocity with which he or she is moving. GPS receivers have become small and inexpensive enough to be carried by just about everyone in cars and boats. Handheld GPS receivers are plentiful and have even been incorporated into popular cellular phone units.

How Does GPS Work?

A GPS receiver operates by measuring its distance from a group of satellites in space, which are acting as precise reference points. Since the GPS system consists of 24 satellites, there will always be more than four orbiting bodies visible from anywhere on Earth. The 24 satellites

* The P spreading code rate is 10.23 Mchip/s with a spreading gain of $L = 204,600$. The P code period is 6.1871×10^{12} bits long. In fact, at the chip rate of 10.23 Mchip/s, the code period is one week long!

are located in six orbital planes at a height of 22,200 km. Each satellite circles the earth in 12 hours. The satellites are constantly monitored by the U.S. Department of Defense, which knows their exact locations and speeds at every moment. This information is relayed back to the satellites. All the satellites have atomic clocks of unbelievable precision on board and are synchronized to generate the same PN code at the same time. The satellites are continuously transmitting this PN code and the information about their locations and time. A GPS receiver on the ground is also generating the same PN code, although not in synchronism with that of the satellites. This is because of the necessity to make GPS receivers inexpensive. Hence, the timing of the PN code generated by the receiver will be off by an amount of α seconds (timing bias) from that of the PN code of the satellites.

To begin, let us assume that the timing bias $\alpha = 0$. By measuring the time delay between its own PN code and that received from one satellite, the receiver can compute its distance d from that satellite. This information places the receiver anywhere on a sphere of radius d centered at the satellite location (which is known), as shown in Fig. 11.20a. Simultaneous measurements from three satellites place the receiver on the three spheres centered at the three known satellite locations. The intersection of two spheres is a circle (Fig. 11.20b), and the intersection of this circle with the third sphere narrows down the location to just two points, as shown in Fig. 11.20c. One of these points is the correct location. But which one? Fortunately, one of the two points would give a ridiculous answer. The incorrect point may not be on Earth, or it may indicate an impossibly high receiver velocity. The computer in a GPS receiver has various techniques for distinguishing the correct point from the incorrect one.

In practice, the timing bias α is not zero. To solve this problem, we need a distance measurement from a fourth satellite. A user locates his or her position by receiving the signal from four of the possible 24 satellites, as shown in Fig. 11.20d. There are four unknowns, the coordinates in the three-dimensional space of the user along with a timing bias in the user's receiver. These four unknowns can be solved by using four range equations to each of the four satellites.

Since DSSS signals consist of a sequence of extremely short pulses, it is possible to measure their arrival times accurately. The GPS system can result in accuracies of 10 meters anywhere on Earth. The use of **differential GPS** can provide accuracy within centimeters. In this case we use one terrestrial location whose position is known exactly. Comparison of its known coordinates with those read by a GPS receiver (for the same location) gives us the error (bias) of the GPS system, which can be used to correct the errors of GPS measurements of other locations. This is based on the fact that satellite orbits are so high that any errors measured by one receiver will be almost exactly the same for any other receiver in the same locale. Differential GPS is currently used in such diverse applications as surveying, laying petroleum pipelines, aviation systems, marine navigation systems, and preparing highly accurate maps of everything from underground electric cabling to power poles.

Why Spread Spectrum in GPS?

The use of spread spectrum in the GPS system accomplishes three tasks. First, the signals from the satellites can be kept from unauthorized use. Second, and more important in a practical sense, the inherent processing gain of spread spectrum allows reasonable power levels to be used. Since the cost of a satellite is proportional to its weight, it is desirable to reduce the power required as much as possible. In addition, since each satellite must see an entire hemisphere, very little antenna gain is possible. For high accuracy, short pulses are required to provide fine resolution. This results in high spectrum occupancy and a received signal that is several decibels below the noise floor. Since range information needs to be calculated only about once every second, the data bandwidth need be only about 100 Hz. This is a natural match for

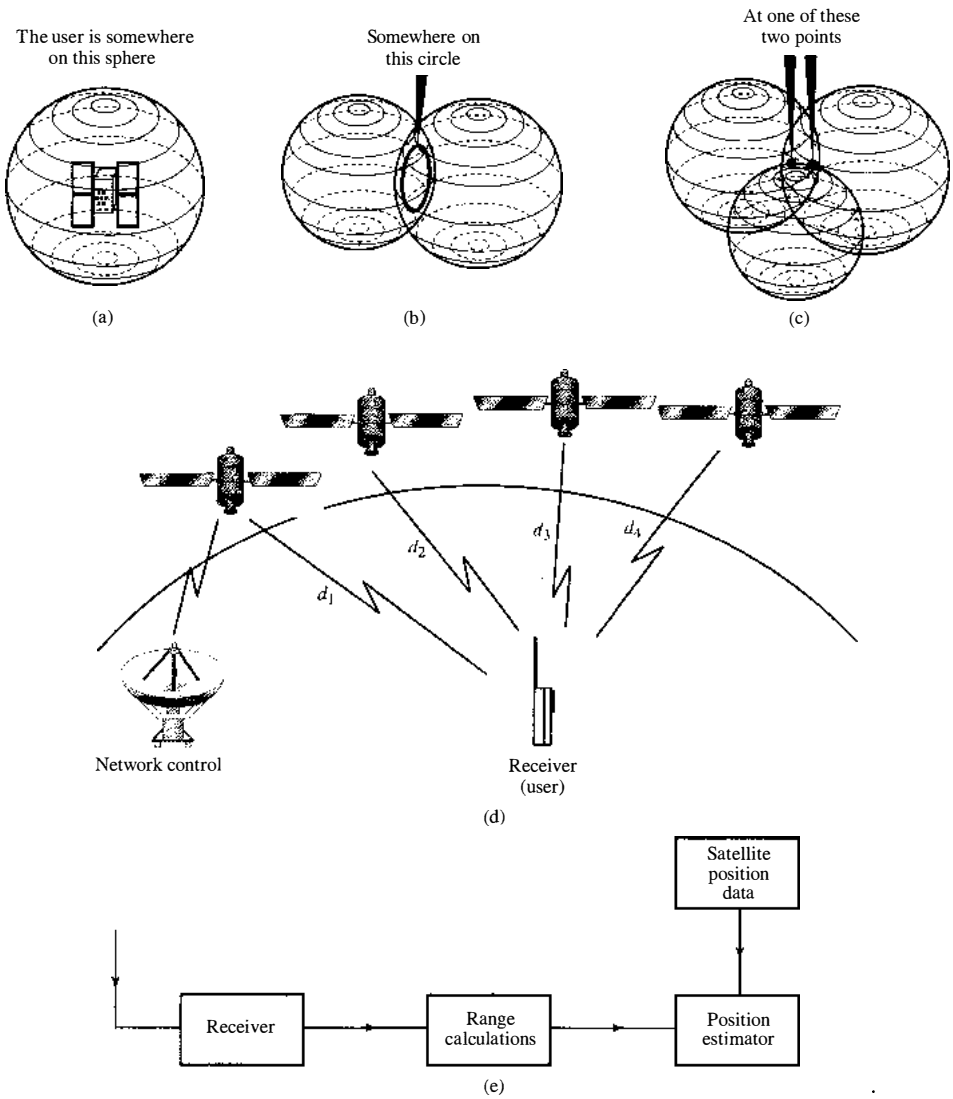


Figure 11.20
(a) Receiver location from one satellite measurement. (b) Location narrowed down by two satellite measurements. (c) Location narrowed down by three satellite measurements. (d) Practical global positioning system using four satellites. (e) Block diagram of a GPS receiver.

spread spectrum. Despreading the received signal in the receiver, in turn, yields a significant processing gain, thus allowing good reception at reasonable power levels. The third reason for spread spectrum is that each satellite can use the same frequency band, yet there is no mutual interference owing to the near orthogonality of each user's signal.

Each satellite circles the earth in 12 hours and emits two PN sequences modulated in phase quadrature at two frequencies. Two frequencies are needed to correct for the delay introduced by the ionosphere.

11.8.3 IEEE 802.11b Standard for Wireless LAN

IEEE 802.11b is a commercial standard developed for wireless local area networks (WLAN) to provide high-speed wireless connection to (typically) laptop computers.

Like its predecessor IEEE 802.11, IEEE 802.11b operates in the license-free ISM band of 2.4 to 2.4835 GHz. Similar to cellular networks, all laptop computers within a small coverage area form 1-to-1 communication links with an "access point." The access point is typically connected to the Internet via a high-speed connection that can deliver the traffics to and from laptop computers. In this way, the access point serves as a bridge between the computers and the Internet.

The ISM band is populated with signals from many unlicensed wireless devices such as microwave ovens, baby monitors, cordless phones, and wireless controllers. Hence, to transmit WLAN data, interference resistance against these unlicensed transmission is essential. For this reason, spread spectrum is a very effective technology.

The simple FSK used in the FHSS IEEE 802.11 provides up to 2 Mbit/s data rate and is simple to implement. Still, the link data rate is quite low. Because the laptop is a relative powerful device capable of supplying moderate levels of power and computation, it can support more complex and faster modulation. IEEE 802.11b eliminates the FHSS option and fully adopts the DSSS transmission. It pushes the data rate up to 11 Mbit/s, which is reasonably satisfactory to most computer connections.

Internationally, there are 14 DSSS channels defined over the ISM band, although not all channel are available in every country. In North America, there are 11 (overlapping) channels of bandwidth 22 MHz. The channel spacing is 5 MHz. Table 11.2 illustrates the 11 DSSS channels.

The chip rate of IEEE 802.11b is 11 MHz, and the spread spectrum transmission bandwidth is approximately 25 MHz. The 802.11b data rate can be 1, 2, 5.5, and 11 Mbit/s. For 1 and 2 Mbit/s data rates, differential BPSK and differential QPSK are used, respectively. At high data rates of 5.5 and 11 Mbit/s, a more sophisticated complementary code keying (CCK) was developed. The link data rate is established based on how good the channel condition is. The different spreading gains for the 802.11b DSSS modulation are given in Table 11.3.

Note that each access point may serve multiple links. Additionally, there may be more than one access point at a given area. To avoid spectral overlap, different network links must be separated by a minimum of five channel numbers. For example, channel 1, channel 6, and channel 11 can coexist without mutual interference. Often, a neighborhood may be very

TABLE 11.2
2.4 GHz ISM Channel Assignment in IEEE 802.11b

Channel	1	2	3	4	5	6	7	8	9	10	11
Center f, GHz	2.412	2.417	2.422	2.427	2.432	2.437	2.442	2.447	2.452	2.457	2.462

TABLE 11.3
Modulation Format and the Spreading Factor in IEEE 802.11b Transmission

Chip rate	11 MHz			
Data rate	1 Mbit/s	2 Mbit/s	5.5 Mbit/s	11 Mbit/s
Modulation	Differential BPSK	Differential QPSK	CCK	CCK
Spreading gain	11	11	2	1

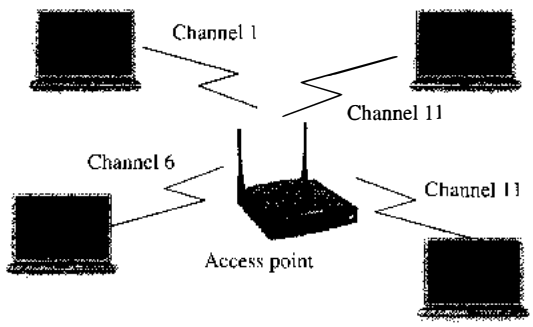


Figure 11.21
A wireless LAN with one access point and four computer nodes.

congested with multiple network coverage. Thus, spectral overlapping becomes unavoidable. When different networks utilize spectrally overlapping channels, signal collisions may take place. Data collisions are not resolved by radio transmitters and receivers (physical layer). Rather, network protocols are developed to force all competing networks and users to back off (i.e., to wait for a timer to expire before transmitting a finite data packet). In 802.11 WLAN, the timer is set to a random value based on a traffic-dependent uniform distribution. This backoff protocol to resolve data collisions in WLAN is known as the distributed coordinator function (DCF).

To allow multiple links to share the same channel, DCF forces each link to vacate the channel for a random period of time. This means that the maximum data rate of 11 Mbit/s cannot be achieved by any of the competing users. As shown in Fig. 11.21, the two computers both using channel 11 to connect to the access point must resort to DCF to reduce their access time and effectively lower their effective data rate. In this case, perfect coordination would be able to allocate 11 Mbit/s equally between the two users. This idealistic situation is really impossible under the distributed protocol of DCF. Under DCF, the maximum throughput of either user would be much lower than 5.5 Mbit/s.

IEEE 802.11b is without a question one of the most successful of the wireless standards that are responsible for opening up the commercial WLAN market. Nevertheless, to further improve the spectral efficiency and to increase the possible data rate, a new modulation scheme known as orthogonal frequency division multiplexing (OFDM) was incorporated into the follow-up standards of IEEE 802.11a and IEEE 802.11g.* The principles and analysis of OFDM will be discussed next in Chapter 12.

11.9 MATLAB EXERCISES

In this section of computer exercise, we provide some opportunities for readers to learn firsthand about the implementation and behavior of spread spectrum communications. We consider the cases of frequency hopping spread spectrum (FHSS), direct sequence spread spectrum (DSSS) or CDMA, and multiuser CDMA systems. We test the narrowband jamming effect on spread spectrum communications and the near-far effect on multiuser CDMA systems.

* IEEE 802.11g operates in the same ISM band as in IEEE 802.11b and must be backward compatible. Thus, IEEE 802.11g includes both the CDMA and the OFDM mechanisms. IEEE 802.11a, however, operates in the 5 GHz band and uses OFDM exclusively.

COMPUTER EXERCISE 11.1: FHSS FSK COMMUNICATION UNDER PARTIAL BAND JAMMING

The first MATLAB program, `Ex11_1.m`, implements an FHSS communication system that utilizes FSK and noncoherent detection receivers. By providing an input value of 1 (with jamming) and 0 (without jamming), we can illustrate the effect of FHSS against partial band jamming signals.

TABLE 11.4
Parameters Used in Computer Exercise 11.1

Number of users	$m = 1$
Spreading factor (number of FSK bands)	$L = 8$
Number of hops per symbol per bit	$L_h = 1$
Modulation	BFSK
Detection	Noncoherent
Partial band jamming	1 fixed FSK band

In `Ex11_1.m`, the parameters of the FHSS system are given in Table 11.4. When partial band jamming is turned on, a fixed but randomly selected FSK channel is blanked out by jamming. Under additive white Gaussian channel noise, the effect of partial band jamming on the FHSS user is shown in Fig. 11.22. Clearly, we can see that without jamming, the FHSS performance matches that of the FSK analysis in Sec. 11.1 and Chapter 10. When partial jamming is turned on, the BER of the FHSS system has a floor of $1/(2L)$ as shown in Eq. (11.4). As L increase from 4 to 8, and to 16, the performance clearly improves.

```
% MATLAB PROGRAM <Ex11_1.m>
% This program provides simulation for FHSS signaling using
% non-coherent detection of FSK.
% The jammer will jam 1 of the L frequency bands and
```

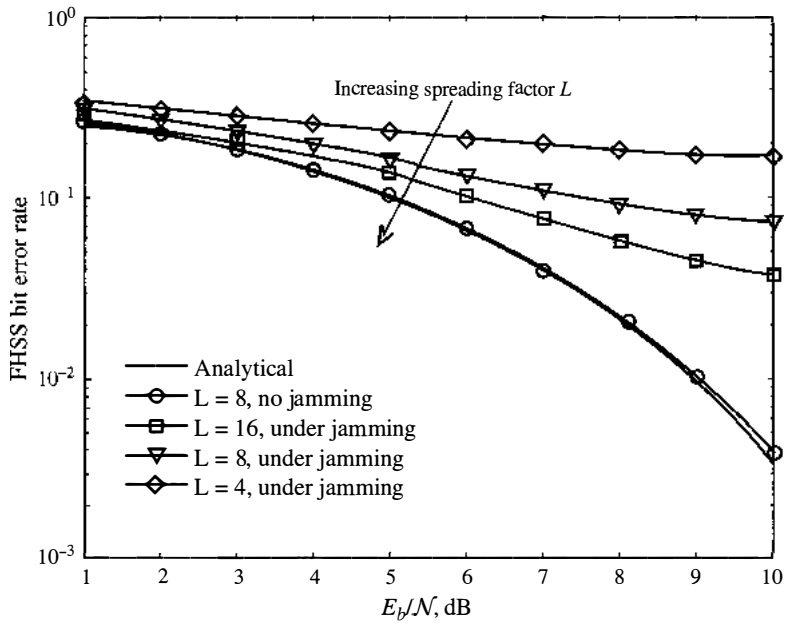


Figure 11.22 Performance of FHSS noncoherent detection under partial band jamming.

```
% can be turned on or off by inputting jamming=1 or 0
% Non-coherent MFSK detection
% only needs to compare the magnitude of each frequency bin.
%
clear;clf
n=10000;       %Number of data symbols in the simulation
L=8;            % Number of frequency bands
Lh=1;           % Number of hops per symbol (bit)
m=1;            % Number of users
% Generating information bits
s_data=round(rand(n,m));
% Turn partial band jamming on or off
jamming=input('jamming=? (Enter 1 for Yes, 0 for No)');
%  Generating random phases on the two frequencies
xbase1=[exp(j*2*pi*rand(Lh*n,1))];
xbase0=[exp(j*2*pi*rand(Lh*n,1))];
%  Modulating two orthogonal frequencies
xmodsig=[kron(s_data,ones(Lh,1)).*xbase1 kron((1-s_data),ones(Lh,1)).*xbas
clear xbase0 xbase1;
%  Generating a random hopping sequence nLh long
Phop=round(rand(Lh*n,1)*(L-1))+1;      % PN hopping pattern;
Xsiga=sparse(1:Lh*n,Phop,xmodsig(:,1));
Xsigb=sparse(1:Lh*n,Phop,xmodsig(:,2));
%  Generating noise sequences for both frequency channels
noise1=randn(Lh*n,1)+j*randn(Lh*n,1);
noise2=randn(Lh*n,1)+j*randn(Lh*n,1);
Nsiga=sparse(1:Lh*n,Phop,noise1);
Nsigb=sparse(1:Lh*n,Phop,noise2);
clear noise1 noise2 xmodsig;
BER=[];
BER_az=[];
%  Add a jammed channel (randomly picked)
if (jamming)
nch=round(rand*(L-1))+1;
Xsiga(:,nch)=Xsiga(:,nch)*0;
Xsigb(:,nch)=Xsigb(:,nch)*0;
Nsiga(:,nch)=Nsiga(:,nch)*0;
Nsigb(:,nch)=Nsigb(:,nch)*0;
end
% Generating the channel noise (AWGN)
for i=1:10,
    Eb2N(i)=i;                          %(Eb/N in dB)
    Eb2N_num=10^(Eb2N(i)/10);           % Eb/N in numeral
    Var_n=1/(2*Eb2N_num);               %1/SNR is the noise variance
    signois=sqrt(Var_n);                % standard deviation
    ych1=Xsiga+signois*Nsiga;           % AWGN complex channels
    ych2=Xsigb+signois*Nsigb;           % AWGN channels
    % Non-coherent detection

    for kk=0:n-1,
        Yvec1=[];Yvec2=[];
        for kk2=1:Lh,
            Yvec1=[Yvec1 ych1(kk*Lh+kk2,Phop(kk*Lh+kk2))];
            Yvec2=[Yvec2 ych2(kk*Lh+kk2,Phop(kk*Lh+kk2))];
```

```
        end
        ydim1=Yvec1*Yvec1';
        ydim2=Yvec2*Yvec2';
        dec(kk+1)=(ydim1>ydim2);
    end
    clear ych1 ych2;
    % Compute BER from simulation
    BER=[BER; sum(dec'~=s_data)/n];
    % Compare against analytical BER.
    BER_az=[BER_az; 0.5*exp(-Eb2N_num/2)];
end
figber=semilogy(Eb2N,BER_az,'k-',Eb2N,BER,'k-o');
set(figber,'Linewidth',2);
legend('Analytical BER', 'FHSS simulation');
fx=xlabel('E_b/N (dB)');
fy=ylabel('Bit error rate');
set(fx,'FontSize',11); set(fy,'Fontsize',11);
```

COMPUTER EXERCISE 11.2: DSSS TRANSMISSION OF QPSK

In this exercise, we performance a DSSS baseband system test under narrowband jamming. For spreading in this case, we apply the Barker code of length 11

```
pcode = [1 1 1 -1 -1 -1 1 -1 -1 1 -1]
```

for spreading because of its nice spectrum spreading property as a short code. We assume that the channel noises are additive white Gaussian. MATLAB program `Ex11_2b.m` provides the results of a DSSS user with QPSK modulation under a narrowband jamming.

```
% MATLAB PROGRAM <Ex11_2b.m>
% This program provides simulation for DS-CDMA signaling using
% coherent QAM detection.
% To illustrate the CDMA spreading effect, a single user is spread by
% PN sequence of different lengths.  Jamming is added as a narrowband;
% Changing spreading gain Lc;
clear;clf
Ldata=20000;        % data length in simulation; Must be divisible by 8
Lc=11;             % spreading factor vs data rate
                   % can also use the shorter Lc=7
%  Generate QPSK modulation symbols
data_sym=2*round(rand(Ldata,1))-1+j*(2*round(rand(Ldata,1))-1);
jam_data=2*round(rand(Ldata,1))-1+j*(2*round(rand(Ldata,1))-1);
% Generating a spreading code
pcode=[1 1 1 -1 -1 -1 1 -1 -1 1 -1]';
% Now spread
x_in=kron(data_sym,pcode);
%  Signal power of the channel input is 2*Lc
% Jamming power is relative
SIR=10;              % SIR in dB
Pj=2*Lc/(10^(SIR/10));

% Generate noise (AWGN)
noiseq=randn(Ldata*Lc,1)+j*randn(Ldata*Lc,1); % Power is 2
```

```
% Add jamming sinusoid sampling frequency is fc = Lc
jam_mod=kron(jam_data,ones(Lc,1));   clear jam_data;
jammer= sqrt(Pj/2)*jam_mod.*exp(j*2*pi*0.12*(1:Ldata*Lc)).'; %fj/fc=0.12.
clear jam_mod;
[P,x]=pwelch(x_in,[],[],[4096],Lc,'twoside');
figure(1);
semilogy(x-Lc/2,fftshift(P));
axis([-Lc/2 Lc/2 1.e-2 1.e2]);
grid;
xfont=xlabel('frequency (in unit of 1/T_s)');
yfont=ylabel('CDMA signal PSD');
set(xfont,'FontSize',11);set(yfont,'FontSize',11);
[P,x]=pwelch(jammer+x_in,[],[],[4096],Lc,'twoside');
figure(2);semilogy(x-Lc/2,fftshift(P));
grid;
axis([-Lc/2 Lc/2 1.e-2 1.e2]);
xfont=xlabel('frequency (in unit of 1/T_s)');
yfont=ylabel('CDMA signal + narrowband jammer PSD');
set(xfont,'FontSize',11);set(yfont,'FontSize',11);

BER=[];
BER_az=[];

for i=1:10,
    Eb2N(i)=(i-1);                          %(Eb/N in dB)
    Eb2N_num=10^(Eb2N(i)/10);           % Eb/N in numeral
    Var_n=Lc/(2*Eb2N_num);               %1/SNR is the noise variance
    signois=sqrt(Var_n);                % standard deviation
    awgnois=signois*noiseq;             % AWGN
    % Add noise to signals at the channel output
    y_out=x_in+awgnois+jammer;
    Y_out=reshape(y_out,Lc,Ldata).';   clear y_out awgnois;

    % Despread first
    z_out=Y_out*pcode;

    % Decision based on the sign of the samples
    dec1=sign(real(z_out))+j*sign(imag(z_out));
    % Now compare against the original data to compute BER
    BER=[BER;sum([real(data_sym)~=real(dec1);...
        imag(data_sym)~=imag(dec1)])/(2*Ldata)];
    BER_az=[BER_az;0.5*erfc(sqrt(Eb2N_num))];          %analytical
end
figure(3)
figber=semilogy(Eb2N,BER_az,'k-',Eb2N,BER,'k-o');
legend('No jamming','Narrowband jamming (-10 dB)');
set(figber,'LineWidth',2);
xfont=xlabel('E_b/N (dB)');
yfont=ylabel('Bit error rate');
title('DSSS (CDMA) with spreading gain = 11');
```

Because the spreading factor in this case is $L = 11$, the DSSS signal occupies a bandwidth approximately 11 times wider. From the user signal carrier, we add a narrowband QPSK jamming signal with a carrier frequency offset of $1.32/T$. The signal-to-interference ratio (SIR) can be adjusted. In Fig. 11.23,

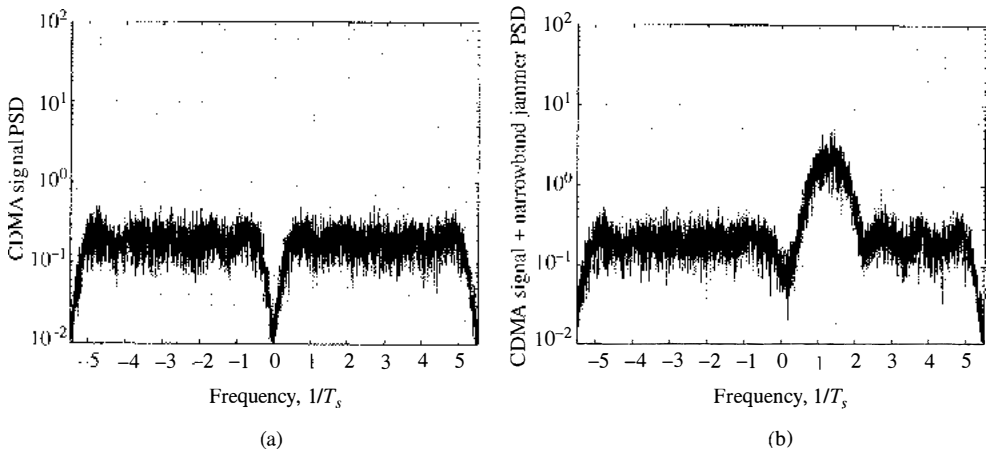


Figure 11.23 (a) Power spectral densities of DSSS signal using Barker code of length 11 for spreading: (a) without narrowband jamming; (b) with narrowband jamming at SIR = 10 dB.

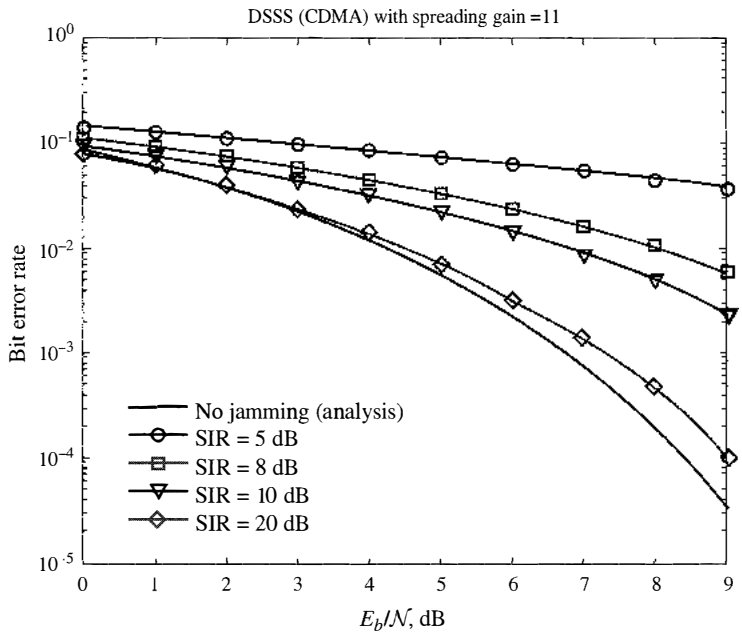


Figure 11.24 Bit error probabilities of DSSS with QPSK modulation under narrowband jamming.

we can witness power spectral densities before and after the addition of the jamming signal when SIR = 10 dB. Despreading at the receiver enables us to find the resulting BER of the QPSK signal under different jamming levels (Fig. 11.24). As the jamming signal becomes stronger and stronger, we will need to apply larger spreading factors to mitigate the degrading effect on the BER.

COMPUTER EXERCISE 11.3: MULTIUSER DS-CDMA SYSTEM

To implement DS-CDMA systems, we must select multiple spreading codes with good cross-correlation and autocorrelation properties. Gold sequences are a very well-known class of such good spreading codes. Note that the Gold sequences are not mutually orthogonal. They have some nonzero but small cross-correlations that can degrade the multiuser detection performance. We select four Gold sequences to spread four QPSK users of equal transmission power. No near-far effect is considered in this example.

The first MATLAB program, `gold31code.m`, assigns four Gold sequences of length 31 to the four QPSK modulated user signals:

```
% MATLAB PROGRAM <gold31code.m>
% to generate a table of 4 Gold sequence
% with length 31 each.
GPN=[1     1     1    -1
    -1     1    -1     1
    -1    -1     1     1
     1     1    -1    -1
    -1    -1    -1    -1
     1     1     1     1
     1     1    -1    -1
    -1    -1    -1     1
    -1     1    -1    -1
     1    -1    -1     1
    -1    -1     1     1
     1     1    -1     1
     1    -1    -1    -1
    -1     1     1     1
     1    -1     1     1
    -1     1     1    -1
    -1    -1     1    -1
    -1     1     1    -1
     1     1     1    -1
     1    -1     1     1
     1    -1     1    -1
     1     1    -1    -1
     1     1     1     1
     1     1    -1    -1
    -1    -1    -1    -1
     1     1    -1     1
     1    -1    -1    -1
    -1     1    -1     1
     1     1    -1    -1
     1     1     1     1
     1     1     1     1];
```

The main MATLAB program, `Ex11_3.m`, completes the spreading of the four user signals. The four spread CDMA signals are summed together at the receiver before detection. Each of the four users will apply the conventional despreader (matched filter) at the receiver before making the symbol-by-symbol decision. We provide the resulting BER of all four users in Fig. 11.25 under additive white Gaussian noise. We also give the single-user BER in AWGN channel as a reference. All four users have the same BER. The small degradation of the multiuser BER from the single-user BER is caused by the nonorthogonal spreading codes.

```
% MATLAB PROGRAM <Ex11_3.m>
% This program provides simulation for multiuser DS-CDMA signaling using
% coherent QPSK for 4 users.
%
%clear;clf
Ldata=10000;        % data length in simulation; Must be divisible by 8
Lc=31;             % spreading factor vs data rate
%User number = 4;
```

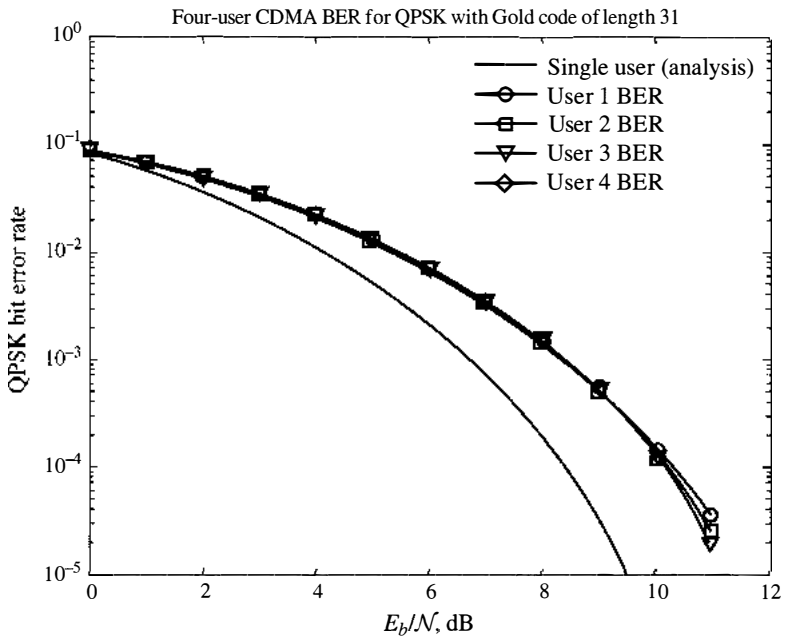


Figure 11.25 Performance of DS-CDMA conventional single-user detection without the near-far effect.

```
%  Generate QPSK modulation symbols
data_sym=2*round(rand(Ldata,4))-1+j*(2*round(rand(Ldata,4))-1);

% Select 4 spreading codes (Gold Codes of  Length 11)
gold31code;
pcode=GPN;
% Spreading codes are now in matrix pcode of 31x4
PowerMat=diag(sqrt([1 1 1 1]));
pcodew=pcode*PowerMat;
% Now spread
x_in=kron(data_sym(:,1),pcodew(:,1))+kron(data_sym(:,2),pcodew(:,2)) +...
kron(data_sym(:,3),pcodew(:,3))+kron(data_sym(:,4),pcodew(:,4));

%  Signal power of the channel input is 2*Lc

% Generate noise (AWGN)
noiseq=randn(Ldata*Lc,1)+j*randn(Ldata*Lc,1); % Power is 2

BER1=[];
BER2=[];
BER3=[];
BER4=[];
BER_az=[];

for i=1:12,
    Eb2N(i)=(i-1);                          %(Eb/N in dB)
    Eb2N_num=10^(Eb2N(i)/10);           % Eb/N in numeral
    Var_n=Lc/(2*Eb2N_num);               %1/SNR is the noise variance
    signois=sqrt(Var_n);                % standard deviation
    awgnois=signois*noiseq;             % AWGN
    % Add noise to signals at the channel output
```

```
    y_out=x_in+awgnois;
    Y_out=reshape(y_out,Lc,Ldata).';   clear y_out awgnois;

    % Despread first
    z_out=Y_out*pcode;

    % Decision based on the sign of the samples
    dec=sign(real(z_out))+j*sign(imag(z_out));
    % Now compare against the original data to compute BER
    BER1=[BER1;sum([real(data_sym(:,1))~=real(dec(:,1));...
        imag(data_sym(:,1))~=imag(dec(:,1))])/(2*Ldata)];
    BER2=[BER2;sum([real(data_sym(:,2))~=real(dec(:,2));...
        imag(data_sym(:,2))~=imag(dec(:,2))])/(2*Ldata)];
    BER3=[BER3;sum([real(data_sym(:,3))~=real(dec(:,3));...
        imag(data_sym(:,3))~=imag(dec(:,3))])/(2*Ldata)];
    BER4=[BER4;sum([real(data_sym(:,4))~=real(dec(:,4));...
        imag(data_sym(:,4))~=imag(dec(:,4))])/(2*Ldata)];
    BER_az=[BER_az;0.5*erfc(sqrt(Eb2N_num))];          %analytical
end
BER=[BER1 BER2 BER3 BER4];
figure(1)
figber=semilogy(Eb2N,BER_az,'k-',Eb2N,BER1,'k-o',Eb2N,BER2,'k-s',...
    Eb2N,BER3,'k-v',Eb2N,BER4,'k-*');
legend('Single-user (analysis)','User 1 BER','User 2 BER',
'User 3 BER','User 4 BER')
axis([0 12 0.99e-5 1.e0]);
set(figber,'LineWidth',2);
xlabel('E_b/N (dB)');ylabel('QPSK bit error rate')
title('4-user CDMA BER with Gold code of length 31');
```

COMPUTER EXERCISE 11.4: MULTIUSER CDMA DETECTION IN NEAR-FAR ENVIRONMENT

We can now modify the program in Computer Exercise 11.3 to include the near-far effect. Among the four users, user 2 and user 4 have the same power and are the weaker users from far transmitters. User 1 is 10 dB stronger, while user 3 is 7 dB stronger. In this near-far environment, both users 2 and 4 suffer from strong interference (users 1 and 3) signals due to the lack of code orthogonality. Note that the two weak users do not have the same level of multiuser interference (MUI) from other users because of the difference in their correlations.

MATLAB program `Ex11_4a.m` compares the performance of the conventional single-user receiver with the performance of the decorrelator multiuser detector (MUD) described in Sec. 11.7. We show the performance results of user 2 and user 4 in Fig. 11.26.

```
% MATLAB PROGRAM <Ex11_4a.m>
% This program provides simulation for multiuser CDMA system
% that experiences the near-far effect due to user Tx power
% variations.
%
% Decorrelator receivers are
% applied to mitigate the near-far effect
%
%clear;clf
```

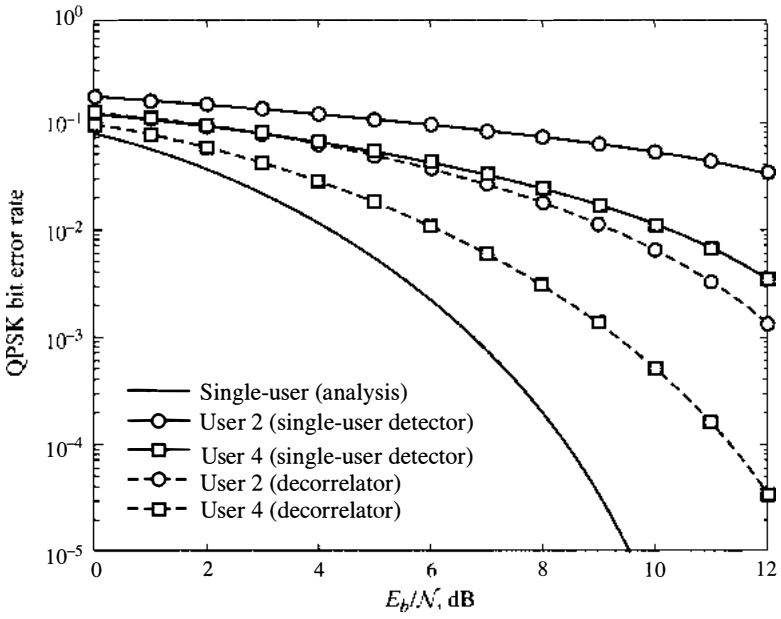


Figure 11.26 Performance comparison of decorrelator MUD in comparison with the conventional single-user receiver.

```
Ldata=100000;        % data length in simulation; Must be divisible by 8
Lc=31;             % spreading factor vs data rate
%User number = 4;
%  Generate QPSK modulation symbols
data_sym=2*round(rand(Ldata,4))-1+j*(2*round(rand(Ldata,4))-1);

% Select 4 spreading codes (Gold Codes of  Length 11)
gold31code;
pcode=GPN;
% Spreading codes are now in matrix pcode of 31x4
PowerMat=diag(sqrt([10 1 5 1]));
pcodew=pcode*PowerMat;
Rcor=pcodew'*pcodew;
Rinv=pinv(Rcor);
% Now spread
x_in=kron(data_sym(:,1),pcodew(:,1))+kron(data_sym(:,2),pcodew(:,2)) +...
kron(data_sym(:,3),pcodew(:,3))+kron(data_sym(:,4),pcodew(:,4));

%  Signal power of the channel input is 2*Lc

% Generate noise (AWGN)
noiseq=randn(Ldata*Lc,1)+j*randn(Ldata*Lc,1); % Power is 2

BERb2=[];
BERa2=[];
BERb4=[];
BERa4=[];
BER_az=[];

for i=1:13,
    Eb2N(i)=(i-1);                          %(Eb/N in dB)
```

```
    Eb2N_num=10^(Eb2N(i)/10);           % Eb/N in numeral
    Var_n=Lc/(2*Eb2N_num);               %1/SNR is the noise variance
    signois=sqrt(Var_n);                % standard deviation
    awgnois=signois*noiseq;             % AWGN
    % Add noise to signals at the channel output
    y_out=x_in+awgnois;
    Y_out=reshape(y_out,Lc,Ldata).';   clear y_out awgnois;
    % Despread first and apply decorrelator Rinv
    z_out=(Y_out*pcode);        % despreader (conventional) output
    clear Y_out;
    z_dcr=z_out*Rinv;           % decorrelator output

    % Decision based on the sign of the single receivers
    dec1=sign(real(z_out))+j*sign(imag(z_out));
    dec2=sign(real(z_dcr))+j*sign(imag(z_dcr));
    % Now compare against the original data to compute BER of user 2
    % and user 4 (weaker ones).
    BERa2=[BERa2;sum([real(data_sym(:,2))~=real(dec1(:,2));...
        imag(data_sym(:,2))~=imag(dec1(:,2))])/(2*Ldata)];
    BERa4=[BERa4;sum([real(data_sym(:,4))~=real(dec1(:,4));...
        imag(data_sym(:,4))~=imag(dec1(:,4))])/(2*Ldata)];
    BERb2=[BERb2;sum([real(data_sym(:,2))~=real(dec2(:,2));...
        imag(data_sym(:,2))~=imag(dec2(:,2))])/(2*Ldata)];
    BERb4=[BERb4;sum([real(data_sym(:,4))~=real(dec2(:,4));...
        imag(data_sym(:,4))~=imag(dec2(:,4))])/(2*Ldata)];
    BER_az=[BER_az;0.5*erfc(sqrt(Eb2N_num))];          %analytical
end
figure(1)
figber=semilogy(Eb2N,BER_az,'k-',Eb2N,BERa2,'k-o',Eb2N,BERa4,' k-s',...
    Eb2N,BERb2,'k--o',Eb2N,BERb4,'k--s');
legend('Single-user (analysis)','User 2 (single user detector)',...
    'User 4 (single user detector)','User 2 (decorrelator)',...
    'User 4 (decorrelator)')
axis([0 12 0.99e-5 1.e0]);
set(figber,'LineWidth',2);
xlabel('E_b/N (dB)');ylabel('QPSK bit error rate')
title('Weak-user BER comparisons');
```

We also implement the decision feedback MUD of Sec. 11.7 in MATLAB program `Ex11_4b.m`. The decision feedback MUD performance of the two users is shown in Fig. 11.27.

```
% MATLAB PROGRAM <Ex11_4b.m>
% This program provides simulation for multiuser CDMA
% systems.  The 4 users have different powers to illustrate the
% near-far effect in single user conventional receivers
%
% Decision feedback detectors are  tested to show its
% ability to overcome the near-far problem.
%
%clear;clf
Ldata=100000;        % data length in simulation; Must be divisible by 8
Lc=31;             % spreading factor vs data rate
%User number = 4;
%  Generate QPSK modulation symbols
```

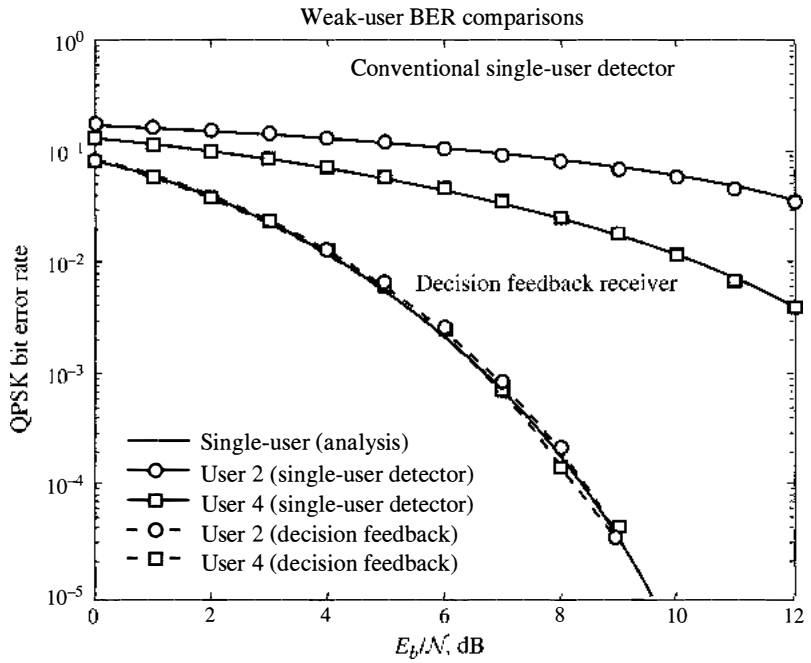


Figure 11.27 Performance comparison of decision feedback MUD in comparison with the conventional single-user receiver.

```
data_sym=2*round(rand(Ldata,4))-1+j*(2*round(rand(Ldata,4))-1);

% Select 4 spreading codes (Gold Codes of  Length 11)
gold31code;
pcode=GPN;
% Spreading codes are now in matrix pcode of 31x4
PowerMat=diag(sqrt([10 1 5 1]));
pcodew=pcode*PowerMat;
Rcor=pcodew'*pcodew;
% Now spread
x_in=kron(data_sym(:,1),pcodew(:,1))+kron(data_sym(:,2),pcodew(:,2)) +...
kron(data_sym(:,3),pcodew(:,3))+kron(data_sym(:,4),pcodew(:,4));

%  Signal power of the channel input is 2*Lc

% Generate noise (AWGN)
noiseq=randn(Ldata*Lc,1)+j*randn(Ldata*Lc,1); % Power is 2

BER_c2=[];
BER2=[];
BER_c4=[];
BER4=[];
BER_az=[];

for i=1:13,
    Eb2N(i)=(i-1);                          %(Eb/N in dB)
    Eb2N_num=10^(Eb2N(i)/10);           % Eb/N in numeral
    Var_n=Lc/(2*Eb2N_num);               %1/SNR is the noise variance
    signois=sqrt(Var_n);                % standard deviation
```

```
    awgnois=signois*noiseq;              % AWGN
    % Add noise to signals at the channel output
    y_out=x_in+awgnois;
    Y_out=reshape(y_out,Lc,Ldata).';   clear y_out awgnois;
    % Despread first
    z_out=Y_out*pcode;         % despreader (conventional) output
    clear Y_out;
    % Decision based on the sign of the single receivers
    dec=sign(real(z_out))+j*sign(imag(z_out));

    % Decision based on the sign of the samples
    dec1=sign(real(z_out(:,1)))+j*sign(imag(z_out(:,1)));
    z_fk1=z_out-dec1*Rcor(1,:);
    dec3=sign(real(z_fk1(:,3)))+j*sign(imag(z_fk1(:,3)));
    z_fk2=z_fk1-dec3*Rcor(3,:);
    dec2=sign(real(z_fk2(:,2)))+j*sign(imag(z_fk2(:,2)));
    z_fk3=z_fk2-dec2*Rcor(2,:);
    dec4=sign(real(z_fk3(:,4)))+j*sign(imag(z_fk3(:,4)));
    % Now compare against the original data to compute BER
    BER_c2=[BER_c2;sum([real(data_sym(:,2))~=real(dec(:,2));...
        imag(data_sym(:,2))~=imag(dec(:,2))])/(2*Ldata)];
    BER2=[BER2;sum([real(data_sym(:,2))~=real(dec2);...
        imag(data_sym(:,2))~=imag(dec2)])/(2*Ldata)];
    BER_c4=[BER_c4;sum([real(data_sym(:,4))~=real(dec(:,4));...
        imag(data_sym(:,4))~=imag(dec(:,4))])/(2*Ldata)];
    BER4=[BER4;sum([real(data_sym(:,4))~=real(dec4);...
        imag(data_sym(:,4))~=imag(dec4)])/(2*Ldata)];
    BER_az=[BER_az;0.5*erfc(sqrt(Eb2N_num))];          %analytical
end
clear z_fk1 z_fk2 z_fk3 dec1 dec3 dec2 dec4 x_in y_out noiseq;
figure(1)
figber=semilogy(Eb2N,BER_az,'k-',Eb2N,BER_c2,'k-o',Eb2N,BER_c4,'k-s',...
Eb2N,BER2,'k--o',Eb2N,BER4,'k--s');
legend('Single-user (analysis)','User 2 (single user detector)',...
    'User 4 (single user detector)','User 2 (decision feedback)',...
    'User 4 (decision feedback)')
axis([0 12 0.99e-5 1.e0]);
set(figber,'LineWidth',2);
xlabel('E_b/N (dB)');ylabel('QPSK bit error rate')
title('Weak-user BER comparisons');
```

REFERENCES

1. E. O. Geronoitis and M. B. Pursley, "Error Probabilities for Slow Frequency-Hopped Spread-Spectrum Multiple Access Communications over Fading Channels," *IEEE Trans. Commun.*, vol. 30, no. 5, pp. 996–1009, 1982.
2. Matthew S. Gast, *802.11 Wireless Networks: The Definitive Guide,* O'Reilly & Associates, Sebastopol, CA, 2002.
3. Brent A. Miller and Chatschik Bisdikian, *Bluetooth Revealed,* Upper Saddle River, NJ, Prentice-Hall, 2001.

4. http://www.bluetooth.com.
5. http://www.physics.princeton.edu/trothman/Broertjes_patent.pdf.
6. David Wallace, "Hedy Lamarr," *Lost Magazine*, October 2006.
7. J. S. Lehnert, "An Efficient Technique for Evaluating Direct-Sequence Spread-Spectrum Communications," *IEEE Trans. on Commun.*, vol. 37, pp. 851–858, August 1989.
8. R. Lupas and S. Verdú, "Near-Far Resistance of Multiuser Detectors in Asynchronous Channels," *IEEE Trans. Commun.*, vol. COM-38, no. 4, pp. 496–508, April 1990.
9. S. Verdú, *Multiuser Detection*, Cambridge University Press, New York, 1998.
10. S. Verdú, "Optimum Multiuser Asymptotic Efficiency," *IEEE Trans. Commun.*, vol. COM-34, no. 9, pp. 890–897, Sept. 1986.
11. R. Lupas and S. Verdu, "Linear Multiuser Detectors for Synchronous Code-Division Multiple-Access Channel," *IEEE Trans. Inform. Theory*, vol. 35, pp. 123–136, Jan. 1989.
12. Z. Xie, R. T. Short, and C. K. Rushforth, "A Family of Suboptimum Detectors for Coherent Multiuser Communications," *IEEE Journal on Selected Areas of Communications*, vol. 8, pp. 683–690, May 1990.
13. M. K. Varanasi and B. Aazhang, "Near-Optimum Detection in Synchronous Code-Division Multiple-Access Systems," *IEEE Trans. Commun.*, vol. 39, pp. 825–836, May 1991.
14. A. J. Viterbi, "Very Low Rate Convolutional Codes for Maximum Theoretical Performance of Spread-Spectrum Multiple-Access Channels," *IEEE J. Select. Areas Commun.*, vol. 8, no 4, May 1990, pp. 641–649.
15. R. Kohno, H. Imai, M. Hatori, and S. Pasupathy, "Combination of an Adaptive Array Antenna and a Canceler of Interference for Direct-Sequence Spread-Spectrum Multiple-Access System," *IEEE J. Select. Areas Commun.*, vol. 8, no. 4, May 1990, pp. 675–682.
16. Juha Korhonen, *Introduction to 3G Mobile Communications*, Artech House, Boston, 2001.
17. S. C. Yang, *3G CDMA2000 Wireless System Engineering*, Artech House, Boston, 2004.
18. Keiji Tachikawa, ed. *W-CDMA Mobile Communications System*, Wiley, 2002.
19. 3rd Generation Partnership Project 2, "Physical Layer Standard for cdma2000 Spread Spectrum Systems," *3GGP2 C.S0002-D*, Version 1.0, Feb. 13, 2004.

PROBLEMS

11.1-1 Consider a fast hopping binary ASK system. The AWGN spectrum equals $S_n(f) = 10^{-6}$ and the binary signal amplitudes are 0 and 2 V, respectively. The ASK uses a data rate of 100 kbit/s and is detected noncoherently. The ASK requires 100 kHz bandwidth for transmission. However, the frequency hopping is over 12 equal ASK bands with bandwidth totaling 1.2 MHz. The partial band jammer can generate a strong Gaussian noise–like interference with total power of 27 dBm.

(a) If a partial band jammer randomly jams one of the 12 FH channels, derive the BER of the FH-ASK if the ASK signal hops 6 bands per bit period.

(b) If a partial band jammer randomly jams two of the 12 FH channels, derive the BER of the FH-ASK if the ASK signal hops 6 bands per bit period.

(c) If a partial band jammer jams all 12 FH channels, derive the BER of the FH-ASK if the ASK signal hops 6 bands per bit period.

11.1-2 Repeat Prob. 11.1-1 if the ASK signal hops 12 bands per bit period.

11.1-3 Repeat Prob. 11.1-1 if the ASK signal hops one band per bit period.

11.2-1 In a multiuser FHSS system that applies BFSK for each user transmission, consider each interfering user as a partial band jammer. There are M users and L total signal bands for synchronous frequency hopping. The desired user under consideration hops L_h bands within each bit period.

(a) Find the probability that exactly 1 of the signal bands used by the desired user during a signal bit is jammed by the interfering signals.

(b) Determine the probability that none of the signal bands used by the desired user during a signal bit will be jammed by the interfering signals.

(c) Assume that when a partial signal band is jammed, we can compute the BER effect by discarding the signal energy within the jammed band. Find the BER of a given user within the system.

11.4-1 Let the AWGN noise $\mathrm{n}(t)$ have spectrum $\mathcal{N}/2$. If the AWGN noise $\mathrm{n}(t)$ is ideally band-limited to $1/2T_c$ Hz, show that if the spreading signal $c(t)$ has autocorrelation function

$$R_c(\tau) = \sum_i \delta(\tau - i \cdot LT_c)$$

then the PSD of $\mathrm{x}(t) = \mathrm{n}(t)c(t)$ is approximately

$$S_{\mathrm{x}}(f) = \int_{-\infty}^{\infty} S_{\mathrm{n}}(\nu) S_c(f - \nu)\, d\nu = \frac{\mathcal{N}}{2}$$

11.5-1 Consider DSSS systems with interference signal $i(t)$. At the receiver, the despread signal $c(t) = \pm 1$ with bandwidth B_c.

(a) Show that $i(t)$ and the despread interference

$$i_a(t) = i(t)c(t)$$

have identical power.

(b) If $i(t)$ has bandwidth B_s and the spreading factor is L such that $B_c = L \cdot B_s$, show that the power spectrum of $i_a(t)$ is L times lower but L times wider.

11.6-1 In a multiuser CDMA system of DSSS, all transmitters are at equal distance from the receivers. In other words, g_i = constant. The additive white Gaussian noise spectrum equals $S_{\mathrm{n}}(f) = 5. \times 10^{-6}$. BPSK is the modulation format of all users at the rate of 16 kbit/s.

(a) If the spreading codes are all mutually orthogonal, find the desired user signal power P_i required to achieve BER of 10^{-5}.

(b) If the spreading codes are not orthogonal, more specifically,

$$R_{ii} = 1 \qquad R_{ij} = -1/16 \quad i \neq j$$

Determine the required user signal power to achieve the same BER of 10^{-5} by applying Gaussian approximation of the nonorthogonal MAI.

11.6-2 Repeat Prob. 11.6-1, if one of the 15 interfering transmitter is 2 times closer to the desired receiver such that its gain g_ℓ is 4 times stronger.

11.7-1 For the multiuser CDMA system of Prob. 11.6-3, design the corresponding decorrelator and the MMSE detectors.

12 DIGITAL COMMUNICATIONS UNDER LINEARLY DISTORTIVE CHANNELS

In our earlier discussion and analysis of digital communication systems, we have made the rather idealistic assumption that the communication channel introduces no distortion. Moreover, the only channel impairment under consideration has been additive white Gaussian noise (AWGN). In reality, however, communication channels are far from ideal. Among a number of physical channel distortions, *multipath* is arguably the most serious problem encountered in wireless communications. In analog communication systems, multipath represents an effect that can often be tolerated by human ears (as echos) and eyes (as shadows). In digital communications, however, multipath leads to linear channel distortions that manifest as intersymbol interferences (ISI). This is because multipath leads to multiple copies of the same signal arriving at the receiver with different delays. Thus, one symbol pulse is delayed, which affects one or more adjacent symbols, causing ISI. As we have discussed, ISI can severely affect the accuracy of the receivers. To combat the effects of ISI due to multipath channels, we discuss, in this chapter, two highly effective tools: **equalization** and **OFDM** (orthogonal frequency division modulation).

12.1 LINEAR DISTORTIONS OF WIRELESS MULTIPATH CHANNELS

Digital communication requires that digital signals be transmitted over a specific medium between the transmitter and the receiver. The physical media (channels) in real world are analog. Because of practical limitations, however, analog channels are usually imperfect and can introduce unwanted distortions. Examples of nonideal analog media include telephone lines, coaxial cables, underwater acoustics, and radio-frequency (RF) wireless channels at various frequencies. Figure 12.1 demonstrates a simple case in which transmission from a base station to a mobile unit encounters a two-ray multipath channel: one ray from the line-of-sight and one from the ground reflection. At the receiver, there are two copies of the transmitted signal, one of which is a delayed version of the other.

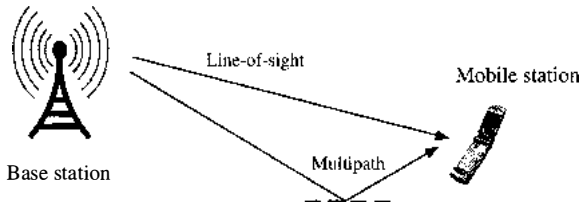


Figure 12.1 Simple illustration of a two-ray multipath channel.

To understand the effect of multipath in this example, we denote the line-of-sight signal arrival and the reflective arrival, respectively, as

$$s(t) = m(t)\cos\omega_c t \qquad \text{and} \qquad \alpha_1 s(t-\tau_1) = \alpha_1 m(t-\tau_1)\cos\omega_c(t-\tau_1)$$

Here we assumed that the modulation is DSB with PAM message signal (Chapter 7)

$$m(t) = \sum_k a_k p(t-kT)$$

where T is the PAM symbol duration. Note also that we use α_1 and τ_1, respectively, to represent the multipath loss and the delay relative to the line-of-sight signal. Hence, the receiver RF input signal is

$$\mathrm{r}(t) = m(t)\cos\omega_c t + \alpha_1 m(t-\tau_1)\cos\omega_c(t-\tau_1) + \mathrm{n}_c(t)\cos\omega_c t + \mathrm{n}_s(t)\sin\omega_c t \quad (12.1)$$

In Eq. (12.1), $\mathrm{n}_c(t)$ and $\mathrm{n}_s(t)$ denote the in-phase and quadrature components of the bandpass noise, respectively (Sec. 9.9). By applying coherent detection, the receiver baseband output signal becomes

$$\begin{aligned} y(t) &= \mathrm{LPF}\{2\mathrm{r}(t)\cos\omega_c t\} \\ &= m(t) + \alpha_1(\cos\omega_c\tau_1)m(t-\tau_1) + \mathrm{n}_c(t) \qquad (12.2a) \\ &= \sum_k a_k p(t-kT) + (\alpha_1\cdot\cos\omega_c\tau_1)\sum_k a_k p(t-kT-\tau_1) + \mathrm{n}_c(t) \\ &= \sum_k a_k\,[p(t-kT) + (\alpha_1\cos\omega_c\tau_1)\,p(t-kT-\tau_1)] + \mathrm{n}_c(t) \qquad (12.2b) \end{aligned}$$

By defining a baseband waveform

$$q(t) = p(t) + (\alpha_1\cos\omega_c\tau_1)\,p(t-\tau_1)$$

we can simplify Eq. (12.2b)

$$y(t) = \sum_k a_k\, q(t-kT) + \mathrm{n}_c(t) \qquad (12.2c)$$

Effectively, this multipath channel has converted the original pulse shape $p(t)$ into $q(t)$. If $p(t)$ was designed (as in Chapter 7) to satisfy Nyquist's first criterion of zero ISI,

$$p(nT) = \begin{cases} 1 & n = 0 \\ 0 & n = \pm 1, \pm 2, \ldots \end{cases}$$

then the new pulse shape $q(t)$ will certainly have ISI as

$$q(nT) = p(nT) + (\alpha_1\cdot\cos\omega_c\tau_1)\,p(nT-\tau_1) \neq 0 \qquad n = \pm 1, \pm 2, \ldots$$

To generalize, if there are $K+1$ different paths, then the effective channel response is

$$q(t) = p(t) + \sum_{i=1}^{K} [\alpha_i \cos \omega_c \tau_i]\, p(t - \tau_i)$$

in which the line-of-sight path delay is assumed to be $\tau_0 = 0$ with unit path gain $\alpha_0 = 1$. The ISI effect caused by the K summations in $q(t)$ depends on (a) the relative strength of the multipath gains $\{\alpha_i\}$; and (b) the multipath delays $\{\tau_i\}$.

General QAM Models

For conserving bandwidth in both wire-line and wireless communications, QAM is an efficient transmission. We again let the QAM symbol rate be $1/T$ and its symbol duration be T. Under QAM, the data symbols $\{s_k\}$ are complex-valued, and the quadrature bandpass RF signal transmission is

$$s(t) = \left[\sum_k \text{Re}\{s_k\} p(t - kT)\right] \cos \omega_c t + \left[\sum_k \text{Im}\{s_k\} p(t - kT)\right] \sin \omega_c t \tag{12.3}$$

Thus, under multipath channels with $K+1$ paths and impulse response

$$\delta(t) + \sum_{i=1}^{K} \alpha_i \delta(t - \tau_i)$$

the received bandpass signal for QAM is

$$\text{r}(t) = s(t) + \sum_{i=1}^{K} \alpha_i s(t - \tau_i) + \text{n}_c(t) \cos \omega_c t + \text{n}_s(t) \sin \omega_c t \tag{12.4}$$

Applying coherent detection, the QAM demodulator has two baseband outputs LPF $\{2\text{r}(t) \cos \omega_c t\}$ and LPF$\{2r(t) \sin \omega_c t\}$. These two (in-phase and quadrature) outputs are real-valued and can be written as a single complex-valued output:

$$y(t) = \text{LPF}\{2\text{r}(t) \cos \omega_c t\} + j \cdot \text{LPF}\{2\text{r}(t) \sin \omega_c t\} \tag{12.5a}$$

$$\begin{aligned}
&= \sum_k \text{Re}\{s_k\} \left[\sum_{i=0}^{K} (\alpha_i \cos \omega_c \tau_i) p(t - kT - \tau_i)\right] \\
&\quad + \sum_k \text{Im}\{s_k\} \left[\sum_{i=0}^{K} (\alpha_i \sin \omega_c \tau_i) p(t - kT - \tau_i)\right] \\
&\quad - j \cdot \sum_k \text{Re}\{s_k\} \left[\sum_{i=0}^{K} (\alpha_i \sin \omega_c \tau_i) p(t - kT - \tau_i)\right] \\
&\quad + j \cdot \sum_k \text{Im}\{s_k\} \left[\sum_{i=0}^{K} (\alpha_i \cos \omega_c \tau_i) p(t - kT - \tau_i)\right] + \text{n}_c(t) + j\,\text{n}_s(t)
\end{aligned}$$

$$= \sum_k s_k \left[\sum_{i=0}^{K} \alpha_i \exp(-j\omega_c \tau_i) p(t - kT - \tau_i)\right] + \text{n}_c(t) + j\,\text{n}_s(t) \tag{12.5b}$$

Once again, we can define a baseband (complex) impulse response

$$q(t) = \sum_{i=0}^{K} \alpha_i \exp(j\omega_c \tau_i) p(t - kT - \tau_i) \tag{12.6a}$$

and the baseband complex noise

$$\mathrm{n}_e(t) = \mathrm{n}_c(t) + j\,\mathrm{n}_s(t) \tag{12.6b}$$

The receiver demodulator output signal at the baseband can be then written simply as

$$y(t) = \sum_{k} s_k q(t - kT) + \mathrm{n}_e(t) \tag{12.7}$$

in which all variables are complex-valued. Clearly, the original pulse $p(t)$ that was designed to be free of ISI has been transformed by the multipath channel route into $q(t)$. In the frequency domain, we can see that

$$Q(f) = \sum_{i=0}^{K} \alpha_i \exp\left[-j(2\pi f - \omega_c)\tau_i\right] \cdot P(f) \tag{12.8}$$

This means that the original frequency response $P(f)$ encounters a frequency-dependent transfer function because of multipath response

$$\sum_{i=0}^{K} \alpha_i \exp(j\omega_c \tau_i) \exp\left[-j2\pi f\, \tau_i\right]$$

Therefore, the channel distortion is a function of the frequency f. Communication channels that introduce frequency-dependent distortions are known as *frequency-selective* channels. Frequency-selective channels can exhibit substantial ISI, which can lead to significant increase of detection errors.

Wire-Line ISI

Although we have just demonstrated how multipath in wireless communications can lead to ISI and linear channel distortions, wire-line systems are not entirely immune to such problems. Indeed, wire-line systems do not have a multipath environment because all signals are transmitted by dedicated cables. However, when the cables have multiple unused open terminals, impedance mismatch at these open terminals can also generate reflective signals that will arrive as delayed copies at the receiver terminals. Therefore, ISI due to linear channel distortion can also be a problem in wire-line systems. Cable internet service is one example.

Equalization and OFDM

Because ISI channels lead to serious signal degradation and poor detection performance, their effects must be compensated either at the transmitter or at the receiver. In most cases, transmitters in an uncertain environment are not aware of the actual conditions of propagation. Thus, it is up to the receivers to identify the unknown multipath channel $q(t)$ and to find effective means to combat the ISI. The two most common and effective tools against ISI channels are **channel equalization** and **OFDM**.

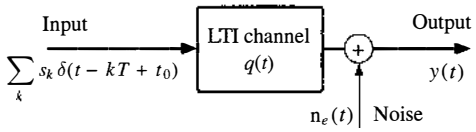


Figure 12.2 Baseband representation of QAM transmission over a linear time-invariant channel with ISI.

12.2 RECEIVER CHANNEL EQUALIZATION

It is convenient for us to describe the problem of channel equalization in the stationary channel case. Once the fundamentals of linear time-invariant (LTI) channel equalization is understood, adaptive technology can handle time-varying channels.

When the channel is LTI, we use the simple system diagram of Fig. 12.2 to describe the problem of channel equalization. In general, channel equalization is studied for the (spectrally efficient) digital QAM systems. The baseband model for a typical QAM (quadrature amplitude modulated) data communication system consists of an unknown LTI channel $q(t)$, which represents the physical interconnection between the transmitter and the receiver in baseband.

The baseband transmitter generates a sequence of complex-valued random input data $\{s_k\}$, each element of which belongs to the constellation $\mathcal{A}$ of QAM symbols. The data sequence $\{s_k\}$ is sent through the baseband channel that is LTI with impulse response $q(t)$. Because QAM symbols $\{s_k\}$ are complex-valued, the baseband channel impulse response $q(t)$ is also complex-valued in general.

Under the causal and complex-valued LTI communication channel with impulse response $q(t)$, the input-output relationship of the QAM system can be written as

$$y(t) = \sum_{k=-\infty}^{\infty} s_k q(t - kT + t_0) + n_e(t) \quad s_k \in \mathcal{A} \tag{12.9}$$

Typically the baseband channel noise $n_e(t)$ is assumed to be stationary, Gaussian, and independent of the channel input s_k. Given the received baseband signal $y(t)$ at the receiver, the job of the channel equalizer is to estimate the original data $\{s_k\}$ from the received signal $y(t)$.

In what follows, we present the common framework within which channel equalization is typically accomplished. Without loss of generality, we let $t_o = 0$.

12.2.1 Antialiasing Filter vs. Matched Filter

We showed in Secs. 10.1 and 10.6 that the optimum receiver filter should be matched to the total response $q(t)$. This filter serves to maximize the SNR of the sampled signal at the filter output. Even if the response $q(t)$ has ISI, Forney[1] has established the optimality* of the matched filter receiver, as shown in Fig. 12.3. With a matched filter $q(-t)$ and symbol (baud) rate sampling at $t = nT$, the receiver obtains an output sequence relationship between the transmitter data $\{s_k\}$ and the receiver samples as

$$z[n] = \sum_k s_k h(nT - kT) \tag{12.10}$$

* Forney proved[1] that sufficient statistics for input symbol estimation is retained by baud rate sampling at $t = nT$ of matched filter output signal. This result forms the basis of the well-known single-input–single-output (SISO) system model obtained by matched filter sampling. However, when $q(t)$ is unknown, the optimality no longer applies.

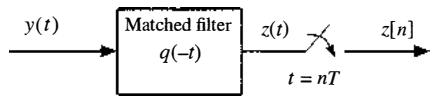


Figure 12.3 Optimal matched filter receiver.

where

$$h(t) = q(t) * q(-t) \tag{12.11}$$

If we denote the samples of $h(t)$

$$h[n] = h(nT)$$

then Eq. (12.10) can be simplified

$$z[n] = \sum_k s_k h[n-k] = h[n] * s[n] \tag{12.12}$$

In short, the channel (input-output) signals are related by a single-input–single-output (SISO) linear discrete channel with transfer function

$$H(z) = \sum_n h[n]\, z^{-n} \tag{12.13}$$

The SISO discrete representation of the linear QAM signal leads to the standard T-spaced equalizer (TSE). The term *T-spaced equalization* refer to processing of the received signal sampled at the rate of $1/T$. Therefore, the time separation between successive samples equals the baud (symbol) period T.

The optimal matched filter receiver faces a major practical obstacle that the total pulse shape response $q(t)$ depends on the multipath channel environment. In reality, it is practically difficult to adjust the receiver filter according to the time-varying $q(t)$ because channel environment may undergo significant and possibly rapid changes. Moreover, the receivers generally do not have a priori information on the channel that affects $q(t)$. As a result, it does not make sense to implement the optimum receiver filter $q(-t)$ in a dynamic channel environment. It makes better sense to design and implement a time-invariant receiver filter. Therefore, the important task is to select a receiver filter without losing any signal information in $y(t)$.

To find a solution, recall the QAM channel input signal

$$x(t) = \sum_k s_k p(t - kT)$$

We have learned from Section 7.2 [see Eq. (7.9)] that the power spectral density of an amplitude-modulated pulse train is

$$S_x(f) = |P(f)|^2 \frac{1}{T}\left[\sum_{n=-\infty}^{\infty} R_s[n] e^{-jn2\pi fT}\right] \tag{12.14a}$$

$$R_s[n] = \overbrace{s_{k+n} s_k^*}^{\sim\sim\sim\sim} \tag{12.14b}$$

by simply substituting the pulse amplitude a_k with the QAM symbol s_k. The signal spectrum of Eq. (12.14a) shows that the signal component in $y(t)$ is limited by the bandwidth of $p(t)$ or $P(f)$.

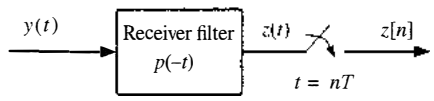


Figure 12.4 Commonly used receiver filter matched instead to the transmission pulse.

Therefore, the receiver filter must not filter out any valuable signal component and should have bandwidth equal to the bandwidth of $P(f)$. On the other hand, if we let the receiver filter have a bandwidth larger than $P(f)$, then more noise will pass through the filter, with no benefit to the signal. For these reasons, a good receiver filter should have bandwidth *exactly identical to* the bandwidth of $P(f)$. Of course many such filters exist. One is the filter matched to the transmission pulse $p(t)$ given by

$$p(-t) \Longleftrightarrow P^*(f)$$

Another consideration is that, if the channel introduces no additional distortions, then $q(t) = p(t)$. In this case, the optimum receiver would be the filter $p(-t)$ matched to $p(t)$. Consequently, it makes sense to select $p(-t)$ as a standard receiver filter (Fig. 12.4) for two reasons:

(a) The filter $p(-t)$ retains all the signal spectral component in the received signal $y(t)$.

(b) The filter $p(-t)$ is optimum if the environment happens to exhibit no channel distortions.

Therefore, we often apply the receiver filter $p(-t)$ matched to the transmission pulse shape $p(t)$. This means that the total channel impulse response consists of

$$h(t) = q(t) * p(-t)$$

Notice that because of the filtering $z(t) = p(-t) * y(t)$. The signal $z(t)$ now becomes

$$z(t) = \sum_k s_k h(t - kT) + \mathrm{w}(t) \tag{12.15}$$

in which the filtered noise term $\mathrm{w}(t)$ arises from

$$\mathrm{w}(t) = p(-t) * \mathrm{n}_e(t) \tag{12.16}$$

with power spectral density

$$S_{\mathrm{w}}(f) = |P(f)|^2 \, S_{\mathrm{n}_e}(f)$$

Finally, the relationship between the sampled output $z[k]$ and the communication symbols s_k is

$$\begin{aligned} z[n] &= \sum_k h[n-k]s_k + \mathrm{w}[n] \\ &= \sum_k h[k]\, s_{n-k} + \mathrm{w}[n] \end{aligned} \tag{12.17}$$

where the discrete noise samples are denoted by $\mathrm{w}[n] = \mathrm{w}(nT)$.

Generally, there are two approaches to the problem of channel input recovery (i.e., equalization) under ISI channels. The first approach is to determine the optimum receiver based on channel and noise models. This approach leads to maximum likelihood sequence estimation (MLSE), which is computationally demanding. A low-cost alternative is to design filters known as channel equalizers to compensate for the channel distortion. In what follows, we first describe the essence of the MLSE method for symbol recovery. By illustrating its typically high computational complexity, we provide the necessary motivation for the subsequent discussions on various complexity channel equalizers.

12.2.2 Maximum Likelihood Sequence Estimation (MLSE)

The receiver output samples $\{z[n]\}$ depend on the unknown input QAM symbols $\{s_n\}$ according to the relationship of Eq. (12.17). The optimum (MAP) detection of $\{s_n\}$ from $\{z[n]\}$ requires the maximization of joint conditional probability [Eq. (10.81)]:

$$\max_{\{s_n\}} p(\ldots, s_{n-1}, s_n, s_{n+1}, \ldots \Big| \ldots, z[n-1], z[n], z[n+1], \ldots) \tag{12.18}$$

Unlike the optimum symbol-by-symbol detection for AWGN channels derived and analyzed in Sec. 10.6, the interdependent relationship in Eq. (12.17) means that the optimum receiver must detect the entire sequence $\{s_n\}$ from a sequence of received signal samples $\{z[n]\}$.

To simplify this optimum receiver, we first note that in most communication systems and applications, each QAM symbol s_n is randomly selected from its constellation $\mathcal{A}$ with equal probability. Thus, the MAP detector can be translated into a maximum likelihood sequence estimation (MLSE):

$$\max_{\{s_n\}} p\left(\ldots, z[n-1], z[n], z[n+1], \ldots \Big| \ldots, s_{n-1}, s_n, s_{n+1}, \ldots\right) \tag{12.19}$$

If the original channel noise $\mathrm{n}_e(t)$ is white Gaussian, then the discrete noise $\mathrm{w}[n]$ is also Gaussian because Eq. (12.16) shows that $\mathrm{w}(t)$ is filtered output of $\mathrm{n}_e(t)$. In fact, we can define the power spectral density of the white noise $\mathrm{n}_e(t)$ as

$$S_{\mathrm{n}_e}(f) = \frac{\mathcal{N}}{2}$$

Then the power spectral density of the filtered noise $\mathrm{w}(t)$ is

$$S_{\mathrm{w}}(f) = |P(f)|^2 \, S_{\mathrm{n}_e}(f) = \frac{\mathcal{N}}{2} \, |P(f)|^2 \tag{12.20}$$

From this information, we can observe that the autocorrelation function between the noise samples is

$$\begin{aligned} R_{\mathrm{w}}[\ell] &= \overline{\mathrm{w}[\ell + n]\mathrm{w}^*[n]} \\ &= \overline{\mathrm{w}(\ell T + nT)\mathrm{w}^*(nT)} \\ &= \int_{-\infty}^{\infty} S_{\mathrm{w}}(f) e^{-j2\pi f \, \ell T} df \\ &= \frac{\mathcal{N}}{2} \int_{-\infty}^{\infty} |P(f)|^2 \, e^{-j2\pi f \, \ell T} df \end{aligned} \tag{12.21}$$

In general, the autocorrelation between two noise samples in Eq. (12.21) depends on the receiver filter which is, in this case, $p(-t)$. In Sec. 7.3, the ISI-free pulse design based on Nyquist's first criterion is of particular interest. Nyquist's first criterion requires that the total response from the transmitter to the receiver be free of intersymbol interferences. Without channel distortion, the QAM system in our current study has a total impulse response of

$$p(t) * p(-t) \iff |P(f)|^2$$

For this combined pulse shape to be free of ISI, we can apply the first Nyquist criterion in the frequency domain

$$\frac{1}{T}\sum_{k=-\infty}^{\infty}\left|P\left(f+\frac{k}{T}\right)\right|^2 = 1 \tag{12.22a}$$

This is equivalent to the time domain requirement

$$p(t) * p(-t)\Big|_{t=\ell T} = \begin{cases} 1 & \ell = 0 \\ 0 & \ell = \pm 1, \pm 2, \ldots \end{cases} \tag{12.22b}$$

In other words, the Nyquist pulse-shaping filter is equally split between the transmitter and the receiver. According to Eq. (12.22a), the pulse-shaping frequency response $P(f)$ is the square root of a pulse shape that satisfies Nyquist's first criterion in the frequency domain. If the raised-cosine pulse shape of Section 7.3 is adopted, then $P(f)$ would be known as the **root-raised-cosine** pulse. For a given roll-off factor r, the root-raised-cosine pulse in the time domain is

$$p_{\text{rrc}}(t) = \frac{2r}{\pi\sqrt{T}} \frac{\cos\left[(1+r)\frac{\pi t}{T}\right] + \left(4r\frac{t}{T}\right)^{-1}\sin(1-r)\frac{\pi t}{T}}{\left[1-\left(4r\frac{t}{T}\right)^2\right]} \tag{12.23}$$

Based on the ISI-free conditions of Eq. (12.22b), we can derive from Eq. (12.21) that

$$\begin{aligned} R_{\text{w}}[\ell] &= \frac{\mathcal{N}}{2}\int_{-\infty}^{\infty}|P(f)|^2 e^{-j2\pi f\,\ell T}df \\ &= \frac{\mathcal{N}}{2}p(t) * p(-t)\Big|_{t=\ell T} \\ &= \begin{cases} \frac{\mathcal{N}}{2} & \ell = 0 \\ 0 & \ell = \pm 1, \pm 2, \ldots \end{cases} \end{aligned} \tag{12.24}$$

This means that the noise samples $\{\text{w}[n]\}$ are uncorrelated. Since the noise samples $\{\text{w}[n]\}$ are Gaussian, they are also independent. As a result, the conditional joint probability of Eq. (12.19) becomes much simpler

$$\begin{aligned} &p\left(\ldots, z[n-1], z[n], z[n+1], \ldots \Big| \ldots, s_{n-1}, s_n, s_{n+1}, \ldots\right) \\ &\quad = \prod_i p\left(z[n-i]\Big| \ldots, s_{n-1}, s_n, s_{n+1}, \ldots\right) \end{aligned} \tag{12.25}$$

Indeed, Eq. (12.24) tells us that $z[n-i]$ is Gaussian with equal variance $\mathcal{N}/2$ and mean value of

$$\sum_k h[k]s_{n-i-k}$$

Therefore, the MLSE optimum receiver under Gaussian channel noise and root-raised-cosine pulse shape $p_{\text{rrc}}(t)$ [Eq. (12.23)],

$$\max_{\{s_n\}} \ln \left[\prod_i p\left(z[n-i] \middle| \ldots, s_{n-1}, s_n, s_{n+1}, \ldots\right) \right]$$

$$\Longleftrightarrow \max_{\{s_n\}} \left\{ -\frac{2}{\mathcal{N}} \sum_i \left| z[n-i] - \sum_k h[k]s_{n-i-k} \right|^2 \right\} \tag{12.26a}$$

Thus, MLSE is equivalent to

$$\min_{\{s_n\}} \sum_i \left| z[n-i] - \sum_k h[k]s_{n-i-k} \right|^2 \tag{12.26b}$$

For a vast majority of communication channels, the impulse response $h[k]$ can be closely approximated as a finite impulse response (FIR) filter of some finite order. If the maximum channel order is L such that

$$H(z) = \sum_{k=0}^{L} h[k]\, z^{-k}$$

then the MLSE receiver needs to solve

$$\min_{\{s_n\}} \sum_i \left| z[n-i] - \sum_{k=0}^{L} h[k]s_{n-i-k} \right|^2 \tag{12.27}$$

We note that the MLSE algorithm requires that the receiver possess the knowledge of the discrete channel coefficients $\{h[k]\}$. When exact channel knowledge is not available, the receiver must first complete the important task of channel estimation.

MLSE Complexity and Practical Implementations

Despite the apparent high complexity of the MLSE algorithm [Eq. (12.27)], there exists a much more efficient solution given by Viterbi[2] based on the *dynamic programming* principle of Bellman.[3] This algorithm, often known as the Viterbi algorithm, does not have an exponentially growing complexity as the data length grows. Instead, if the QAM constellation size is M, then the complexity of the Viterbi algorithm grows according to M^L. The Viterbi algorithm is a very powerful tool, particularly when the channel order L is not very long and the constellation size M is not huge. The details of the Viterbi algorithm will be explained in Chapter 14 when we present the decoding of convolutional codes.

MLSE is very common in practical applications. Most notably, many GSM cellular receivers perform the MLSE detection described here against multipath distortions. Because GSM uses binary constellations in voice transmission, the complexity of the MLSE receivers is reasonably low for common cellular channels that can be approximated as FIR responses of order 3 to 8.

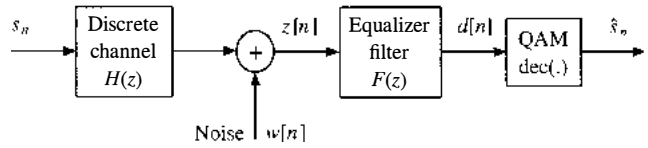


Figure 12.5 A SISO discrete linear channel model for TSE.

On the other hand, the modulation formats adopted in high-speed dial-up modems are highly complex. For example, the V.32bis (14.4 kbit/s) modem uses a trellis-coded QAM constellation of size 128 (with 64 distinct symbols) at the symbol rate of 2400 baud (symbols/s). In such applications, even a relatively short $L = 5$ FIR channel would require MLSE to have over 1 billion states. In fact, at higher bit rates, dial-up modems can use size 256 QAM or even size 960 QAM. As a result, the large number of states in MLSE makes it completely unsuitable as a receiver in such systems. Consequently, suboptimal equalization approaches with low complexity are much more attractive. The design of simple and cost effective equalizers (deployed in applications including voiceband dial-up modems) is discussed next.

12.3 LINEAR *T*-SPACED EQUALIZATION (TSE)

When the receiver filter is matched to the transmission pulse $p(t)$ only, it is no longer optimum.* Even if the ideal matched filter $q(-t)$ is known and applied, it is quite possible in practice for the sampling instant to have an offset t_0 such that the sampling takes place at $t = nT + t_0$. Such a sampling offset is known as a *timing error*. When there is a timing error, the receiver is also not optimum. It is in fact commonplace for practical communication systems to have unknown distortive channels and timing jitters. Nevertheless, T-spaced equalization is simpler to implement. Here we discuss the fundamental aspects of TSE design.

Because T-spaced sampling leads to a simple discrete time linear system Eq. (12.17) as shown in Fig. 12.5, the basic linear equalizer is simply a linear filter $F(z)$ followed by a direct QAM decision device. The operational objective of the equalizer (filter) $F(z)$ is to remove as much ISI as possible from its output $d[n]$. We begin our discussion on the T-spaced equalizer (TSE) by denoting the (causal) equalizer transfer function

$$F(z) = \sum_i f[i]z^{-i}$$

If the channel noise w$[n]$ is included, the TSE output is

$$d[n] = F(z)z[n] = \underbrace{F(z)H(z)s_n}_{\text{signal term}} + \underbrace{F(z)\mathrm{w}[n]}_{\text{noise term}} \tag{12.28}$$

We denote the joint channel equalizer transfer function as

$$C(z) = F(z)H(z) = \sum_{i=0}^{\infty} c_i z^{-i}$$

The goal of the equalizer $F(z)$ is to clean up the ISI in $d[n]$ to achieve an error-free decision

$$\hat{s}_n = \text{dec}\ (d[n]) = s_{n-u} \tag{12.29}$$

---

* The sufficient statistics shown by G. D. Forney[1] are not necessarily retained.

where u is a fixed delay in the equalizer output. Because both the channel and the equalizer must be causal, the inclusion of a possible delay u provides opportunities for simpler and better equalizer designs.

To better understand the design of the TSE filter $F(z)$, we can divide the TSE output into different terms

$$\begin{aligned} d[n] &= \sum_{i=0}^{\infty} c_i s_{n-i} + \sum_{i=0}^{\infty} f[i]\mathrm{w}[n-i] \\ &= c_u s_{n-u} + \underbrace{\sum_{i=0,\, i\neq u}^{\infty} c_i s_{n-i}}_{\text{ISI term}} + \underbrace{\sum_{i=0}^{\infty} f[i]\mathrm{w}[n-i]}_{\text{noise term}} \end{aligned} \tag{12.30}$$

The equalizer filter output $d[n]$ consists of the desired signal component with the right delay, plus the ISI and noise terms. If both the ISI and noise terms are zero, then the QAM decision device will always make correct detections without any error. Therefore, the design of this linear equalizer filter $F(z)$ should aim to minimize effect of the ISI and the noise terms. In practice, there are two very popular types of linear equalizer: zero-forcing (ZF) design and minimum mean square error (MMSE) design.

12.3.1 Zero-Forcing TSE

The principle of zero-forcing equalizer design is to eliminate the ISI term without considering the noise effect. In principle, a perfect ZF equalizer $F(z)$ should force

$$\sum_{i=0,\, i\neq u}^{\infty} c_i s_{n-i} = 0$$

In other words, all ISI terms are eliminated

$$c_i = \begin{cases} 1 & i = u \\ 0 & i \neq u \end{cases} \tag{12.31a}$$

Equivalently in frequency domain, the ZF equalizer requires

$$C(z) = F(z)H(z) = z^{-u} \tag{12.31b}$$

Notice that the linear equalizer $F(z)$ is basically an inverse filter of the discrete ISI channel $H(z)$ with appropriate delay u

$$F(z) = \frac{z^{-u}}{H(z)} \tag{12.31c}$$

If the ZF filter of Eq. (12.31c) is causal and can be implemented, then the ISI is completely eliminated from $z[n]$. This appears to be an excellent solution, since the only decision that the decision device now must make is based on

$$z[n] = s_{n-u} + F(z)\mathrm{w}[n]$$

without any ISI. One major drawback of the ZF equalizer lies in the remaining noise term $F(z)\mathrm{w}[n]$. If the noise power in $z[n]$ is weak, then the QAM decision would be highly accurate. Problems arise when the transfer function $F(z)$ has strong gains at certain frequencies. As a result, the noise term $F(z)\mathrm{w}[n]$ may be amplified at those frequencies. In fact, when the frequency response of $H(z)$ has spectral nulls, that is,

$$H(e^{j\omega_o}) = 0 \quad \text{for some } \omega_o \in [0, \pi]$$

then the ZF equalizer $F(z)$ at ω_o would have infinite gain, and substantially amplify the noise component at ω_o.

A different perspective is to consider the filtered noise variance. If $\mathrm{w}[n]$ are independent identically distributed (i.i.d.) Gaussian with zero mean and variance $\mathcal{N}/2$, then the filtered noise term equals

$$\tilde{w}[n] = F(z)w[n] = \sum_{i=0}^{\infty} f[i]\mathrm{w}[n-i]$$

The noise term $\tilde{w}[n]$ remains Gaussian with mean

$$\overline{\sum_{i=0}^{\infty} f[i]\mathrm{w}[n-i]} = \sum_{i=0}^{\infty} f[i]\overline{\mathrm{w}[n-i]} = 0$$

and variance

$$\overline{\left|\sum_{i=0}^{\infty} f[i]\mathrm{w}[n-i]\right|^2} = \mathcal{N}/2 \cdot \sum_{i=0}^{\infty} |f[i]|^2$$

Because the ZF equalizer output is

$$z[n] = s_{n-u} + \tilde{w}[n]$$

the probability of decision error in dec $(z[n])$ can therefore be analyzed by applying the same tools used in Chapter 10 (Sec. 10.6). In particular, under BPSK modulation, $s_n = \pm\sqrt{E_b}$ with equal probability. Then the probability of detection error is

$$P_b = Q\left(\sqrt{\frac{2E_b}{\mathcal{N} \cdot \sum_{i=0}^{\infty} |f[i]|^2}}\right) \tag{12.32}$$

where the ZF equalizer parameters can be obtained via the inverse-Z transform

$$\begin{aligned} f[i] &= \frac{1}{2\pi j} \oint F(z) z^{i-1} dz \\ &= \frac{1}{2\pi j} \oint \frac{z^{i-1-u}}{H(z)} dz \end{aligned} \tag{12.33}$$

If $F(e^{j\omega})$ has spectral nulls, then $f[i]$ from Eq. (12.33) may become very large, causing a serious increase of P_b.

Example 12.1 Consider a first-order channel

$$H(z) = 1 + z^{-2}$$

Determine the noise amplification effect on the ZF equalizer for a BPSK transmission.

Because $H(e^{j2\pi f}) = 0$ when $f = \pm 1/4$, it is clear that $H(z)$ has spectral nulls. By applying the ZF equalizer, we have

$$f[i] = \frac{1}{2\pi j} \oint \frac{z^{i-1-u}}{1+z^{-2}} dz$$
$$= \begin{cases} 0 & i < u \\ (-1)^{i-u} & i \geq u \end{cases}$$

Therefore,

$$\sum_{i=0}^{\infty} |f[i]|^2 = \sum_{i=u}^{\infty} (1) = \infty$$

This means that the BER of the BPSK transmission equals

$$P_b = Q(0) = 0.5$$

The noise amplification is so severe that the detection is completely random.

Example 12.1 clearly shows the significant impact of noise amplification due to ZF equalization. The noise amplification effect strongly motivates other design methodologies for equalizers. One practical solution is the minimum mean square error (MMSE) design.

12.3.2 TSE Design Based on MMSE

Because of the noise amplification effect in ZF equalization, we must not try to eliminate the ISI without considering the negative impact from the noise term. In fact, we can observe the equalizer output in Eq. (12.30) and quantify the *overall distortion* in $d[n]$ by considering the difference (or error)

$$d[n] - s_{n-u} = \sum_{i=0, i\neq u}^{\infty} c_i s_{n-i} - s_{n-u} + \sum_{i=0}^{\infty} f[i] \mathrm{w}[n-i] \tag{12.34}$$

To reduce the number of decision errors when

$$\mathrm{dec}\,(d[n]) \neq s_{n-u}$$

it would be sensible to design an equalizer that would minimize the mean square error between $d[n]$ and s_{n-u}. In other words, the MMSE equalizer design should minimize

$$\overline{|d[n] - s_{n-u}|^2} \tag{12.35}$$

Let us now proceed to find an equalizer filter that can minimize the mean square error of Eq. (12.35). Once again, we will apply the principle of orthogonality in optimum estimation (Sec. 8.5), that the error (difference) signal must be orthogonal to the signals used in the filter input. Because $d[n] = \sum_{i=0}^{\infty} f[i]z[n-i]$, we must have

$$(d[n] - s_{n-u}) \perp z[n-\ell] \qquad \ell = 0, 1, \ldots$$

In other words,

$$\overline{(d[n] - s_{n-u})\, z^*[n-\ell]} = 0 \qquad \ell = 0, 1, \ldots \tag{12.36}$$

Therefore, the equalizer parameters $\{f[i]\}$ must satisfy

$$\overline{\left(\sum_{i=0}^{\infty} f[i]z[n-i] - s_{n-u}\right) z^*[n-\ell]} = 0 \qquad \ell = 0, 1, \ldots$$

Note that the signal s_n and the noise $\mathrm{w}[n]$ are independent. Moreover, $\{s_n\}$ are also i.i.d. with zero mean and variance E_s. Therefore, $\overline{s_{n-u}\mathrm{w}^*[n]} = 0$, and we have

$$\begin{aligned}
\overline{s_{n-u}\, z^*[n-\ell]} &= \overline{s_{n-u}\left(\sum_{j=0}^{\infty} h[j]^* s^*_{n-j-\ell} + \mathrm{w}[n-\ell]^*\right)} \\
&= \sum_{j=0}^{\infty} h[j]^* \overline{s_{n-u}\, s^*_{n-j-\ell}} + 0 \\
&= \begin{cases} E_s \cdot h[u-\ell]^* & 0 \le \ell \le u \\ 0 & \ell > u \end{cases}
\end{aligned} \tag{12.37}$$

Let us also denote

$$R_z[m] = \overline{z[n+m]\, z^*[n]} \tag{12.38}$$

Then the MMSE equalizer is the solution to linear equations

$$\sum_{i=0}^{\infty} f[i]\, R_z[\ell - i] = \begin{cases} E_s\, h[u-\ell]^* & \ell = 0, 1, \ldots, u \\ 0 & \ell = u+1, u+2, \ldots, \infty \end{cases} \tag{12.39}$$

Based on the channel output signal model, we can show that

$$\begin{aligned}
R_z[m] &= \overline{\left(\sum_{i=0}^{\infty} h_i\, s_{n+m-i} + \mathrm{w}[n+m]\right)\left(\sum_{j=0}^{\infty} h_j\, s_{n-j} + \mathrm{w}[n]\right)^*} \\
&= E_s \sum_{j=0}^{\infty} h_{m+j} h_j^* + \frac{\mathcal{N}}{2}\delta[m]
\end{aligned} \tag{12.40}$$

Minimum MSE and Optimum Delay

Because of the orthogonality condition Eq. (12.36), we have

$$\overline{(d[n] - s_{n-u})\, d[n]^*} = 0$$

Hence, the resulting minimum mean square error is shown to be

$$\begin{aligned} \text{MSE}(u) &= \overline{(s_{n-u} - d[n])\, s_{n-u}^*} \\ &= E_s\,(1 - c_u) \\ &= E_s \left(1 - \sum_{i=0}^{\infty} h_i f[u-i]\right) \end{aligned} \tag{12.41}$$

It is clear that MMSE equalizers of different delays can lead to different mean square error results. To find the delay that achieves the least mean square error, the receiver can determine the optimum delay according to

$$u_o = \arg\max_u \sum_{i=0}^{\infty} h_i f[u-i] \tag{12.42}$$

Finite Length MMSE Equalizers

Because we require the equalizer $F(z)$ to be causal, the MMSE equalizer based on the solution of Eq. (12.39) does not have a simple closed form. The reason is that $\{f[i]\}$ is causal while $R_z[m]$ is not. Fortunately, practical implementation of the MMSE equalizer often assumes the form of a finite impulse response (FIR) filter. When $F(z)$ is FIR, the MMSE equalizer can be numerically determined from Eq. (12.39). Let

$$F(z) = \sum_{i=0}^{M} f[i] z^{-i}$$

The orthogonality condition of Eq. (12.39) then is reduced to a finite set of linear equations

$$\sum_{i=0}^{M} f[i]\, R_z[\ell - i] = \begin{cases} E_s\, h[u-\ell]^* & \ell = 0, 1, \ldots, u \\ 0 & \ell = u+1, u+2, \ldots, M \end{cases} \tag{12.43a}$$

Alternatively, we can write the MMSE condition into matrix form for $u < M$:

$$\begin{bmatrix} R_z[0] & R_z[-1] & \cdots & R_z[-M] \\ R_z[1] & R_z[0] & \cdots & R_z[1-M] \\ \vdots & \vdots & \ddots & \vdots \\ R_z[M] & R_z[M-1] & \cdots & R_z[0] \end{bmatrix} \begin{bmatrix} f[0] \\ f[1] \\ \vdots \\ f[M] \end{bmatrix} = E_s \left\{ \begin{bmatrix} h[u]^* \\ h[u-1]^* \\ \vdots \\ h[0]^* \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\} M+1 \text{ rows} \tag{12.43b}$$

Of course, if the delay u exceeds M, then the right hand side of Eq. (12.43b) becomes

$$\begin{bmatrix} R_z[0] & R_z[-1] & \cdots & R_z[-M] \\ R_z[1] & R_z[0] & \cdots & R_z[1-M] \\ \vdots & \vdots & \ddots & \vdots \\ R_z[M] & R_z[M-1] & \cdots & R_z[0] \end{bmatrix} \begin{bmatrix} f[0] \\ f[1] \\ \vdots \\ f[M] \end{bmatrix} = \begin{bmatrix} h[u]^* \\ h[u-1]^* \\ \vdots \\ h[u-M]^* \end{bmatrix} \tag{12.43c}$$

The solution is unique so long as the autocorrelation matrix in Eq. (12.43c) has full rank.

MMSE vs. ZF

Note that if we simply set the noise spectral level to $\mathcal{N} = 0$, the MMSE equalizer design of Eqs. (12.39) and (12.43c) is easily reduced to the ZF design. In other words, the only design change from MMSE to ZF is to replace $R_z[0]$ from the noisy to the noise-free case of

$$R_z[0] = E_s \sum_{j=0}^{\infty} |h_j|^2$$

All other procedures can be directly followed to numerically obtain the ZF equalizer parameters.

It is important to understand, however, that the design of finite length ZF equalizers according to Eq. (12.43c) may or may not achieve the objective of forcing all ISI to zero. In fact, if the channel $H(z)$ has finite order L, then ZF design would require

$$F(z)H(z) = \sum_{i=0}^{M} f[i]z^{-i} \sum_{i=0}^{L} h[i]z^{-i} = z^{-u}$$

This equality would be impossible for any stable causal equalizer to achieve. The reason is quite simple if we consider the basics of polynomials. The left-hand side is a polynomial of order $M+L$. Hence, it has a total of $M+L$ roots, whose locations depends on the channel and the equalizer transfer functions. On the other hand, the right-hand side has a root at ∞ only. It is therefore impossible to fully achieve this zero-forcing equality. Thus, one would probably ask the following question: *What would a finite length equalizer achieve if designed according to Eq. (12.43c)?*

The answer can in fact be found in the MMSE objective function when the noise is zero. Specifically, the equalizer is designed to minimize

$$\overline{|d[n] - s_{n-u}|^2} = \overline{|F(z)H(z)s_n - s_{n-u}|^2}$$

when the channel noise is not considered. Hence, the solution to Eq. (12.43c) would lead to a finite length equalizer that achieves the minimum difference between $F(z)H(z)$ and a pure delay z^{-u}. In terms of the time domain, the finite length ZF design based on Eq. (12.43c) will minimize the ISI distortion that equals

$$\begin{aligned} &|c_u - 1|^2 + \sum_{j \neq u} |c_j|^2 \\ &= \left| \sum_{i=0}^{M} f[i]h[u-i] - 1 \right|^2 + \sum_{j \neq u} \left| \sum_{i=0}^{M} f[i]h[j-i] \right|^2 \end{aligned}$$

In other words, this equalizer will minimize the contribution of ISI to the mean square error in $d[n]$.

Finite Data Design

The MMSE (and ZF) design of Eqs. (12.39) and (12.43c) assumes statistical knowledge of $R_z[m]$ and $s_{n-u}z^*[n-\ell]$. In practice, such information is not always readily available and may require real-time estimation. Instead, it is more common for the transmitter to send a short sequence of training (or pilot) symbols that the receiver can use to determine the optimum equalizer. We now describe how the previous design can be directly extended to cover this scenario.

Suppose a training sequence $\{s_n,\ n = n_1,\ n_1+1,\ \ldots, n_2\}$ is transmitted. To design an FIR equalizer

$$F(z) = f[0] + f[1]z^{-1} + \cdots + f[M]z^{-M}$$

we can minimize the average square error

$$J = \frac{1}{n_2 - n_1 + 1} \sum_{n=u+n_1}^{u+n_2} |d[n] - s_{n-u}|^2$$

where

$$d[n] = \sum_{i=0}^{M} f[i]z[n-i]$$

To minimize J, we can take its gradient with respect to $f[j]$. By setting the gradient to zero, we can derive the conditions required by the optimum equalizer parameters

$$\sum_{i=0}^{M} f[i] \frac{1}{n_2 - n_1 + 1} \sum_{n=u+n_1}^{u+n_2} z[n-i]z^*[n-j] = \frac{1}{n_2 - n_1 + 1} \times \sum_{n=u+n_1}^{u+n_2} s_{n-u}z^*[n-j] \qquad j = 0,\ 1,\ \ldots,\ M \tag{12.44}$$

These $M+1$ equations can be written more compactly as

$$\begin{bmatrix} \tilde{R}_z[0,0] & \tilde{R}_z[1,0] & \cdots & \tilde{R}_z[M,0] \\ \tilde{R}_z[0,1] & \tilde{R}_z[1,1] & \cdots & \tilde{R}_z[M,1] \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{R}_z[0,M] & \tilde{R}_z[1,M] & \cdots & \tilde{R}_z[M,M] \end{bmatrix} \begin{bmatrix} f[0] \\ f[1] \\ \vdots \\ f[M] \end{bmatrix} = \begin{bmatrix} \tilde{R}_{sz}[-u] \\ \tilde{R}_{sz}[-u+1] \\ \vdots \\ \tilde{R}_{sz}[-u+M] \end{bmatrix} \tag{12.45}$$

where we denote the time average approximations of the correlation functions (for $i, j = 0,\ 1,\ \ldots,\ M$):

$$\tilde{R}_z[i,j] = \frac{1}{n_2 - n_1 + 1} \sum_{n=u+n_1}^{u+n_2} z[n-i]z^*[n-j]$$

$$\tilde{R}_{sz}[-u+j] = \frac{1}{n_2 - n_1 + 1} \sum_{n=u+n_1}^{u+n_2} s_{n-u}z^*[n-j]$$

It is quite clear from comparing Eqs. (12.45) and (12.43c) that under a short training sequence (preamble), the optimum equalizer can be obtained by replacing the exact values of the correlation function with their time average approximations. If matrix inverse is to be avoided for complexity reasons, adaptive channel equalization is a viable technology. Adaptive channel equalization was first developed by Lucky at Bell Labs[4, 5] for telephone channels. It belongs to the field of adaptive filtering. Interested readers can refer to the book by Ding and Li[6] and the references therein.

12.4 LINEAR FRACTIONALLY SPACED EQUALIZERS (FSE)

We have shown that when the channel response is unknown to the receiver, TSE is likely to lose important signal information. In fact, this point is quite clear from the sampling theory. As shown by Gitlin and Weinstein,[7] when the transmitted signal (or pulse shape) does have spectral content beyond a frequency of $1/(2T)$ Hz, baud rate sampling at the frequency of $1/T$ is below the Nyquist rate and can lead to spectral aliasing. Consequently, receiver performance may be poor because of information loss.

In most cases, when the transmission pulse satisfies Nyquist's first criterion of zero ISI, the received signal component is certain to possess frequency content above $1/(2T)$ Hz. For example, when a raised-cosine (or a root-raised-cosine) pulse $p_{\text{rrc}}(t)$ is adopted with roll-off factor r [Eq. (12.23)], the signal component bandwidth is

$$\frac{1+r}{2T} \text{ Hz}$$

For this reason, sampling at $1/T$ will certainly cause spectral aliasing and information loss unless we use the perfectly matched filter $q(-t)$ and the ideal sampling moments $t = kT$. Hence, the use of faster samplers has great significance. When the actual sampling period is an integer fraction of the baud period T, the sampled signal under linear modulation can be equivalently represented by a single-input–multiple-output (SIMO) discrete system model. The resulting equalizers are known as the fractionally spaced equalizers (or FSE).

12.4.1 The Single-Input–Multiple-Output (SIMO) Model

An FSE can be obtained from the system in Fig. 12.6 if the channel output is sampled at a rate faster than the baud or symbol rate $1/T$. Let m be an integer such that the sampling interval becomes $\Delta = T/m$. In general, because of the (root) raised-cosine pulse has bandwidth B:

$$\frac{1}{2T} \leq B = \frac{1+r}{2T} \leq \frac{1}{T}$$

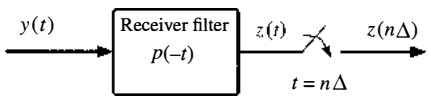


Figure 12.6 Fractionally spaced sampling receiver front end for FSE.

Any sampling rate of the form $1/\Delta = m/T$ $(m > 1)$ will be above the Nyquist sampling rate and can avoid aliasing. For analysis, denote the sequence of channel output samples as

$$\begin{aligned} z(k\Delta) &= \sum_{n=0}^{\infty} s_n h(k\Delta - nT) + \mathrm{w}(k\Delta) \\ &= \sum_{n=0}^{\infty} s_n h(k\Delta - n\,m\,\Delta) + \mathrm{w}(k\Delta) \end{aligned} \tag{12.46}$$

To simplify our notation, the oversampled channel output $z(k\Delta)$ can be reorganized (decimated) into m parallel subsequences

$$\begin{aligned} z_i[k] &\triangleq z(kT + i\Delta) \\ &= z(km\Delta + i\Delta) \\ &= \sum_{n=0}^{\infty} s_n h(km\Delta + i\Delta - n\,m\,\Delta) + \mathrm{w}(km\Delta + i\Delta). \\ &= \sum_{n=0}^{\infty} s_n h(kT - nT + i\Delta) + \mathrm{w}(kT + i\Delta) \qquad i = 1, \ldots, m \end{aligned} \tag{12.47}$$

Each subsequence $z_i[k]$ is related to the original data via

$$z_i[k] \triangleq z(kT + i\Delta) = s_k * h(kT + i\Delta) + \mathrm{w}(kT + i\Delta)$$

In effect, each subsequence is an output of a linear *subchannel*. By denoting each subchannel response as

$$h_i[k] \triangleq h(kT + i\Delta) \iff H_i(z) = \sum_{k=0}^{\infty} h_i[k] z^{-k}$$

and the corresponding subchannel noise as

$$\mathrm{w}_i[k] \triangleq \mathrm{w}(kT + i\Delta)$$

then the reorganized m subchannel outputs are

$$\begin{aligned} z_i[k] &= \sum_{n=0}^{\infty} s_n h_i[k - n] + \mathrm{w}_i[k] \\ &= \sum_{n=0}^{\infty} h_i[n] s_{n-k} + \mathrm{w}_i[k] \quad i = 1, \ldots, m \end{aligned} \tag{12.48}$$

Thus, these m subsequences can be viewed as stationary outputs of m discrete channels with a common input sequence $s[k]$ as shown in Fig. 12.7. Naturally, this represents a single-input–multiple-output (SIMO) system analogous to a physical receiver with m antennas. The FSE is in fact a bank of m filters $\{F_i(z)\}$ that jointly attempts to minimize the channel distortion shown in Fig. 12.7.

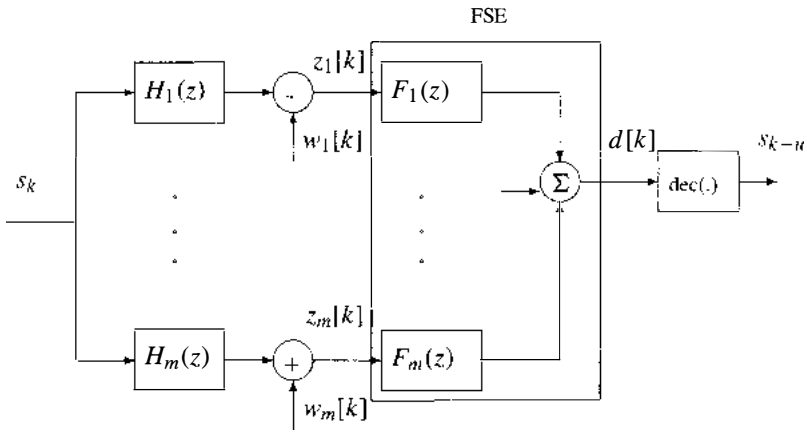


Figure 12.7 Equivalent structure of fractionally spaced equalizers (FSE).

12.4.2 FSE Designs

Based on the SIMO representation of the FSE in Fig. 12.7, one FSE filter is provided for each subsequence $z_i[k]$. In fact, the actual equalizer is a vector of filters

$$F_i(z) = \sum_{k=0}^{M} f_i[k]z^{-k} \qquad i = 1, \ldots, m \tag{12.49}$$

The m filter outputs are summed to form the stationary equalizer output

$$y[k] = \sum_{i=1}^{m} \sum_{n=0}^{M} f_i[n]z_i[k-n] \tag{12.50}$$

Given the linear relationship between equalizer output and equalizer parameters, any TSE design criterion can be generalized to the FSE design.

ZF Design

To design a ZF FSE, the goal is to eliminate all ISI at the input of the decision device. Because there are now m parallel subchannels, the ZF filters should satisfy

$$C(z) = \sum_{i=1}^{m} F_i(z)H_i(z) = z^{-u} \tag{12.51}$$

This zero-forcing condition means that the decision output will have a delay of integer u.

A closer observation of this ZF requirement reveals its connection to a well-known equality known as the *Bezout identity*. In the Bezout identity, suppose there are two polynomials of orders up to L.

$$A_1(z) = \sum_{i=0}^{L} a_{1,i}z^{-i} \quad \text{and} \quad A_2(z) = \sum_{i=0}^{L} a_{2,i}z^{-i}$$

If $A_1(z)$ and $A_2(z)$ do not share any common root, then they are called **coprime**. The Bezout identity states that if $A_1(z)$ and $A_2(z)$ are coprime, then there must exist two polynomials

$$B_1(z) = \sum_{i=0}^{M} b_{1,i} z^{-i} \quad \text{and} \quad B_2(z) = \sum_{i=0}^{M} b_{2,i} z^{-i}$$

such that

$$B_1(z)A_1(z) + B_2(z)A_2(z) = 1$$

The order requirement is that $M \geq L-1$. The solution of $B_1(z)$ and $B_2(z)$ need not be unique. It is evident from the classic text by Kailath[8] that the ZF design requirement of Eq. (12.51) is an m-channel generalization of the Bezout identity. To be precise, let $\{H_i(z),\ i = 1, 2, \ldots, m\}$ be a set of finite order polynomials of z^{-1} with maximum order L. If the m-subchannel transfer functions $\{H_i(z)\}$ are coprime, then there exists a set of filters $\{F_i(z)\}$ with orders $M \geq L-1$ such that

$$\sum_{i=1}^{m} F_i(z)H_i(z) = z^{-u} \tag{12.52}$$

where the delay can be selected from the range $u = 0, 1, \ldots, M + L - 1$. Note that the equalizer filters $\{F_i(z)\}$ vary with the desired delay u. Moreover, for each delay u, the ZF equalizer filters $\{F_i(z)\}$ are not necessarily unique.

We now describe the numerical approach to finding the equalizer filter parameters. Instead of continuing with the polynomial representation in the z-domain, we can equivalently find the matrix representation of Eq. (12.52) as

$$\underbrace{\begin{bmatrix} h_1[0] & & & \cdots & \cdots & h_m[0] & & \\ h_1[1] & \ddots & & \cdots & \cdots & h_m[1] & \ddots & \\ \vdots & \ddots & h_1[0] & \cdots & \cdots & \vdots & \ddots & h_m[0] \\ h_1[L] & \ddots & h_1[1] & \cdots & \cdots & h_m[L] & \ddots & h_m[1] \\ & \ddots & \vdots & \cdots & \cdots & & \ddots & \vdots \\ & & h_1[L] & \cdots & \cdots & & & h_m[L] \end{bmatrix}}_{\mathcal{H}:\ (L+M)\times m(M+1)} \underbrace{\begin{bmatrix} f_1[0] \\ f_1[1] \\ \vdots \\ f_1[M] \\ \vdots \\ \vdots \\ f_m[0] \\ f_m[1] \\ \vdots \\ f_m[M] \end{bmatrix}}_{m(M+1)\times 1} = \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{(M+L)\times 1} \leftarrow u\text{th} \tag{12.53}$$

The numerical design as a solution to this ZF design exists if and only if $\mathcal{H}$ has full row rank, that is, if the rows of $\mathcal{H}$ are linearly independent. This condition is satisfied for FSE (i.e., $m > 1$) if $M \geq L$ and $\{H_i(z)\}$ are coprime.[6]

MMSE FSE Design

We will apply a similar technique to provide the MMSE FSE design. The difference between FSE and TSE lies in the output signal

$$d[n] = \sum_{i=1}^{m} \sum_{k=0}^{M} f_i[k] z_i[n-k]$$

To minimize the MSE $\overline{|d[n] - s_{n-u}|^2}$, the principle of orthogonality leads to

$$\overline{(d[n] - s_{n-u})\, z_j^*[n-\ell]} = 0 \qquad \ell = 0, 1, \ldots, M, \quad j = 1, \ldots, m \tag{12.54}$$

Therefore, the equalizer parameters $\{f_i[k]\}$ must satisfy

$$\sum_{i=1}^{m} \sum_{k=0}^{M} f_i[k] \overline{z_i[n-k] z_j^*[n-\ell]} = \overline{s_{n-u} z_j^*[n-\ell]} \qquad \ell = 0, 1, \ldots, M, \quad j = 1, 2, \ldots, m$$

There are $m(M+1)$ equations for the $m(M+1)$ unknown parameters $\{f_i[k]\}$, $i = 1, \ldots, m$, $k = 0, \ldots, M$. The MMSE FSE can be found as a solution to this set of linear equations. In terms of practical issues, we should also make the following observations:

- When we have only finite length data to estimate the necessary statistics,

$$\overline{s_{n-u}\, z_j^*[n-\ell]} \quad \text{and} \quad \overline{z_i[n-k]\, z_j^*[n-\ell]}$$

  can be replaced by their time averages from the limited data collection. This is similar to the TSE design.
- Also similar to the MMSE TSE design, different values of delay u will lead to different mean square errors. To find the optimum delay, we can evaluate the MSE for all possible delays $u = 0, 1, \ldots, M + L - 1$ and choose the delay that results in the lowest MSE value.

Since their first appearance,[7] adaptive equalizers have often been implemented as FSE. When training data can be had, FSE has the advantage of suppressing timing phase sensitivity.[7] Unlike the case in TSE, linear FSE does not necessarily amplify the channel noise. Indeed, the noise amplification effect depends strongly on the coprime channel condition. In some cases, the subchannels in a set do not strictly share any common zero. However, there is at least one point z_a that is almost the root of all the subchannels, that is,

$$H_i(z_a) \approx 0 \qquad i = 1, \ldots, m$$

then we say that the subchannels are close to being singular. When the subchannels are coprime but are close to being singular, the noise amplification effect can still be quite severe.

12.5 CHANNEL ESTIMATION

Thus far, we have focused on the direct equalizer design approach in which the equalizer filter parameters are directly estimated from the channel input signal s_n and the channel output

signals $z_i[n]$. We should recognize that if MLSE receiver is implemented, the MLSE algorithm requires the knowledge of channel parameters $\{h[k]\}$. When exact channel knowledge is not available, the receiver must first complete the important first step of channel estimation.

In channel estimation, it is most common to consider FIR channels of finite order L. Similar to the linear estimation of equalizer parameters introduced in the last section, channel estimation should first consider the channel input-output relationship

$$z[n] = \sum_{k=0}^{L} h[k]s_{n-k} + \mathrm{w}[n] \tag{12.55}$$

If consecutive pilot symbols $\{s_n,\ n = n_1, n_1 + 1, \ldots, n_2\}$ are transmitted, then because of the finite channel order L, the following channel output samples

$$\{z[n], \quad n = n_1 + L, n_1 + L + 1, \ldots, n_2\}$$

depend on these pilot data and noise only. We can apply the principle of MMSE to estimate the channel coefficients $\{h[k]\}$ to minimize the average estimation error:

$$J(h[0], h[1], \ldots, h[L]) = \frac{1}{n_2 - n_1 - L + 1} \sum_{n_1+L}^{n_2} \left| z[n] - \sum_{k=0}^{L} h[k]s_{n-k} \right|^2 \tag{12.56}$$

This MMSE estimation can be simplified by setting to zero the derivative of the $J(h[0], h[1], \ldots, h[M])$ with respect to each $h[j]$. Removing redundant constants, we have

$$\left(\sum_{n_1+L}^{n_2} z[n]s_{n-j}^* \right) - \sum_{k=0}^{L} h[k] \cdot \left(\sum_{n_1+L}^{n_2} s_{n-k}s_{n-j}^* \right) = 0 \qquad j = 0, 1, \ldots, L$$

Therefore, by defining

$$\tilde{r}_{zs}[j] \triangleq \sum_{n_1+L}^{n_2} z[n]s_{n-j}^* \quad \text{and} \quad \tilde{R}_s[j,k] \triangleq \sum_{n_1+L}^{n_2} s_{n-k}s_{n-j}^* \qquad j = 0, 1, \ldots, L$$

we can simplify the MMSE channel estimation into a compact matrix expression:

$$\begin{bmatrix} \tilde{R}_z[0,0] & \tilde{R}_z[0,1] & \cdots & \tilde{R}_z[0,L] \\ \tilde{R}_z[1,0] & \tilde{R}_z[1,1] & \cdots & \tilde{R}_z[1,L] \\ \vdots & \vdots & \cdots & \vdots \\ \tilde{R}_z[L,0] & \tilde{R}_z[L,1] & \cdots & \tilde{R}_z[L,L] \end{bmatrix} \begin{bmatrix} h[0] \\ h[1] \\ \vdots \\ h[L] \end{bmatrix} = \begin{bmatrix} \tilde{r}_{sz}[0] \\ \tilde{r}_{sz}[1] \\ \vdots \\ \tilde{r}_{sz}[M] \end{bmatrix} \tag{12.57}$$

Eq. (12.57) can be solved by matrix inversion to estimate the channel parameters $h[i]$.

In the more general case of FSE, the same method can be used to estimate the ith subchannel parameters by simply replacing $z[n-k]$ with $z_i[n-k]$.

12.6 DECISION FEEDBACK EQUALIZER

The TSE and FSE we have discussed thus far are known as linear equalizers because the equalization consists of a linear filter followed by a memoryless decision device. These linear

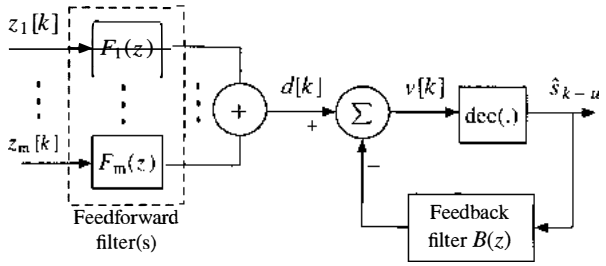


Figure 12.8 A decision feedback equalizer with fractionally spaced samples.

equalizers are also known as feedforward (FFW) equalizers. The advantages of FFW equalizers lie in their simple implementation as FIR filters and in the straightforward design approaches they accommodate. FFW equalizers require much lower computational complexity than the nonlinear MLSE receivers.

On the other hand, FFW equalizers do suffer from several major weaknesses. First, the TSE or FSE in their FFW forms can cause severe noise amplifications depending on the underlying channel conditions. Second, depending on the roots of the channel polynomials, the FFW equalizer(s) may need to be very long to be effective, particular when the channels are nearly singular. To achieve simple and effective channel equalization without risking noise amplification, a *decision feedback equalizer* (DFE) proves to be a very useful tool.

Recall that FFW equalizers generally serve as a channel inverse filter (in ZF design) or a regularized channel inverse filter (in MMSE design). The DFE, however, comprises another feedback filter in addition to a feedforward filter. The feedforward filter is identical to linear TSE or FSE, whereas the feedback filter attempts to cancel ISI from previous data samples using data estimates generated by a memoryless decision device. The feedforward filter may be operating on fractionally spaced samples. Hence, there may be m parallel filters as shown in Fig. 12.8.

The basic idea behind the inclusion of a feedback filter $B(z)$ is motivated by awareness that the feedforward filter output $d[k]$ may contain some residual ISI that can be more effectively *regenerated* by the feedback filter output and canceled from $v[k]$. More specifically, consider the case in which the feedforward filter output $d[k]$ consists of

$$d[k] = s_{k-u} + \underbrace{\sum_{i=u+1}^{N} c_i s_{k-i}}_{\text{residual ISI}} + \underbrace{\tilde{w}[n]}_{\text{noise}} \tag{12.58}$$

There is a residual ISI term and a noise term. If the decision output is very accurate such that

$$\hat{s}_{k-u} = s_{k-u}$$

then the feedback filter input will equal to the actual data symbol. If we denote the feedback filter as

$$B(z) = \sum_{i=1}^{N-u} b_i z^{-i}$$

then we have

$$\begin{aligned} v[k] &= d[k] - \sum_{i=1}^{N-u} b_i \hat{s}_{k-u-i} \\ &= s_{k-u} + \sum_{i=u+1}^{N} c_i s_{k-i} - \sum_{i=1}^{N-u} b_i \hat{s}_{k-u-i} + \tilde{w}[n] \\ &= s_{k-u} + \sum_{i=u+1}^{N} c_i s_{k-i} - \sum_{i=1}^{N-u} b_i s_{k-u-i} + \tilde{w}[n] \\ &= s_{k-u} + \sum_{i=1}^{N-u} (c_{u+i} - b_i) s_{k-u-i} + \tilde{w}[n] \end{aligned} \tag{12.59}$$

To eliminate the residual ISI, the feedback filter should have coefficients

$$b_i = c_{u+i} \quad i = 1, 2, \ldots, N-u-1$$

With these matching DFE parameters, the residual ISI is completely canceled. Hence, the input to the decision device

$$v[k] = s_{k-u} + \tilde{w}[n]$$

contains zero ISI. The only nuisance that remains in $v[k]$ is the noise. Because the noise term in $d[k]$ is not affected or amplified by the feedback filter, the decision output for the next time instant would be much more accurate after all residual ISI has been canceled.

Our DFE analysis so far has focused on the ideal operation of DFE when the decision results are correct. Traditionally, the design and analysis of DFE has often been based on such an idealized operating scenario. The design of DFE filters must include both the feedforward filters and the feedback filter. Although historically there have been a few earlier attempts to fully decouple the design of the feedforward filter and the feedback filter, the more recent work by Al Dhahir and Cioffi[9] provides a comprehensive and rigorous discussion.

In the analysis of a DFE, the assumption of correct decision output leads to the removal of ISI in $v[k]$, and hence, a better likelihood that the decision output is accurate. One cannot help but notice this circular "chicken or egg" argument. The truth of the matter is that the DFE is inherently a nonlinear system. More importantly the hard decision device is not even differentiable. As a result, most traditional analytical tools developed for linear and nonlinear systems no longer apply. For this reason, the somewhat ironic chicken-egg analysis becomes the last resort. Fortunately, for high-SNR systems, this circular argument does yield analytical results that can be closely matched by experiments.

Error Propagation in DFE

Because of its feedback structure, the DFE does suffer from the particular phenomenon known as error propagation. For example, when the decision device makes an error, the erroneous symbol will be sent to the feedback filter and used for ISI cancellation in Eq. (12.59). However, because the symbol is incorrect, instead of canceling the ISI caused by this symbol, the canceling subtraction may instead strengthen the ISI in $v[k]$. As a result, the decision device is more likely to make another subsequent error, and so on. This is known as *error propagation.*

Error propagation means that the actual DFE performance will be worse than the prediciton of analytical results derived from the assumption of perfect decision. Moreover, the effect of error propagation means that DFE is more likely to make a short burst of decision errors before recovery from the error propagation mode. The recovery time from error propagation depends on the channel response and was investigated by Kennedy and Anderson.[10]

12.7 OFDM (MULTICARRIER) COMMUNICATIONS

As we have learned from the design of TSE and FSE, channel equalization is exclusively the task of the receivers. The only assistance provided by the transmitter to receiver equalization is the potential transmission of training or pilot symbols. In a typically uncertain environment, it makes sense for the receivers to undertake the task of equalization because the transmitter normally has little or no knowledge of the channel response it uses.* Still, despite their simpler implementation compared with the optimum MLSE, equalizers such as the feedforward and decision feedback types often lead to less than satisfactory performance. More importantly, the performance of the FFW and decision feedback equalizers is too sensitive to all the parameters in their transversal structure. If even one parameter fails to hold the desired value, an entire equalizer could crumble.

In a number of applications, however, the transmitters have partial information regarding the channel characteristics. One of the most important piece of partial channel information is the channel delay spread; that is, for a finite length channel

$$H(z) = \sum_{k=0}^{L} h[k]z^{-k}$$

the channel order L is known at the transmitter while $\{h[k]\}$ are still unknown. Given this partial channel information, the particular transmission technique known as orthogonal frequency division modulation (OFDM) can be implemented at the transmitter. With the application of OFDM, the task of receiver equalization is significantly simplified.

12.7.1 Principles of OFDM

Consider a transmitter that is in charge of transmitting a sequence of data signals $\{s_k\}$ over the FIR channel $H(z)$ of order up to L. Before we begin to describe the fundamentals of OFDM, we note that the frequency response of the FIR channel can be represented as

$$H(e^{j2\pi fT}) = \sum_{k=0}^{L} h[k]e^{-j2\pi fkT} \tag{12.60}$$

where T is the symbol duration and also the sampling period. Because $H(e^{j2\pi fT})$ is the frequency response of the channel $h[k] = h(kT)$, it is a periodic function of f with period $1/T$.

The discrete Fourier transform (DFT) is a sampled function of the channel frequency response. Let N be the total number of uniform samples in each frequency period $1/T$. Then

* In a stationary environment (e.g., DSL lines) the channels are quite stable, and the receivers can use a reverse link channel to inform the transmitter its forward channel information. This channel state information (CSI) feedback, typically performed at a low bit rate to ensure accuracy, can consume rather valuable bandwidth resources.

the frequency f is sampled at

$$f_0 = 0 \cdot \frac{1}{NT} = 0$$

$$f_1 = 1 \cdot \frac{1}{NT} = \frac{1}{NT}$$

$$\vdots$$

$$f_{N-1} = (N-1) \cdot \frac{1}{NT} = \frac{(N-1)}{NT}$$

We can use a simpler notation to denote the DFT sequence by letting $\omega_n = 2\pi n/NT$

$$\begin{aligned} H[n] &= H(e^{j\omega_n T}) \\ &= \sum_{k=0}^{L} h[k] \exp(-j\omega_n Tk) \\ &= \sum_{k=0}^{L} h[k] \exp\left(-j2\pi \frac{n}{NT} kT\right) \\ &= \sum_{k=0}^{L} h[k] \exp\left(-j2\pi \frac{nk}{N}\right) \qquad n = 0, 1, \ldots, (N-1) \end{aligned} \tag{12.61}$$

From Eq. (12.61), it is useful to notice that $H[n]$ is periodic with period N (Fig. 12.9). Hence,

$$H[-n] = \sum_{k=0}^{L} h[k] \exp\left(j2\pi \frac{nk}{N}\right) \tag{12.62a}$$

$$\begin{aligned} &= \sum_{k=0}^{L} h[k] \exp\left(j2\pi \frac{nk}{N} - j2\pi \frac{Nk}{N}\right) \\ &= \sum_{k=0}^{L} h[k] \exp\left[-j2\pi \frac{(N-n)k}{N}\right] \\ &= H[N-n] \end{aligned} \tag{12.62b}$$

Based on the linear convolutional relationship between the channel input $\{s_k\}$ and output

$$z[k] = \sum_{i=0}^{L} h[i] s_{k-i} + \mathrm{w}[k]$$

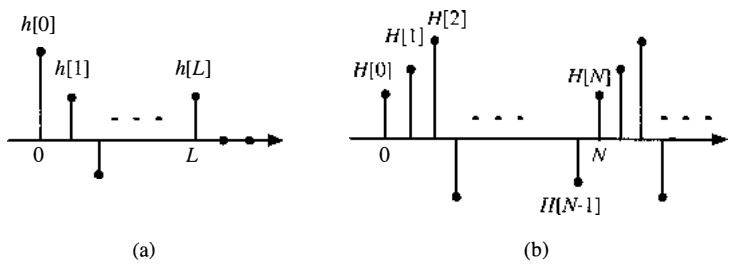


Figure 12.9 (a) Discrete time domain channel response and (b) its corresponding periodic DFT.

a vector of N output symbols can be written in matrix form as

$$\begin{bmatrix} z[N] \\ z[N-1] \\ \vdots \\ z[L] \\ \vdots \\ z[1] \end{bmatrix} = \begin{bmatrix} h[0] & h[1] & \cdots & h[L] & & & & & \\ & h[0] & h[1] & \cdots & h[L] & & & & \\ & & \ddots & \ddots & \ddots & \ddots & & & \\ & & & h[0] & h[1] & \cdots & h[L] & & \\ & & & & \ddots & \ddots & \ddots & \ddots & \\ & & & & & h[0] & h[1] & \cdots & h[L] \end{bmatrix} \times \begin{bmatrix} s_N \\ s_{N-1} \\ \vdots \\ s_1 \\ s_0 \\ \vdots \\ s_{-(L-1)} \end{bmatrix} + \begin{bmatrix} \mathrm{w}[N] \\ \mathrm{w}[N-1] \\ \vdots \\ \mathrm{w}[L] \\ \vdots \\ \mathrm{w}[1] \end{bmatrix} \tag{12.63}$$

The key step in OFDM is to introduce what is known as the *cyclic prefix* in the transmitted data.* This step replaces the M leading elements

$$s_0, s_{-1}, \ldots, s_{-(L-1)}$$

of the $(N + L)$-dimensional data vector by the trailing symbols

$$\{s_N, s_{N-1}, \ldots, s_{N-L+1}\} \longrightarrow \{s_0, s_{-1}, \ldots, s_{-(L-1)}\}$$

By inserting the cyclic prefix, we can then rewrite Eq. (12.63) as

$$\begin{bmatrix} z[N] \\ z[N-1] \\ \vdots \\ z[L] \\ \vdots \\ z[1] \end{bmatrix} = \begin{bmatrix} h[0] & h[1] & \cdots & h[L] & & & & & \\ & h[0] & h[1] & \cdots & h[L] & & & & \\ & & \ddots & \ddots & \ddots & \ddots & & & \\ & & & h[0] & h[1] & \cdots & h[L] & & \\ & & & & \ddots & \ddots & \ddots & \ddots & \\ & & & & & h[0] & h[1] & \cdots & h[L] \end{bmatrix} \times \begin{bmatrix} s_N \\ s_{N-1} \\ \vdots \\ s_1 \\ s_N \\ \vdots \\ s_{N-L+1} \end{bmatrix} + \begin{bmatrix} \mathrm{w}[N] \\ \mathrm{w}[N-1] \\ \vdots \\ \mathrm{w}[L] \\ \vdots \\ \mathrm{w}[1] \end{bmatrix} \tag{12.64a}$$

* Besides the use cyclic prefix, zero padding is an alternative but equivalent approach.

$$= \underbrace{\begin{bmatrix} h[0] & h[1] & \cdots & h[L] & 0 & \cdots & 0 \\ 0 & h[0] & h[1] & \cdots & h[L] & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & h[0] & h[1] & \cdots & h[L] \\ h[L] & \ddots & \ddots & 0 & h[0] & \cdots & h[L-1] \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ h[1] & \cdots & h[L] & 0 & \cdots & 0 & h[0] \end{bmatrix}}_{\mathcal{H}_{\text{cp}} : (N \times N)} \begin{bmatrix} s_N \\ s_{N-1} \\ \vdots \\ s_1 \end{bmatrix} + \begin{bmatrix} \text{w}[N] \\ \text{w}[N-1] \\ \vdots \\ \text{w}[L] \\ \vdots \\ \text{w}[1] \end{bmatrix} \tag{12.64b}$$

The critical role of the cyclic prefix is to convert the convolution channel matrix in Eq. (12.64a) into a well-structured $N \times N$ cyclic matrix $\mathcal{H}_{\text{cp}}$ in Eq. (12.64b).

Next, we need to introduce the N-point DFT matrix and the corresponding inverse DFT matrix. First, it is more convenient to denote

$$W_N = \exp\left(-j\frac{2\pi}{N}\right)$$

This complex number W_N has some useful properties:

- $W_N^N = 1$
- $W_N^{-i} = W_N^{N-i}$

If we take the DFT of the N-dimensional vector

$$\boldsymbol{v} = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{N-1} \end{bmatrix}$$

then we have the DFT

$$V[n] = \sum_{k=0}^{N-1} v_i \exp\left(-j2\pi\frac{nk}{N}\right) = \sum_{k=0}^{N-1} v_i W_N^{nk} \qquad n = 0, 1, \ldots, (N-1)$$

and

$$V[-n] = \sum_{k=0}^{N-1} v_i \exp\left(j2\pi\frac{nk}{N}\right) = \sum_{k=0}^{N-1} v_i W_N^{-nk} \qquad n = 0, 1, \ldots, (N-1)$$

The inverse DFT can also be simplified as

$$v_k = \frac{1}{N}\sum_{n=0}^{N-1} V[n] \exp\left(j2\pi\frac{nk}{N}\right) = \frac{1}{N}\sum_{k=0}^{N-1} V[n] W_N^{-nk} \qquad k = 0, 1, \ldots, (N-1)$$

Thus, the N-point DFT of $\boldsymbol{v}$ can be written in the matrix form

$$\boldsymbol{V} = \begin{bmatrix} V[0] \\ V[1] \\ \vdots \\ V[N-1] \end{bmatrix} = \begin{bmatrix} W_N^{0\cdot 0} & W_N^{0\cdot 1} & \cdots & W_N^{0\cdot(N-1)} \\ W_N^{1\cdot 0} & W_N^{1\cdot 1} & \cdots & W_N^{1\cdot(N-1)} \\ \vdots & \vdots & \cdots & \vdots \\ W_N^{(N-1)\cdot 0} & W_N^{(N-1)\cdot 1} & \cdots & W_N^{(N-1)\cdot(N-1)} \end{bmatrix} \cdot \boldsymbol{v} \tag{12.65}$$

If we denote the $N \times N$ DFT matrix as

$$\boldsymbol{W}_N \triangleq \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & \cdots & W_N^{(N-1)} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & W_N^{(N-1)} & \cdots & W_N^{(N-1)^2} \end{bmatrix} \tag{12.66a}$$

then $\boldsymbol{W}_N$ also has an inverse

$$\boldsymbol{W}_N^{-1} = \frac{1}{N} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N^{-1} & \cdots & W_N^{-(N-1)} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & W_N^{-(N-1)} & \cdots & W_N^{-(N-1)^2} \end{bmatrix} \tag{12.66b}$$

This can be verified (Prob. 12.7-1) by showing that

$$\boldsymbol{W}_N \cdot \boldsymbol{W}_N^{-1} = \boldsymbol{I}_{N\times N}$$

Given this notation, we have the relationship of

$$\boldsymbol{V} = \boldsymbol{W}_N \cdot \boldsymbol{v}$$
$$\boldsymbol{v} = \boldsymbol{W}_N^{-1} \cdot \boldsymbol{V}$$

An amazing property of the cyclic matrix $\mathcal{H}_{\text{cp}}$ can be established by applying the DFT and IDFT matrices.

$$\mathcal{H}_{\text{cp}} \cdot \boldsymbol{W}_N^{-1} = \begin{bmatrix} h[0] & h[1] & \cdots & h[L] & 0 & \cdots & 0 \\ 0 & h[0] & h[1] & \cdots & h[L] & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & h[0] & h[1] & \cdots & h[L] \\ h[L] & \ddots & \ddots & 0 & h[0] & \cdots & h[L-1] \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ h[1] & \cdots & h[L] & 0 & \cdots & 0 & h[0] \end{bmatrix}$$

$$\times \frac{1}{N}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N^{-1} & \cdots & W_N^{-(N-1)} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & W_N^{-(N-1)} & \cdots & W_N^{-(N-1)^2} \end{bmatrix}$$

$$= \frac{1}{N}\begin{bmatrix} H[0] & H[-1] & \cdots & H[-N+1] \\ H[0] & H[-1]W_N^{-1} & \cdots & H[-N+1]W_N^{-(N-1)} \\ \vdots & \vdots & \cdots & \vdots \\ H[0] & H[-1]W_N^{-(N-1)} & \cdots & H[-N+1]W_N^{-(N-1)(N-1)} \end{bmatrix}$$

$$= \frac{1}{N}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N^{-1} & \cdots & W_N^{-(N-1)} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & W_N^{-(N-1)} & \cdots & W_N^{-(N-1)^2} \end{bmatrix} \begin{bmatrix} H[0] & & & \\ & H[-1] & & \\ & & \ddots & \\ & & & H[-N+1] \end{bmatrix}$$

$$= \boldsymbol{W}_N^{-1} \cdot \boldsymbol{D}_H \tag{12.67a}$$

where we have defined the diagonal matrix with the channel DFT entries as

$$\boldsymbol{D}_H = \begin{bmatrix} H[0] & & & \\ & H[-1] & & \\ & & \ddots & \\ & & & H[-N+1] \end{bmatrix} = \begin{bmatrix} H[N] & & & \\ & H[N-1] & & \\ & & \ddots & \\ & & & H[1] \end{bmatrix}$$

The last equality follows from the periodic nature of $H[n]$ given in Eq. (12.62b). We leave it as homework to show that any cyclic matrix of size $N \times N$ can be diagonalized by premultiplication with $\boldsymbol{W}_N$ and postmultiplication with $\boldsymbol{W}_N^{-1}$ (Prob. 12.7-2).

Based on Eq. (12.67a) we have established the following very important relationship for OFDM:

$$\begin{aligned} \mathcal{H}_{\text{cp}} &= \boldsymbol{W}_N^{-1} \cdot \boldsymbol{D}_H \cdot \boldsymbol{W}_N \\ &= \left(\frac{1}{\sqrt{N}}\boldsymbol{W}_N\right)^{-1} \cdot \boldsymbol{D}_H \cdot \frac{1}{\sqrt{N}}\boldsymbol{W}_N \end{aligned} \tag{12.67b}$$

Recall that after the cyclic prefix has been added, the channel input-output relationship is reduced to Eq. (12.64b). As a result,

$$\begin{bmatrix} z[N] \\ z[N-1] \\ \vdots \\ z[1] \end{bmatrix} = \left(\frac{1}{\sqrt{N}}\boldsymbol{W}_N\right)^{-1} \cdot \boldsymbol{D}_H \cdot \left(\frac{1}{\sqrt{N}}\boldsymbol{W}_N\right) \begin{bmatrix} s_N \\ s_{N-1} \\ \vdots \\ s_1 \end{bmatrix} + \begin{bmatrix} \text{w}[N] \\ \text{w}[N-1] \\ \vdots \\ \text{w}[1] \end{bmatrix}$$

This means that if we put the information source data into

$$\tilde{\boldsymbol{s}} \triangleq \begin{bmatrix} \tilde{s}_N \\ \tilde{s}_{N-1} \\ \vdots \\ \tilde{s}_1 \end{bmatrix} = \left(\frac{1}{\sqrt{N}}\boldsymbol{W}_N\right) \begin{bmatrix} s_N \\ s_{N-1} \\ \vdots \\ s_1 \end{bmatrix}$$

then we can obtain the OFDM transmission symbols via

$$\boldsymbol{s} \triangleq \begin{bmatrix} s_N \\ s_{N-1} \\ \vdots \\ s_1 \end{bmatrix} = \left(\frac{1}{\sqrt{N}}\boldsymbol{W}_N\right)^{-1} \begin{bmatrix} \tilde{s}_N \\ \tilde{s}_{N-1} \\ \vdots \\ \tilde{s}_1 \end{bmatrix}$$

Despite the minor scalar $1/\sqrt{N}$, we can call the matrix transformation of $\sqrt{N}\boldsymbol{W}_N^{-1}$ the IDFT (inverse DFT) operation. In other words, we apply IDFT on the information source data $\tilde{\boldsymbol{s}}$ at the OFDM transmitter to obtain $\boldsymbol{s}$ before adding the cyclic prefix.

Similarly, we can also transform the channel output vector via

$$\tilde{\boldsymbol{z}} \triangleq \begin{bmatrix} \tilde{z}[N] \\ \tilde{z}[N-1] \\ \vdots \\ \tilde{z}[1] \end{bmatrix} = \left(\frac{1}{\sqrt{N}}\boldsymbol{W}_N\right) \begin{bmatrix} z[N] \\ z[N-1] \\ \vdots \\ z[1] \end{bmatrix}$$

Corresponding to the IDFT, this operation can also be named the DFT: Finally, we note that the noise vector at the channel output also undergoes the DFT:

$$\tilde{\mathbf{w}} \triangleq \begin{bmatrix} \tilde{\mathrm{w}}[N] \\ \tilde{\mathrm{w}}[N-1] \\ \vdots \\ \tilde{\mathrm{w}}[1] \end{bmatrix} = \left(\frac{1}{\sqrt{N}}\boldsymbol{W}_N\right) \begin{bmatrix} \mathrm{w}[N] \\ \mathrm{w}[N-1] \\ \vdots \\ \mathrm{w}[1] \end{bmatrix}$$

We now can see the very simple relationship between the source data and the channel output vector, which has undergone the DFT:

$$\tilde{\boldsymbol{z}} = \boldsymbol{D}_H\tilde{\boldsymbol{s}} + \tilde{\mathrm{w}} \tag{12.68a}$$

Because $\boldsymbol{D}_H$ is diagonal, this matrix product is essentially element-wise multiplication:

$$\tilde{z}[n] = H[n]\,\tilde{s}_n + \tilde{\mathrm{w}}[n] \qquad n = 1, \ldots, N \tag{12.68b}$$

This shows that we now equivalently have N parallel (sub)channels, each of which is just a scalar channel with gain $H[n]$. Each vector of N data symbols in OFDM transmission is known as an *OFDM frame* or an OFDM symbol. Each subchannel $H[n]$ is also known as a subcarrier.

Thus, by applying the IDFT on the source data vector and the DFT on the channel output vector, OFDM converts an ISI channel of order L into N parallel subchannels without ISI. We no longer have to deal with the complex convolution that involves the time domain channel response. Instead, every subchannel is a non-frequency-selective gain only. There is no ISI within each subchannel. The N parallel subchannels are independent of one another because their noises are independent. This is why such a modulation is known as orthogonal frequency division modulation (OFDM). The block diagram of an N-point OFDM system implementation with a linear FIR channel of order L is given in Fig. 12.10.

12.7.2 OFDM Channel Noise

According to Eq. (12.68b), each of the N channels acts like a separate carrier of frequency $f = n/NT$ with channel gain $H[n]$. Effectively, the original data symbols $\{\tilde{s}_n\}$ are split into N

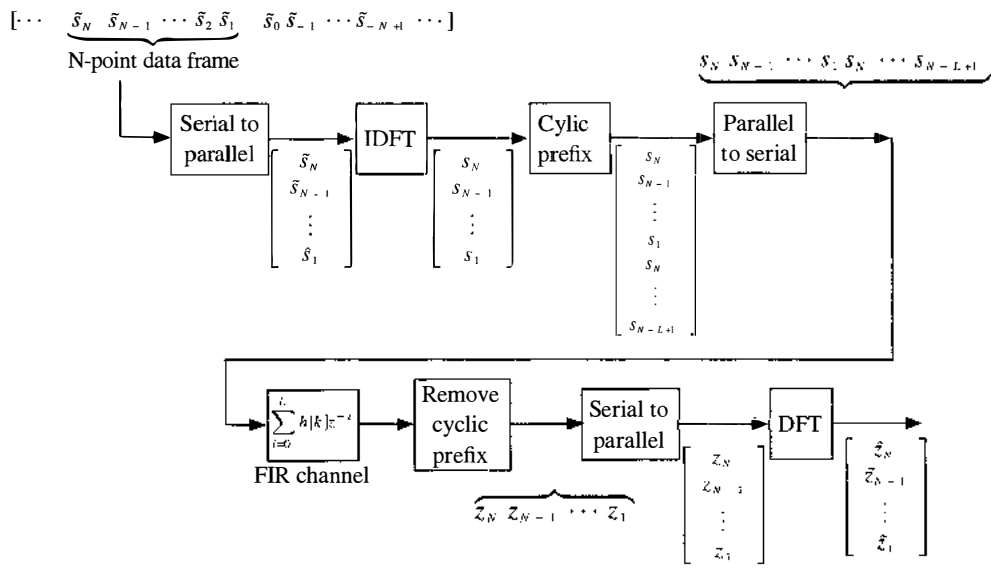


Figure 12.10 Illustration of an N-point OFDM transmission system.

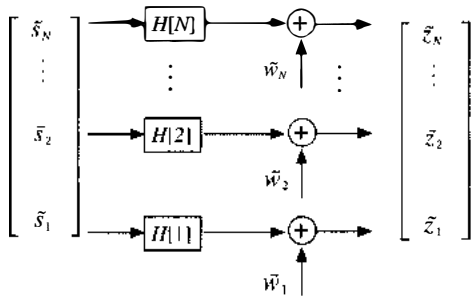


Figure 12.11 N independent AWGN channels generated by OFDM without ISI.

sequences and transmitted over N *subcarriers*. For this apparent reason, OFDM is also commonly known as a *multicarrier* communication system. Simply put, OFDM utilizes IDFT and cyclic prefix to effectively achieve multicarrier communications without the need to actually generate and modulate multiple (sub)carriers. The effective block diagram of OFDM appears in Fig. 12.11.

Now we can study the relationship between the transformed noise samples $\tilde{w}[n]$ in Fig. 12.11. First, notice that

$$\tilde{w}[N-j] = \sum_{k=0}^{N-1} W_N^{k\cdot(N-j)} w[N-k]$$
$$= \sum_{k=0}^{N-1} W_N^{-kj} w[N-k] \qquad j = 0, 1, \ldots, (N-1)$$

They are linear combinations of jointly distributed Gaussian noise samples $\{w[N-k]\}$. Therefore, $\{\tilde{w}[N-j]\}$ remains Gaussian. In addition, because $w[n]$ has zero mean, we have

$$\overline{\tilde{w}[N-j]} = \sum_{k=0}^{N-1} W_N^{-k\cdot j}\,\overline{w[N-k]} = 0, \qquad j = 0, 1, \ldots, (N-1)$$

$$\overline{\tilde{w}[N-i]\tilde{w}[N-j]^*} = \frac{1}{N}\overline{\sum_{k_1=0}^{N-1} W_N^{-k_1\cdot i} w[N-k_1] \sum_{k_2=0}^{N-1} W_N^{k_2\cdot j} w[N-k_2]^*}$$

$$
\begin{aligned}
&= \frac{1}{N}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1} W_N^{k_2 j - k_1 i}\,\overline{\mathrm{w}[N-k_1]\,\mathrm{w}[N-k_2]}^{*} \\
&= \frac{1}{N}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1} W_N^{k_2 j - k_1 i}\cdot\frac{\mathcal{N}}{2}\delta[k_1-k_2] \\
&= \mathcal{N}/2N\sum_{k_1=0}^{N-1} W_N^{k_1(j-i)} \\
&= \mathcal{N}/2N\cdot\begin{cases} N & i=j \\ 0 & i\neq j \end{cases} \\
&= \frac{\mathcal{N}}{2}\delta[i-j]
\end{aligned}
\tag{12.69}
$$

Because $\{\tilde{\mathrm{w}}[n]\}$ are zero mean with zero correlation, they are uncorrelated according to Eq. (12.69). Moreover, $\{\tilde{\mathrm{w}}[n]\}$ are also Gaussian noises. Since uncorrelated Gaussian random variables are also independent, $\{\tilde{\mathrm{w}}[n]\}$ are independent Gaussian noises with zero mean and identical variance of $\mathcal{N}/2$. The independence of the N channel noises demonstrates that OFDM converts an FIR channel with ISI and order up to L into N parallel, independent, and AWGN channels as shown in Fig. 12.11.

12.7.3 Zero-Padded OFDM

We have shown that by introducing a cyclic prefix of length L, a circular convolution channel matrix can be established. Because any circular matrix of size $N \times N$ can be diagonalized by IDFT and DFT (Prob. 12.7-2), the ISI channel of order less than or equal to L is transformed into N parallel independent subchannels.

There is also an alternative approach to the use of cyclic prefix. This method is known as **zero padding**. The transmitter first perform an IDFT on the N input data. Then, instead of repeating the last L symbols as in Eq. (12.64b) to transmit

$$
\begin{bmatrix} s_N \\ s_{N-1} \\ \vdots \\ s_1 \\ s_N \\ \vdots \\ s_{N-L+1} \end{bmatrix}
$$

we can simply replace the cyclic prefix with L zeros and transmit

$$
\left.\begin{bmatrix} s_N \\ s_{N-1} \\ \vdots \\ s_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right\} (N+L)\times 1
$$

The rest of the OFDM transmission steps remain unchanged. At the receiver end, we can stack up the received symbols in

$$y = \begin{bmatrix} z[N] \\ z[N-1] \\ \vdots \\ z[L] \\ \vdots \\ z[1] \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ z[N+L] \\ \vdots \\ z[N+1] \end{bmatrix}$$

We then can show (Prob. 12.7-4) that

$$\tilde{z} = \left(\frac{1}{\sqrt{N}} W_N \right) y \tag{12.70}$$

would achieve the same multichannel relationship of Eq. (12.68b).

12.7.4 Cyclic Prefix Redundancy in OFDM

The two critical steps of OFDM at the transmitter are the insertion of the cyclic prefix and the use of N-point IDFT. The necessary length of cyclic prefix L depends on the order of the FIR channel. Since the channel order may vary in practical systems, the OFDM transmitter must be aware of the maximum channel order information a priori.

Although it is acceptable for OFDM transmitters to use an overestimated channel order, the major disadvantage of inserting a longer-than-necessary cyclic prefix is the waste of channel bandwidth. To understand this drawback, notice that in OFDM, the cyclic prefix makes possible the successful transmission of N data symbols $\{\tilde{s}_1, \ldots, \tilde{s}_N\}$ with time duration $(N+L)T$. The L cyclic prefix symbols are introduced by OFDM as redundancy to remove the ISI in the original frequency-selective channel $H(z)$. Because $(N+L)$ symbol periods are now being used to transmit the N information data, the effective data rate of OFDM equals

$$\frac{N}{N+L} \frac{1}{T}$$

If L is overestimated, the effective data rate is reduced, and the transmission of the unnecessarily long cyclic prefix wastes some channel bandwidth. For this reason, OFDM transmitters require accurate knowledge about the channel delay spread to achieve good bandwidth efficiency. If the cyclic prefix is shorter than L, then the receiver is required to include a time domain filter known as the channel-shortening filter to reduce the effective channel-filter response to within LT.

12.7.5 OFDM Equalization

We have shown that OFDM converts an ISI channel into N parallel AWGN subchannels as shown in Fig. 12.11. Each of the N subchannels has an additive white Gaussian noise of zero mean and variance $\mathcal{N}/2$. The subchannel gain equals $H[k]$, which is the FIR frequency response at k/NT Hz. Strictly speaking, these N parallel channels do not have any ISI. Hence,

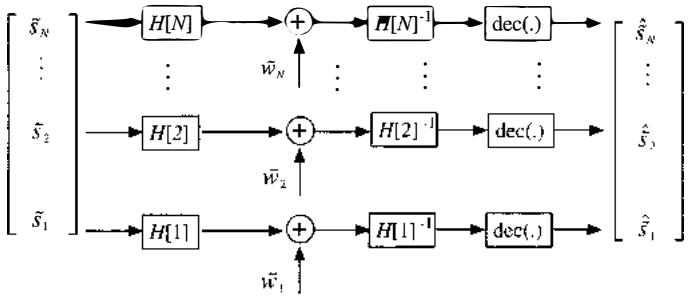


Figure 12.12 Using a bank of receiver gain adjustors for N independent AWGN channels in OFDM to achieve gain equalization.

channel equalization is not necessary. However, because each subchannel has a different gain, the optimum detection of $\{\tilde{s}_n\}$ from

$$\tilde{z}[n] = H[n]\tilde{s}_n \quad n = 1, \ldots, N$$

would require knowledge of the channel gain $H[n]$

$$\hat{\tilde{s}}_n = \text{dec}\left(H[n]^{-1}\tilde{z}[n]\right) \quad n = 1, \ldots, N$$

This resulting OFDM receiver is shown in Fig. 12.12. For each subchannel, a one-tap gain adjustment can be applied to compensate the subchannel scaling. In fact, this means that we need to implement a bank of N gain adjustment taps. The objective is to compensate the N subchannels such that the total gain of each data symbol equals unity before the QAM decision device. In fact, the gain equalizers scale both the subchannel signal and the noise equally. They do not change the subchannel SNR and do not change the detection accuracy. Indeed, equalizers are used only to facilitate the use of the same modular decision device on all subchannels. Oddly enough, this bank of gain elements at the receiver is exactly the same as the *equalizer* in a high-fidelity audio amplifier. This structure is known henceforth as a one-tap equalizer for OFDM receivers.

12.8 DISCRETE MULTITONE (DMT) MODULATIONS

A slightly different form of OFDM is called **discrete multitone** (**DMT**) modulation. In DMT, the basic signal processing operations are essentially identical to OFDM. The *only difference* between DMT and a standard OFDM is that DMT transmitters are given knowledge of the subchannel information. As a result, DMT transmits signals of differing constellations on different subchannels (known as subcarriers). As shown in Fig. 12.13, the single RF channel is split into N subchannels or subcarriers by OFDM or DMT. Each subcarrier conveys a distinct data sequence:

$$\{\cdots \tilde{s}_i[k+1]\ \tilde{s}_i[k]\ \tilde{s}_i[k-1]\ \cdots\}$$

The QAM constellations of the N sequences can often be different.

Because the original channel distortion is frequency selective, subchannel gains are generally different across the bandwidth. Thus, even though DMT or OFDM converts the channel with ISI distortion into N parallel independent channels without ISI, symbols transmitted over

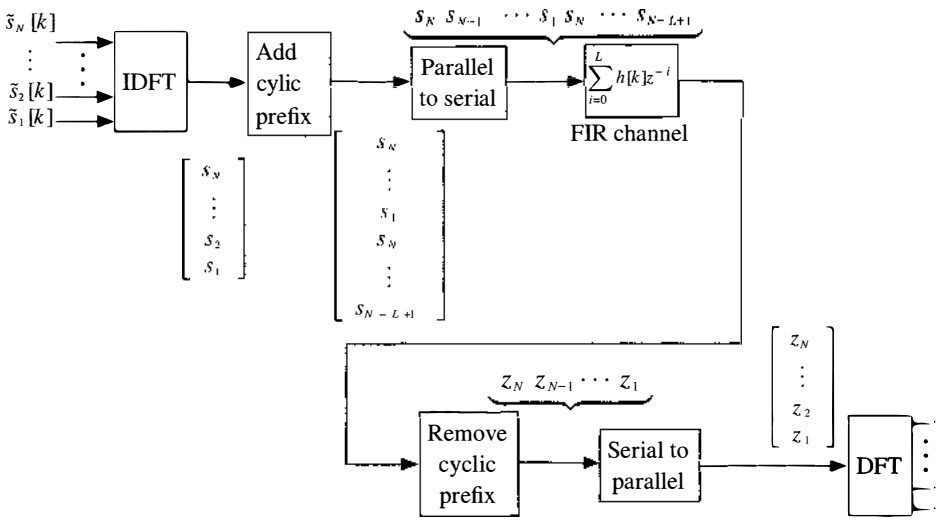


Figure 12.13 DMT transmission of N different symbol streams over a single FIR channel.

different subcarriers will encounter different SNRs at the receiver end. In DMT, the receivers are responsible for conveying to the transmitter all the subchannel information. As a result, the transmitter can implement compensatory measures to optimize various performance metrics. We mention two common approaches adopted at DMT transmitters:

- Subcarrier power loading to maximize average receiver SNR.
- Subcarrier bit loading to equalize the bit error rate (BER) across subcarriers.

Transmitter Power Loading for Maximizing Receiver SNR

To describe the idea of power loading at the transmitter for maximizing total receiver SNR, let $s_i[k]$ be the data stream carried by the ith subchannel and call $\{s_i[k]\}$ an independent data sequence in time k. Let us further say that all data sequences $\{s_i[k]\}$ are also independent of one another. Let the average power of $s_i[k]$ be

$$P_i = \overline{|s_i[k]|^2}$$

The total channel input power is

$$\sum_{i=1}^{M} P_i$$

whereas the corresponding channel output power at the receiver equals

$$\sum_{i=1}^{M} |H[i]|^2 \cdot P_i$$

Hence, the total channel output SNR is

$$\frac{\sum_{i=1}^{M} |H[i]|^2 \cdot P_i}{N\mathcal{N}/2} = \frac{2}{\mathcal{N}N} \sum_{i=1}^{M} |H[i]|^2 \cdot P_i$$

To determine the optimum power distribution, we would like to maximize the output SNR. Because the channel input power is limited, the optimization requires

$$\max_{\{P_i \geq 0\}} \sum_{i=1}^{N} |H[i]|^2 \cdot P_i \tag{12.71}$$

$$\text{subject to } \sum_{i=1}^{N} P_i = P$$

Once again, we can invoke the Cauchy-Schwartz inequality

$$\left| \sum_i a_i b_i \right|^2 \leq \sum_i |a_i|^2 \sum_i |b_i|^2$$

with equality if and only if $b_i = \lambda a_i^*$

Based on the Cauchy-Schwartz inequality,

$$\max_{\{P_i \geq 0\}} \sum_{i=1}^{N} |H[i]|^2 \cdot P_i = \sqrt{\sum_{i=1}^{N} |H[i]|^4 \cdot \sum_{i=1}^{N} |P_i|^2} \tag{12.72a}$$

if

$$P_i = \lambda |H[i]|^2 \tag{12.72b}$$

Because of the input power constraint $\sum_{i=1}^{N} P_i = P$, the optimum input power distribution should be

$$\sum_{i=1}^{N} P_i = \lambda \cdot \sum_{i=1}^{N} |H[i]|^2 = P \tag{12.73a}$$

In other words,

$$\lambda = \frac{1}{\sum_{i=1}^{N} |H[i]|^2} P \tag{12.73b}$$

Substituting Eq. (12.73b) into Eq. (12.72b), we can obtain the optimum channel input power loading across the N subchannels as

$$P_i = \frac{|H[i]|^2}{\sum_{i=1}^{N} |H[i]|^2} P \tag{12.74}$$

This optimum distribution of power in OFDM, also known as power loading, makes very good sense. When a channel has high gain, it is able to boost the power of its input much more effectively than a channel with low gain. Hence, the high-gain subchannels will be receiving higher power loading, while low-gain subchannels will receive much less. No power should be wasted on the extreme case of a subchannel that has zero gain, since the output of such a subchannel will make no power contribution to the total received signal power.

In addition to the perspective of maximizing average SNR, information theory can also rigorously prove the optimality of power loading (known as water pouring) in maximizing the capacity of frequency-selective channels. This discussion will be presented later (Sec. 13.7).

Subcarrier Bit Loading in DMT

If the transmitter has obtained the channel information $|H[i]|$, it then becomes possible for the transmitter to predict the detection error probability on the symbols transmitted over each subcarrier. The SNR of each subcarrier is

$$\text{SNR}_i = \frac{2|H[i]|^2}{\mathcal{N}} \overline{|s_i[k]|^2}$$

Therefore, the BER on this particular subcarrier depends on the SNR and the QAM constellation of the subcarrier. Different modulations at different subcarriers can lead to different powers $\overline{|s_i[k]|^2}$.

Consider the general case in which the ith subchannel carries K_i bits in each modulated symbol. Furthermore, we denote the BER of the ith subchannel by $P_b[i]$. Then the average receiver bit error rate across the N subcarriers is

$$P_b = \frac{\sum_{i=1}^{N} K_i \cdot P_b[i]}{\sum_{i=1}^{N} K_i}$$

If all subchannels apply the same QAM constellation, then K_i is constant for all i and

$$P_b = \frac{1}{N} \sum_{i=1}^{N} P_b[i]$$

Clearly, subchannels with a very weak SNR will generate many detection errors, while subchannels with a strong SNR will generate very few detection errors. If there is no power loading, then the ith subchannel SNR is proportional to the subchannel gain $|H[i]|^2$. In other words, BERs of poor subchannels can be larger than the BERs of good subchannels by several orders of magnitude. Hence, the average BER P_b will be dominated by those large $P_b[i]$ from poor subchannels. Based on this observation, we can see that to reduce the overall average BER, it is desirable to "equalize" the subchannel BER. By making each subchannel equally reliable, the average BER of the DMT system will improve. One effective way to "equalize" subchannel BER is to apply the practice of bit loading.[11, 12]

To describe the concept of bit loading, Table 12.1 illustrates the SNR necessary to achieve a detection error probability of 10^{-6} for five familiar constellations. It is clear that small constellations (e.g., BSPK, QPSK) require much lower SNRs than large constellations (e.g., 16-QAM, 32-QAM). This means that subcarriers with low gains should be assigned less complex constellations and should carry fewer bits per symbol. In the extreme case of subchannels with gains close to zero, no bit should be assigned and the subcarriers should be kept vacant.

TABLE 12.1
SNR Required to Achieve Detection Error Probability of 10^{-6}

Constellation	$E_b/\mathcal{N}$ at $P_e = 10^{-6}$, dB
BPSK	10.6
QPSK	10.6
8-PSK	14
16-QAM	14.5
32-QAM	17.4

On the other hand, subcarriers with large gains should be assigned more complex constellations and should carry many more bits in each symbol. This distribution of bits at the transmitter according to subcarrier conditions is called bit loading. In some cases, a subcarrier gain may be a little too low to carry n bits per symbol but too wasteful to carry $n-1$ bits per symbol. In such cases, the transmitter can apply additional power loading to this subcarrier. Therefore, DMT bit loading and power loading are often complementary at the transmitter.[11, 12] Figure 12.14 is a simple block diagram of the highly effective DMT bit-and-power loading.

Cyclic Prefix and Channel Shortening

The principles of OFDM and DMT require that the cyclic prefix be no shorter than the order of the FIR communication channel response. Although this requirement may be reasonable in a well-defined environment, for many applications, channel order or delay spread may have a large variable range. If a long cyclic prefix is always provisioned to target the worst-case (large) delay spread, then the overall bandwidth efficiency of the OFDM/DMT communication systems will be very low.

To overcome this problem, it is more desirable to apply an additional time domain equalizer (TEQ) at the receiver end to shorten the effective channel order. We note that the objective of this time domain equalizer (TEQ) is not to fully eliminate the ISI as in Sec. 13.3. Instead, the purpose of TEQ filter $G_{\text{TEQ}}(z)$ is to *shorten* the effective order of the combined response of channel equalizer such that

$$G_{\text{TEQ}}(z)H(z) \approx \sum_{k=0}^{L_1} q[k]z^{-k} \qquad L_1 < L$$

This channel-shortening task is less demanding than full ISI removal. By forcing L_1 to be (approximately) smaller than the original order L, a shorter cyclic prefix can be used to improve the OFDM/DMT transmission efficiency. The inclusion of a TEQ for channel shortening is illustrated in Fig. 12.15.

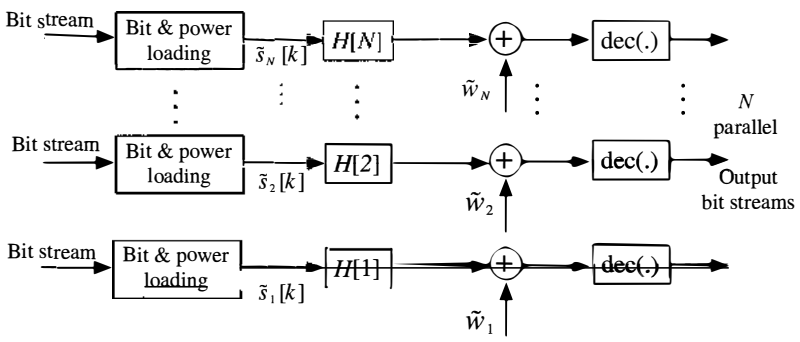


Figure 12.14 Bit and power loading in a DMT (OFDM) transmission system with N subcarriers.

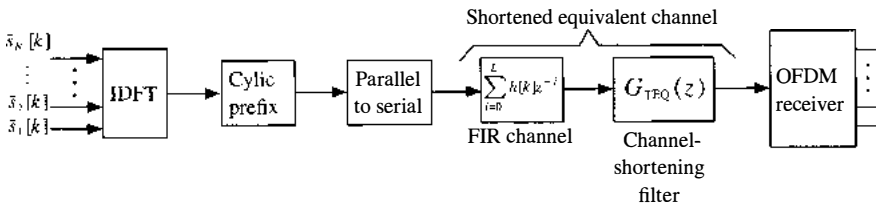


Figure 12.15 Time domain equalizer (TEQ) for channel shortening in DMT (OFDM) transmission system with N subcarriers.

12.9 REAL-LIFE APPLICATIONS OF OFDM AND DMT

OFDM is arguably one of the most successful signaling techniques for digital communications. Combined with transmitter power loading and bit loading, the benefits of OFDM include high spectral efficiency and resiliency against RF interferences and multipath distortion. As a result of the many advantages, there are a number of practical OFDM/DMT communication systems ranging from the wire-line digital subscriber line (DSL) system to the wireless ultrawideband (UWB) radio as well as satellite broadcasting.

Asymmetric Digital Subscriber Line (ADSL)

In the past few years, ADSL has replaced a vast majority of voice modems to become the dominant technology providing internet service to millions of homes. Conventional voice band modems use up to 3.4 kHz of analog bandwidth sampled at 8 kHz by the public switched telephone network (PSTN). These dial-up modems convert bits into waveforms that must fit into this tiny voice band. Because of the very small bandwidth, voice band modems are forced to apply very large QAM constellation (e.g., 960-QAM in V.34 for 28.8kbit/s). Large QAM constellation require very high transmission power and high complexity equalization. For these reasons, voice band modems quickly hit a rate plateau at 56kbit/s in the ITU-T V.90 recommendation.[13]

ADSL, on the other hand, is not limited by the telephone voice band. In fact, ADSL completely bypasses the voice telephone systems by specializing in data service. It relies on the traditional twisted pair of copper phone lines to provide the last-mile connection to individual homes. The main idea is that the copper wire channels in fact have bandwidth much larger than the 4 kHz voice band. However, as distance increases, the copper wire channel degrades rapidly at higher frequency. Hence, DSL can exploit the large telephone wire bandwidth (up to 1 MHz) only when the connection distance is short (1–5 km).[14]

The voice band is sometimes known as the plain-old-telephone-service (POTS) band. POTS and DSL data service are separated in frequency. The voice traffic continues to use the voice band below 3.4 kHz. DSL data uses the frequency band above the voice band. As shown in Fig. 12.16, the separation of the two signals is achieved by a simple (in-line) low-pass filter inserted between the phone outlet and each telephone unit when DSL service is available.

Figure 12.17 illustrates the bandwidth and subcarrier allocation of the ADSL system. From the top of the POTS band to the nominal ADSL upper limit of 1104 kHz, we have 255 equally spaced subchannels (subcarriers) of bandwidth 4.3175 kHz. These subcarriers are labeled 1 to 255. The lower number subcarriers, between 4.3175 and 25.875 kHz, may also be optionally used by some service providers. In typical cases, however, ADSL service providers utilize the

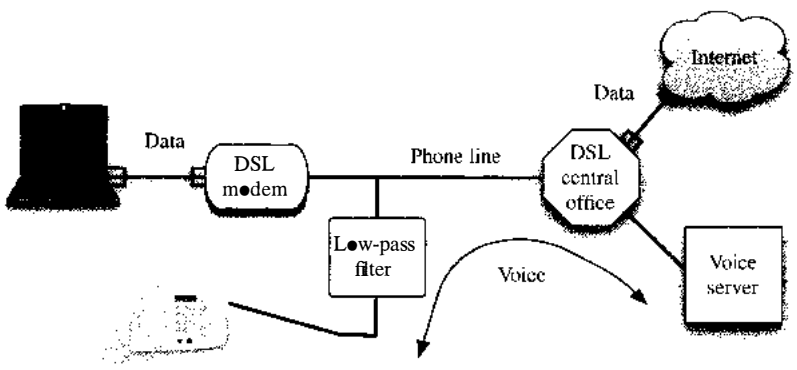


Figure 12.16 Data and voice share the same telephone line via frequency division. The data service is provided by the DSL central office, situated near the DSL modems.

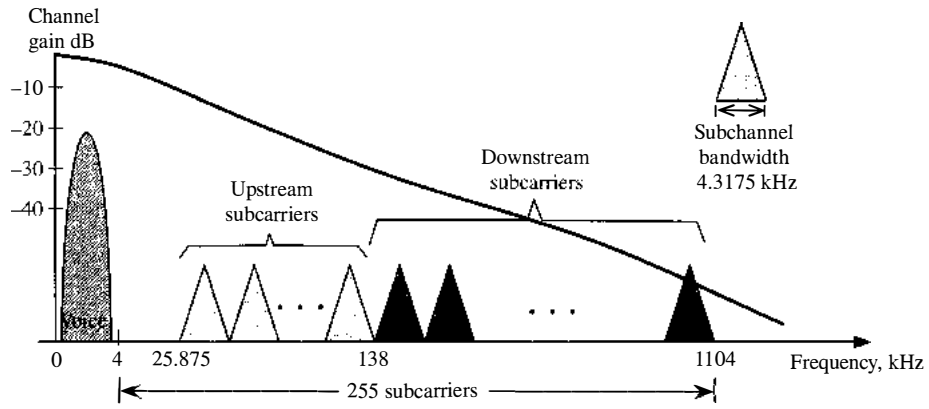


Figure 12.17 Frequency and subcarrier allocation in ADSL services.

TABLE 12.2
Basic ADSL Upstream and Downstream Subcarrier Allocations and Data Rates

	Upstream	Downstream
Modulation (bit loading)	QPSK to 64-QAM (2–6 bits per symbol)	
DMT frame transmission rate	4 kHz	
Pilot subcarrier	No. 64	No. 96
Typical subcarriers	6 to 32	33 to 255
Typical bits per frame	Up to 162 bits	Up to 1326 bits
Maximum possible subcarriers	1 to 63	1 to 255 (excluding 64 and 96)
Maximum bits per frame	Up to 378 bits	Up to 1518 bits
Maximum data rate	4 kHz $\times$378 = 1.512 Mbit/s	4 kHz $\times$1518 bits = 6.072 Mbit/s

nominal band of 25.875 to 1104 kHz (subcarrier 6 to subcarrier 255). These 250 available subcarriers are divided between downstream data transmission (from DSL server to homes) and upstream data (from homes to DSL server).

In today's internet applications, most individual consumers have a higher downstream need than upstream. Unlike business users, these "asymmetric" data service requirements define the objective of ADSL. Therefore in ADSL, the number of downstream subcarriers is greater than the number of upstream subcarriers. In ADSL, subcarriers 6 to 32 (corresponding to 25.875–138 kHz) are generally allocated for upstream data. Subcarrier 64 and subcarrier 96 are reserved for upstream pilot and downstream pilot, respectively. Excluding the two pilot subcarriers, subcarriers 33 to 255 (corresponding to 138–1104 kHz) are allocated for downstream data. The typical carrier allocation and data rates are summarized in Table 12.2. Notice that this table applies only to the basic DSL recommendations by ITU-T (G.992.1). Depending on the channel condition, various service providers may choose to increase the data rate by using higher bandwidth and even more subcarriers above subcarrier 255.

In ADSL, the DMT frame transmission rate is 4 kHz. Upstream DMT utilizes 64-point real-valued IFFT that is equivalent to 32-point complex IFFT. The upstream cyclic prefix has length 4. On downstream, 512 real-valued IFFT is applied, equivalent to 256-point complex IFFT. The downstream cyclic prefix has length 32 (equivalent to 16 complex numbers). Because the channel delay spread is usually larger than the prescribed cyclic prefix, TEQ channel shortening is commonly applied in ADSL with the help of several thousand training symbols (e.g., in downstream) to adapt the TEQ parameters.

Digital Broadcasting

Although North America has decided to adopt the ATSC standard for digital television broadcasting at the maximum rate of 19.39 Mbit/s using 8-VSB modulation, DVB-T (digital video broadcasting—terrestrial) has become a pan-European standard, also gaining acceptance in parts of Asia, Latin America, and Australia. DVB-T was first introduced in 1997,[15] utilizing OFDM over channels 6, 7, or 8 MHz wide.

DVB-T specifies three different OFDM transmission modes with increasing complexity for different target bit rates (video quality). It can use 2048 subcarriers (2K mode), 4096 subcarriers (4k mode), and 8196 subcarriers (8k mode). The cyclic prefix length may be 1/32, 1/16, 1/8, or 1/4 of the FFT length in the three different modes. Each subcarrier can have three modulation formats: QPSK, 16-QAM, or 64-QAM. When subchannel quality is poor, a simpler constellation such as QPSK is used. When subchannel SNR is high, the 64-QAM constellation is used. Different quality channels will bring about different video quality from standard-definition TV (SDTV) to high-definition TV (HDTV).

The DVB-H standard for mobile video reception by handheld mobile phones was published in 2004. The OFDM and QAM subcarrier modulation formats remain identical to those for DVB-T. For lower video quality multimedia services, digital multimedia broadcasting (DMB) also applies OFDM but limits itself to (differential) QPSK subcarrier modulation. Occupying less than 1.7 MHz bandwidth, DMB can use as many as 1536 subcarriers.

Broad OFDM Applications

DSL and DVB-T are only two limited applications of OFDM in digital communication systems. Overall, OFDM has found broad applications in numerous terrestrial wireless communication systems. An impressive list includes digital audio broadcasting (DAB), Wi-Fi (IEEE 802.11a, IEEE 802.11g), WiMAX (IEEE 802.16), ultrawideband (UWB) radio (IEEE 802.15.3a), 3rd Generation Partnership Project (3GPP) long-term-evolution (LTE), and high-speed OFDM packet access (HSOPA). Table 12.3 provides a snapshot of the important roles played by OFDM in various communication systems.

It is noticeable, however, that OFDM has not been very popular in satellite communications using directional antennas and in coaxial cable systems (e.g., cable modems, cable DTV). The reason is in fact quite obvious. Directional satellite channels and coaxial cable channels have very little frequency-selective distortion. In particular, they normally do not suffer from serious multipath effects. Without having to combat significant channel delay spread and ISI, OFDM would in fact be redundant. This is why systems such as digital satellite dish TV services and cable digital services all prefer the basic single-carrier modulation format. Direct broadcasting and terrestrial applications, on the other hand, often encounter multipath distortions and are perfect candidates for OFDM.

Digital Audio Broadcasting

As listed in Table 12.3, the European project Eureka 147 successfully launched OFDM digital audio broadcasting (DAB). Eureka 147 covers both terrestrial digital audio broadcasting and direct satellite audio broadcasting without directional receiving antennas. Receivers are equipped only with traditional omnidirectional antennas. Eureka 147 requires opening a new spectral band of 1.452 to 1.492 MHz in the L-band for both terrestrial and satellite broadcasting.

Despite the success of Eureka in Europe, however, concerns about spectral conflict in the L-band led the United States to decide against using Eureka 147. Instead, DAB in North America has split into satellite radio broadcasting by XM and Sirius, relying on proprietary technologies on the one hand and terrestrial broadcasting using the IBOC (in-band, on-channel) standard recommended by the FCC on the other. XM and Sirius competed as two separate

TABLE 12.3
A Short but Impressive History of OFDM Applications

Year	Events
1995	Digital audio broadcasting standard Eureka 147: first OFDM standard
1996	ADSL standard ANSI T1.413 (later became ITU G.992.1)
1997	DVB-T standard defined by ETSI
1998	Magic WAND project demonstrates OFDM modems for wireless LAN
1999	IEEE 802.11a wireless LAN standard (Wi-Fi)
2002	IEEE 802.11g standard for wireless LAN
2004	IEEE 802.16d standard for wireless MAN (WiMAX)
2004	MediaFLO announced by Qualcomm
2004	ETSI DVB-H standard
2004	Candidate for IEEE 802.15.3a (UWB) standard MB-OFDM
2004	Candidate for IEEE 802.11n standard for next-generation wireless LAN
2005	IEEE 802.16e (improved) standard for WiMAX
2005	Terrestrial DMB (T-DMB) standard (TS 102 427) adopted by ETSI (July)
2005	First T-DMB broadcast began in South Korea (December)
2005	Candidate for 3.75G mobile cellular standards (LTE and HSOPA)
2005	Candidate for CJK (China, Japan, Korea) 4G standard collaboration
2005	Candidate for IEEE P1675 standard for power line communications
2006	Candidate for IEEE 802.16m mobile WiMAX

companies before completing their merger in 2008. The new company, Sirius XM, serves satellite car radios, while IBOC targets traditional home radio customers. Sirius XM uses the 2.3 GHz S-band for direct satellite broadcasting. Under the commercial name of HD Radio developed by iBiquity Digital Corporation, IBOC allows analog FM and AM stations to use the same band to broadcast their content digitally by exploiting the gap between traditional AM and FM radio stations. By October 2008, over 1.5 million HD radio chipsets have been shipped and there were more than 1800 HD Radio Stations in the United States alone.

In satellite radio operation, XM radio uses the bandwidth of 2332.5 to 2345.0 MHz. This 12.5 MHz band is split into six carriers. Four carriers are used for satellite transmission. XM radio uses two geostationary satellites to transmit identical program content. The signals are transmitted with QPSK modulation from each satellite. For reliable reception, the line-of-sight signals transmitted from satellite 1 are received, reformatted to multicarrier modulation (OFDM), and rebroadcast by terrestrial repeaters. Each two-carrier group broadcasts 100 streams of 8 kbit/s. These streams represent compressed audio data. They are combined by means of a patented process to form a variable number of channels using a variety of bit rates.

Sirius satellite radio, on the other hand, uses three orbiting satellites over the frequency band of 2320 to 2332 MHz. These satellite are in lower orbit and are not geostationary. In fact, they follow a highly inclined elliptical Earth orbit (HEO), also known as the Tundra orbit. Each satellite completes one orbit in 24 hours and is therefore said to be geosynchronous. At any given time, two of the three satellites will cover North America. Thus, the 12 MHz bandwidth is equally divided among three carriers: two for the two satellites in coverage and one for terrestrial repeaters. The highly reliable QPSK modulation is adopted for Sirius transmission. Terrestrial repeaters are useful in some urban areas where satellite coverage may be blocked.

For terrestrial HD radio systems, OFDM is also key modulation technology in IBOC for both AM IBOC and the FM IBOC. Unlike satellite DAB, which bundles multiple station programs into a single data stream, AM IBOC and FM IBOC allow each station to use its own spectral allocation to broadcast, just like a traditional radio station. FM IBOC has broader

bandwidth per station and provides a higher data rate. With OFDM, the FM IBOC subchannel bandwidth equals 363.4 Hz, and the maximum number of subcarriers is 1093. Each subcarrier uses QPSK modulation. On the other hand, the AM IBOC subchannel bandwidth is 181.7 Hz (half as wide), and as many as 104 subcarriers may be used. Each subcarrier can apply 16-point QAM (secondary subcarriers) or 64 point QAM (primary subcarriers). Further details on IBOC can be found in the book by Maxson.[16]

12.10 BLIND EQUALIZATION AND IDENTIFICATION

Standard channel equalization and identification at receivers typically require a known (training) signal transmitted by the transmitter to assist in system identification. Alternatively, the training sequence can be used directly to determine the necessary channel equalizer. Figure 12.18 illustrates how a training signal can be used in the initial setup phase of the receiver.

During the training phase, a known sequence is transmitted by the transmitter such that the equalizer output can be compared with the desired input to form an error. The equalizer parameters can be adjusted to minimize the mean square symbol error. At the end of the training phase, the equalizer parameters should be near enough to their optimum values that much of the intersymbol interference (ISI) is removed. Now that the channel input can be correctly recovered from the equalizer output through a memoryless decision device (slicer), real data transmission can begin. The decision output $\hat{s}[k-u]$ can be used as the correct channel input to form the symbol error for continued equalizer adjustment or to track slow channel variations. The adaptive equalizer then obtains its reference signal from the decision output when the equalization system is switched to the *decision-directed* mode (Fig. 12.18). It is evident that this training mechanism can be applied regardless of the equalizer in use, be it TSE, FSE, or DFE.

In many communications, signals are transmitted over time-varying channels. As a result, a periodic training signal is necessary to identify or equalize the time-varying channel response. The drawback of this approach is evident in many communication systems where the use of training sequence can represent significant overhead costs or may even be impractical. For instance, no training signal is available to receivers attempting to intercept enemy communications. In a multicast or a broadcast system, it is highly undesirable for the transmitter to start a training session for each new receiver by temporarily suspending its normal transmission to all existing users. As a result, there is a strong and practical need for a special kind of channel equalizer, known as **blind** equalizers, that do not require the transmission of a training sequence. Digital cable TV and cable modems are excellent examples of such systems that can benefit from blind equalization.

There are a number of different approaches to the problem of blind equalization. In general, blind equalization methods can be classified into direct and indirect approaches. In the direct

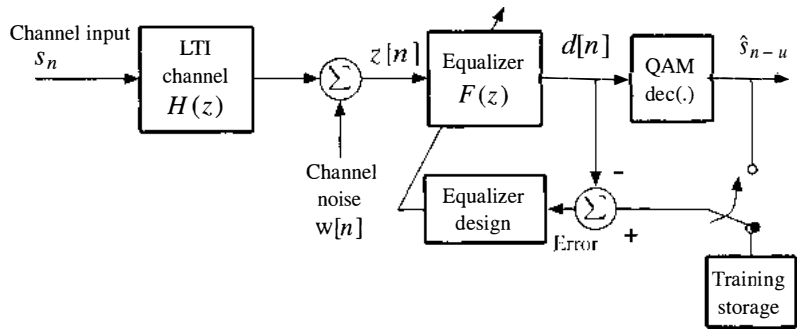


Figure 12.18 Channel equalization based on a training phase before switching to decision feedback mode.

blind equalization approach, equalizer filters are derived directly from input statistics and the observed output signal of the unknown channel. The indirect blind equalization approach first identifies the underlying channel impulse response before designing an appropriate equalizer filter or MLSE metrics. Understanding these subjects require in-depth reading of the literature, including papers from the 1980s by Benveniste et al.,[17, 18] who pioneered the terminology "blind equalization." Another very helpful source of information can be found in the papers by Godard,[19] Picchi and Prati,[20] Shalvi and Weinstein,[21, 22] Rupprecht,[23] Kennedy and Ding,[24] Tong et al.,[25] Moulines et al.,[26] and Brillinger and Rosenblatt.[28] For more systematic coverage, readers are referred to several published books on this topic.[6, 28, 29]

12.11 TIME-VARYING CHANNEL DISTORTIONS DUE TO MOBILITY

Thus far, we have focused on channel distortions that are invariant in time, or invariant at least for the period of concern. In mobile wireless communications, user mobility naturally leads to channel variation. Two main causes lead to time-varying channels: (1) a change of surroundings and (2) the Doppler effect. In most cases, a change of surroundings for a given user takes place at a much slower rate than the Doppler effect. For example, a transmitter/receiver traveling at the speed of 100 km/h, moves less than 2.8 meters in 100 ms. However, for carrier frequency of 900 MHz, the maximum corresponding Doppler frequency shift would be 83 Hz. This means that within 100 ms, the channel could have undergone 8 full cycles of change. Thus, unless the mobile unit suddenly turns a corner or enters a tunnel, the Doppler effect is usually far more severe than the effect of change in surroundings.

Doppler Shifts and Fading Channels

In mobile communications, the mobility of transmitters and receivers can lead to what is known as the *Doppler* effect, described by the nineteenth-century Austrian physicist Christian Doppler. He observed that the frequency of light and sound waves is affected by the relative motion of the source and the receiver. Radio waves experience the same Doppler effect when the transmitter or receiver is in motion. In the case of a narrowband RF transmission of a signal

$$m(t) \cos \omega_c t$$

if the relative velocity of the distance change between the source and the receiver equals v_d, then the received RF signal effectively has a new carrier

$$m(t) \cos (\omega_c + \omega_d)t \qquad \omega_d = \frac{v_d}{c}\omega_c$$

where c is the speed of light. Note that v_d and hence ω_d are negative when the source-to-receiver distance decreases and positive when it increases.

In the multipath environment, if the mobile user is traveling at a given speed v_d, then the line-of-sight path has the highest variation rate. This means that if there are $K + 1$ multipaths in the channel, the ith propagation path distance would vary at the velocity of v_i. The ith signal copy traveling along the ith path should have a Doppler shift

$$\omega_i = \frac{v_i}{c}\omega_c \tag{12.75}$$

Moreover, because

$$-v_d \leq v_i \leq v_d$$

the maximum Doppler shift is bounded by

$$|\omega_i| \leq \omega_{\max} = \frac{|v_d|}{c}\omega_c$$

Based on the Doppler analysis, each path has a Doppler frequency shift ω_i, delay τ_i, and path attenuation α_i. The signal from the ith path can be written as

$$\alpha_i \left[\sum_k \text{Re}\{s_k\} p(t - kT - \tau_i)\right] \cos[(\omega_c + \omega_i)(t - \tau_i)]$$
$$+ \alpha_i \left[\sum_k \text{Im}\{s_k\} p(t - kT - \tau_i)\right] \sin[(\omega_c + \omega_i)(t - \tau_i)] \tag{12.76}$$

As a result, the baseband receiver signal after demodulation is now

$$y(t) = \sum_k s_k \left\{\sum_{i=0}^{K} \underbrace{\alpha_i \exp[-j(\omega_c + \omega_i)\tau_i] \exp(-j\omega_i t)}_{\beta_i(t)} p(t - kT - \tau_i)\right\}$$
$$= \sum_k s_k \left[\sum_{i=0}^{K} \beta_i(t)\, p(t - kT - \tau_i)\right] \tag{12.77}$$

Frequency-Selective Fading Channel

Recall that the original baseband transmission is

$$x(t) = \sum_k s_k\, p(t - kT)$$

In the channel output of Eq. (12.77), if the mobile velocity is zero, then $\omega_i = 0$ and $\beta_i(t) = \beta_i$ are constant. In the case of zero mobility, the baseband channel output simply becomes

$$y(t) = \sum_k s_k \left[\sum_{i=0}^{K} \beta_i\, p(t - kT - \tau_i)\right]$$

This means that corresponding channel is linear time-invariant with impulse response

$$h(t) = \sum_{i=0}^{K} \beta_i\, \delta(t - \tau_i) \tag{12.78}$$

and transfer function

$$H(f) = \sum_{i=0}^{K} \beta_i \exp(-j2\pi f \tau_i) \tag{12.79}$$

This is a frequency-selective channel with intersymbol interference (ISI).

When the mobile speed v_d is not zero, then $\beta_i(t)$ are time-varying. As a result, the channel is no longer linear time-invariant. Instead, the channel is linear time-varying. Suppose the channel input is a pure sinusoid, $x(t) = \exp(j\omega_p t)$. The output of this time-varying channel according to Eq. (12.77) is

$$\sum_{i=0}^{K} \beta_i(t) \exp\left[j\omega_p(t-\tau_i)\right] = \exp(j\omega_p t) \cdot \sum_{i=0}^{K} \beta_i(t) \exp(-j\omega_p \tau_i) \tag{12.80}$$

This relationship shows that the channel response to a sinusoidal input equals a sinusoid of the same frequency but with time-varying amplitude. Moreover, the time-varying amplitude of the channel output also depends on the input frequency (ω_p). For these multipath channels, the channel response is *time-varying* and is *frequency dependent*. In wireless communications, time-varying channels are known as *fading* channels. When the time-varying behaviors are dependent on frequency, the channels are known as *frequency-selective fading channels*. Frequency-selective fading channels, which are characterized by time-varying ISI, are major obstacles to wireless digital communications.

Flat Fading Channels

One special case to consider is when the multipath delays $\{\tau_i\}$ do not have a large spread. In other words, let us assume

$$0 = \tau_0 < \tau_1 < \cdots < \tau_K$$

If the multipath delay spread is small, then $\tau_K \ll T$ and

$$\tau_i \approx 0 \qquad i = 1, 2, \ldots, K$$

In this special case, because $p(t-\tau_i) \approx p(t)$, the received signal $y(t)$ is simply

$$\begin{aligned} y(t) &= \sum_k s_k \left\{ \sum_{i=0}^{K} \alpha_i \exp\left[-j(\omega_c+\omega_i)\tau_i\right] \exp\left(-j\omega_i t\right) p(t-kT-\tau_i) \right\} \\ &\approx \sum_k s_k \left\{ \sum_{i=0}^{K} \alpha_i \exp\left[-j(\omega_c+\omega_i)\tau_i\right] \exp\left(-j\omega_i t\right) p(t-kT) \right\} \\ &= \left\{ \sum_{i=0}^{K} \alpha_i \exp\left[-j(\omega_c+\omega_i)\tau_i\right] \exp\left(-j\omega_i t\right) \right\} \sum_k s_k p(t-kT) \\ &= \rho(t) \cdot \sum_k a_k p(t-kT) \end{aligned} \tag{12.81}$$

where we have defined the time-varying channel gain as

$$\rho(t) = \sum_{i=0}^{K} \alpha_i \exp\left[-j(\omega_c+\omega_i)\tau_i\right] \exp\left(-j\omega_i t\right) \tag{12.82}$$

Therefore, when the multipath delay spread is small, the only distortion in the received signal $y(t)$ is a time-varying gain $\rho(t)$. This time-variation of the received signal strength

is known as fading. Channels that exhibit only a time-varying gain that is dependent on the environment are known as flat fading channels. Flat fading channels do not introduce any ISI and therefore do not require equalization. Instead, since flat fading channels generate output signals that have time-varying strength, periods of error-free detections tend to be followed by periods of error bursts. To overcome burst errors due to flat fading channels, interleaving forward error correction codewords is an effective tool.

Converting Frequency-Selective Fading Channels into Flat Fading Channels

Fast fading frequency-selective channels pose serious challenges to mobile wireless communications. On one hand, the channels introduce ISI. On the other hand, the channel characteristics are also time varying. Although the time domain equalization techniques described in Secs. 12.3 to 12.6 can effectively mitigate the effect of ISI, they require training data to either identify the channel parameters or estimate equalizer parameters. Generally, parameter estimation of channels or equalizers cannot work well unless the parameters stay nearly unchanged between successive training periods. As a result, such time domain channel equalizers are not well equipped to confront fast changing channels.

Fortunately, we do have an alternative. We have shown (in Sec. 12.7) that OFDM can convert a frequency-selective channel into a parallel group of flat channels. When the underlying channel is fast fading and frequency selective, OFDM can effectively converts it into a bank of fast flat-fading channels. As a result, means to combat fast flat-fading channels such as code interleaving can now be successfully applied to fast frequency-selective fading channels.

We should note that for fast fading channels, another very effective means to combat the fading effect is to introduce channel *diversity*. Channel diversity allows the same transmitted data to be sent over a plurality of channels. Channel diversity can be achieved in the time domain by repetition, in the frequency domain by using multiple bands, or in space by applying multiple transmitting and receiving antennas. Because both time diversity and frequency diversity occupy more bandwidth, spatial diversity in the form of multiple-input–multiple-output (MIMO) systems has been particularly attractive recently. Among recent wireless standards, Wi-Fi (IEEE 802.11n), WiMAX (IEEE 802.16e), and cellular LTE (long-term evolution) have all adopted OFDM and MIMO technologies to achieve much higher data rate and better coverage. We shall present some fundamental discussions on MIMO in Chapter 13.

12.12 MATLAB EXERCISES

We provide three different computer exercises in this section; all model a QAM communication system that modulates data using 16-QAM constellation. The 16-QAM signals then pass through linear channels with ISI and encounter additive white Gaussian noise (AWGN) at the channel output.

COMPUTER EXERCIESE 12.1: 16-QAM LINEAR EQUALIZATION

The first MATLAB program, `Ex12_1.m`, generates 1,000,000 points of 16-QAM data for transmission. Each QAM requires T as the symbol period. The transmitted pulse shape is a root-raised cosine with a roll-off factor of 0.5 [Eq. (12.23)]. Thus the bandwidth at the baseband is $0.75/T$ Hz.

```
% Matlab Program <Ex12_1.m>
% This Matlab exercise <Ex12_1.m> performs simulation of
% linear equalization under QAM-16 baseband transmission
% a multipath channel with AWGN.
```

```
% Correct carrier and synchronization is assumed.
% Root-raised cosine pulse of rolloff factor = 0.5 is used
% Matched filter is applied at the receiver front end.
% The program estimates the symbol error rate (SER) at different Eb/N
clear;clf;
L=1000000;            % Total data symbols in experiment is 1 million
% To display the pulse shape, we oversample the signal
% by factor of f_ovsamp=8
f_ovsamp=8;             % Oversampling factor vs data rate
delay_rc=4;
% Generating root-raised cosine pulseshape (rolloff factor = 0.5)
prcos=rcosflt([ 1 ], 1, f_ovsamp, 'sqrt', 0.5, delay_rc); % RRC pulse
prcos=prcos(1:end-f_ovsamp+1);                          % remove 0's
prcos=prcos/norm(prcos);                                 % normalize
pcmatch=prcos(end:-1:1);                                 % MF

% Generating random signal data for polar signaling
s_data=4*round(rand(L,1))+2*round(rand(L,1))-3+...
    +j*(4*round(rand(L,1))+2*round(rand(L,1))-3);
% upsample to match the 'oversampling rate' (normalize by 1/T).
% It is f_ovsamp/T   (T=1 is the symbol duration)
s_up=upsample(s_data,f_ovsamp);

% Identify the decision delays due to pulse shaping
% and matched filters
delayrc=2*delay_rc*f_ovsamp;
% Generate polar signaling of different pulse-shaping
xrcos=conv(s_up,prcos);
[c_num,c_den] = cheby2(12,20,(1+0.5)/8);
% The next commented line finds frequency response
%[H,fnlz]=freqz(c_num,c_den,512,8);

% The lowpass filter is the Tx filter before signal is sent to channel
xchout=filter(c_num,c_den,xrcos);

% We can now plot the power spectral densities of the two signals
%           xrcos and xchout
% This shows the filtering effect of the Tx filter before
% transmission in terms of the signal power spectral densities
% It shows how little lowpass Tx filter may have distorted the signal
plotPSD_comparison

% Apply a 2-ray multipath channel
mpath=[1 0 0 -0.65];         % multipath delta(t)-0.65 delta(t-3T/8)
% time-domain multipath channel
h=conv(conv(prcos,pcmatch),mpath);
hscale=norm(h);

xchout=conv(mpath,xchout);      % apply 2-ray multipath
xrxout=conv(xchout,pcmatch); % send the signal through matched filter
                                % separately from the noise
delaychb=delayrc+3;
out_mf=xrxout(delaychb+1:f_ovsamp:delaychb+L*f_ovsamp);
clear xrxout;
```

```
% Generate complex random noise for channel output
noiseq=randn(L*f_ovsamp,1)+j*randn(L*f_ovsamp,1);
% send AWGN noise into matched filter first
noiseflt=filter(pcmatch,[1],noiseq);   clear noiseq;
% Generate sampled noise after matched filter before scaling it
% and adding to the QAM signal
noisesamp=noiseflt(1:f_ovsamp:L*f_ovsamp,1);

clear noiseq noiseflt;
Es=10*hscale;                  % symbol energy

% Call linear equalizer receiver to work
linear_eq

for ii=1:10;
    Eb2Naz(ii)=2*ii-2;
    Q(ii)=3*0.5*erfc(sqrt((2*10^(Eb2Naz(ii)*0.1)/5)/2));
%Compute the Analytical BER
end
% Now plot results
plotQAM_results
```

The transmission is over a two-ray multipath channel with impulse response

$$h(t) = g(t) - 0.65g(t - 3T/8)$$

where $g(t)$ is the response of a low-pass channel formed by applying a type II Chebyshev filter of order 12, a stopband gap of 20 dB, and bandwidth of $0.75/T$ Hz. The impulse response of this channel is shown in Fig. 12.19.

The main program `Ex12_1.m` will call a subroutine program `plotPSD_comparison.m` to first generate the power spectral densities of the transmitted signal before and after the low-pass Chebyshev filter. The comparison in Fig. 12.20 shows that the root-raised-cosine design is almost ideally band-limited, as the low-pass channel introduces very little change in the **passband** of the transmitted signal spectrum. This means that the multipath environment is solely responsible for the ISI effect.

```
% MATLAB PROGRAM <plotPSD_comparison.m>
% This program computes the PSD of the QAM signal before and after it
% enters a good chebyshev lowpass filter prior to entering the channel
%
[Pdfy,fq]=pwelch(xchout,[],[],1024,8,'twosided'); % PSD before
Tx filter
[Pdfx,fp]=pwelch(xrcos,[],[],1024,8,'twosided');  % PSD after
Tx filter
figure(1);
subplot(211);semilogy(fp-f_ovsamp/2,fftshift(Pdfx),'b-');
axis([-4 4 1.e-10 1.2e0]);
xlabel('Frequency (in unit of 1/T_s)');ylabel('Power Spectrum');
title('(a) Lowpass filter input spectrum')
subplot(212);semilogy(fq-f_ovsamp/2,fftshift(Pdfy),'b-');
axis([-4 4 1.e-10 1.2e0]);
xlabel('Frequency (in unit of 1/T_s)');ylabel('Power Spectrum');
title('(b) Lowpass filter output spectrum')
```

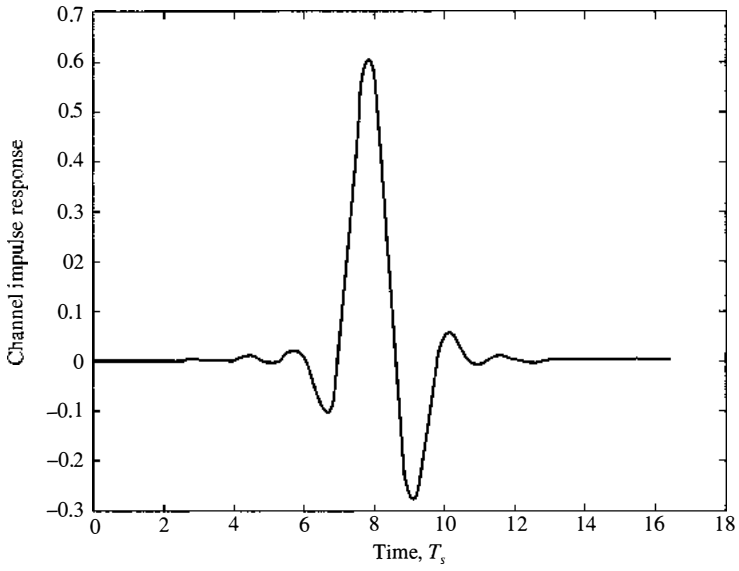
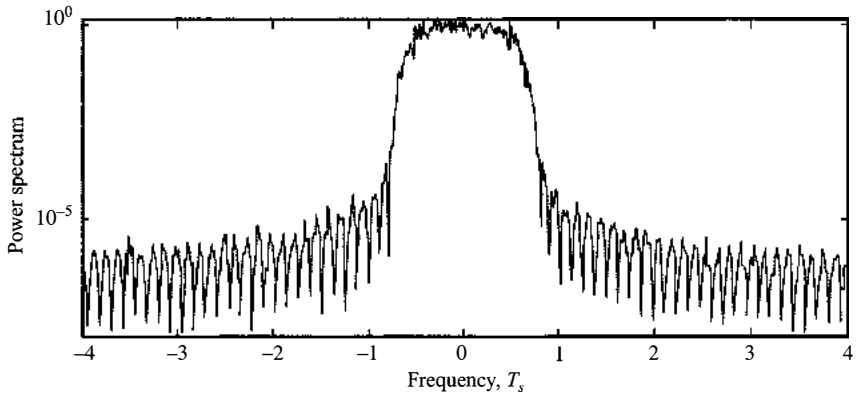
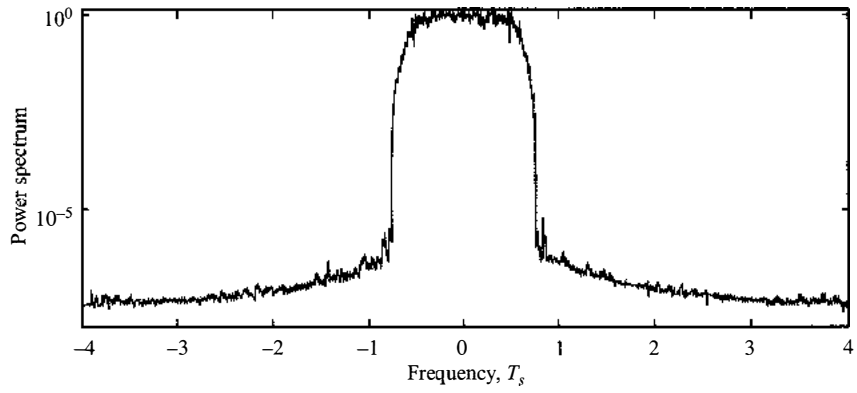


Figure 12.19 Two-ray multipath channel response for QAM transmission.



(a)



(b)

Figure 12.20 Power spectral densities of the root-raised-cosine QAM signal before and after a low-pass channel of bandwidth $0.75/T$: (a) input and (b) output of low-pass filter spectra.

After a matched filter has been applied at the receiver (root-raised cosine), the QAM signal will be sampled, equalized, and decoded. The subroutine program `linear_eq.m` designs a T-spaced finite length MMSE equalizer of order $M = 8$ as described in Sec. 12.3 [Eq. (12.43b)]. The equalizer is designed by applying the first 200 QAM symbols as training data. The equalizer filters the matched filter output before making a 16-QAM decision according to the decision region of Fig. 10.24b in Chapter 10.

```
% MATLAB PROGRAM <linear_eq.m>
% This is the receiver part of the QAM equalization example
%
Ntrain=200;              % Number of training symbols for Equalization
Neq=8;                   % Order of linear equalizer (=length-1)
u=0;                     % equalization delay u must be <= Neq
SERneq=[];
SEReq=[];
for i=1:13,
    Eb2N(i)=i*2-1;                              %(Eb/N in dB)
    Eb2N_num=10^(Eb2N(i)/10);                  % Eb/N in numeral
    Var_n=Es/(2*Eb2N_num);                  %1/SNR is the noise variance
    signois=sqrt(Var_n/2);                     % standard deviation
    z1=out_mf+signois*noisesamp;               % Add noise

    Z=toeplitz(z1(Neq+1:Ntrain),z1(Neq+1:-1:1));  % signal matrx for
                                                    % computing R
    dvec=[s_data(Neq+1-u:Ntrain-u)];    % build training data vector
    f=pinv(Z'*Z)*Z'*dvec;                      % equalizer tap vector
    dsig=filter(f,1,z1);                       % apply FIR equalizer
        % Decision based on the Re/Im parts of the samples
    deq=sign(real(dsig(1:L)))+sign(real(dsig(1:L))-2)+...
        sign(real(dsig(1:L))+2)+...
        j*(sign(imag(dsig(1:L)))+sign(imag(dsig(1:L))-2)+...
        sign(imag(dsig(1:L))+2));
    % Now compare against the original data to compute SER
    % (1) for the case without equalizer
    dneq=sign(real(z1(1:L)))+sign(real(z1(1:L))-2)+...
        sign(real(z1(1:L))+2)+...
        j*(sign(imag(dsig(1:L)))+sign(imag(z1(1:L))-2)+...
        sign(imag(z1(1:L))+2));
    SERneq=[SERneq;sum(abs(s_data~=dneq))/(L)];
    % (2) for the case with equalizer
    SEReq=[SEReq;sum(s_data~=deq)/L];
end
```

Once the linear equalization results are available, the main program `Ex12_1.m` calls another subroutine program, `plotQAM_results.m`, to provide illustrative figures. In Fig. 12.21, the noise-free eye diagram of the in-phase component at the output of the receiver matched filter before sampling shows a strong ISI effect. The QAM signal eye is closed and, without equalization, a simple QAM decision leads to very high probabilities of symbol error (also known as symbol error rate).

```
% MATLAB PROGRAM <plotQAM_results.m>
% This program plots symbol error rate comparison before and after
% equalization
%                    constellation points
%                    eye-diagrams before equalization
figure(2)
```

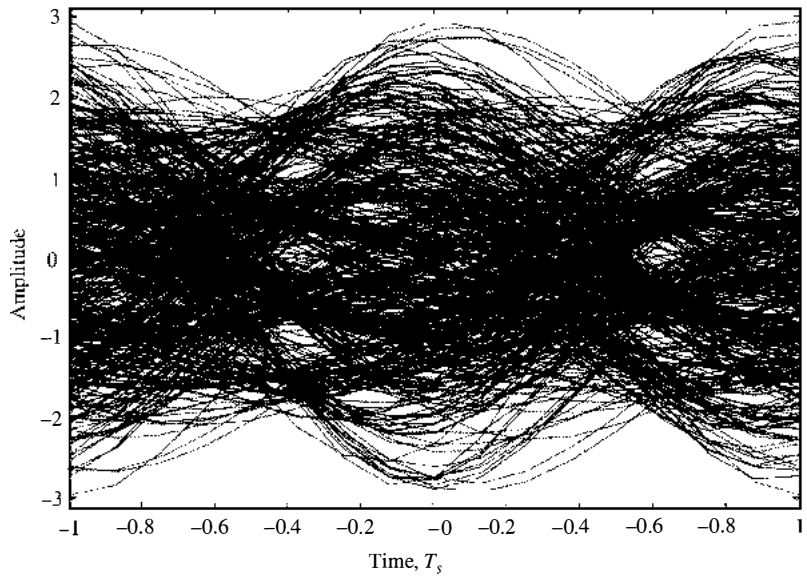


Figure 12.21 Noise-free eye diagram of the in-phase (real) component at the receiver (after matched filter) before sampling: the eyes are closed, and ISI will lead to decision errors.

```
subplot(111)
figber=semilogy(Eb2Naz,Q,'k-',Eb2N,SERneq,'b-o',Eb2N,SEReq,'b-v');
axis([0 26 .99e-5 1]);
legend('Analytical', 'Without equalizer', 'With equalizer');
xlabel('E_b/N (dB)');ylabel('Symbol error probability');
set(figber,'Linewidth',2);
% Constellation plot before and after equalization
figure(3)
subplot(121)
plot(real(z1(1:min(L,4000))),imag(z1(1:min(L,4000))),'.');
axis('square')
xlabel('Real part')
title('(a) Before equalization')
ylabel('Imaginary part');
subplot(122)
plot(real(dsig(1:min(L,4000))),imag(dsig(1:min(L,4000))),'.');
axis('square')
title('(b) After equalization')
xlabel('Real part')
ylabel('Imaginary part');
figure(4)
t=length(h);
plot([1:t]/f_ovsamp,h);
xlabel('time (in unit of T_s)')
title('Multipath channel impulse response');
%  Plot eye diagrams due to multipath channel
eyevec=conv(xchout,prcos);
eyevec=eyevec(delaychb+1:(delaychb+800)*f_ovsamp);
eyediagram(real(eyevec),16,2);
title('Eye diagram (in-phase component)');
xlabel('Time (in unit of T_s)');
```

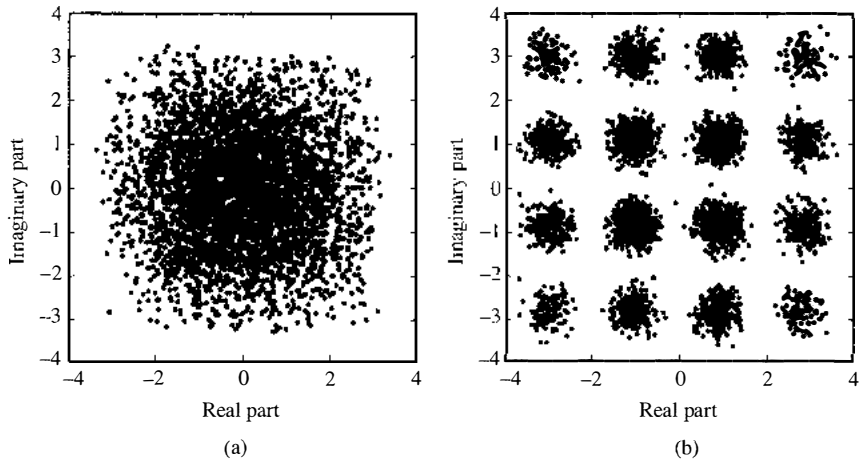


Figure 12.22 Scatter plots of signal samples before (a) and after (b) linear equalization at $E_b/\mathcal{N} = 26$ dB demonstrate effective ISI mitigation by the linear equalizer.

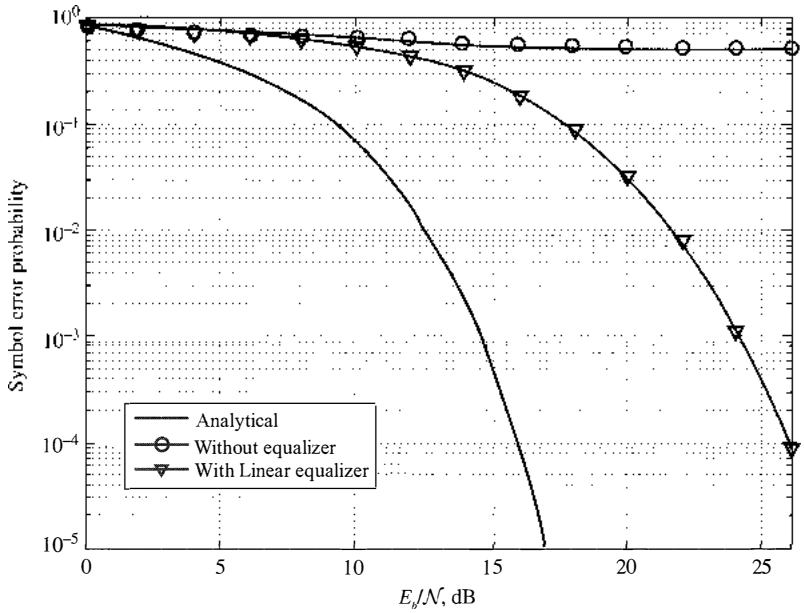


Figure 12.23 Symbol error rate (SER) comparison before and after linear equalization demonstrates its effectiveness in combating multipath channel ISI.

We can suppress a significant amount of ISI by applying the linear equalizer to the sampled matched filter output. Figure 12.22 compares the "scatter plot" of signal samples before and after equalization at $E_b/\mathcal{N} = 26$ dB. The contrast illustrates that the equalizer has effectively mitigated much of the ISI introduced by the multipath channel.

The program `linear_eq.m` also statistically computes the symbol error rate (SER) at different SNR levels. It further computes the ideal SER according to ISI-free AWGN channel (Chapter 10) and, for comparision, the SER without equalization. The results shown in Fig. 12.23 clearly demonstrate the effectiveness of linear equalization in this example.

COMPUTER EXERCISE 12.2: DECISION FEEDBACK EQUALIZATION

In this exercise, we use the main MATLAB program, `Ex12_2.m`, to generate the same kind of data as in the last exercise. The main difference is that we adopt a slightly different two-ray multipath channel

$$h(t) = g(t) - 0.83g(t - 3T/8)$$

in which the ISI is much more severe. At the receiver, instead of using linear equalizers, we will implement and test the decision feedback equalizer (DFE) as described in Sec. 12.6. For simplicity, we will implement only a DFE feedback filter, without using the FFW filter.

```
% Matlab Program <Ex12_2.m>
% This Matlab exercise <Ex12_2.m> performs simulation of
% decision feedback equalization under QAM-16 baseband transmission
% a multipath channel with AWGN.
% Correct carrier and synchronization is assumed.
% Root-raised cosine pulse of rolloff factor = 0.5 is used
% Matched filter is applied at the receiver front end.
% The program estimates the symbol error rate (SER) at different Eb/N
clear;clf;
L=100000;          % Total data symbols in experiment is 1 million
% To display the pulse shape, we oversample the signal
% by factor of f_ovsamp=8
f_ovsamp=8;             % Oversampling factor vs data rate
delay_rc=4;
% Generating root-raised cosine pulseshape (rolloff factor = 0.5)
prcos=rcosflt([ 1 ], 1, f_ovsamp, 'sqrt', 0.5, delay_rc); % RRC pulse
prcos=prcos(1:end-f_ovsamp+1);                              % remove 0's
prcos=prcos/norm(prcos);                                     % normalize
pcmatch=prcos(end:-1:1);                                     % MF

% Generating random signal data for polar signaling
s_data=4*round(rand(L,1))+2*round(rand(L,1))-3+...
    +j*(4*round(rand(L,1))+2*round(rand(L,1))-3);
% upsample to match the 'oversampling rate' (normalize by 1/T).
% It is f_ovsamp/T   (T=1 is the symbol duration)
s_up=upsample(s_data,f_ovsamp);

% Identify the decision delays due to pulse shaping
% and matched filters
delayrc=2*delay_rc*f_ovsamp;
% Generate polar signaling of different pulse-shaping
xrcos=conv(s_up,prcos);
[c_num,c_den] = cheby2(12,20,(1+0.5)/8);
% The next commented line finds frequency response
%[H,fnlz]=freqz(c_num,c_den,512,8);

% The lowpass filter is the Tx filter before signal is sent to channel
xchout=filter(c_num,c_den,xrcos);

% We can now plot the power spectral densities of the two signals
%           xrcos and xchout
% This shows the filtering effect of the Tx filter before
% transmission in terms of the signal power spectral densities
```

```
% It shows how little lowpass Tx filter may have distorted the signal
plotPSD_comparison

% Apply a 2-ray multipath channel
mpath=[1 0 0 -0.83];          % multipath delta(t)-0.83 delta(t-3T/8)
%                                  or use mpath=[1 0 0 .45];
% time-domain multipath channel
h=conv(conv(prcos,pcmatch),mpath);
hscale=norm(h);

xchout=conv(mpath,xchout);       % apply 2-ray multipath
xrxout=conv(xchout,pcmatch); % send the signal through matched filter
                                  % separately from the noise
delaychb=delayrc+3;
out_mf=xrxout(delaychb+1:f_ovsamp:delaychb+L*f_ovsamp);
clear xrxout;

% Generate complex random noise for channel output
noiseq=randn(L*f_ovsamp,1)+j*randn(L*f_ovsamp,1);
% send AWGN noise into matched filter first
noiseflt=filter(pcmatch,[1],noiseq);   clear noiseq;
% Generate sampled noise after matched filter before scaling it
% and adding to the QAM signal
noisesamp=noiseflt(1:f_ovsamp:L*f_ovsamp,1);

clear noiseq noiseflt;
Es=10*hscale;                  % symbol energy

% Call decision feedback equalizer receiver to work
dfe
SERdfe=SEReq;
for ii=1:9;
    Eb2Naz(ii)=2*ii;
    Q(ii)=3*0.5*erfc(sqrt((2*10^(Eb2Naz(ii)*0.1)/5)/2));
%Compute the Analytical BER
end
% use the program plotQAM_results to show results
plotQAM_results
linear_eq
```

At the receiver, once the signal has passed through the root-raised-cosine matched filter, the T-spaced samples will be sent into the DFE. The subroutine program `dfe.m` implements the DFE design and the actual equalization. The DFE design requires the receiver to first estimate the discrete channel response. We use the first 200 QAM symbols as training data for channel estimation. We then compute the SER of the DFE output in `dfe.m`. The necessary program `dfe.m` is given here

```
% MATLAB PROGRAM <dfe.m>
% This is the receiver part of the QAM equalization
% that uses Decision feedback equalizer (DFE)
%
Ntrain=200;              % Number of training symbols for Equalization
Nch=3;                   % Order of FIR channel (=length-1)
SEReq=[]; SERneq=[];
```

```
for i=1:13,
    Eb2N(i)=i*2-1;                              %(Eb/N in dB)
    Eb2N_num=10^(Eb2N(i)/10);                  % Eb/N in numeral
    Var_n=Es/(2*Eb2N_num);                     %1/SNR is the noise
                                                variance
    signois=sqrt(Var_n/2);                     % standard deviation
    z1=out_mf+signois*noisesamp;               % Add noise

    Z=toeplitz(s_data(Nch+1:Ntrain),s_data(Nch+1:-1:1));
    % signal matrx for
                                                          % computing R
    dvec=[z1(Nch+1:Ntrain)];
    % build training data vector
    h_hat=pinv(Z'*Z)*Z'*dvec;
    % find channel estimate tap vector
    z1=z1/h_hat(1);
    % equalize the gain loss
    h_hat=h_hat(2:end)/h_hat(1);
    % set the leading tap to 1

    feedbk=zeros(1,Nch);
    for kj=1:L,
        zfk=feedbk*h_hat;                      % feedback data
        dsig(kj)=z1(kj)-zfk;                   % subtract the feedback
        % Now make decision after feedback
        d_temp=sign(real(dsig(kj)))+sign(real(dsig(kj))-2)+...
        sign(real(dsig(kj))+2)+...
        j*(sign(imag(dsig(kj)))+sign(imag(dsig(kj))-2)+...
        sign(imag(dsig(kj))+2));
        feedbk=[d_temp feedbk(1:Nch-1)];
        % update the feedback data
    end
    %  Now compute the entire DFE decision after decision feedback
    dfeq=sign(real(dsig))+sign(real(dsig)-2)+...
        sign(real(dsig)+2)+...
        j*(sign(imag(dsig))+sign(imag(dsig)-2)+...
        sign(imag(dsig)+2));
    dfeq=reshape(dfeq,L,1);
    % Compute the SER after decision feedback equalization
    SEReq=[SEReq;sum(s_data(1:L)~=dfeq)/L];
    % find the decision without DFE
      dneq=sign(real(z1(1:L)))+sign(real(z1(1:L))-2)+...
        sign(real(z1(1:L))+2)+...
        j*(sign(imag(z1(1:L)))+sign(imag(z1(1:L))-2)+...
        sign(imag(z1(1:L))+2));
    % Compute the SER without equalization
    SERneq=[SERneq;sum(abs(s_data~=dneq))/(L)];
end
```

Once the SER of the DFE has been determined, it is compared against the SER of the linear equalization from the last exercise, along with the SER from ideal AWGN channel and the SER from a receiver without equalization. We provide the results in Fig. 12.24. From the comparison, we can see that both the DFE and the linear equalizer are effective at mitigating channel ISI. The linear equalizer is slightly better at lower SNR because the DFE is more susceptible to error propagation (Sec. 12.6) at lower SNR.

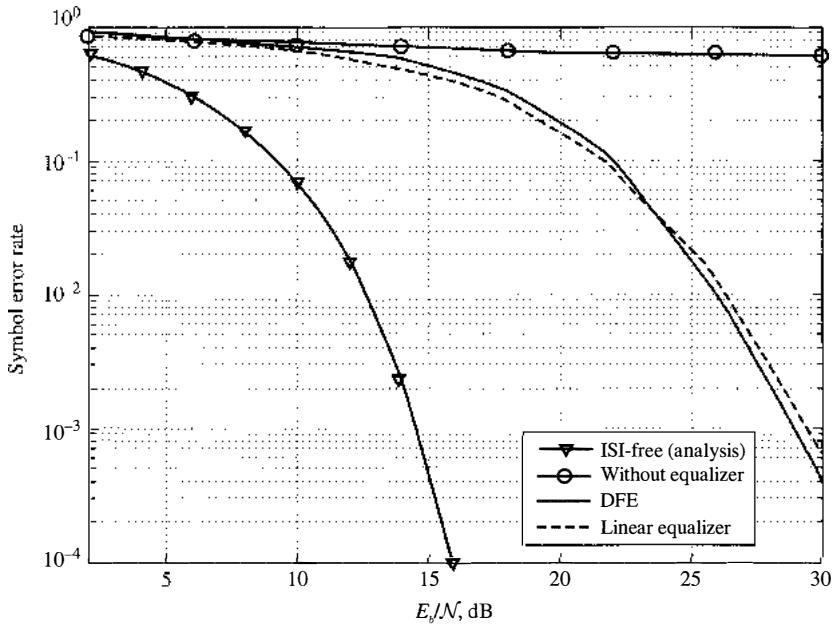


Figure 12.24 Symbol error rate (SER) comparison of DFE, linear equalization, and under ideal channel.

COMPUTER EXERCISE 12.3: OFDM TRANSMISSION OF QAM SIGNALS

In the example, we will utilize OFDM for QAM transmission. We choose the number of subcarriers (and the FFT size) as $N = 32$. We let the finite impulse response (FIR) channel to be

```
channel=[0.3 -0.5 0 1 .2 -0.3]
```

The channel length is 6 ($L = 5$ in Section 12.7). For this reason, we can select the cyclic prefix length to be the minimum length of $L = 5$.

```
% Matlab Program <Ex12_3.m>
% This Matlab exercise <Ex12_3.m> performs simulation of
% an OFDM system that employs QAM-16 baseband signaling
% a multipath channel with AWGN.
% Correct carrier and synchronization is assumed.
% 32 subcarriers are used with channel length of 6
% and cyclic prefix length of 5.
clear;clf;
L=1600000;           % Total data symbols in experiment is 1 million
Lfr=L/32;         % number of data frames
% Generating random signal data for polar signaling
s_data=4*round(rand(L,1))+2*round(rand(L,1))-3+...
    +j*(4*round(rand(L,1))+2*round(rand(L,1))-3);

channel=[0.3 -0.5 0 1 .2 -0.3];      % channel in t-domain
hf=fft(channel,32);                  % find the channel in f-domain

p_data=reshape(s_data,32,Lfr);       % S/P conversion

p_td=ifft(p_data);                   % IFFT to convert to t-domain
p_cyc=[p_td(end-4:end,:);p_td];      % add cyclic prefix
```

```
s_cyc=reshape(p_cyc,37*Lfr,1);      % P/S conversion

Psig=10/32;                         % average channel input power
chsout=filter(channel,1,s_cyc);     % generate channel output signal
clear p_td p_cyc s_data s_cyc;      % release some memory
noiseq=(randn(37*Lfr,1)+j*randn(37*Lfr,1));
SEReq=[];

for ii=1:31,
SNR(ii)=ii-1;                        % SNR in dB
Asig=sqrt(Psig*10^(-SNR(ii)/10))*norm(channel);
x_out=chsout+Asig*noiseq;           % Add noise
x_para=reshape(x_out,37,Lfr);        % S/P conversion
x_disc=x_para(6:37,:);               % discard tails
xhat_para=fft(x_disc);               % FFT back to f-domain

z_data=inv(diag(hf))*xhat_para;      % f-domain equalizing
% compute the QAM decision after equalization
deq=sign(real(z_data))+sign(real(z_data)-2)+sign(real(z_data)+2)+...
    j*(sign(imag(z_data))+sign(imag(z_data)-2)+sign(imag(z_data)+2));
% Now compare against the original data to compute SER
SEReq=[SEReq sum(p_data~=deq,2)/Lfr];
end

for ii=1:9,
    SNRa(ii)=2*ii-2;
    Q(ii)=3*0.5*erfc(sqrt((2*10^(SNRa(ii)*0.1)/5)/2));
%Compute the Analytical BER
end

% call another program to display OFDM Analysis
ofdmAz
```

The main MATLAB program `Ex12_3.m` completes OFDM modulation, equalization, and detection. Because the subcarriers (subchannels) have different gain and, consequently, different SNR, each of the 32 subcarriers may have a different SER. Thus, simply comparing the overall SER does not tell the full story. For this reason, we can call another program `ofdmAz.m` to analyze the results of this OFDM system.

```
% MATLAB PROGRAM <ofdmAz.m>
% This program is used to analyze the OFDM subcarriers and their
% receiver outputs.

% Plot the subcarrier gains
figure(2);
stem(abs(hf));
xlabel('Subcarrier label');
title('Subchannel gain');

% Plot the subchannel constellation scattering after OFDM
figure(3);
subplot(221);plot(z_data(1,1:800),'.')    % subchannel 1 output
ylabel('Imaginary');
title('(a) Subchannel 1 output');axis('square');
subplot(222);plot(z_data(10,1:800),'.');  % subchannel 10 output
```

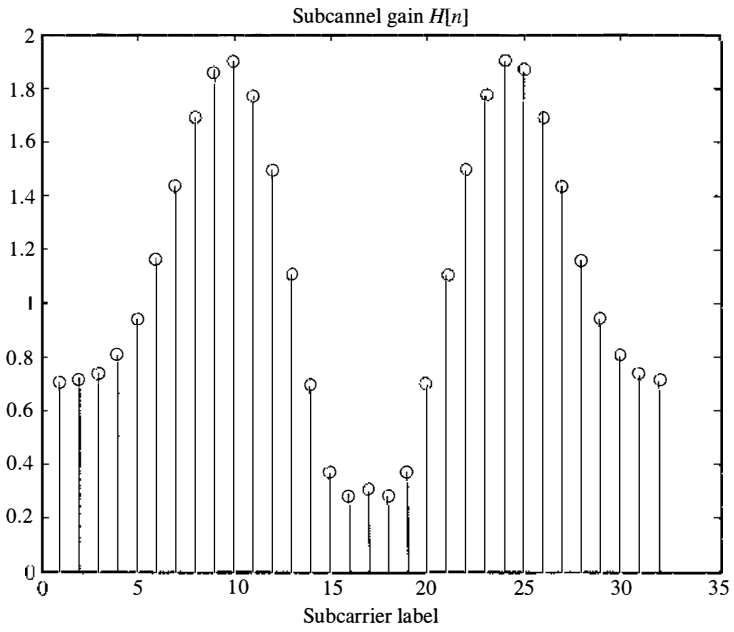


Figure 12.25 Comparison of the in channel gain for 32 subcarriers.

```
ylabel('Imaginary');
title('(b) Subchannel 10 output');axis('square');
subplot(223);plot(z_data(15,1:800),'.');  % subchannel 15 output
xlabel('Real');ylabel('Imaginary');
title('(c) Subchannel 15 output');axis('square');
subplot(224);plot(z_data(:,1:800),'b.');  % mixed subchannel output
xlabel('Real');ylabel('Imaginary');
title('(d) Mixed OFDM output');axis('square');

% Plot the average OFDM SER versus SER under "ideal channel"
% By Disabling 5 poor subcarriers, average SER can be reduced.
figure(4);
figc=semilogy(SNRa,Q,'k-',SNR,mean(SEReq),'b-o',...
    SNR,mean([SEReq(1:14,:);SEReq(20:32,:)]),'b-s');
set(figc,'LineWidth',2);
legend('Ideal channel','Using all subcarriers','Disabling 5 poor
subcarriers')
title('Average OFDM SER');
axis([1 30 1.e-4 1]);hold off;
xlabel('SNR (dB)');ylabel('Symbol Error Rate (SER)');
```

First, we display the subchannel gain $H[n]$ in Fig. 12.25. We can clearly see that, among the 32 subchannels, the 5 near the center have the lowest gains and hence the lowest SNR. We therefore expect them to exhibit the worst performance. By fixing the average channel SNR at 30 dB, we can take a quick peek at the equalizer outputs of the different subcarrier equalizers. In particular, we select subchannels 1, 10, and 15 because they represent the moderate, good, and poor channels, respectively. Scatter plots of the output samples (Fig. 12.26a–c) clearly demonstrate the quality contrast among them. If we do not make any distinction among subchannels, we can see from Fig. 12.26d that the overall OFDM performance is dominated mainly by the poor subchannels.

We can also look at the SER of all 32 individual subcarriers in Fig. 12.27. We see very clearly that the 5 worst channels are responsible for the 5 worst SER performances. Naturally if we average the SER

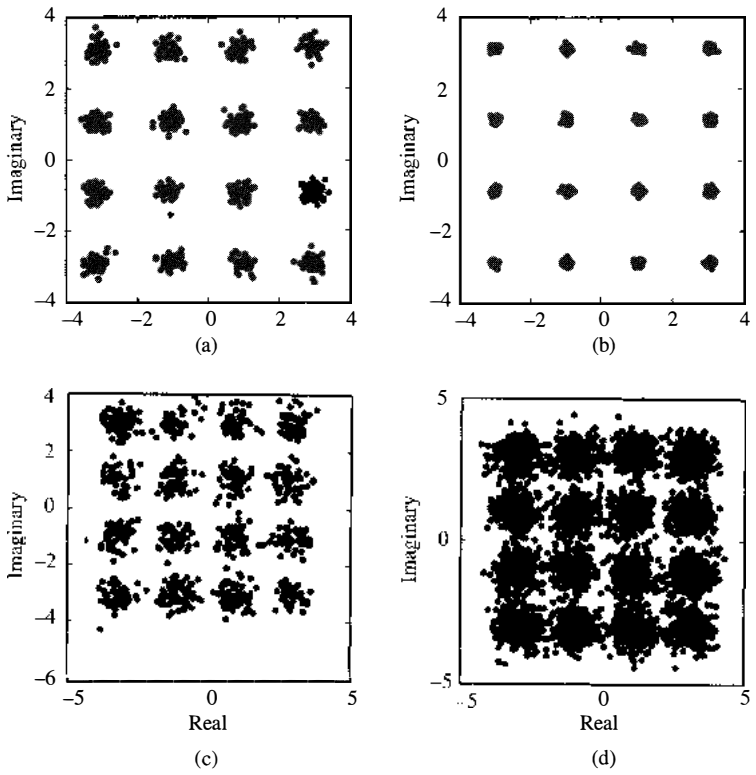


Figure 12.26 Different in channel quality as showen by scatter plots of the following OFDM channel outputs: (a) subchannel 1, (b) subchannel 10, (c) subchannel 15, and (d) mixed.

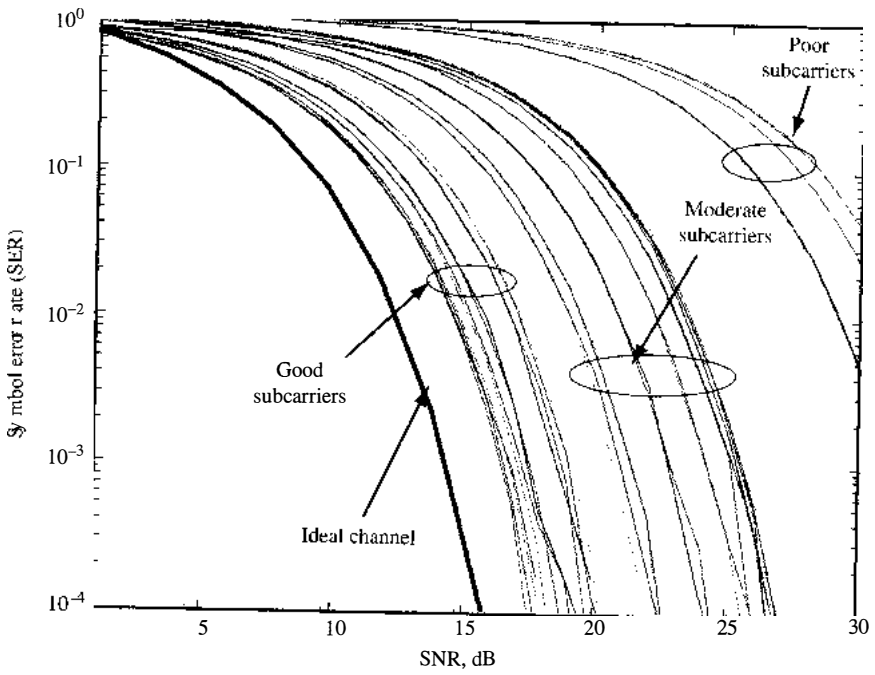


Figure 12.27 Symbol error rate (SER) of all 32 subcarriers over the multipath channel.

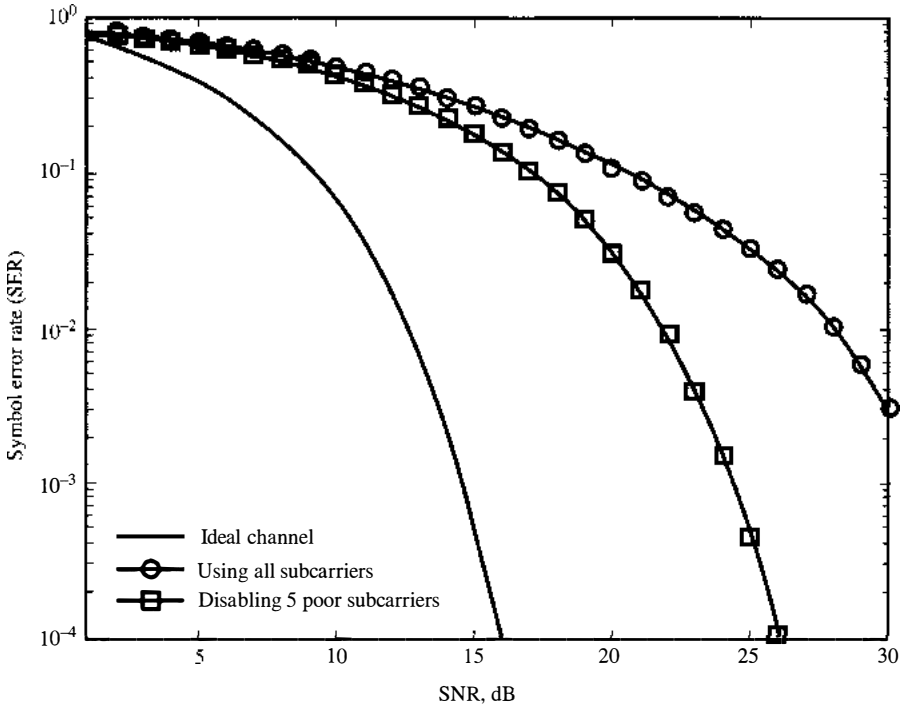


Figure 12.28 Average SER of the OFDM subcarriers before and after disabling five worst channels.

across all 32 subchannels, the larger SERs tend to dominate and make the overall SER of the OFDM system much higher.

To make the OFDM system more reliable, one possible approach is to apply bit loading. In fact, one extreme case of bit loading is to disable all the poor subchannels (i.e., to send nothing on the subchannels with very low gains). We can see from the SER comparison of Fig. 12.28 that by disabling 5 of the worst channels among the 32 subcarriers, the overall SER is significantly reduced (improved).

REFERENCES

1. G. D. Forney, Jr.,"Maximum Likelihood Sequence estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 363–378, May 1972.
2. Andrew J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. Inform. Theory*, vol. 13, no. 2, pp. 260–269, April 1967.
3. Richard Bellman, "Sequential Machines, Ambiguity, and Dynamic Programming," *J. ACM,* vol. 7 no. 1, pp. 24–28, Jan. 1960.
4. R. W. Lucky, "Automatic Equalization for Digital Communication," *Bell Syst. Tech. J.*, vol. 44, pp. 547–588, April 1965.
5. R. W. Lucky, "Techniques for Adaptive Equalization of Digital Communication Systems," *Bell Syst. Tech. J.*, vol. 45, pp. 255–286, Feb. 1966.
6. Z. Ding and Y. Li, *Blind Equalization and Identification*, CRC Press, New York, 2001.

7. R. D. Gitlin and S. B. Weinstein, "Fractionally-Spaced Equalization: An Improved Digital Transversal Equalizer," *Bell Syst. Tech. J.*, vol. 60, pp. 275-296, 1981.
8. T. Kailath, *Linear Systems*, Chapter 5, Prentice Hall, Englewood Cliffs, NJ, 1979.
9. N. Al-Dhahir and J. Cioffi, "MMSE Decision Feedback Equalizers and Coding: Finite-Length Results," *IEEE Trans. Inform. Theory*, vol. 41, no. 4, pp. 961–976, July 1995.
10. R. A. Kennedy and B. D. O. Anderson, "Tight Bounds on the Error Probabilities of Decision Feedback Equalizers," *IEEE Trans. Commun.*, vol. COM-35, pp. 1022–1029, Oct. 1987.
11. P. Chow, J. Cioffi, and J. Bingham, "A Practical Discrete Multitone Transceiver Loading Algorithm for Data Transmission over Spectrally Shaped Channels," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 773–775, Feb./Mar./Apr. 1995.
12. A. Leke and J. M. Cioffi, "A Maximum Rate Loading Algorithm for Discrete Multitone Modulation Systems," *Proceedings of IEEE Globecom*, pp. 1514–1518, Phoenix. AZ, 1997.
13. International Telecommunication Union, ITU-T Recommendation V.90, September 1998.
14. International Telecommunication Union, ITU-T Recommendation G.992.1, June 1999.
15. ETSI, "Digital Video Broadcasting: Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television," European Telecommunication Standard EN 300 744 V1.5, Nov. 2004.
16. D. P. Maxson, *The IBOC Handbook*, Elsevier Amsterdam, 2007.
17. A. Benveniste, M. Goursat, and G. Ruget, "Robust Identification of a Non-minimum Phase System: Blind Adjustment of a Linear Equalizer in Data Communications," *IEEE Trans. Autom. Control*, vol. AC-25, pp. 385–399, June 1980.
18. A. Benveniste and M. Goursat, "Blind Equalizers," *IEEE Trans. Commun.*, vol. 32, pp. 871–882, Aug. 1982.
19. D. N. Godard, "Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems," *IEEE Trans. Commun.*, vol. COM-28, pp. 1867–1875, Nov. 1980.
20. G. Picchi and G. Prati, "Blind Equalization and carrier Recovery Using a 'Stop-and-Go' Decision-Directed Algorithm," *IEEE Trans. Commun.*, vol. COM-35, pp. 877–887, Sept. 1987.
21. O. Shalvi and E. Weinstein, "New Criteria for Blind Deconvolution of Non-minimum Phase Systems (Channels)," *IEEE Trans. Inform., Theory*, vol. IT-36, pp. 312–321, March 1990.
22. O. Shalvi and E. Weinstein, "Super-exponential Methods for Blind Deconvolution," *IEEE Trans. Inform., Theory*, vol. IT-39, pp. 504–519, March 1993.
23. W. T. Rupprecht, "Adaptive Equalization of Binary NRZ-Signals by Means of Peak Value Minimization," In *Proc. 7th Eu. Conf. on Circuit Theory Design*, Prague, 1985, pp. 352–355.
24. R. A. Kennedy and Z. Ding, "Blind Adaptive Equalizers for QAM Communication Systems Based on Convex Cost Functions," *Opt. Eng.*, pp. 1189–1199, June 1992.
25. L. Tong, G. Xu, and T. Kailath, "Blind Channel Identification and Equalization Based on Second-Order Statistics: A Time-Domain Approach," *IEEE Trans. Inform., Theory*, vol. IT-40, pp. 340–349, March 1994.
26. E. Moulines, P. Duhamel, J.-F. Cardoso, and S. Mayrargue, "Subspace Methods for the Blind Identification of Multichannel FIR Filters," *IEEE Trans. Signal Process.*, vol. SP-43, pp. 516–525, Feb. 1995.
27. D. R. Brillinger and M. Rosenblatt, "Computation and Interpretation of kth Order Spectra," In *Spectral Analysis of Time Series*, B. Harris, E. Wiley, New York, 1967.
28. C.-Y. Chi, C.-C. Feng, C.-H. Chen, and C.-Y. Chen, *Blind Equalization and System Identification*, Springer, Berlin, 2006.
29. Simon Haykin, Ed. *Blind Deconvolution*, Prentice-Hall, Englewood Cliffts, NJ, 1994.

PROBLEMS

12.1-1 In a QAM transmission of symbol rate $1/T = 1$ MHz, assume that $p(t)$ is a raised-cosine pulse with roll-off factor of 0.5. The carrier frequency in use is 2.4 GHz.

(a) Derive the resulting baseband pulse part $q(t)$ when the multipath channel impulse response is given by

$$0.95\delta(t) - 0.3\delta(t - T/2)$$

(b) Show whether the eye is open for QPSK transmission in part **(a)** when the channel outputs are sampled at $t = kT$.

12.2-1 Consider the signal transmission model of Prob. 12.1-1.

(a) Determine the matched filter for the equivalent baseband pulse resulting from the multipath channel.

(b) Determine the equivalent discrete time linear system transfer function $H(z)$ between the QAM input symbols and the matched filter output sampled at $t = kT$.

12.2-2 In a digital QAM system, the received baseband pulse shape is $q(t) = \Delta\left(\frac{t}{2T}\right)$. The channel noise (before the matched filter) is AWGN with spectrum of $\mathcal{N}/2$.

(a) Find the power spectral density of the noise $\mathrm{w}(t)$ at the matched filter output.

(b) Determine the mean and the variance of the sampled noise $\mathrm{w}[kT]$ at the matched filter output.

(c) Show whether the noise samples $\mathrm{w}[kT]$ are independent.

12.3-1 In a BPSK baseband system, the discrete time channel is specified by

$$H(z) = 1 + 0.6z^{-1}$$

The received signal samples are

$$z[k] = H(z)s_k + \mathrm{w}[k]$$

The BPSK signal is $s_k = \pm 1$ with equal probability. The discrete channel noise $\mathrm{w}[k]$ is additive white Gaussian with zero mean and variance $\mathcal{N}/2$ such that the $E_b/\mathcal{N} = 18$.

(a) Find the probability of error if $z[k]$ is directly sent into a BPSK decision device.

(b) Find the probability of error if $z[k]$ first passes through a zero-forcing equalizer before a BPSK decision device.

12.3-2 Repeat Prob. 12.3-2 if the discrete channel

$$H(z) = 1 + 0.9z^{-1}$$

12.3-3 Compare the BER results of Probs. 12.3-1 and 12.3-2. Observe the different depth of the channel spectral nulls and explain their BER difference based on the different the noise amplification effect.

12.3-4 For the channel of Prob. 12.3-1, find the response of a six-tap MMSE equalizer. Determine the resulting minimum MSE. What is the corresponding MSE if the ZF equalizer is applied instead?

12.3-5 Repeat Prob. 12.3-3 for the FIR channel of Prob. 12.3-2.

12.4-1 In a fractionally sampled channel, the sampling frequency is chosen to be $2/T$ (i.e., there are two samples for every transmitted symbol s_k). The two sampled subchannel responses are

$$H_1(z) = 1 + 0.9z^{-1} \qquad H_2(z) = -0.3 + 0.5z^{-1}$$

Both subchannels have additive white Gaussian noises that are independent with zero mean and identical variance $\sigma_w^2 = 0.2$. The input symbol s_k is a PAM-4 with equal probability of being $(\pm 1, \pm 3)$.

(a) Show that $F_1(z) = 0.3$ and $F_2(z) = 1$ form a zero-forcing equalizer.

(b) Show that $F_1(z) = 1$ and $F_2(z) = -1.8$ also form a zero-forcing equalizer.

(c) Show which of the two previous fractionally spaced ZF equalizers delivers better performance. This result shows that ZF equalizers of different delays can lead to different performance.

12.4-2 For the same system of Prob. 12.4-1, complete the following.

(a) Find the ZF equalizers of delays 0, 1, and 2, respectively when the ZF equalizer filters have order 1, that is,

$$F_i(z) = f_i[0] + f_i[1]z^{-1} \qquad i = 1, 2$$

(b) Find the resulting noise distribution at the equalizer output for each of the three fractionally spaced ZF equalizers.

(c) Determine the probability of symbol error if hard PAM decision is taken from the equalizer output.

12.6-1 In a DFE for binary polar signaling, $s_k = \pm 1$ with equal probability. The feedforward filter output $d[k]$ is given by

$$d[k] = s_{k-2} + 0.8s_{k-3} + \mathrm{w}[k]$$

where $\mathrm{w}[k]$ is white Gaussian with zero mean and variance 0.04.

(a) Determine the DFE filter coefficient.

(b) Find the DFE output BER when the decisions in feedback are error free.

(c) If the decision device is not error free, then there will be error propagation. Find the probability of error of the next decision on symbol s_{k-2} when the previous decision s_{k-3} is known to be wrong.

12.7-1 Prove that $\boldsymbol{W}_N \cdot \boldsymbol{W}_N^{-1} = I_{N\times N}$.

12.7-2 A cyclic matrix is a matrix that is completely specified by its first row (or column). Row i is a circular shift of the elements in row $i-1$. In other words, if the first row of matrix C is $a_1, \ldots, a_{N-1}, a_N$, then its second row is $a_N, a_1, \ldots, a_{N-1}$, and so on. Prove that any cyclic matrix of size $N \times N$ can be diagonalized by $\boldsymbol{W}_N$ and $\boldsymbol{W}_N^{-1}$, that is,

$$\boldsymbol{W}_N \cdot C \cdot \boldsymbol{W}_N^{-1} = \text{diagonal}$$

12.7-3 Consider an FIR channel with impulse response

$$h[0] = 1.0, \qquad h[1] = -0.5, \qquad h[2] = 0.3$$

The channel noise is additive white Gaussian with spectrum $\mathcal{N}/2$. Design an OFDM system with $N = 16$ by **(a)** specifying the length of the cyclic prefix; **(b)** determining the N subchannel gains; **(c)** deriving the bit error rate of each subchannels for BPSK modulations; **(d)** finding the average bit error rate of the entire OFDM system.

12.7-4 Consider an FIR channel of order up to L. First, we apply the usual IDFT on the source data vector via

$$\boldsymbol{s} = \left(\frac{1}{\sqrt{N}}\boldsymbol{W}_N\right)^{-1}\tilde{\boldsymbol{s}}$$

Next, instead of applying a cyclic prefix as in Eq. (12.64b), we insert a string of L zeros in front of every N data before transmission as in

$$\left.\begin{bmatrix} s_N \\ s_{N-1} \\ \vdots \\ s_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right\} (N+L)\times 1$$

This zero-padded data vector is transmitted normally over the FIR channel $\{h[k]\}$. At the receiver end, we stack up the received symbols $\{z[n]\}$ into

$$\boldsymbol{y} = \begin{bmatrix} z[N] \\ z[N-1] \\ \vdots \\ z[L] \\ \vdots \\ z[1] \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ z[N+L] \\ \vdots \\ z[N+1] \end{bmatrix}$$

Prove that

$$\tilde{\boldsymbol{z}} = \left(\frac{1}{\sqrt{N}}\boldsymbol{W}_N\right)\boldsymbol{y} = \begin{bmatrix} H[N] & & & \\ & H[N-1] & & \\ & & \ddots & \\ & & & H[+1] \end{bmatrix}\tilde{\boldsymbol{s}}$$

This illustrates the equivalence between zero padding and cyclic prefix in OFDM.

12.7-5 Show that in AWGN channels, cyclic OFDM and zero-padded OFDM achieve identical SNRs for the same channel input power.

13 INTRODUCTION TO INFORMATION THEORY

Among all the means of communication discussed thus far, none produces error-free communication. We may be able to improve the accuracy in digital signals by reducing the error probability P_e. But it appears that as long as channel noise exists, our communications cannot be free of errors. For example, in all the digital systems discussed thus far, P_e varies as e^{-kE_b} asymptotically. By increasing E_b, the energy per bit, we can reduce P_e to any desired level. Now, the signal power is $S_i = E_b R_b$, where R_b is the bit rate. Hence, increasing E_b means either increasing the signal power S_i (for a given bit rate), decreasing the bit rate R_b (for a given power), or both. Because of physical limitations, however, S_i cannot be increased beyond a certain limit. Hence, to reduce P_e further, we must reduce R_b, the rate of transmission of information digits. Thus, the price to be paid for reducing P_e is a reduction in the transmission rate. To make P_e approach 0, R_b also approaches 0. Hence, it appears that in the presence of channel noise it is impossible to achieve error-free communication. Thus thought communication engineers until the publication of Shannon's seminal paper[1] in 1948. Shannon showed that for a given channel, as long as the rate of information digits per second to be transmitted is maintained within a certain limit determined by the physical channel (known as the channel capacity), it is possible to achieve error-free communication. That is, to attain $P_e \to 0$, it is not necessary to make $R_b \to 0$. Such a goal ($P_e \to 0$) can be attained by maintaining R_b below C, the channel capacity (per second). The gist of Shannon's paper is that the presence of random disturbance in a channel does not, by itself, define any limit on transmission accuracy. Instead, it defines a limit on the information rate for which an arbitrarily small error probability ($P_e \to 0$) can be achieved.

We have been using the phrase "rate of information transmission" as if information could be measured. This is indeed so. We shall now discuss the information content of a message as understood by our "common sense" and also as it is understood in the "engineering sense." Surprisingly, both approaches yield the same measure of information in a message.

13.1 MEASURE OF INFORMATION

Commonsense Measure of Information

Consider the following three hypothetical headlines in a morning paper:

1. There will be daylight tomorrow.
2. United States invades Iran.
3. Iran invades the United States.

The reader will hardly notice the first headline unless he or she lives near the North or the South Pole. The reader will be very, very interested in the second. But what really catches the reader's attention is the third headline. This item will attract much more interest than the other two headlines. From the viewpoint of "common sense," the first headline conveys hardly any information; the second conveys a large amount of information; and the third conveys yet a larger amount of information. If we look at the probabilities of occurrence of these three events, we find that the probability of occurrence of the first event is unity (a certain event), that of the second is low (an event of small but finite probability), and that of the third is practically zero (an almost impossible event). If an event of low probability occurs, it causes greater surprise and, hence, conveys more information than the occurrence of an event of larger probability. Thus, the information is connected with the element of surprise, which is a result of uncertainty, or unexpectedness. The more unexpected the event, the greater the surprise, and hence the more information. The probability of occurrence of an event is a measure of its unexpectedness and, hence, is related to the information content. Thus, from the point of view of common sense, the amount of information received from a message is directly related to the uncertainty or inversely related to the probability of its occurrence. If P is the probability of occurrence of a message and I is the information gained from the message, it is evident from the preceding discussion that when $P \to 1, I \to 0$ and when $P \to 0, I \to \infty$, and, in general a smaller P gives a larger I. This suggests the following information measure:

$$I \sim \log \frac{1}{P} \tag{13.1}$$

Engineering Measure of Information

We now show that from an engineering point of view, the information content of a message is consistent with the intuitive measure [Eq. (13.1)]. What do we mean by an engineering point of view? An engineer is responsible for the efficient transmission of messages. For this service the engineer will charge a customer an amount proportional to the information to be transmitted. But in reality the engineer will charge the customer in proportion to the time that the message occupies the channel bandwidth for transmission. In short, from an engineering point of view, the amount of information in a message is proportional to the (minimum) time required to transmit the message. We shall now show that this concept of information also leads to Eq. (13.1). This implies that a message with higher probability can be transmitted in a shorter time than that required for a message with lower probability. This fact may be verified by the example of the transmission of alphabetic symbols in the English language using Morse code. This code is made up of various combinations of two symbols (such as a dash and a dot in Morse code, or pulses of height A and $-A$ volts). Each letter is represented by a certain combination of these symbols, called the **codeword**, which has a certain length. Obviously, for efficient transmission, shorter codewords are assigned to the letters e, t, a, and o, which occur more frequently. The longer codewords are assigned to letters x, k, q, and z, which occur less frequently. Each letter may be considered to be a message. It is obvious that the letters that occur more frequently (with higher probability of occurrence) need a shorter time to transmit (shorter codewords) than those with smaller probability of occurrence. We shall now show that on the average, the time required to transmit a symbol (or a message) with probability of occurrence P is indeed proportional to $\log (1/P)$.

For the sake of simplicity, let us begin with the case of binary messages m_1 and m_2, which are equally likely to occur. We may use binary digits to encode these messages, representing m_1 and m_2 by the digits **0** and **1**, respectively. Clearly, we must have a minimum of one binary digit (which can assume two values) to represent each of the two equally likely messages. Next, consider the case of the four equiprobable messages m_1, m_2, m_3, and m_4. If these messages are

encoded in binary form, we need a minimum of two binary digits per message. Each binary digit can assume two values. Hence, a combination of two binary digits can form the four codewords **00**, **01**, **10**, **11**, which can be assigned to the four equiprobable messages m_1, m_2, m_3, and m_4, respectively. It is clear that each of these four messages takes twice as much transmission time as that required by each of the two equiprobable messages and, hence, contains twice as much information. Similarly, we can encode any one of eight equiprobable messages with a minimum of three binary digits. This is because three binary digits form eight distinct codewords, which can be assigned to each of the eight messages. It can be seen that, in general, we need $\log_2 n$ binary digits to encode each of n equiprobable messages.* Because all the messages are equiprobable, P, the probability of any one message occurring, is $1/n$. Hence, to encode each message (with probability P), we need $\log_2(1/P)$ binary digits. Thus, from the engineering viewpoint, the information I contained in a message with probability of occurrence P is proportional to $\log_2 (1/P)$,

$$I = k \log_2 \frac{1}{P} \tag{13.2}$$

where k is a constant to be determined. Once again, we come to the conclusion (from the engineering viewpoint) that the information content of a message is proportional to the logarithm of the reciprocal of the probability of the message.

We shall now define the information conveyed by a message according to Eq. (13.2). The proportionality constant is taken as unity for convenience, and the information is then in terms of binary units, abbreviated **bit** (**bi**nary uni**t**),

$$I = \log_2 \frac{1}{P} \quad \text{bits} \tag{13.3}$$

According to this definition, the information I in a message can be interpreted as the minimum number of binary digits required to encode the message. This is given by $\log_2 (1/P)$, where P is the probability of occurrence of the message. Although here we have shown this result for the special case of equiprobable messages, we shall show in the next section that it is true for nonequiprobable messages also.

Next, we shall consider the case of r-ary digits instead of binary digits for encoding. Each of the r-ary digits can assume r values (**0**, **1**, **2**, . . . , $\boldsymbol{r-1}$). Each of n messages (encoded by r-ary digits) can then be transmitted by a particular sequence of r-ary signals. Because each r-ary digit can assume r values, k r-ary digits can form a maximum of r^k distinct codewords. Hence, to encode each of the n equiprobable messages, we need a minimum of $k = \log_r n$ r-ary digits.† But $n = 1/P$, where P is the probability of occurrence of each message. Obviously, we need a minimum of $\log_r(1/P)$ r-ary digits. The information I per message is therefore

$$I = \log_r \frac{1}{P} \quad r\text{-ary units} \tag{13.4}$$

From Eqs. (13.3) and (13.4) it is evident that

$$I = \log_2 \frac{1}{P} \quad \text{bits} = \log_r \frac{1}{P} \quad r\text{-ary units}$$

---

* Here we are assuming that the number n is such that $\log_2 n$ is an integer. Later on we shall observe that this restriction is not necessary.

† Here again we are assuming that n is such that $\log_r n$ is an integer. As we shall see later, this restriction is not necessary.

Hence,*

$$1\ r\text{-ary unit} = \log_2 r \text{ bits} \tag{13.5}$$

A Note on the Unit of Information: Although it is tempting to use the r-ary unit as a general unit of information, the binary unit bit ($r = 2$) is commonly used in the literature. There is, of course, no loss of generality in using $r = 2$. These units can always be converted into any other units by using Eq. (13.5). Henceforth, unless otherwise stated, we shall use the binary unit (bit) for information. The bases of the logarithmic functions will be omitted, but will be understood to be 2.

Average Information per Message: Entropy of a Source

Consider a memoryless source m emitting messages $m_1, m_2, \ldots, m_n$ with probabilities $P_1, P_2, \ldots, P_n$, respectively ($P_1 + P_2 + \cdots + P_n = 1$). A **memoryless source** implies that each message emitted is independent of the previous message(s). By the definition in Eq. (13.3) [or Eq. (13.4)], the information content of message m_i is I_i, given by

$$I_i = \log \frac{1}{P_i} \quad \text{bits} \tag{13.6}$$

The probability of occurrence of m_i is P_i. Hence, the mean, or average, information per message emitted by the source is given by $\sum_{i=1}^{n} P_i I_i$ bits. The average information per message of a source m is called its **entropy**, denoted by $H(\text{m})$. Hence,

$$H(\text{m}) = \sum_{i=1}^{n} P_i I_i \quad \text{bits}$$

$$= \sum_{i=1}^{n} P_i \log \frac{1}{P_i} \quad \text{bits} \tag{13.7a}$$

$$= -\sum_{i=1}^{n} P_i \log P_i \quad \text{bits} \tag{13.7b}$$

The entropy of a source is a function of the message probabilities. It is interesting to find the message probability distribution that yields the maximum entropy. Because the entropy is a measure of uncertainty, the probability distribution that generates the maximum uncertainty will have the maximum entropy. On qualitative grounds, one expects entropy to be maximum when all the messages are equiprobable. We shall now show that this is indeed true.

---

* In general,

$$1\ r\text{-ary unit} = \log_s r \quad s\text{-ary units}$$

The 10-ary unit of information is called the **hartley** in honor of R. V. L. Hartley,[2] who was one of the pioneers (along with Nyquist[3] and Carson) in the area of information transmission in the 1920s. The rigorous mathematical foundations of information theory, however, were established by C. E. Shannon[1] in 1948:

$$1 \text{ hartley} = \log_2 10 = 3.32 \quad \text{bits}$$

Sometimes the unit **nat** is used:

$$1 \text{ nat} = \log_2 e = 1.44 \quad \text{bits}$$

Because $H(\mathrm{m})$ is a function of $P_1, P_2, \ldots, P_n$, the maximum value of $H(\mathrm{m})$ is found from the equation $dH(\mathrm{m})/dP_i = 0$ for $i = 1, 2, \ldots, n$, with the constraint that

$$1 = P_1 + P_2 + \cdots + P_{n-1} + P_n \tag{13.8}$$

Because the function for maximization is

$$H(\mathrm{m}) = -\sum_{i=1}^{n} P_i \log P_i \tag{13.9}$$

we need to use the Lagrangian to form a new function

$$f(P_1, P_2, \ldots, P_n) = -\sum_{i=1}^{n} P_i \log P_i + \lambda(P_1 + P_2 + \cdots + P_{n-1} + P_n - 1)$$

Hence,

$$\begin{aligned} \frac{df}{dP_j} &= -P_j\left(\frac{1}{P_j}\right)\log e - \log P_j + \lambda \\ &= -\log P_j + \lambda - \log e \qquad j = 1, 2, \ldots, n \end{aligned}$$

Setting the derivatives to zero leads to

$$P_1 = P_2 = \cdots = P_n = \frac{2^{\lambda}}{e}$$

By invoking the probability constraint of Eq. (13.8), we have

$$n\frac{2^{\lambda}}{e} = 1$$

Thus,

$$P_1 = P_2 = \cdots = P_n = \frac{1}{n} \tag{13.10}$$

To show that Eq. (13.10) yields $[H(\mathrm{m})]_{\max}$ and not $[H(\mathrm{m})]_{\min}$, we note that when $P_1 = 1$ and $P_2 = P_3 = \cdots = P_n = 0$, $H(\mathrm{m}) = 0$, whereas the probabilities in Eq. (13.10) yield

$$H(\mathrm{m}) = -\sum_{i=1}^{n} \frac{1}{n} \log \frac{1}{n} = \log n$$

The Intuitive (Commonsense) and the Engineering Interpretations of Entropy: Earlier we observed that both the intuitive and the engineering viewpoints lead to the same definition of the information associated with a message. The conceptual bases, however, are entirely different for the two points of view. Consequently, we have two physical interpretations of information. According to the engineering point of view, the information content of any message is equal to the minimum number of digits required to encode the message, and, therefore, the entropy $H(\mathrm{m})$ is equal to the minimum number of digits per message required,

on the average, for encoding. From the intuitive standpoint, on the other hand, information is thought of as being synonymous with the amount of surprise, or uncertainty, associated with the event (or message). A smaller probability of occurrence implies more uncertainty about the event. Uncertainty is, of course, associated with surprise. Hence intuitively, the information associated with a message is a measure of the uncertainty (unexpectedness) of the message. Therefore, $\log (1/P_i)$ is a measure of the uncertainty of the message m_i, and $\sum_{i=1}^{n} P_i \log (1/P_i)$ is the average uncertainty (per message) of the source that generates messages $m_1, m_2, \ldots, m_n$ with probabilities $P_1, P_2, \ldots, P_n$. Both these interpretations prove useful in the qualitative understanding of the mathematical definitions and results in information theory. Entropy may also be viewed as a function associated with a random variable m that assumes values $m_1, m_2, \ldots, m_n$ with probabilities $P(m_1), P(m_2), \ldots, P(m_n)$:

$$H(\mathrm{m}) = \sum_{i=1}^{n} P(m_i) \log \frac{1}{P(m_i)} = \sum_{i=1}^{n} P_i \log \frac{1}{P_i}$$

Thus, we can associate an entropy with every discrete random variable.

If the source is not memoryless (i.e., in the event that a message emitted at any time is dependent of the previous messages emitted), then the source entropy will be less than $H(\mathrm{m})$ in Eq. (13.9). This is because the dependence of a message on previous messages reduces its uncertainty.

13.2 SOURCE ENCODING

The minimum number of binary digits required to encode a message was shown to be equal to the source entropy $\log(1/P)$ if all the messages of the source are equiprobable (each message probability is P). We shall now generalize this result to the case of nonequiprobable messages. We shall show that the average number of binary digits per message required for encoding is given by $H(\mathrm{m})$ (in bits) for an arbitrary probability distribution of the messages.

Let a source m emit messages $m_1, m_2, \ldots, m_n$ with probabilities $P_1, P_2, \ldots, P_n$, respectively. Consider a sequence of N messages with $N \to \infty$. Let k_i be the number of times message m_i occurs in this sequence. Then according to the relative frequency interpretation (or law of large numbers),

$$\lim_{N \to \infty} \frac{k_i}{N} = P_i$$

Thus, the message m_i occurs NP_i times in a sequence of N messages (provided $N \to \infty$). Therefore, in a typical sequence of N messages, m_1 will occur NP_1 times, m_2 will occur NP_2 times, $\ldots, m_n$ will occur NP_n times. All other compositions are extremely unlikely to occur ($P \to 0$). Thus, any typical sequence (where $N \to \infty$) has the same proportion of the n messages, although in general the order will be different. We shall assume a memoryless source; that is, we assume that the message is emitted from the source independently of the previous messages. Consider now a typical sequence S_N of N messages from the source. Because the n messages (of probability $P_1, P_2, \ldots, P_n$) occur $NP_1, NP_2, \ldots, NP_n$ times, and because each message is independent, the probability of occurrence of a typical sequence S_N is given by

$$P(S_N) = (P_1)^{NP_1} (P_2)^{NP_2} \cdots (P_n)^{NP_n} \tag{13.11}$$

Because all possible sequences of N messages from this source have the same composition, all the sequences (of N messages) are equiprobable, with probability $P(S_N)$. We can consider these long sequences as new messages (which are now equiprobable). To encode one such sequence we need L_N binary digits, where

$$L_N = \log \left[\frac{1}{P(S_N)} \right] \quad \text{binary digits} \tag{13.12}$$

Substituting Eq. (13.11) into Eq. (13.12), we obtain

$$L_N = N \sum_{i=1}^{n} P_i \log \frac{1}{P_i} = NH(\text{m}) \quad \text{binary digits}$$

Note that L_N is the length (number of binary digits) of the codeword required to encode N messages in sequence. Hence, L, the average number of digits required per message, is L_N/N and is given by

$$L = \frac{L_N}{N} = H(\text{m}) \quad \text{binary digits} \tag{13.13}$$

Thus, by encoding N successive messages, it is possible to encode a sequence of source messages using, on the average, $H(\text{m})$ binary digits per message, where $H(\text{m})$ is the entropy of the source message (in bits). Moreover, one can show that $H(\text{m})$ is indeed, on the average, the minimum number of digits required to encode this message source. It is impossible to find any uniquely decodable code whose average length is less than $H(\text{m})$.[4, 5]

Huffman Code

The source encoding theorem says that to encode a source with entropy $H(\text{m})$, we need, on the average, a minimum of $H(\text{m})$ binary digits per message. The number of digits in the codeword is the **length** of the codeword. Thus, the average word length of an optimum code is $H(\text{m})$. Unfortunately, to attain this length, in general, we have to encode a sequence of N messages ($N \to \infty$) at a time. If we wish to encode each message directly without using longer sequences, then, in general, the average length of the codeword per message will be greater than $H(\text{m})$. In practice, it is not desirable to use long sequences, since they cause transmission delay and add to equipment complexity. Hence, it is preferable to encode messages directly, even if the price has to be paid in terms of increased word length. In most cases, the price turns out to be small. The following is a procedure, given without proof, for finding the optimum source code, called the Huffman code. The proof that this code is optimum can be found elsewhere.[4–6]

We shall illustrate the procedure with an example using a binary code. We first arrange the messages in the order of descending probability, as shown in Table 13.1. Here we have six messages with probabilities 0.30, 0.25, 0.15, 0.12, 0.08, and 0.10, respectively. We now aggregate the last two messages into one message with probability $P_5 + P_6 = 0.18$. This leaves five messages with probabilities, 0.30, 0.25, 0.18, 0.15, and 0.12. These messages are now rearranged in the second column in the order of descending probability. We repeat this procedure by aggregating the last two messages in the second column and rearranging them in the order of descending probability. This is done until the number of messages is reduced to two. These two (reduced) messages are now assigned **0** and **1** as their first digits in the code

TABLE 13.1

Original Source		Reduced Sources			
Messages	**Probabilities**	S_1	S_2	S_3	S_4
m_1	0.30	0.30	0.30	0.43	0.57
m_2	0.25	0.25	0.27	0.30	0.43
m_3	0.15	0.18	0.25	0.27	
m_4	0.12	0.15	0.18		
m_5	0.08	0.12			
m_6	0.10				

TABLE 13.2

Original Source			Reduced Sources			
Messages	**Probabilities**	**Code**	S_1	S_2	S_3	S_4
m_1	0.30	**00**	0.30 **00**	0.30 **00**	0.43 **1**	0.57 **0**
m_2	0.25	**10**	0.25 **10**	0.27 **01**	0.30 **00**	0.43 **1**
m_3	0.15	**010**	0.18 **11**	0.25 **10**	0.27 **01**	
m_4	0.12	**011**	0.15 **010**	0.18 **11**		
m_5	0.08	**110**	0.12 **011**			
m_6	0.10	**111**				

sequence. We now go back and assign the numbers **0** and **1** to the second digit for the two messages that were aggregated in the previous step. We keep regressing in this way until the first column is reached. The code finally obtained (for the first column) can be shown to be optimum. The complete procedure is shown in Tables 13.1 and 13.2.

The optimum (Huffman) code obtained this way is also called a **compact code**. The average length of the compact code in the present case is given by

$$L = \sum_{i=1}^{n} P_i L_i = 0.3(2) + 0.25(2) + 0.15(3) + 0.12(3) + 0.1(3) + 0.08(3)$$

$$= 2.45 \text{ binary digits}$$

The entropy $H(\text{m})$ of the source is given by

$$H(\text{m}) = \sum_{i=1}^{n} P_i \log_2 \frac{1}{P_i}$$

$$= 2.418 \text{ bits}$$

Hence, the minimum possible length (attained by an infinitely long sequence of messages) is 2.418 binary digits. By using direct coding (the Huffman code), it is possible to attain an average length of 2.45 bits in the example given. This is a close approximation of the optimum performance attainable. Thus, little is gained by complex coding of a number of messages in this case.

The merit of any code is measured by its average length in comparison to $H(\mathrm{m})$ (the average minimum length). We define the **code efficiency** η as

$$\eta = \frac{H(\mathrm{m})}{L}$$

where L is the average length of the code. In our present example,

$$\eta = \frac{2.418}{2.45}$$
$$= 0.976$$

The **redundancy** γ is defined as

$$\gamma = 1 - \eta$$
$$= 0.024$$

Even though the Huffman code is a variable length code, it is uniquely decodable. If we receive a sequence of Huffman-coded messages, it can be decoded only one way, that is, without ambiguity. For instance, if the source in this exercise were to emit the message sequence $m_1 m_5 m_2 m_1 m_4 m_3 m_6 \ldots$, it would be encoded as **001101000011010111**.... The reader may verify that this message sequence can be decoded only one way, viz, $m_1 m_5 m_2 m_1 m_4 m_3 m_6 \ldots$, even if there is no demarcation between individual messages. This uniqueness is assured by the special property that no codeword is a prefix of another (longer) codeword.

A similar procedure is used to find a compact r-ary code. In this case we arrange the messages in descending order of probability, combine the last r messages into one message, and rearrange the new set (reduced set) in the order of descending probability. We repeat the procedure until the final set reduces to r messages. Each of these messages is now assigned one of the r numbers **0, 1, 2,..., r − 1**. We now regress in exactly the same way as in the binary case until each of the original messages has been assigned a code.

For an r-ary code, we will have exactly r messages left in the last reduced set if, and only if, the total number of original messages is $r + k(r - 1)$, where k is an integer. This is because each reduction decreases the number of messages by $r - 1$. Hence, if there is a total of k reductions, the total number of original messages must be $r + k(r - 1)$. In case the original messages do not satisfy this condition, we must add some dummy messages with zero probability of occurrence until this condition is fulfilled. For example, if $r = 4$ and the number of messages n is 6, then we must add one dummy message with zero probability of occurrence to make the total number of messages 7, that is, $[4 + 1(4 - 1)]$, and proceed as usual. The procedure is illustrated in Example 13.1.

Example 13.1 A memoryless source emits six messages with probabilities 0.3, 0.25, 0.15, 0.12, 0.1, and 0.08. Find the 4-ary (quaternary) Huffman code. Determine its average word length, the efficiency, and the redundancy.



In this case, we need to add one dummy message to satisfy the required condition of $r + k(r - 1)$ messages and proceed as usual. The Huffman code is found in Table 13.3. The length L of this code is

$$L = 0.3(1) + 0.25(1) + 0.15(1) + 0.12(2) + 0.1(2) + 0.08(2) + 0(2)$$
$$= 1.3 \quad \text{4-ary digits}$$

TABLE 13.3

Original Source			Reduced Sources	
Messages	**Probabilities**	**Code**		
m_1	0.30	**0**	0.30	**0**
m_2	0.25	**2**	0.30	**1**
m_3	0.15	**3**	0.25	**2**
m_4	0.12	**10**	0.15	**3**
m_5	0.10	**11**		
m_6	0.08	**12**		
m_7	0.00	**13**		

Also,

$$H_4(\mathrm{m}) = -\sum_{i=1}^{6} P_i \log_4 P_i$$
$$= 1.209 \quad \text{4-ary units}$$

The code efficiency η is given by

$$\eta = \frac{1.209}{1.3} = 0.93$$

The redundancy $\gamma = 1 - \eta = 0.07$.

To achieve code efficiency $\eta \to 1$, we need $N \to \infty$. The Huffman code uses $N = 1$, but its efficiency is, in general, less than 1. A compromise exists between these two extremes of $N = 1$ and $N = \infty$. We can encode a group of $N = 2$ or 3 messages. In most cases, the use of $N = 2$ or 3 can yield an efficiency close to 1, as the following example shows.

Example 13.2 A memoryless source emits messages m_1 and m_2 with probabilities 0.8 and 0.2, respectively. Find the optimum (Huffman) binary code for this source as well as for its **second-** and **third-order extensions** (i.e., for $N = 2$ and 3). Determine the code efficiencies in each case.

The Huffman code for the source is simply **0** and **1**, giving $L = 1$, and

$$H(\mathrm{m}) = -(0.8 \log 0.8 + 0.2 \log 0.2)$$
$$= 0.72 \quad \text{bit}$$

Hence,

$$\eta = 0.72$$

For the second-order extension of the source ($N = 2$), there are four possible composite messages, m_1m_1, m_1m_2, m_2m_1, and m_2m_2, with probabilities 0.64, 0.16, 0.16, and 0.04, respectively. The Huffman code is obtained in Table 13.4.

TABLE 13.4

Original Source							
Messages	**Probabilities**	**Code**		**Reduced Source**			
m_1m_1	0.64	**0**	0.64	**0**	0.64	**0**	
m_1m_2	0.16	**11**	0.20	**10**	0.36	**1**	
m_2m_1	0.16	**100**	0.16	**11**			
m_2m_2	0.04	**101**					

TABLE 13.5

Messages	Probabilities	Code
$m_1m_1m_1$	0.512	0
$m_1m_1m_2$	0.128	100
$m_1m_2m_1$	0.128	101
$m_2m_1m_1$	0.128	110
$m_1m_2m_2$	0.032	11100
$m_2m_1m_2$	0.032	11101
$m_2m_2m_1$	0.032	11110
$m_2m_2m_2$	0.008	11111

In this case the average word length L' is

$$\begin{aligned} L' &= 0.64(1) + 0.16(2) + 0.16(3) + 0.04(3) \\ &= 1.56 \end{aligned}$$

This is the word length for two messages of the original source. Hence L, the word length per message, is

$$L = \frac{L'}{2} = 0.78$$

and

$$\eta = \frac{0.72}{0.78} = 0.923$$

If we proceed with $N = 3$ (the third-order extension of the source), we have eight possible messages, and following the Huffman procedure, we find the code as shown in Table 13.5. The word length L'' is

$$\begin{aligned} L'' &= (0.512)1 + (0.128 + 0.128 + 0.128)3 \\ &\quad + (0.032 + 0.032 + 0.032)5 + (0.008)5 \\ &= 2.184 \end{aligned}$$

Then,

$$L = \frac{L''}{3} = 0.728$$

and

$$\eta = \frac{0.72}{0.728} = 0.989$$

13.3 ERROR-FREE COMMUNICATION OVER A NOISY CHANNEL

As seen in the previous section, messages of a source with entropy $H(\mathrm{m})$ can be encoded by using an average of $H(\mathrm{m})$ digits per message. This encoding has zero redundancy. Hence, if we transmit these coded messages over a noisy channel, some of the information will be received erroneously. There is absolutely no possibility of error-free communication over a noisy channel when messages are encoded with zero redundancy. The use of redundancy, in general, helps combat noise. This can be seen from a simple example of a **single parity check code**, in which an extra binary digit is added to each codeword to ensure that the total number of **1**s in the resulting codeword is always even (or odd). If a single error occurs in the received codeword, the parity is violated, and the receiver requests retransmission. This is a rather simple example to demonstrate the utility of redundancy. More complex coding procedures, which can correct up to n digits, will be discussed in the next chapter.

The addition of an extra digit increases the average word length to $H(\mathrm{m}) + 1$, giving $\eta = H(\mathrm{m})/[H(\mathrm{m}) + 1]$, and the redundancy is $1 - \eta = 1/[H(\mathrm{m}) + 1]$. Thus, the addition of an extra check digit increases redundancy, but it also helps combat noise. Immunity against channel noise can be increased by increasing the redundancy. Shannon has shown that it is possible to achieve error-free communication by adding sufficient redundancy.

Transmission over Binary Symmetric Channels

We consider a binary symmetric channel (BSC) with an error probability P_e, then for error-free communication over this channel, messages from a source with entropy $H(\mathrm{m})$ must be encoded by binary codes with a word length of at least $H(\mathrm{m})/C_s$, where

$$C_s = 1 - \left[P_e \log \frac{1}{P_e} + (1 - P_e) \log \frac{1}{1 - P_e}\right] \tag{13.14}$$

The parameter C_s ($C_s < 1$) is called the **channel capacity** (to be discussed next in Sec. 13.4).

Because of the intentional addition of redundancy for error protection, the efficiency of these codes is always below $C_s < 1$. If a certain binary channel has $C_s = 0.4$, a code that can achieve error-free communication must have at least 2.5 $H(\mathrm{m})$ binary digits per message, which is 2.5 times as many digits as are required for coding without redundancy. This means there are 1.5 $H(\mathrm{m})$ redundant bits per message bit. Thus, on the average, for every 2.5 digits transmitted, one digit is the information digit and 1.5 digits are redundant, or check, digits, giving a redundancy of $1 - C_s = 0.6$.

As discussed in the beginning of this chapter, P_e, the error probability of binary signaling, varies as e^{-kE_b} and, hence, to make $P_e \to 0$, either $S_i \to \infty$ or $R_b \to 0$. Because S_i must be finite, $P_e \to 0$ only if $R_b \to 0$. But Shannon's results state that it is really not necessary to let $R_b \to 0$ for error-free communication over bandwidth B. All that is required is to hold R_b below C, the channel capacity per second ($C = 2BC_s$). Where is the discrepancy? To answer

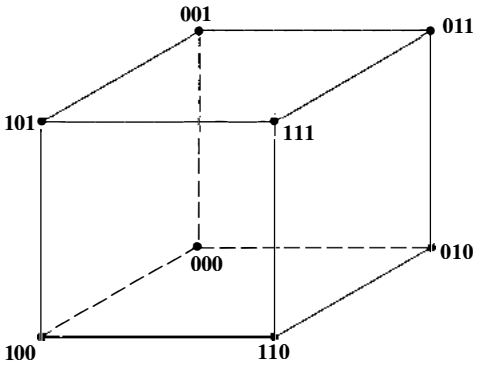


Figure 13.1 Three-dimensional cube in Hamming space.

this question let us investigate carefully the role of redundancy in error-free communication. Although the discussion here is with reference to a binary scheme, it is quite general and can be extended to the M-ary case.

Consider a simple method of reducing P_e by repeating a given digit an odd number of times. For example, we can transmit **0** and **1** as **000** and **111**. The receiver uses the majority rule to make the decision; that is, if at least two out of three digits are **1**, the decision is **1**, and if at least two out of three digits are **0**, the decision is **0**. Thus, if fewer than two of the three digits are in error, the information is received error-free. Similarly, to correct two errors, we need five repetitions. In any case, repetitions cause redundancy but improve P_e (Example 8.8).

It will be instructive to understand the situation just described from a graphic point of view. Consider the case of three repetitions. We can show all eight possible sequences of three binary digits graphically as the vertices of a cube (Fig. 13.1). It is convenient to map binary sequences as shown in Fig. 13.1 and to talk in terms of what is called the **Hamming distance** between binary sequences. If two binary sequences of the same length differ in j places (j digits), then the Hamming distance between the sequences is considered to be j. Thus, the Hamming distance between **000** and **010** (or **001** and **101**) is 1, and is 3 between **000** and **111**. In the case of three repetitions, we transmit binary **1** by **111** and binary **0** by **000**. The Hamming distance between these two sequences is 3. Observe that of the eight possible vertices, we are occupying only two (**000** and **111**) for transmitted messages. At the receiver, however, because of channel noise, we are liable to receive any one of the eight sequences. The majority decision rule can be interpreted as a rule that decides in favor of the message (**000** or **111**) that is at the closest Hamming distance to the received sequence. Sequences **000**, **001**, **010**, and **100** are within 1 unit of the Hamming distance from **000** but are at least 2 units away from **111**. Hence, when we receive any one of these four sequences, our decision is binary **0**. Similarly, when any one of the sequences **110**, **111**, **011**, or **101** is received, the decision is binary **1**.

We can now see why the error probability is reduced in this scheme. Of the eight possible vertices, we have used only two, which are separated by 3 Hamming units. If we draw a Hamming sphere of unit radius around each of these two vertices (**000** and **111**), the two Hamming spheres* will be nonoverlapping. The channel noise can cause a distance between the received sequence and the transmitted sequence, and as long as this distance is equal to or less than 1 unit, we can still detect the message without error. In a similar way, the case of five repetitions can be represented by a hypercube of five dimensions. The transmitted sequences **00000** and **11111** occupy two vertices separated by five units, and the Hamming

* Note that the Hamming sphere is not a true geometrical hypersphere because the Hamming distance is not a true geometrical distance (e.g., sequences **001**, **010**, and **100** lie on a Hamming sphere centered at **111** and having a radius 2).

spheres of 2-unit radius drawn around each of these two vertices would be nonoverlapping. In this case, even if channel noise causes two errors, we can still detect the message correctly. Hence, the reason for the reduction in error probability is that we have not used all the available vertices for messages. Had we occupied all the available vertices for messages (as is the case without redundancy, or repetition), then if channel noise caused even one error, the received sequence would occupy a vertex assigned to another transmitted sequence, and we would inevitably make a wrong decision. Precisely because we have left the neighboring vertices of the transmitted sequence unoccupied, are we able to detect the sequence correctly, despite channel errors within a certain limit. The smaller the fraction of vertices used, the smaller the error probability. It should also be remembered that redundancy (or repetition) is what makes it possible to have unoccupied vertices.

Repetition Is Inefficient

If we continue to increase n, the number of repetitions, we will reduce P_e, but we will also reduce R_b by the factor n. But no matter how large we make n, the error probability never becomes zero. The trouble with this scheme is that it is inefficient because we are adding redundant (or check) digits to each information digit. To give an analogy, redundant (or check) digits are like guards protecting the information digit. To hire guards for each information digit is somewhat similar to a case of families living on a certain street that has been hit by several burglaries. Each family panics and hires a guard. This is obviously expensive and inefficient. A better solution would be for all the families on the street to hire one guard and share the expense. One guard can check on all the houses on the street, assuming a reasonably short street. If the street is too long, it might be necessary to hire a team of guards. But it is certainly not necessary to hire one guard per house. In using repetitions, we had a similar situation. Redundant (or repeated) digits were used to help (or check on) only one message digit. Using the clue from the preceding analogy, it might be more efficient if we used redundant digits not to check (guard) any one individual transmitted digit but, rather, a block of digits. Herein lies the key to our problem. Let us consider a group of information digits over a certain time interval of T seconds, and let us add some redundant digits to check on all these digits.

Suppose we need to transmit α binary information digits per second. Then over a period of T seconds, we have a block of αT binary information digits. If to this block of information digits we add $(\beta - \alpha)T$ check digits (i.e., $\beta - \alpha$ check digits, or redundant digits, per second), then we need to transmit βT ($\beta > \alpha$) digits for every αT information digits. Therefore over a T-second interval, we have

$$\begin{aligned} \alpha T &= \text{information digits} \\ \beta T &= \text{total transmitted digits } (\beta > \alpha) \\ (\beta - \alpha)T &= \text{check digits} \end{aligned} \tag{13.15}$$

Thus, instead of transmitting one binary digit every $1/\alpha$ second we let αT digits accumulate over T seconds. Now consider this as a message to be transmitted. There are a total of $2^{\alpha T}$ such supermessages. Thus, every T seconds we need to transmit one of the $2^{\alpha T}$ possible supermessages. These supermessages are transmitted by a sequence of βT binary digits. There are in all $2^{\beta T}$ possible sequences of βT binary digits, and they can be represented as vertices of a βT-dimensional hypercube. Because we have only $2^{\alpha T}$ messages to be transmitted, whereas $2^{\beta T}$ vertices are available, we occupy only a $2^{-(\beta-\alpha)T}$ fraction of the vertices of the βT-dimensional hypercube. Observe that we have reduced the transmission rate by a factor of α/β. This rate reduction factor α/β is independent of T. The fraction of the vertices occupied (occupancy factor) by transmitted messages is $2^{-(\beta-\alpha)T}$ and can be made as small as possible

simply by increasing T. In the limit as $T \to \infty$, the occupancy factor approaches 0. This will make the error probability go to 0, and we have the possibility of error-free communication.

One important question, however, still remains unanswered. What must be the rate reduction ratio α/β for this dream to come true? To answer this question, we observe that increasing T increases the length of the transmitted sequence (βT digits). If P_e is the digit error probability, then it can be seen from the relative frequency definition (or the law of large numbers) that as $T \to \infty$, the total number of digits in error in a sequence of βT digits ($\beta T \to \infty$) is exactly $\beta T P_e$. Hence, the received sequences will be at a Hamming distance of $\beta T P_e$ from the transmitted sequences. Therefore, for error-free communication, we must leave all the vertices unoccupied within spheres of radius $\beta T P_e$ drawn around each of the $2^{\alpha T}$ occupied vertices. In short, we must be able to pack $2^{\alpha T}$ nonoverlapping spheres, each of radius $\beta T P_e$, into the Hamming space of dimensions βT. This means that for a given β, α cannot be increased beyond some limit without causing overlap in the spheres and the consequent failure of the error correction scheme. Shannon's theorem states that for this scheme to work, α/β must be less than the constant (channel capacity) C_s, which physically is a function of the channel noise and the signal power:

$$\frac{\alpha}{\beta} < C_s \tag{13.16}$$

It must be remembered that such perfect, error-free communication is not practical. In this system we accumulate the information digits for T seconds before encoding them, and because $T \to \infty$, for error-free communication we would have to wait until eternity to start encoding. Hence, there will be an infinite delay at the transmitter and an additional delay of the same amount at the receiver. Second, the equipment needed for the storage, encoding, and decoding sequence of infinite digits would be monstrous. Needless to say, the dream of error-free communication cannot be achieved in practice. Then what is the use of Shannon's result? For one thing, it indicates the upper limit on the rate of error-free communication that can be achieved on a channel. This in itself is monumental. Second, it indicates that we can reduce the error probability below an *arbitrarily* small level by allowing only a small reduction in the rate of transmission of information digits. We can therefore seek a compromise between error-free communication with infinite delay and *virtually* error-free communication with a finite delay.

13.4 CHANNEL CAPACITY OF A DISCRETE MEMORYLESS CHANNEL

This section treats discrete memoryless channels. Consider a source that generates a message that contains r symbols $x_1, x_2, \ldots, x_r$. The receiver receives symbols $y_1, y_2, \ldots, y_s$. The set of symbols $\{y_k\}$ may or may not be identical to the set $\{x_k\}$, depending on the nature of the receiver. If we use receivers of the types discussed in Chapter 10, the set of received symbols will be the same as the set transmitted. This is because the optimum receiver, upon receiving a signal, decides which of the r symbols $x_1, x_2, \ldots, x_r$ has been transmitted. Here we shall be more general and shall not constrain the set $\{y_k\}$ to be identical to the set $\{x_k\}$.

If the channel is noiseless, then the reception of some symbol y_j uniquely determines the message transmitted. Because of noise, however, there is a certain amount of uncertainty regarding the transmitted symbol when y_j is received. If $P(x_i|y_j)$ represents the conditional probabilities that x_i was transmitted when y_j is received, then there is an uncertainty of $\log [1/P(x_i|y_j)]$ about x_i when y_j is received. When this uncertainty is averaged over all

x_i and y_j, we obtain $H(\mathrm{x}|\mathrm{y})$, which is the average uncertainty about the transmitted symbol x when a symbol y is received. Thus,

$$H(\mathrm{x}|\mathrm{y}) = \sum_i \sum_j P(x_i, y_j) \log \frac{1}{P(x_i|y_j)} \quad \text{bits per symbol} \tag{13.17}$$

For noiseless channels, the uncertainty would be zero.* Obviously, this uncertainty, $H(\mathrm{x}|\mathrm{y})$, is caused by channel noise. Hence, it is the average loss of information about a transmitted symbol when a symbol is received. We call $H(\mathrm{x}|\mathrm{y})$ the **conditional entropy** of x given y (i.e., the amount of uncertainty about x once y is known).

Note that $P(y_j|x_i)$ represents the a priori probability that y_j is received when x_i is transmitted. This is a characteristic of the channel and the receiver. Thus, a given channel (with its receiver) is specified by the **channel matrix**:

$$\begin{array}{cc} & \text{Outputs} \\ \text{Inputs} & \begin{array}{c} \\ x_1 \\ x_2 \\ \vdots \\ x_r \end{array} \begin{pmatrix} y_1 & y_2 & \cdots & y_s \\ P(y_1|x_1) & P(y_2|x_1) & \cdots & P(y_s|x_1) \\ P(y_1|x_2) & P(y_2|x_2) & \cdots & P(y_s|x_2) \\ \vdots & \vdots & \cdots & \vdots \\ P(y_1|x_r) & P(y_2|x_r) & \cdots & P(y_s|x_r) \end{pmatrix} \end{array}$$

We can use Bayes' rule to obtain the a posteriori (or reverse) conditional probabilities $P(x_i|y_j)$:

$$P(x_i|y_j) = \frac{P(y_j|x_i)P(x_i)}{P(y_j)} \tag{13.18a}$$

$$= \frac{P(y_j|x_i)P(x_i)}{\sum_i P(x_i, y_j)} \tag{13.18b}$$

$$= \frac{P(y_j|x_i)P(x_i)}{\sum_i P(x_i)P(y_j|x_i)} \tag{13.18c}$$

Thus, if the input symbol probabilities $P(x_i)$ and the channel matrix are known, the a posteriori conditional probabilities can be computed from Eqs. (13.18). The a posteriori conditional probability $P(x_i|y_j)$ is the probability that x_i was transmitted when y_j is received.

For a noise-free channel, the average amount of information received would be $H(\mathrm{x})$ bits (entropy of the source) per received symbol. Note that $H(\mathrm{x})$ is the average information transmitted over the channel per symbol. Because of channel noise, even when receiving y we still have some uncertainty about x in the average amount of $H(\mathrm{x}|\mathrm{y})$ bits of information per symbol. Therefore, in this transaction of receiving y, the amount of information the receiver receives is, on the average, $I(\mathrm{x};\mathrm{y})$ bits per received symbol, where

$$I(\mathrm{x};\mathrm{y}) = H(\mathrm{x}) - H(\mathrm{x}|\mathrm{y}) \quad \text{bits per symbol} \tag{13.19}$$

* This can be verified from the fact that for a noiseless channel all the probabilities in Eq. (13.17) are either 0 or 1. If $P(x_i|y_j) = 1$, then $\log [1/P(x_i|y_j)] = 0$ and if $P(x_i|y_j) = 0$, then $P(x_i, y_j) = P(y_j)P(x_i|y_j) = 0$. This shows that $H(\mathrm{x}|\mathrm{y}) = 0$.

$I(\mathrm{x};\mathrm{y})$ is called the **mutual information** of x and y. Because

$$H(\mathrm{x}) = \sum_i P(x_i) \log \frac{1}{P(x_i)} \qquad \text{bits}$$

we have

$$I(\mathrm{x};\mathrm{y}) = \sum_i P(x_i) \log \frac{1}{P(x_i)} - \sum_i \sum_j P(x_i, y_j) \log \frac{1}{P(x_i|y_j)}$$

Also because

$$\sum_j P(x_i, y_j) = P(x_i)$$

We have

$$\begin{aligned} I(\mathrm{x};\mathrm{y}) &= \sum_i \sum_j P(x_i, y_j) \log \frac{1}{P(x_i)} - \sum_i \sum_j P(x_i, y_j) \log \frac{1}{P(x_i|y_j)} \\ &= \sum_i \sum_j P(x_i, y_j) \log \frac{P(x_i|y_j)}{P(x_i)} \qquad (13.20a) \\ &= \sum_i \sum_j P(x_i, y_j) \log \frac{P(x_i, y_j)}{P(x_i)P(y_j)} \qquad (13.20b) \end{aligned}$$

Alternatively, by using Bayes' rule in Eq. (13.20a), we can express $I(\mathrm{x};\mathrm{y})$ as

$$I(\mathrm{x};\mathrm{y}) = \sum_i \sum_j P(x_i, y_j) \log \frac{P(y_j|x_i)}{P(y_j)} \qquad (13.20c)$$

or we may substitute Eq. (13.18c) into Eq. (13.20a):

$$I(\mathrm{x};\mathrm{y}) = \sum_i \sum_j P(x_i)P(y_j|x_i) \log \frac{P(y_j|x_i)}{\sum_i P(x_i)P(y_j|x_i)} \qquad (13.20d)$$

Equation (13.20d) expresses $I(\mathrm{x};\mathrm{y})$ in terms of the input symbol probabilities and the channel matrix.

The units of $I(\mathrm{x};\mathrm{y})$ should be carefully noted. Since $I(\mathrm{x};\mathrm{y})$ is the average amount of information received per symbol transmitted, its units are bits per symbol. If we use binary digits at the input, then the symbol is a binary digit, and the units of $I(\mathrm{x};\mathrm{y})$ are bits per binary digit.

Because $I(\mathrm{x};\mathrm{y})$ in Eq. (13.20b) is symmetrical with respect to x and y, it follows that

$$\begin{aligned} I(\mathrm{x};\mathrm{y}) &= I(\mathrm{y};\mathrm{x}) \qquad (13.21a) \\ &= H(\mathrm{y}) - H(\mathrm{y}|\mathrm{x}) \qquad (13.21b) \end{aligned}$$

The quantity $H(\mathrm{y}|\mathrm{x})$ is the conditional entropy of y given x and is the average uncertainty about the received symbol when the transmitted symbol is known. Equation (13.21b) can be rewritten as

$$H(\mathrm{x}) - H(\mathrm{x}|\mathrm{y}) = H(\mathrm{y}) - H(\mathrm{y}|\mathrm{x}) \tag{13.21c}$$

From Eq. (13.20d) it is clear that $I(\mathrm{x};\mathrm{y})$ is a function of the transmitted symbol probabilities $P(x_i)$ and the channel matrix. For a given channel, $I(\mathrm{x};\mathrm{y})$ will be maximum for some set of probabilities $P(x_i)$. This maximum value is the **channel capacity** C_s,

$$C_s = \max_{P(x_i)} I(\mathrm{x};\mathrm{y}) \quad \text{bits per symbol} \tag{13.22}$$

Thus, because we have allowed the channel input to choose any symbol probabilities $P(x_i)$, C_s represents the maximum information that can be transmitted by one symbol over the channel. These ideas will become clear from the following example of a binary symmetric channel (BSC).

Example 13.3 Find the channel capacity of the BSC shown in Fig. 13.2.

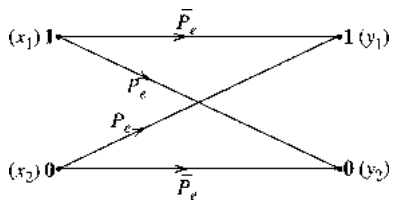


Figure 13.2
Binary symmetric channel.

Let $P(x_1) = \alpha$ and $P(x_2) = \bar{\alpha} = (1 - \alpha)$. Also,

$$P(y_1|x_2) = P(y_2|x_1) = P_e$$
$$P(y_1|x_1) = P(y_2|x_2) = \bar{P}_e = 1 - P_e$$

Substitution of these probabilities into Eq. (13.20d) gives

$$\begin{aligned} I(\mathrm{x};\mathrm{y}) &= \alpha\bar{P}_e \log\left(\frac{\bar{P}_e}{\alpha\bar{P}_e + \bar{\alpha}P_e}\right) + \alpha P_e \log\left(\frac{P_e}{\alpha P_e + \bar{\alpha}\bar{P}_e}\right) \\ &\quad + \bar{\alpha}P_e \log\left(\frac{P_e}{\alpha\bar{P}_e + \bar{\alpha}P_e}\right) + \bar{\alpha}\bar{P}_e \log\left(\frac{\bar{P}_e}{\alpha P_e + \bar{\alpha}\bar{P}_e}\right) \\ &= (\alpha P_e + \bar{\alpha}\bar{P}_e) \log\left(\frac{1}{\alpha P_e + \bar{\alpha}\bar{P}_e}\right) \\ &\quad + (\alpha\bar{P}_e + \bar{\alpha}P_e) \log\left(\frac{1}{\alpha\bar{P}_e + \bar{\alpha}P_e}\right) \\ &\quad - \left(P_e \log\frac{1}{P_e} + \bar{P}_e \log\frac{1}{\bar{P}_e}\right) \end{aligned}$$

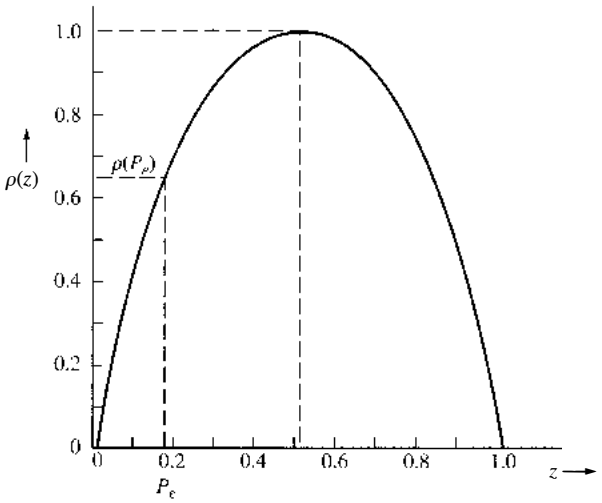


Figure 13.3 Plot of $\rho(z)$.

If we define

$$\rho(z) = z \log \frac{1}{z} + \bar{z} \log \frac{1}{\bar{z}}$$

with $\bar{z} = 1 - z$, then

$$I(\mathrm{x};\mathrm{y}) = \rho(\alpha P_e + \bar{\alpha}\bar{P}_e) - \rho(P_e) \tag{13.23}$$

The function $\rho(z)$ vs. z is shown in Fig. 13.3 It can be seen that $\rho(z)$ is maximum at $z = \frac{1}{2}$. (Note that we are interested in the region $0 < z < 1$ only.) For a given P_e, $\rho(P_e)$ is fixed. Hence from Eq. (13.23) it follows that $I(\mathrm{x};\mathrm{y})$ is maximum when $\rho(\alpha P_e + \bar{\alpha}P_e)$ is maximum. This occurs when

$$\alpha P_e + \bar{\alpha}\bar{P}_e = 0.5$$

or

$$\alpha P_e + (1 - \alpha)(1 - P_e) = 0.5$$

This equation is satisfied when

$$\alpha = 0.5 \tag{13.24}$$

For this value of α, $\rho(\alpha P_e + \bar{\alpha}\bar{P}_e) = 1$ and

$$\begin{aligned} C_s &= \max_{P(x_i)} I(\mathrm{x};\mathrm{y}) = 1 - \rho(P_e) \\ &= 1 - \left[P_e \log \frac{1}{P_e} + (1 - P_e) \log \left(\frac{1}{1 - P_e} \right) \right] \end{aligned} \tag{13.25}$$

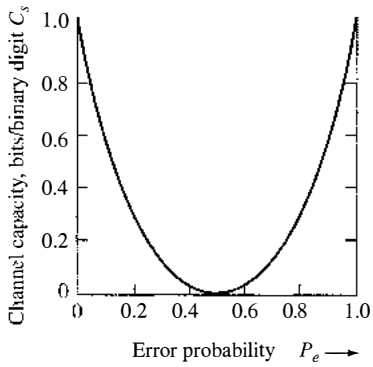


Figure 13.4 Binary symmetric channel capacity as a function of error probability P_e.

From Fig. 13.4, which shows C_s vs. P_e, it follows that the maximum value of C_s is unity. This means we can transmit at most 1 bit of information per binary digit. This is the expected result, because one binary digit can convey one of the two equiprobable messages. The information content of one of the two equiprobable messages is $\log_2 2 = 1$ bit. Second, we observe that C_s is maximum when the error probability $P_e = 0$ or $P_e = 1$. When the error probability $P_e = 0$, the channel is noiseless, and we expect C_s to be maximum. But surprisingly, C_s is also maximum when $P_e = 1$. This is easy to explain, because a channel that consistently and with certainty makes errors is as good as a noiseless channel. All we have to do to have error-free reception is reverse the decision that is made; that is, if **0** is received, we decide that **1** was actually sent, and vice versa. The channel capacity C_s is zero (minimum) when $P_e = \frac{1}{2}$. If the error probability is $\frac{1}{2}$, then the transmitted symbols and the received symbols are statistically independent. If we received **0**, for example, either **1** or **0** is equally likely to have been transmitted, and the information received is zero.

Channel Capacity per Second

The channel capacity C_s in Eq. (13.22) gives the maximum possible information transmitted when one symbol (digit) is transmitted. If K symbols are being transmitted per second, then the maximum rate of transmission of information per second is KC_s. This is the channel capacity in information units per seconds and will be denoted by C (in bits per second):

$$C = KC_s$$

A Comment on Channel Capacity: Channel capacity is the property of a particular physical channel over which the information is transmitted. This is true provided the term *channel* is correctly interpreted. A channel means not only the transmission medium, it also includes the specifications of the kind of signals (binary, r-ary, etc., or orthogonal, simplex, etc.) and the kind of receiver used (the receiver determines the error probability). All these specifications are included in the channel matrix. A channel matrix completely specifies a channel. If we decide to use, for example, 4-ary digits instead of binary digits over the same physical channel, the channel matrix changes (it becomes a 4×4 matrix), as does the channel capacity. Similarly, a change in the receiver or the signal power or noise power will change the channel matrix and, hence, the channel capacity.

Measuring Channel Capacity

The channel capacity C_s is the maximum value of $H(\mathrm{x}) - H(\mathrm{x}|\mathrm{y})$; naturally, $C_s \leq \max H(\mathrm{x})$ [because $H(\mathrm{x}|\mathrm{y}) \geq 0$]. But $H(\mathrm{x})$ is the average information per input symbol. Hence, C_s is always less than (or equal to) the maximum average information per input symbol. If we use binary symbols at the input, the maximum value of $H(\mathrm{x})$ is 1 bit, occurring when $P(x_1) = P(x_2) = \frac{1}{2}$. Hence, for a binary channel, $C_s \leq 1$ bit per binary digit. If we use r-ary symbols, the maximum value of $H_r(\mathrm{x})$ is 1 r-ary unit. Hence, $C_s \leq 1$ r-ary unit per symbol.

Verification of Error-Free Communication over a BSC

We have shown that over a noisy channel, C_s bits of information can be transmitted per symbol. If we consider a binary channel, this means that for each binary digit (symbol) transmitted, the received information is C_s bits ($C_s \leq 1$). Thus, to transmit 1 bit of information, we need to transmit at least $1/C_s$ binary digits. This gives a code efficiency C_s and redundancy $1 - C_s$. Here, the transmission of information means error-free transmission, since $I(\mathrm{x}; \mathrm{y})$ was defined as the transmitted information minus the loss of information caused by channel noise.

The problem with this derivation is that it is based on a certain speculative definition of information [Eq. (13.1)]. And based on this definition, we defined the information lost during the transmission over the channel. We really have no direct proof that the information lost over the channel will oblige us in this way. Hence, the only way to ensure that this whole speculative structure is sound is to verify it. If we can show that C_s bits of error-free information can be transmitted per symbol over a channel, the verification will be complete. A general case will be discussed later. Here we shall verify the results for a BSC.

Let us consider a binary source emitting messages at a rate of α digits per second. We accumulate these information digits over T seconds to give a total of αT digits. Because αT digits form $2^{\alpha T}$ possible combinations, our problem is now to transmit one of these $2^{\alpha T}$ supermessages every T seconds. These supermessages are transmitted by a code of word length βT digits, with $\beta > \alpha$ to ensure redundancy. Because βT digits can form $2^{\beta T}$ distinct patterns (vertices of a βT-dimensional hypercube), and we have only $2^{\alpha T}$ messages, we are utilizing only a $2^{-(\beta-\alpha)T}$ fraction of the vertices. The remaining vertices are deliberately unused, to combat noise. If we let $T \to \infty$, the fraction of vertices used approaches 0. Because there are βT digits in each transmitted sequence, the number of digits received in error will be exactly $\beta T P_e$ when $T \to \infty$. We now construct Hamming spheres of radius $\beta T P_e$ each around the $2^{\alpha T}$ vertices used for the messages. When any message is transmitted, the received message will be in the Hamming sphere surrounding the vertex corresponding to that message. We use the following decision rule: If a received sequence falls inside or on a sphere surrounding message m_i, then the decision is "m_i is transmitted." If $T \to \infty$, the decision will be without error if all the $2^{\alpha T}$ spheres are nonoverlapping.

Of all the possible sequences of βT digits, the number of sequences that differ from a given sequence by exactly j digits is $\binom{\beta T}{j}$ (see Example 8.6). Hence, K, the total number of sequences that differ from a given sequence by less than or equal to $\beta T P_e$ digits, is

$$K = \sum_{j=0}^{\beta T P_e} \binom{\beta T}{j} \tag{13.26}$$

Here we use an inequality often used in information theory:[4, 7]

$$\sum_{j=0}^{\beta TP_e} \binom{\beta T}{j} \leq 2^{\beta T \rho(P_e)} \qquad P_e < 0.5$$

Hence,

$$K \leq 2^{\beta T \rho(P_e)} \tag{13.27}$$

with the definition that

$$\rho(P_e) = P_e \log \frac{1}{P_e} + (1 - P_e) \log \frac{1}{1 - P_e}$$

From the $2^{\beta T}$ possible vertices we choose $2^{\alpha T}$ vertices to be assigned to the supermessages. How shall we select these vertices? From the decision procedure it is clear that if we assign a particular vertex to a supermessage, then none of the other vertices lying within a sphere of radius βTP_e can be assigned to another supermessage. Thus, when we choose a vertex for m_1, the corresponding K vertices [Eq. (13.26)] become ineligible for consideration. We must choose, from the remaining $2^{\beta T} - K$ vertices, another vertex for m_2. We proceed in this way until all the $2^{\beta T}$ vertices have been exhausted. This is a rather tedious procedure. Let us see what happens if we choose the required $2^{\alpha T}$ vertices randomly from the $2^{\beta T}$ vertices. In this procedure there is the danger of selecting more than one vertex lying within a distance βTP_e. If, however, α/β is sufficiently small, the probability of making such a choice is extremely small as $T \to \infty$. The probability of choosing any particular vertex $\boldsymbol{s}_1$ as one of the $2^{\alpha T}$ vertices from $2^{\beta T}$ vertices is $2^{\alpha T}/2^{\beta T} = 2^{-(\beta - \alpha)T}$.

Remembering that K vertices lie within a distance of βTP_e digits from $\boldsymbol{s}_1$, the probability that we may also choose another vertex $\boldsymbol{s}_2$ that is within the distance βTP_e from each of these K vertices (that form the Hamming sphere around $\boldsymbol{s}_1$) is

$$P = K 2^{-(\beta - \alpha)T}$$

From Eq. (13.27) it follows that

$$P \leq 2^{-[\beta[1 - \rho(P_e)] - \alpha]T}$$

Hence, as $T \to \infty$, $P \to 0$ if

$$\beta[1 - \rho(P_e)] > \alpha$$

that is, if

$$\frac{\alpha}{\beta} < 1 - \rho(P_e) \tag{13.28a}$$

But $1 - \rho(P_e)$ is C_s, the channel capacity of a BSC [Eq. (13.25)]. Therefore,

$$\frac{\alpha}{\beta} < C_s \tag{13.28b}$$

Hence, the probability of choosing two sequences randomly within a distance βTP_e approaches 0 as $T \to \infty$ provided $\alpha/\beta < C_s$, and we have error-free communication. We can choose $\alpha/\beta = C_s - \epsilon$, where ϵ is arbitrarily small.

13.5 CHANNEL CAPACITY OF A CONTINUOUS MEMORYLESS CHANNEL

For a discrete random variable x taking on values $x_1, x_2, \ldots, x_n$ with probabilities $P(x_1)$, $P(x_2), \ldots, P(x_n)$, the entropy $H(\mathrm{x})$ was defined as

$$H(\mathrm{x}) = -\sum_{i=1}^{n} P(x_i) \log P(x_i) \tag{13.29}$$

For analog data, we have to deal with continuous random variables. Therefore, we must extend the definition of entropy to continuous random variables. One is tempted to state that $H(\mathrm{x})$ for continuous random variables is obtained by using the integral instead of discrete summation in Eq. (13.29)*:

$$H(\mathrm{x}) = \int_{-\infty}^{\infty} p(x) \log \frac{1}{p(x)} dx \tag{13.30}$$

We shall see that Eq. (13.30) is indeed the meaningful definition of entropy for a continuous random variable. We cannot accept this definition, however, unless we show that it has the meaningful interpretation as uncertainty. A random variable x takes a value in the range $(n\Delta x, (n+1)\Delta x)$ with probability $p(n\Delta x)\,\Delta x$ in the limit as $\Delta x \to 0$. The error in the approximation will vanish in the limit as $\Delta x \to 0$. Hence $H(\mathrm{x})$, the entropy of a continuous random variable x, is given by

$$\begin{aligned} H(\mathrm{x}) &= \lim_{\Delta x \to 0} \sum_n p(n\Delta x)\,\Delta x \log \frac{1}{p(n\Delta x)\,\Delta x} \\ &= \lim_{\Delta x \to 0} \left[\sum_n p(n\Delta x) \Delta x \log \frac{1}{p(n\Delta x)} - \sum_n p(n\Delta x)\,\Delta x \log \Delta x \right] \\ &= \int_{-\infty}^{\infty} p(x) \log \frac{1}{p(x)}\, dx - \lim_{\Delta x \to 0} \log \Delta x \int_{-\infty}^{\infty} p(x)\, dx \\ &= \int_{-\infty}^{\infty} p(x) \log \frac{1}{p(x)}\, dx - \lim_{\Delta x \to 0} \log \Delta x \end{aligned} \tag{13.31}$$

In the limit as $\Delta x \to 0$, $\log \Delta x \to -\infty$. It therefore appears that the entropy of a continuous random variable is infinite. This is quite true. The magnitude of uncertainty associated with a continuous random variable is infinite. This fact is also apparent intuitively. A continuous random variable assumes an uncountable infinite number of values, and, hence, the uncertainty is on the order of infinity. Does this mean that there is no meaningful definition of entropy for a continuous random variable? On the contrary, we shall see that the first term in Eq. (13.31) serves as a meaningful measure of the entropy (average information) of a continuous random variable x. This may be argued as follows. We can consider $\int p(x) \log [1/p(x)]\, dx$ as a relative entropy with $-\log \Delta x$ serving as a datum, or reference. The information transmitted over a channel is actually the difference between the two terms $H(\mathrm{x})$ and $H(\mathrm{x}|\mathrm{y})$. Obviously, if we have a common datum for both $H(\mathrm{x})$ and $H(\mathrm{x}|\mathrm{y})$, the difference $H(\mathrm{x}) - H(\mathrm{x}|\mathrm{y})$ will be the same

* Throughout this discussion, the PDF $p_{\mathrm{x}}(x)$ will be abbreviated as $p(x)$; this practice causes no ambiguity and improves the clarity of the equations.

as the difference between their relative entropies. We are therefore justified in considering the first term in Eq. (13.31) as the **differential** entropy of x. We must, however, always remember that this is a relative entropy and not the absolute entropy. Failure to realize this subtle point generates many apparent fallacies, one of which will be given in Example 13.4.

Based on this argument, we define $H(\mathrm{x})$, the differential entropy of a continuous random variable x, as

$$H(\mathrm{x}) = \int_{-\infty}^{\infty} p(x) \log \frac{1}{p(x)}\, dx \quad \text{bits} \tag{13.32a}$$

$$= -\int_{-\infty}^{\infty} p(x) \log p(x)\, dx \quad \text{bits} \tag{13.32b}$$

Although $H(\mathrm{x})$ is the differential (relative) entropy of x, we shall call it the entropy of random variable x for brevity.

Example 13.4 A signal amplitude x is a random variable uniformly distributed in the range $(-1, 1)$. This signal is passed through an amplifier of gain 2. The output y is also a random variable, uniformly distributed in the range $(-2, 2)$. Determine the (differential) entropies $H(\mathrm{x})$ and $H(\mathrm{y})$.

We have

$$P(x) = \begin{cases} \frac{1}{2} & |x| < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$P(y) = \begin{cases} \frac{1}{4} & |y| < 2 \\ 0 & \text{otherwise} \end{cases}$$

Hence,

$$H(\mathrm{x}) = \int_{-1}^{1} \frac{1}{2} \log 2\, dx = 1 \text{ bit}$$

$$H(\mathrm{y}) = \int_{-2}^{2} \frac{1}{4} \log 4\, dx = 2 \text{ bits}$$

The entropy of the random variable y is 1 bit higher than that of x. This result may come as a surprise, since a knowledge of x uniquely determines y, and vice versa, because y = 2x. Hence, the average uncertainty of x and y should be identical. Amplification itself can neither add nor subtract information. Why, then, is $H(\mathrm{y})$ twice as large as $H(\mathrm{x})$? This becomes clear when we remember that $H(\mathrm{x})$ and $H(\mathrm{y})$ are differential (relative) entropies, and they will be equal if and only if their datum (or reference) entropies are equal. The reference entropy R_1 for x is $-\log \Delta x$, and the reference entropy R_2 for y is $-\log \Delta y$

(in the limit as $\Delta x,\ \Delta y \to 0$),

$$R_1 = \lim_{\Delta x \to 0} -\log \Delta x$$
$$R_2 = \lim_{\Delta y \to 0} -\log \Delta y$$

and

$$\begin{aligned} R_1 - R_2 &= \lim_{\Delta x, \Delta y \to 0} \log \left(\frac{\Delta y}{\Delta x} \right) \\ &= \log \left(\frac{dy}{dx} \right) \\ &= \log 2 = 1 \text{ bit} \end{aligned}$$

Thus, R_1, the reference entropy of x, is higher than the reference entropy R_2 for y. Hence, if x and y have equal absolute entropies, their differential (relative) entropies must differ by 1 bit.

Maximum Entropy for a Given Mean Square Value of x

For discrete random variables, we observed that entropy was maximum when all the outcomes (messages) were equally likely (uniform probability distribution). For continuous random variables, there also exists a PDF $p(x)$ that maximizes $H(\mathrm{x})$ in Eqs. (13.32). In the case of a continuous distribution, however, we may have additional constraints on x. Either the maximum value of x or the mean square value of x may be given. We shall find here the PDF $p(x)$ that will yield maximum entropy when $\overline{\mathrm{x}^2}$ is given to be a constant σ^2. The problem, then, is to maximize $H(\mathrm{x})$:

$$H(\mathrm{x}) = \int_{-\infty}^{\infty} p(x) \log \frac{1}{p(x)}\, dx \tag{13.33}$$

with the constraints

$$\int_{-\infty}^{\infty} p(x)\, dx = 1 \tag{13.34a}$$
$$\int_{-\infty}^{\infty} x^2 p(x)\, dx = \sigma^2 \tag{13.34b}$$

To solve this problem, we use a theorem from the calculus of variation. Given the integral I,

$$I = \int_a^b F(x,p)\, dx \tag{13.35}$$

subject to the following constraints:

$$\begin{aligned}
&\int_a^b \varphi_1(x,p)\,dx = \lambda_1 \\
&\int_a^b \varphi_2(x,p)\,dx = \lambda_2 \\
&\qquad\vdots \\
&\int_a^b \varphi_k(x,p)\,dx = \lambda_k
\end{aligned} \tag{13.36}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_k$ are given constants. The result from the calculus of variation states that the form of $p(x)$ that maximizes I in Eq. (13.35) with the constraints in Eq. (13.36) is found from the solution of the equation

$$\frac{\partial F}{\partial p} + \alpha_1 \frac{\partial \varphi_1}{\partial p} + \alpha_2 \frac{\partial \varphi_2}{\partial p} + \cdots + \alpha_k \frac{\partial \varphi_k}{\partial p} = 0 \tag{13.37}$$

The quantities $\alpha_1, \alpha_2, \ldots, \alpha_k$ are adjustable constants, called **undetermined multipliers**, which can be found by substituting the solution of $p(x)$ [obtained from Eq. (13.37)] in Eq. (13.36). In the present case,

$$\begin{aligned}
F(p,x) &= p \log \frac{1}{p} \\
\varphi_1(x,p) &= p \\
\varphi_2(x,p) &= x^2 p
\end{aligned}$$

Hence, the solution for p is given by

$$\frac{\partial}{\partial p}\left(p \log \frac{1}{p}\right) + \alpha_1 + \alpha_2 \frac{\partial}{\partial p} x^2 p = 0$$

or

$$-(1 + \log p) + \alpha_1 + \alpha_2 x^2 = 0$$

Solving for p, we have

$$p = e^{(\alpha_1 - 1)} e^{\alpha_2 x^2} \tag{13.38}$$

Substituting Eq. (13.38) into Eq. (13.34a), we have

$$\begin{aligned}
1 &= \int_{-\infty}^{\infty} e^{\alpha_1 - 1} e^{\alpha_2 x^2}\,dx \\
&= 2e^{\alpha_1 - 1} \int_0^{\infty} e^{\alpha_2 x^2}\,dx \\
&= 2e^{\alpha_1 - 1}\left(\frac{1}{2}\sqrt{\frac{\pi}{-\alpha_2}}\right)
\end{aligned}$$

provided α_2 is negative, or

$$e^{\alpha_1 - 1} = \sqrt{\frac{-\alpha_2}{\pi}} \tag{13.39}$$

Next we substitute Eqs. (13.38) and (13.39) into Eq. (13.34b):

$$\begin{aligned}\sigma^2 &= \int_{-\infty}^{\infty} x^2 \sqrt{\frac{-\alpha_2}{\pi}} e^{\alpha_2 x^2} dx \\ &= 2\sqrt{\frac{-\alpha_2}{\pi}} \int_0^{\infty} x^2 e^{\alpha_2 x^2} dx \\ &= -\frac{1}{2\alpha_2}\end{aligned}$$

or

$$\alpha_2 = -\frac{1}{2\sigma^2} \tag{13.40a}$$

and

$$e^{\alpha_1 - 1} = \sqrt{\frac{1}{2\pi\sigma^2}} \tag{13.40b}$$

Substituting Eqs. (13.40) into Eq. (13.38), we have

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2} \tag{13.41}$$

We therefore conclude that for a given mean square value, the maximum entropy (or maximum uncertainty) is obtained when the distribution of x is Gaussian. This maximum entropy, or uncertainty, is given by

$$H(\mathrm{x}) = \int_{-\infty}^{\infty} p(x) \log_2 \frac{1}{p(x)} dx$$

Note that

$$\begin{aligned}\log \frac{1}{p(x)} &= \log \left(\sqrt{2\pi\sigma^2}\, e^{x^2/2\sigma^2}\right) \\ &= \frac{1}{2} \log (2\pi\sigma^2) + \frac{x^2}{2\sigma^2} \log e\end{aligned}$$

Hence,

$$\begin{aligned}H(x) &= \int_{-\infty}^{\infty} p(x) \left[\frac{1}{2} \log (2\pi\sigma^2) + \frac{x^2}{2\sigma^2} \log e\right] dx \qquad (13.42a) \\ &= \frac{1}{2} \log (2\pi\sigma^2) \int_{-\infty}^{\infty} p(x)\, dx + \frac{\log e}{2\sigma^2} \int_{-\infty}^{\infty} x^2 p(x)\, dx\end{aligned}$$

$$= \frac{1}{2}\log(2\pi\sigma^2) + \frac{\log e}{2\sigma^2}\sigma^2$$

$$= \frac{1}{2}\log(2\pi e\sigma^2) \tag{13.42b}$$

$$= \frac{1}{2}\log(17.1\sigma^2) \tag{13.42c}$$

To reiterate, for a given mean square value $\overline{x^2}$, the entropy is maximum for a Gaussian distribution, and the corresponding entropy is $\frac{1}{2}\log(2\pi e\sigma^2)$.

The reader can similarly show (Prob. 13-5.1) that if x is constrained to some peak value M $(-M < x < M)$, then the entropy is maximum when x is uniformly distributed:

$$p(x) = \begin{cases} \dfrac{1}{2M} & -M < x < M \\ 0 & \text{otherwise} \end{cases}$$

Entropy of a Band-Limited White Gaussian Noise

Consider a band-limited white Gaussian noise $n(t)$ with power spectral density (PSD) $\mathcal{N}/2$. Because

$$R_n(\tau) = \mathcal{N}B\,\text{sinc}\,(2\pi B\tau)$$

we know that sinc $(2\pi B\tau)$ is zero at $\tau = \pm k/2B$ (k integer). Therefore,

$$R_n\left(\frac{k}{2B}\right) = 0 \qquad k = \pm 1, \pm 2, \pm 3, \ldots$$

Hence,

$$R_n\left(\frac{k}{2B}\right) = \overline{n(t)n\left(t + \frac{k}{2B}\right)} = 0 \qquad k = \pm 1, \pm 2, \ldots$$

Because $n(t)$ and $n(t + k/2B)$ $(k = \pm 1, \pm 2, \ldots)$ are Nyquist samples of $n(t)$, it follows that all Nyquist samples of $n(t)$ are uncorrelated. Because $n(t)$ is Gaussian, uncorrelatedness implies independence. Hence, all Nyquist samples of $n(t)$ are independent. Note that

$$\overline{n^2} = R_n(0) = \mathcal{N}B$$

Hence, the variance of each Nyquist sample is $\mathcal{N}B$. From Eq. (13.42b) it follows that the entropy $H(n)$ of each Nyquist sample of $n(t)$ is

$$H(n) = \frac{1}{2}\log(2\pi e\mathcal{N}B) \quad \text{bits per sample} \tag{13.43a}$$

Because $n(t)$ is completely specified by $2B$ Nyquist samples per second, the entropy per second of $n(t)$ is the entropy of $2B$ Nyquist samples. Because all the samples are independent,

knowledge of one sample gives no information about any other sample. Hence, the entropy of $2B$ Nyquist samples is the sum of the entropies of the $2B$ samples, and

$$H'(\mathrm{n}) = B \log (2\pi e \mathcal{N} B) \quad \text{bit/s} \tag{13.43b}$$

where $H'(\mathrm{n})$ is the entropy per second of $\mathrm{n}(t)$.

From the results derived thus far, we can draw one significant conclusion. Among all signals band-limited to B Hz and constrained to have a certain mean square value σ^2, the white Gaussian band-limited signal has the largest entropy per second. To understand the reason for this, recall that for a given mean square value, Gaussian samples have the largest entropy; moreover, all the $2B$ samples of a Gaussian band-limited process are independent. Hence, the entropy per second is the sum of the entropies of all the $2B$ samples. In processes that are not white, the Nyquist samples are correlated, and, hence, the entropy per second is less than the sum of the entropies of the $2B$ samples. If the signal is not Gaussian, then its samples are not Gaussian, and, hence, the entropy per sample is also less than the maximum possible entropy for a given mean square value. To reiterate, for a class of band-limited signals constrained to a certain mean square value, the white Gaussian signal has the largest entropy per second, or the largest amount of uncertainty. This is also the reason why white Gaussian noise is the worst possible noise in terms of interference with signal transmission.

Mutual Information $I(\mathrm{x}; \mathrm{y})$

The ultimate test of any concept is its usefulness. We shall now show that the relative entropy defined in Eqs. (13.32) does lead to meaningful results when we consider $I(\mathrm{x}; \mathrm{y})$, the mutual information of continuous random variables x and y. We wish to transmit a random variable x over a channel. Each value of x in a given continuous range is now a message that may be transmitted, for example, as a pulse of height x. The message recovered by the receiver will be a continuous random variable y. If the channel were noise free, the received value y would uniquely determine the transmitted value x. But channel noise introduces a certain uncertainty about the true value of x. Consider the event that at the transmitter, a value of x in the interval $(x, x + \Delta x)$ has been transmitted ($\Delta x \to 0$). The probability of this event is $p(x)\Delta x$ in the limit $\Delta x \to 0$. Hence, the amount of information transmitted is $\log [1/p(x)\Delta x]$. Let the value of y at the receiver be y and let $p(x|y)$ be the conditional probability density of x when $\mathrm{y} = y$. Then $p(x|y)\Delta x$ is the probability that x will lie in the interval $(x, x + \Delta x)$ when $\mathrm{y} = y$ (provided $\Delta x \to 0$). Obviously, there is an uncertainty about the event that x lies in the interval $(x, x + \Delta x)$. This uncertainty, $\log [1/p(x|y)\Delta x]$, arises because of channel noise and therefore represents a loss of information. Because $\log [1/p(x)\Delta x]$ is the information transmitted and $\log [1/p(x|y)\Delta x]$ is the information lost over the channel, the net information received is $I(x; y)$ given by

$$\begin{aligned} I(x; y) &= \log \left[\frac{1}{p(x)\Delta x}\right] - \log \left[\frac{1}{p(x/y)\Delta x}\right] \\ &= \log \frac{p(x|y)}{p(x)} \end{aligned} \tag{13.44}$$

Note that this relation is true in the limit $\Delta x \to 0$. Therefore, $I(x; y)$, represents the information transmitted over a channel if we receive y ($\mathrm{y} = y$) when x is transmitted ($\mathrm{x} = x$). We are interested in finding the average information transmitted over a channel when some x is transmitted and a certain y is received. We must therefore average $I(x; y)$ over all values of

x and y. The average information transmitted will be denoted by $I(\mathrm{x};\mathrm{y})$, where

$$I(\mathrm{x};\mathrm{y}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)I(x;y)\,dx\,dy \tag{13.45a}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\log\frac{p(x|y)}{p(x)}\,dx\,dy \tag{13.45b}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\log\frac{1}{p(x)}\,dx\,dy + \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\log p(x|y)\,dx\,dy$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x)p(y|x)\log\frac{1}{p(x)}\,dx\,dy + \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\log p(x|y)\,dx\,dy$$

$$= \int_{-\infty}^{\infty} p(x)\log\frac{1}{p(x)}dx\int_{-\infty}^{\infty} p(y|x)\,dy + \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\log p(x|y)\,dx\,dy$$

Note that

$$\int_{-\infty}^{\infty} p(y|x)\,dy = 1 \qquad \text{and} \qquad \int_{-\infty}^{\infty} p(x)\log\frac{1}{p(x)}\,dx = H(\mathrm{x})$$

Hence,

$$I(\mathrm{x};\mathrm{y}) = H(\mathrm{x}) + \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\log p(x|y)\,dx\,dy \tag{13.46a}$$

$$= H(\mathrm{x}) - \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\log\frac{1}{p(x|y)}\,dx\,dy \tag{13.46b}$$

The integral on the right-hand side is the average over x and y of $\log[1/p(x|y)]$. But $\log[1/p(x|y)]$ represents the uncertainty about x when y is received. This, as we have seen, is the information lost over the channel. The average of $\log[1/p(x|y)]$ is the average loss of information when some x is transmitted and some y is received. This, by definition, is $H(\mathrm{x}|\mathrm{y})$, the conditional (differential) entropy of x given y,

$$H(\mathrm{x}|\mathrm{y}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\log\frac{1}{p(x|y)}\,dx\,dy \tag{13.47}$$

Hence,

$$I(\mathrm{x};\mathrm{y}) = H(\mathrm{x}) - H(\mathrm{x}|\mathrm{y}) \tag{13.48}$$

Thus, when some value of x is transmitted and some value of y is received, the average information transmitted over the channel is $I(\mathrm{x};\mathrm{y})$, given by Eq. (13.48). We can define the channel capacity C_s as the maximum amount of information that can be transmitted, on the average, per sample or per value transmitted:

$$C_s = \max I(\mathrm{x};\mathrm{y}) \tag{13.49}$$

For a given channel, $I(\mathrm{x};\mathrm{y})$ is a function of the input probability density $p(x)$ alone. This can be shown as follows:

$$p(x,y) = p(x)p(y|x) \tag{13.50}$$

$$\frac{p(x|y)}{p(x)} = \frac{p(y|x)}{p(y)}$$

$$= \frac{p(y|x)}{\int_{-\infty}^{\infty} p(x,y)\,dx}$$

$$= \frac{p(y|x)}{\int_{-\infty}^{\infty} p(x)p(y|x)\,dx} \tag{13.51}$$

Substituting Eqs. (13.50) and (13.51) into Eq. (13.45b), we obtain

$$I(\mathrm{x};\mathrm{y}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x)p(y|x)\log\left(\frac{p(y|x)}{\int_{-\infty}^{\infty} p(x)p(y|x)\,dx}\right) dx\,dy \tag{13.52}$$

The conditional probability density $p(y|x)$ is characteristic of a given channel. Hence, for a given channel specified by $p(y|x)$, $I(\mathrm{x};\mathrm{y})$ is a function of the input probability density $p(x)$ alone. Thus,

$$C_s = \max_{p(x)} I(\mathrm{x};\mathrm{y})$$

If the channel allows the transmission of K values per second, then C, the channel capacity per second, is given by

$$C = KC_s \text{ bit/s} \tag{13.53}$$

Just as in the case of discrete variables, $I(\mathrm{x};\mathrm{y})$ is symmetrical with respect to x and y for continuous random variables. This can be seen by rewriting Eq. (13.45b) as

$$I(\mathrm{x};\mathrm{y}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}\,dx\,dy \tag{13.54}$$

This equation shows that $I(\mathrm{x};\mathrm{y})$ is symmetrical with respect to x and y. Hence,

$$I(\mathrm{x};\mathrm{y}) = I(\mathrm{y};\mathrm{x})$$

From Eq. (13.48) it now follows that

$$I(\mathrm{x};\mathrm{y}) = H(\mathrm{x}) - H(\mathrm{x}|\mathrm{y}) = H(\mathrm{y}) - H(\mathrm{y}|\mathrm{x}) \tag{13.55}$$

Capacity of a Band-Limited AWGN Channel

The channel capacity C is, by definition, the maximum rate of information transmission over a channel. The mutual information $I(\mathrm{x};\mathrm{y})$ is given by Eq. (13.55):

$$I(\mathrm{x};\mathrm{y}) = H(\mathrm{y}) - H(\mathrm{y}|\mathrm{x}) \tag{13.56}$$

The channel capacity C is the maximum value of the mutual information $I(\mathrm{x};\mathrm{y})$ per second. Let us first find the maximum value of $I(\mathrm{x};\mathrm{y})$ per sample. We shall find here the capacity

of a channel band-limited to B Hz and disturbed by a white Gaussian noise of PSD $\mathcal{N}/2$. In addition, we shall constrain the signal power (or its mean square value) to S. The disturbance is assumed to be additive; that is, the received signal $\mathrm{y}(t)$ is given by

$$\mathrm{y}(t) = \mathrm{x}(t) + \mathrm{n}(t) \tag{13.57}$$

Because the channel is band-limited, both the signal $\mathrm{x}(t)$ and the noise $\mathrm{n}(t)$ are band-limited to B Hz. Obviously, $\mathrm{y}(t)$ is also band-limited to B Hz. All these signals can therefore be completely specified by samples taken at the uniform rate of $2B$ samples per second. Let us find the maximum information that can be transmitted per sample. Let x, n, and y represent samples of $\mathrm{x}(t)$, $\mathrm{n}(t)$, and $\mathrm{y}(t)$, respectively. The information $I(\mathrm{x};\mathrm{y})$ transmitted per sample is given by Eq. (13.56):

$$I(\mathrm{x};\mathrm{y}) = H(\mathrm{y}) - H(\mathrm{y}|\mathrm{x})$$

We shall now find $H(\mathrm{y}|\mathrm{x})$. By definition [Eq. (13.47)],

$$\begin{aligned} H(\mathrm{y}|\mathrm{x}) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\log\frac{1}{p(y|x)}\,dx\,dy \\ &= \int_{-\infty}^{\infty} p(x)\,dx \int_{-\infty}^{\infty} p(y|x)\log\frac{1}{p(y|x)}\,dy \end{aligned}$$

Because

$$y = x + n$$

for a given x, y is equal to n plus a constant (x). Hence, the distribution of y when x has a given value is identical to that of n except for a translation by x. If $p_{\mathrm{n}}(\cdot)$ represents the PDF of noise sample n, then

$$p(y|x) = p_{\mathrm{n}}(y-x) \tag{13.58}$$

$$\int_{-\infty}^{\infty} p(y|x)\log\frac{1}{p(y|x)}\,dy = \int_{-\infty}^{\infty} p_{\mathrm{n}}(y-x)\log\frac{1}{p_{\mathrm{n}}(y-x)}\,dy$$

Letting $y - x = z$, we have

$$\int_{-\infty}^{\infty} p(y|x)\log\frac{1}{p(y|x)}\,dy = \int_{-\infty}^{\infty} p_{\mathrm{n}}(z)\log\frac{1}{p_{\mathrm{n}}(z)}\,dz$$

The right-hand side is the entropy $H(\mathrm{n})$ of the noise sample n. Hence,

$$\begin{aligned} H(\mathrm{y}|\mathrm{x}) &= H(\mathrm{n})\int_{-\infty}^{\infty} p(x)\,dx \\ &= H(\mathrm{n}) \end{aligned} \tag{13.59}$$

In deriving Eq. (13.59), we made no assumptions about the noise. Hence, Eq. (13.59) is very general and applies to all types of noise. The only condition is that the noise disturb the channel in an additive fashion. Thus,

$$I(\mathrm{x};\mathrm{y}) = H(\mathrm{y}) - H(\mathrm{n}) \quad \text{bits per sample} \tag{13.60}$$

We have assumed that the mean square value of the signal x(t) is constrained to have a value S, and the mean square value of the noise is N. We shall also assume that the signal x(t) and the noise n(t) are independent. In such a case, the mean square value of y will be the sum of the mean square values of x and n. Hence,

$$\overline{\mathrm{y}^2} = S + N$$

For a given noise [given $H(\mathrm{n})$], $I(\mathrm{x};\mathrm{y})$ is maximum when $H(\mathrm{y})$ is maximum. We have seen that for a given mean square value of y ($\overline{\mathrm{y}^2} = S + N$), $H(\mathrm{y})$ will be maximum if y is Gaussian, and the maximum entropy $H_{\max}(\mathrm{y})$ is then given by

$$H_{\max}(\mathrm{y}) = \frac{1}{2}\log\left[2\pi e(S+N)\right] \tag{13.61}$$

Because

$$\mathrm{y} = \mathrm{x} + \mathrm{n}$$

and n is Gaussian, y will be Gaussian only if x is Gaussian. As the mean square value of x is S, this implies that

$$p(x) = \frac{1}{\sqrt{2\pi S}} e^{-x^2/2S}$$

and

$$\begin{aligned} I_{\max}(\mathrm{x};\mathrm{y}) &= H_{\max}(\mathrm{y}) - H(\mathrm{n}) \\ &= \frac{1}{2}\log\left[2\pi e(S+N)\right] - H(\mathrm{n}) \end{aligned}$$

For a white Gaussian noise with mean square value N,

$$H(\mathrm{n}) = \frac{1}{2}\log 2\pi eN \qquad N = \mathcal{N}B$$

and

$$C_s = I_{\max}(\mathrm{x};\mathrm{y}) = \frac{1}{2}\log\left(\frac{S+N}{N}\right) \tag{13.62a}$$

$$= \frac{1}{2}\log\left(1 + \frac{S}{N}\right) \tag{13.62b}$$

The channel capacity per second will be the maximum information that can be transmitted per second. Equations (13.62) represent the maximum information transmitted per sample. If all the samples are statistically independent, the total information transmitted per second will be $2B$ times C_s. If the samples are not independent, then the total information will be less than $2BC_s$. Because the channel capacity C represents the maximum possible information transmitted per second,

$$\begin{aligned} C &= 2B\left[\frac{1}{2}\log\left(1 + \frac{S}{N}\right)\right] \\ &= B\log\left(1 + \frac{S}{N}\right) \quad \text{bit/s} \end{aligned} \tag{13.63}$$

The samples of a band-limited Gaussian signal are independent if and only if the signal power spectral density (PSD) is uniform over the band (Example 9.2 and Prob. 9.2-3). Obviously, to transmit information at the maximum rate [Eq. (13.63)], the PSD of signal y(t) must be uniform. The PSD of y is given by

$$S_y(f) = S_x(f) + S_n(f)$$

Because $S_n(f) = \mathcal{N}/2$, the PSD of x(t) must also be uniform. Thus, the maximum rate of transmission (C bit/s) is attained when x(t) is also a white Gaussian signal.

To recapitulate, when the channel noise is additive, white, and Gaussian with mean square value N ($N = \mathcal{N}B$), the channel capacity C of a band-limited channel under the constraint of a given signal power S is given by

$$C = B \log \left(1 + \frac{S}{N}\right) \qquad \text{bit/s}$$

where B is the channel bandwidth in hertz. The maximum rate of transmission (C bit/s) can be realized only if the input signal is a white Gaussian signal.

Capacity of a Channel of Infinite Bandwidth

Superficially, Eq. (13.63) seems to indicate that the channel capacity goes to ∞ as the channel's bandwidth B goes to ∞. This, however, is not true. For white noise, the noise power $N = \mathcal{N}B$. Hence, as B increases, N also increases. It can be shown that in the limit as $B \to \infty$, C approaches a limit:

$$\begin{aligned} C &= B \log \left(1 + \frac{S}{N}\right) \\ &= B \log \left(1 + \frac{S}{\mathcal{N}B}\right) \\ \lim_{B\to\infty} C &= \lim_{B\to\infty} B \log \left(1 + \frac{S}{\mathcal{N}B}\right) \\ &= \lim_{B\to\infty} \frac{S}{\mathcal{N}} \left[\frac{\mathcal{N}B}{S} \log \left(1 + \frac{S}{\mathcal{N}B}\right)\right] \end{aligned}$$

This limit can be found by noting that

$$\lim_{x\to\infty} x \log_2 \left(1 + \frac{1}{x}\right) = \log_2 e = 1.44$$

Hence,

$$\lim_{B\to\infty} C = 1.44 \frac{S}{\mathcal{N}} \quad \text{bit/s} \tag{13.64}$$

Thus, for a white Gaussian channel noise, the channel capacity C approaches a limit of $1.44S/\mathcal{N}$ as $B \to \infty$. The variation of C with B is shown in Fig. 13.5. It is evident that the capacity can be made infinite only by increasing the signal power S to infinity. For finite signal and noise powers, the channel capacity always remains finite.

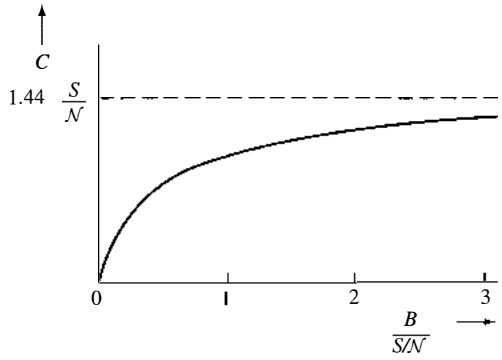


Figure 13.5 Channel capacity vs. bandwidth for a channel with white Gaussian noise and fixed signal power.

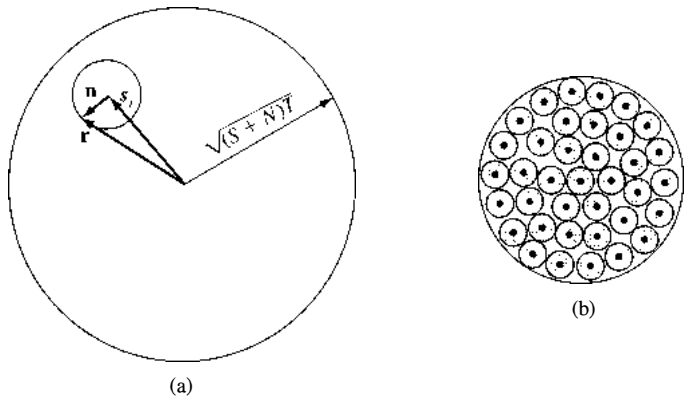


Figure 13.6 (a) Signal space representation of transmitted and received signals and noise signal. (b) Choice of signals for error-free communication.

Verification of Error-Free Communication over a Continuous Channel

Using the concepts of information theory, we have shown that it is possible to transmit error-free information at a rate of $B \log_2 (1 + S/N)$ bit/s over a channel band-limited to B Hz. The signal power is S, and the channel noise is white Gaussian with power N. This theorem can be verified in a way similar to that used for the verification of the channel capacity of a discrete case. This verification using signal space is so general that it is in reality an alternate proof of the capacity theorem.

Let us consider M-ary communication with M equiprobable messages $m_1, m_2, \ldots, m_M$ transmitted by signals $s_1(t), s_2(t), \ldots, s_M(t)$. All signals are time-limited with duration T and have an essential bandwidth B Hz. Their powers are less than or equal to S. The channel is band-limited to B, and the channel noise is white Gaussian with power N.

All the signals and noise waveforms have $2BT + 1$ dimensions. In the limit we shall let $T \to \infty$. Hence $2BT \gg 1$, and the number of dimensions will be taken as $2BT$ in our future discussion. Because the noise power is N, the energy of the noise waveform of T-second duration is NT. Given signal power S, the maximum signal energy is ST. Because signals and noise are independent, the maximum received energy is $(S + N)T$. Hence, all the received signals will lie in a $2BT$-dimensional hypersphere of radius $\sqrt{(S + N)T}$ (Fig. 13.6a). A typical received signal $s_i(t) + n(t)$ has an energy $(S_i + N)T$, and the point $\boldsymbol{r}$ representing this signal lies at a distance of $\sqrt{(S_i + N)T}$ from the origin (Fig. 13.6a). The signal vector $\boldsymbol{s}_i$, the noise vector $\boldsymbol{n}$, and the received vector $\boldsymbol{r}$ are shown in Fig. 13.6a. Because

$$|\boldsymbol{s}_i| = \sqrt{S_i T}, \qquad |\boldsymbol{n}| = \sqrt{NT}, \qquad |\boldsymbol{r}| = \sqrt{(S_i + N)T} \tag{13.65}$$

it follows that vectors $\boldsymbol{s}_i$, $\boldsymbol{n}$, and $\boldsymbol{r}$ form a right triangle. Also, $\mathbf{n}$ lies on the sphere of radius $\sqrt{NT}$, centered at $\boldsymbol{s}_i$. Note that because $\mathbf{n}$ is random, it can lie anywhere on the sphere centered at $\boldsymbol{s}_i$.*

We have M possible transmitted vectors located inside the big sphere. For each possible $\boldsymbol{s}$, we can draw a sphere of radius $\sqrt{NT}$ around $\boldsymbol{S}$. If a received vector $\boldsymbol{r}$ lies on one of the small spheres, the center of that sphere is the transmitted waveform. If we pack the big sphere with M nonoverlapping and nontouching spheres, each of radius $\sqrt{NT}$ (Fig. 13.6b), and use the centers of these M spheres for the transmitted waveforms, we will be able to detect all these M waveforms correctly at the receiver simply by using the maximum likelihood receiver. The maximum likelihood receiver looks at the received signal point $\boldsymbol{r}$ and decides that the transmitted signal is that one of the M possible transmitted points that is closest to $\boldsymbol{r}$ (smallest error vector). Every received point $\boldsymbol{r}$ will lie on the surface of one of the M nonoverlapping spheres, and using the maximum likelihood criterion, the transmitted signal will be chosen correctly as the point lying at the center of the sphere on which $\boldsymbol{r}$ lies.

Hence, our task is to find out how many such nonoverlapping small spheres can be packed into the big sphere. To compute this number, we must determine the volume of a sphere of D dimensions.

Volume of a D-Dimensional Sphere

A D-dimensional sphere is described by the equation

$$x_1^2 + x_2^2 + \cdots + x_D^2 = R^2$$

where R is the radius of the sphere. We can show that the volume $V(R)$ of a sphere of radius R is given by

$$V(R) = R^D V(1) \tag{13.66}$$

where $V(1)$ is the volume of a D-dimensional sphere of unit radius and, thus, is constant. To prove this, we have by definition

$$V(R) = \int_{x_1^2+x_2^2+\cdots+x_D^2 \le R^2} \cdots \int dx_1\, dx_2 \cdots dx_D$$

Letting $y_j = x_j/R$, we have

$$\begin{aligned} V(R) &= R^D \int_{y_1^2+y_2^2+\cdots+y_D^2 \le 1} \cdots \int dy_1\, dy_2 \cdots dy_D \\ &= R^D V(1) \end{aligned}$$

Hence, the ratio of the volumes of two spheres of radii $\hat{R}$ and R is

$$\frac{V(\hat{R})}{V(R)} = \left(\frac{\hat{R}}{R}\right)^D$$

* Because N is the average noise power, the energy over an interval T is $NT + \epsilon$, where $\epsilon \to 0$ as $T \to \infty$. Hence, we can assume that $\mathbf{n}$ lies on the sphere.

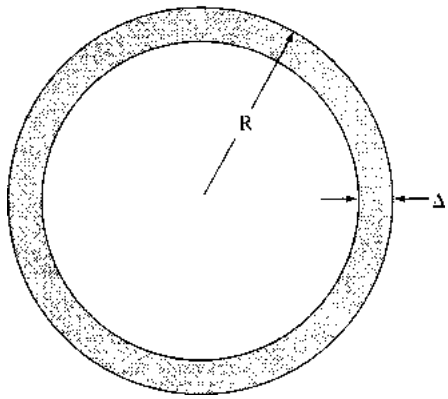


Figure 13.7 Volume of a shell of a D-dimensional hypersphere.

As direct consequence of this result, when D is large, almost all of the volume of the sphere is concentrated at the surface. This is because if $\hat{R}/R < 1$, then $(\hat{R}/R)^D \to 0$ as $D \to \infty$. This ratio approaches zero even if $\hat{R}$ differs from R by a very small amount Δ (Fig. 13.7). This means that no matter how small Δ is, the volume within radius $\hat{R}$ is a negligible fraction of the total volume within radius R if D is large enough. Hence, for a large D, almost all of the volume of a D-dimensional sphere is concentrated at the surface. Such a result sounds strange, but a little reflection will show that it is reasonable. This is because the volume is proportional to the Dth power of the radius. Thus, for large D, a small increase in R can increase the volume tremendously, and all the increase comes from a tiny increase in R near the surface of the sphere. This means that most of the volume must be concentrated at the surface.

The number of nonoverlapping spheres of radius $\sqrt{NT}$ that can be packed into a sphere of radius $\sqrt{(S+N)T}$ is bounded by the ratio of the volume of the signal sphere to the volume of the noise sphere. Hence,

$$M \leq \frac{\left[\sqrt{(S+N)T}\right]^{2BT} V(1)}{\left(\sqrt{NT}\right)^{2BT} V(1)} = \left(1 + \frac{S}{N}\right)^{BT} \tag{13.67}$$

Each of the M-ary signals carries the information of $\log_2 M$ binary digits. Hence, the transmission of one of the M signals every T seconds is equivalent to the information rate C given by

$$C = \frac{\log M}{T} \leq B \log\left(1 + \frac{S}{N}\right) \quad \text{bit/s} \tag{13.68}$$

This equation gives the upper limit of C.

To show that we can actually receive error-free information at a rate of $B \log (1 + S/N)$, we use the argument proposed by Shannon.[8] Instead of choosing the M transmitted messages at the centers of nonoverlapping spheres (Fig. 13.6b), Shannon proposed selecting the M points randomly located in the signal sphere I_s of radius $\sqrt{ST}$ (Fig. 13.8). Consider one particular transmitted signal s_k. Because the signal energy is assumed to be $\leq S$, point s_k will lie somewhere inside the signal sphere I_s of radius $\sqrt{ST}$. Because all the M signals are picked randomly from this sphere, the probability of finding a signal within a volume ΔV is $\min(1, M\Delta V/V_s)$, where V_s is the volume of I_s. But because for large D all of the volume of the sphere is concentrated at the surface, all M signal points selected randomly would lie

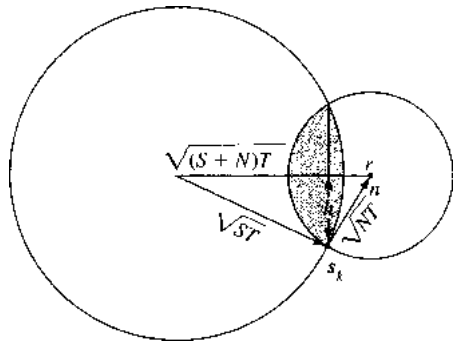


Figure 13.8 Derivation of channel capacity.

near the surface of I_s. Figure 13.8 shows the transmitted signal $\boldsymbol{s}_k$, the received signal $\boldsymbol{r}$, and the noise $\boldsymbol{n}$. We draw a sphere of radius $\sqrt{NT}$ with $\boldsymbol{r}$ as the center. This sphere intersects the sphere I_s and forms a common lens-shaped region. The signal $\boldsymbol{s}_k$ lies on the surface of both spheres. We shall use a maximum likelihood receiver. This means that when $\boldsymbol{r}$ is received, we shall make the decision that "$\boldsymbol{s}_k$ was transmitted," provided none of the remaining $M-1$ signal points are closer to $\boldsymbol{r}$ than $\boldsymbol{s}_k$. The probability of finding any one signal in the lens is V_{lens}/V_s. Hence P_e, the error probability in the detection of $\boldsymbol{s}_k$ when $\boldsymbol{r}$ is received, is

$$\begin{aligned} P_e &= (M-1)\frac{V_{\text{lens}}}{V_s} \\ &< M\frac{V_{\text{lens}}}{V_s} \end{aligned}$$

From Fig. 13.8, we observe that $V_{\text{lens}} < V(h)$, where $V(h)$ is the volume of the D-dimensional sphere of radius h. Because $\boldsymbol{r}$, $\boldsymbol{s}_k$, and $\boldsymbol{n}$ form a right triangle,

$$h\sqrt{(S+N)T} = \sqrt{(ST)(NT)} \qquad \text{and} \qquad h = \sqrt{\frac{SNT}{S+N}}$$

Hence,

$$V(h) = \left(\frac{SNT}{S+N}\right)^{BT} V(1)$$

Also,

$$V_s = (ST)^{BT} V(1)$$

and

$$P_e < M\left(\frac{N}{S+N}\right)^{BT}$$

If we choose

$$M = \left[k\left(1+\frac{S}{N}\right)\right]^{BT}$$

then

$$P_e < [k]^{BT}$$

If we let $k = 1 - \Delta$, where Δ is a positive number chosen as small as we wish, then

$$P_e \to 0 \qquad \text{as} \qquad BT \to \infty$$

This means that P_e can be made arbitrarily small by increasing T, provided M is chosen arbitrarily close to $(1 + S/N)^{BT}$. Thus,

$$\begin{aligned} C &= \frac{1}{T} \log_2 M \\ &= \left[B \log \left(1 + \frac{S}{N} \right) - \epsilon \right] \qquad \text{bit/s} \end{aligned} \tag{13.69}$$

where ϵ is a positive number chosen as small as we please. This leads to $k = 2^{-\epsilon T}$ and proves the desired result. A more rigorous derivation of this result can be found in Wozencraft and Jacobs.[9]

Because the M signals are selected randomly from the signal space, they tend to acquire the statistics of white noise[8] (i.e., a white Gaussian random process).

Comments on Channel Capacity

According to the result derived in this chapter, theoretically we can communicate error-free up to C bit/s. There are, however, practical difficulties in achieving this rate. In proving the capacity formula, we assumed that communication is effected by signals of duration T. This means we must wait T seconds to accumulate the input data and then encode it by one of the waveforms of duration T. Because the capacity rate is achieved only in the limit as $T \to \infty$, we have a long wait at the receiver to get the information. Moreover, because the number of possible messages that can be transmitted over interval T increases exponentially with T, the transmitter and receiver structures increase in complexity beyond imagination as $T \to \infty$.

The channel capacity indicated by Shannon's equation [Eq. (13.69)] is the maximum error-free communication rate achievable on an optimum system without any restrictions (except for bandwidth B, signal power S, and Gaussian white channel noise power N). If we have any other restrictions, this maximum rate will not be achieved. For example, if we consider a binary channel (a channel restricted to transmit only binary signals), we will not be able to attain Shannon's rate, even if the channel is optimum. In Sec. 13.9, MATLAB Computer Exercise 13.2 supplies numerical confirmation. The channel capacity formula [Eq. (13.63)] indicates that the transmission rate is a monotonically increasing function of the signal power S. If we use a binary channel, however, we will find increasing the transmitted power beyond a certain point buys very little advantage. Hence, on a binary channel, increasing S will not increase the error-free communication rate beyond some value. This does not mean that the channel capacity formula has failed. It simply means that when we have a large amount of power (with a finite bandwidth) available, the binary scheme is not the optimum communication scheme.

One last comment: Shannon's results tell us the upper theoretical limit of error-free communication. But they do not tell us precisely how this can be achieved. To quote the words of Abramson, written in 1963; "[This is one of the problems] which has persisted to mock information theorists since Shannon's original paper in 1948. Despite an enormous amount of effort spent since that time in quest of this Holy Grail of information theory, a *deterministic* method of generating the codes promised by Shannon is still to be found."[4] Amazingly, 30 years later,

the introduction of turbo codes and the rediscovery of the low-density parity check (LDPC) codes would completely alter the landscape. We shall introduce these codes in Chapter 14.

13.6 PRACTICAL COMMUNICATION SYSTEMS IN LIGHT OF SHANNON'S EQUATION

It would be instructive to determine the ideal law for the exchange between the SNR and the transmission bandwidth by using the channel capacity equation. Consider a message of bandwidth B that is used for modulation (or coding), with the resulting modulated signal of bandwidth B_T. This signal is received at the input of an ideal demodulator with signal and noise powers of S_i and N_i, respectively* (Fig. 13.9). The demodulator output bandwidth is B, and the SNR is S_o/N_o. Because an SNR S/N and a bandwidth B can transmit ideally $B \log (1 + S/N)$ bits of information, the ideal information rates of the signals at the input and the output of the demodulator are $B_T \log (1 + S_i/N_i)$ bits and $B \log (1 + S_o/N_o)$ bits, respectively. Because the demodulator neither creates nor destroys information, the two rates should be equal, that is,

$$B_T \log \left(1 + \frac{S_i}{N_i}\right) = B \log \left(1 + \frac{S_o}{N_o}\right)$$

and

$$\left(1 + \frac{S_o}{N_o}\right) = \left(1 + \frac{S_i}{N_i}\right)^{B_T/B} \tag{13.70a}$$

In practice, for the majority of systems, S_o/N_o as well as $S_i/N_i \gg 1$, and

$$\frac{S_o}{N_o} \simeq \left(\frac{S_i}{N_i}\right)^{B_T/B} \tag{13.70b}$$

Also,

$$\frac{S_i}{N_i} = \frac{S_i}{\mathcal{N} B_T}$$
$$= \left(\frac{S_i}{\mathcal{N} B}\right)\left(\frac{B}{B_T}\right) = \frac{B}{B_T}\gamma \qquad \gamma = \frac{S_i}{\mathcal{N} B}$$

Hence, Eqs. (13.70) become

$$\frac{S_o}{N_o} = \left(1 + \frac{\gamma}{B_T/B}\right)^{B_T/B} - 1 \tag{13.71a}$$

$$\simeq \left(\frac{\gamma}{B_T/B}\right)^{B_T/B} \tag{13.71b}$$

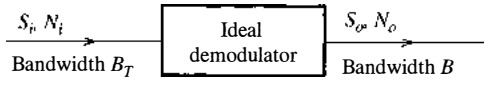


Figure 13.9 Ideal exchange between SNR and bandwidth.

* An additive white Gaussian channel noise is assumed.

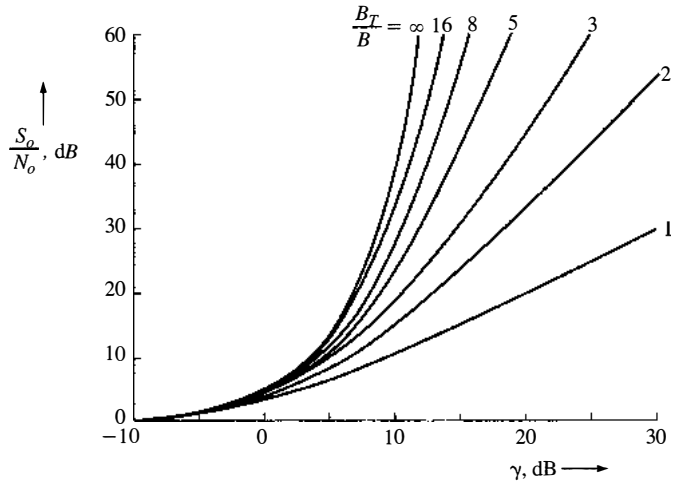


Figure 13.10 Ideal behavior of SNR vs. γ for various ratios of B_T to B.

Equations (13.70) and (13.71) give the ideal law of exchange between the SNR and the bandwidth. The output SNR S_o/N_o is plotted in Fig. 13.10 as a function of γ for various values of B_T/B.

The output SNR increases exponentially with the bandwidth expansion factor B_T/B. This means that to maintain a given output SNR, the transmitted signal power can be reduced exponentially with the bandwidth expansion factor. Thus, for a small increase in bandwidth, we can cut the transmitted power considerably. On the other hand, for a small reduction in bandwidth, we need to increase the transmitted power considerably.

Let us now investigate how two digital systems fare in comparison to the ideal system.

PCM

As seen earlier, M-ary PCM shows a saturation effect unless we go to higher values of M as γ increases. If the message signal is quantized in L levels, then each sample can be encoded by $\log_M L$ number of M-ary pulses. If B is the bandwidth of the message signal, we need to transmit $2B$ samples per second. Consequently, R_M, the number of M-ary pulses per second, is

$$R_M = 2B \log_M L$$

Also, the transmission bandwidth B_T is half the number of (M-ary) pulses per second. Hence,

$$B_T = \frac{R_M}{2} = B \log_M L \tag{13.72a}$$

From Eq. (10.98a), the power S_i is found as

$$S_i = \frac{M^2 - 1}{3} E_p R_M \tag{13.72b}$$

Also,

$$N_i = \mathcal{N} B_T = \frac{\mathcal{N} R_M}{2} \tag{13.73}$$

Each of the M-ary pulses carries the information of $\log_2 M$ bits, and we are transmitting $2B \log_M L$ number of M-ary pulses per second. Hence, we are transmitting information at a rate of R_b bits, where

$$\begin{aligned} R_b &= (2B \log_M L)(\log_2 M) \\ &= 2B_T \log_2 M \\ &= B_T \log_2 M^2 \quad \text{bit/s} \end{aligned}$$

Substitution of Eqs. (13.72b) and (13.73) into this equation yields

$$R_b = B_T \log_2 \left(1 + \frac{3\mathcal{N}}{2E_p} \frac{S_i}{N_i}\right) \quad \text{bit/s} \tag{13.74}$$

We are transmitting the information equivalent of R_b binary digits per second over the M-ary PCM channel. The reception is not error free, however. The pulses are detected with an error probability P_{eM} given in Eq. (10.99c). If P_{eM} is on the order of 10^{-6}, we could consider the reception to be essentially error free. From Eq. (10.99c),

$$P_{eM} \simeq 2Q\left(\sqrt{\frac{2E_p}{\mathcal{N}}}\right) = 10^{-6} \qquad M \gg 1$$

This gives

$$\frac{2E_p}{\mathcal{N}} = 24$$

Substitution of this value in Eq. (13.74) gives

$$R_b = B_T \log_2 \left(1 + \frac{1}{8} \frac{S_i}{N_i}\right) \quad \text{bit/s} \tag{13.75}$$

Thus, over a channel of bandwidth B_T with an SNR of S_i/N_i, a PCM system can transmit information at a rate of R_b in Eq. (13.75). The ideal channel with bandwidth B_T and SNR S_i/N_i transmits information at a rate of C bit/s, where

$$C = B_T \log_2 \left(1 + \frac{S_i}{N_i}\right) \quad \text{bit/s} \tag{13.76}$$

It follows that PCM uses roughly eight times (9 dB) as much power as the ideal system. This performance is still much superior to that of FM. Figure 13.11 shows R_b/B_T as a function of S_i/N_i. For the ideal system, we have

$$\frac{R_b}{B_T} = \frac{C}{B_T} = \log_2 \left(1 + \frac{S_i}{N_i}\right) \tag{13.77}$$

PCM at the threshold is 9 dB inferior to the ideal curve.

When PCM is in saturation, the detection error probability approaches 0. Each M-ary pulse transmits $\log_2 M$ bits, and there are $2B_T$ pulses per second. Hence,

$$R_b = 2B_T \log_2 M \tag{13.78}$$

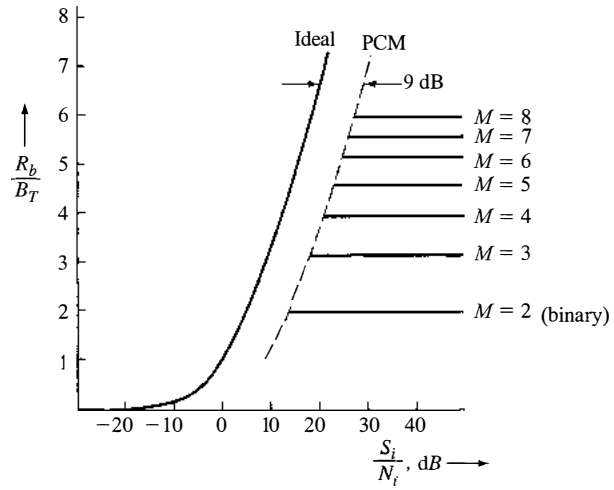


Figure 13.11 Comparison of ideal system behavior to that of PCM.

or

$$\frac{R_b}{B_T} = 2\log_2 M \tag{13.79}$$

This is clearly seen in Fig. 13.11 (solid horizontal lines).

Orthogonal Signaling

We have already shown that [Eq. (10.122)] for M-ary orthogonal signaling, the error-free communication rate is

$$R_b \leq 1.44\frac{S_i}{\mathcal{N}} \quad \text{bit/s} \tag{13.80}$$

We showed in Eq. (13.64) that this is precisely the rate of error-free communication over an ideal channel with infinite bandwidth. Therefore, as $M \to \infty$, the bandwidth of an M-ary scheme also approaches infinity, and its rate of communication approaches that of an ideal channel.

13.7 FREQUENCY-SELECTIVE CHANNEL CAPACITY

Thus far, we have limited the discussion of capacity to distortionless channels of finite bandwidth under white Gaussian noise. Such a channel model is suitable for application when channels are either flat or flat fading. In reality, we often face many types of complex channels. In particular, we have shown, in Chapter 12, that most wireless communication channels in the presence of significant multipath tend to be frequency-selective channels. We now take a look at the capacity of frequency-selective channels that do not exhibit a distortionless (flat) spectrum.

First, consider a band-limited AWGN channel whose random output is

$$\mathrm{y} = H \cdot \mathrm{x} + \mathrm{n}$$

This channel has a constant gain of H across the bandwidth. Based on Eq. (13.63) this band-limited (low-pass) AWGN channel with bandwidth B has capacity

$$C = B \cdot \log\left(1 + |H|^2 \frac{S}{N}\right) \quad \text{bit/s} \tag{13.81}$$

in which S and N are the signal power and the noise power, respectively. Furthermore, in Chapter 4 and Chapter 9, we have demonstrated the equivalence of baseband and passband channels through modulation. Therefore, given the same noise spectrum and bandwidth, AWGN low-pass, band-limited channels and AWGN bandpass channels possess identical channel capacity. We are now ready to describe the capacity of frequency-selective channels.

Consider a bandpass channel of infinitesimal bandwidth Δf centered at a frequency f_i. Within this small band, the channel gain is $H(f_i)$, the signal power spectral density (PSD) is $S_x(f_i)$, and the Gaussian noise PSD is $S_n(f_i)$. Since this small bandwidth is basically a band limited AWGN channel, according to Eq. (13.63), its capacity is

$$\begin{aligned} C(f_i) &= \Delta f \cdot \log\left[1 + |H(f_i)|^2 \frac{S_x(f_i)\Delta f}{S_n(f_i)\Delta f}\right] \\ &= \log\left[1 + \frac{|H(f_i)|^2 S_x(f_i)}{S_n(f_i)}\right] \Delta f \quad \text{bit/s} \end{aligned} \tag{13.82}$$

This means that we can divide a frequency-selective channel $H(f)$ into small disjoint AWGN bandpass channels of bandwidth Δf. Thus, the sum channel capacity is simply approximated by

$$\hat{C} = \sum_i \log\left[1 + \frac{|H(f_i)|^2 S_x(f_i)}{S_n(f_i)}\right] \Delta f \quad \text{bit/s}$$

In fact, the practical OFDM (or DMT) system discussed in Chapter 12 is precisely such a system, which consists of a bank of parallel flat channels with different gains. This capacity is an approximation because the channel response, the signal PSD, or the noise PSD, may not be constant over a nonzero Δf. By taking $\Delta f \to 0$, we can determine the total channel capacity as

$$C = \int_{-\infty}^{\infty} \log\left[1 + \frac{|H(f)|^2 S_x(f)}{S_n(f)}\right] df \tag{13.83}$$

Maximum Capacity Power Loading

In Eq. (13.83), we have established that the capacity of a frequency-selective channel with response $H(f)$ under colored Gaussian noise of power spectral density (PSD) $S_n(f)$ depends on the input PSD $S_x(f)$. For the transmitter to utilize the full channel capacity, we now need to find the optimum input power spectral density (PSD) $S_x(f)$ that can further maximize the integral capacity

$$\int_{-\infty}^{\infty} \log\left[1 + \frac{|H(f)|^2 S_x(f)}{S_n(f)}\right] df$$

To do so, we have noted that it would not be fair to consider arbitrary input PSD $S_x(f)$ because different power spectral densities may lead to different values of total input power. Given two

signals of the same PSD shape, the stronger signal with larger power has an unfair advantage and costs more to transmit. Thus, a fair approach to channel capacity maximization should limit the total input signal power to a transmitter power constraint P_x. Finding the best input PSD under the total power constraint is known as the problem of **maximum capacity power loading**.

The PSD that achieves the maximum capacity power loading is the solution to the optimization problem of:

$$\max_{S_x(f)} \int_{-\infty}^{\infty} \log\left(1 + \frac{|H(f)|^2 S_x(f)}{S_n(f)}\right) df \tag{13.84}$$

$$\text{subject to} \quad \int_{-\infty}^{\infty} S_x(f)\, df \le P_x$$

To solve this optimization problem, we again partition the channel (of bandwidth B) into K narrow flat channels centered at $\{f_i, i = 1, 2, \ldots, K\}$ of bandwidth $\Delta f = B/K$. By denoting

$$N_i = S_n(f_i)\,\Delta f$$
$$H_i = H(f_i)$$
$$S_i = S_x(f_i)\,\Delta f$$

the optimization problem becomes a discrete problem of

$$\max_{\{S_i\}} \sum_{i=1}^{K} \log\left(1 + \frac{|H_i|^2 S_i}{N_i}\right) \Delta f \tag{13.85a}$$

$$\text{subject to} \quad \sum_{i=1}^{K} S_i = P \tag{13.85b}$$

The problem of finding the N optimum power values $\{S_i\}$ is the essence of the optimum power loading problem.

This problem can be dealt by introducing a standard Lagrange multiplier λ to form a modified objective function

$$G(S_1, S_2, \ldots, S_K) = \sum_{i=1}^{K} \log\left(1 + \frac{|H_i|^2 S_i}{N_i}\right) \Delta f + \lambda \cdot \left(P - \sum_{i=1}^{K} S_i\right) \tag{13.86}$$

Taking a partial derivative of $G(S_1, \ldots, S_K)$ with respect to S_j and setting it to zero, we have

$$\frac{\Delta f}{1 + |H_j|^2 S_j / N_j} \frac{|H_j|^2}{N_j} - \lambda \ln 2 = 0 \qquad j = 1, 2, \ldots, K$$

We rewrite this optimality condition into

$$\frac{\Delta f}{\lambda \ln 2} - \frac{N_j}{|H_j|^2} = S_j \qquad j = 1, 2, \ldots, K$$

By defining a new variable $W = (\lambda \ln 2)^{-1}$, we ensure that the optimum power allocation among the K subchannels is

$$S_i = W \cdot \Delta f - \frac{N_i}{|H_i|^2} \qquad i = 1, 2, \ldots, K \tag{13.87a}$$

$$\text{such that} \qquad \sum S_i = P \tag{13.87b}$$

The optimum power loading condition of Eq. (13.87) is not quite yet complete because some S_i may become negative if no special care is taken. Therefore, we must further constrain the solution to ensure that $S_i \geq 0$ via

$$S_i = \max\left(W \cdot \Delta f - \frac{N_i}{|H_i|^2},\ 0\right) \qquad i = 1, 2, \ldots, K \tag{13.88a}$$

$$\text{such that} \qquad \sum S_i = P \tag{13.88b}$$

The two relationships in Eq. (13.88) describe the solution of the power loading optimization problem. We should note that there remains an unknown parameter W that needs to be specified. By enforcing the total power constraint $\sum S_i = P$, we can finally determine the unknown parameter W.

Finally, we take the limit as $\Delta f \to 0$ and $K \to \infty$. Since $S_i = S_x(f_i)\Delta f$ and $N_i = S_n(f_i)\Delta f$, the optimum input signal PSD becomes

$$S_x(f) = \max\left(W - \frac{S_n(f)}{|H(f)|^2},\ 0\right) \tag{13.89a}$$

We note again that there is no closed-form solution given for the optimum constant W. Instead, the optimum W is obtained from the total input power constraint

$$\int_{-\infty}^{\infty} S_x(f)\, df = P \tag{13.89b}$$

or

$$P = \int_{\{f:\, W - S_n(f)/|H(f)|^2 > 0\}} \left(W - \frac{S_n(f)}{|H(f)|^2}\right) df \tag{13.89c}$$

Substituting the optimum PSD Eq. (13.89) into the capacity formula will lead to the maximum channel capacity value of

$$C_{\max} = \int_{\{f:\, W - S_n(f)/|H(f)|^2 > 0\}} \log\left(\frac{W|H(f)|^2}{S_n(f)}\right) df \tag{13.90}$$

Water-Pouring Interpretation of Optimum Power Loading

The optimum channel input PSD must satisfy the power constraint Eq. (13.89c). Once the constant W has been determined, the transmitter can adjust its transmission PSD to Eq. (13.89a), which will maximize the channel capacity. This optimum solution to the channel input PSD optimization problem is known as the water-filling or water-pouring solution.[5]

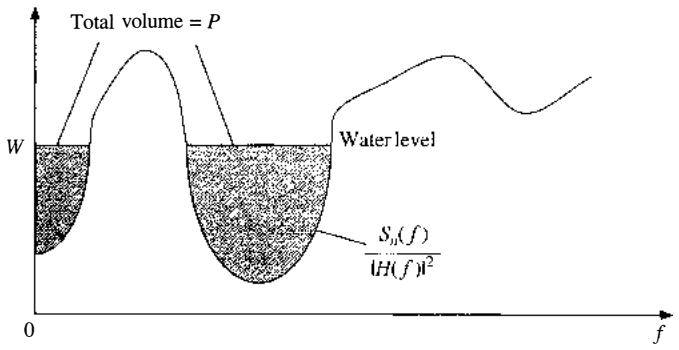


Figure 13.12 Illustration of water-pouring power allocation for maximizing frequency-selective channel capacity.

The literal water-pouring interpretation of optimum PSD design is illustrated by Fig. 13.12. First, plot the frequency response $S_n(f)/|H(f)|^2$. This curve is viewed as shaped like the bottom of a water container. Consider the total power as a bucket of water with total volume P. We can then pour the entire bucket of water into the container to achieve equal water level. The final water level will be raised to W when the bucket is empty. The depth of the water for every frequency f is the desired optimum PSD level $S_x(f)$ as specified in Eq. (13.89a). Clearly, when the noise PSD is large such that $S_n(f)/|H(f)|^2$ is high for some f, then there may be zero water poured at those points. In other words, the optimum PSD for these frequencies will be zero. Notice that a high value of $S_n(f)/|H(f)|^2$ means a low value of channel SNR $|H(f)|^2/S_n(f)$. Conversely, when $S_n(f)/|H(f)|^2$ is low or SNR is high, the optimum PSD value $S_x(f)$ should be kept high. In short, water-pouring power loading allocates more signal power to frequencies at which the channel SNR $|H(f)|^2/S_n(f)$ is high and allocates little or no signal power to frequencies at which the channel SNR $|H(f)|^2/S_n(f)$ is low.

This solution is similar, but not the same with the transmitter power loading for maximum receiver SNR in the DMT system discussed in Sec. 12.8.

Optimum Power Loading in OFDM/DMT

As the water-filling illustration shows, it is impossible to find a closed-form expression of W. Once P has been specified, an iterative water-filling algorithm can be used to eventually determine W and hence the optimum power loading PSD $S_x(f)$. Of course, the approach in practice to determine the water level W is by numerically solving for W. The numerical solution requires dividing the entire channel bandwidth into sufficiently small nonoverlapping bands of width Δf.

Indeed, for practical OFDM or DMT communication systems, the iterative water-filling algorithm is tailor-made to achieve maximum channel capacity. Maximum capacity can be realized for OFDM channels by allocating different powers S_i to the different orthogonal subcarriers. In particular, the power allocated to subcarrier f_i should be

$$S_i = \Delta f \cdot \max\left(W - \frac{S_n(f_i)}{|H(f_i)|^2},\ 0\right)$$

such that $\sum S_i = P$. This optimum power allocation or power loading can be solved by adding incremental power to the subcarriers one at a time until $\sum S_i = P$.

13.8 MULTIPLE-INPUT–MULTIPLE-OUTPUT COMMUNICATION SYSTEMS

In the past decade, one of the important breakthroughs in wireless communications is the advent of multiple-input–multiple-output (MIMO) technologies. In fact, both the Wi-Fi (IEEE 802.11n) standard and the WiMAX (IEEE 802.16e) standard have incorporated MIMO transmitters and receivers (or transceivers). The key advantage of MIMO wireless communication systems lies in their ability to significantly increase wireless channel capacity without either requiring additional bandwidth or substantially increasing the signal power at the transmitter. Interestingly, the MIMO development originates from the fundamentals of information theory. We shall explain this connection here.

13.8.1 Capacity of MIMO Channels

Whereas earlier only a single signal variable was considered for transmission, we now deal with input and output signal vectors. In other words, each signal vector consists of multiple data symbols to be transmitted or received concurrently in MIMO systems. Consider a random signal vector $\mathbf{x} = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_N)^T$. If the random signal vector is discrete with probabilities

$$p_i = P(\mathbf{x} = \boldsymbol{x}_i) \qquad i = 1, 2, \ldots$$

then the entropy of $\mathbf{x}$ is determined by

$$H(\mathbf{x}) = -\sum_i p_i \log p_i \tag{13.91}$$

Similarly, when $\mathbf{x}$ is continuously distributed with probability density function $p(x_1, x_2, \ldots, x_N)$, its differential entropy is defined by

$$H(\boldsymbol{x}) = -\int \cdots \int p(x_1, x_2, \ldots, x_N) \log p(x_1, x_2, \ldots, x_N)\, dx_1, dx_2, \ldots, dx_N \tag{13.92}$$

Consider a real-valued random vector $\mathbf{x}$ consisting of N i.i.d. Gaussian random variables. Let $\mathbf{x}$ have (vector) mean $\boldsymbol{\mu}$ and covariance matrix

$$\mathbf{C}_\mathbf{x} = E\{(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T\}$$

Its differential entropy can be found[5] to be

$$H(\boldsymbol{x}) = \frac{1}{2}[N \cdot \log(2\pi e) + \log\det(\boldsymbol{C}_\mathbf{x})] \tag{13.93}$$

Clearly, the entropy of a random vector is not affected by the mean $\boldsymbol{\mu}$. It is therefore convenient to consider only the random vectors with zero mean. From now on, we will assume that

$$\boldsymbol{\mu} = E\{\mathbf{x}\} = 0$$

Among all the real-valued random variable vectors that have zero mean and satisfy the condition

$$\boldsymbol{C}_{\mathbf{x}} = \text{Cov}(\mathbf{x}, \mathbf{x}) = E\{\mathbf{x}\mathbf{x}^T\}$$

we have[11]

$$\max_{p_{\mathbf{x}}(\boldsymbol{x}):\ \text{Cov}(\mathbf{x}, \mathbf{x}^T)=\boldsymbol{C}_{\mathbf{x}}} H(\mathbf{x}) = \frac{1}{2}\left[N \cdot \log(2\pi e) + \log\det(\boldsymbol{C}_{\mathbf{x}})\right]. \tag{13.94}$$

This means that Gaussian vector distribution has maximum entropy among all real random vectors of the same covariance matrix.

Now consider a flat fading MIMO channel with matrix gain $\boldsymbol{H}$. The $N \times M$ channel matrix $\boldsymbol{H}$ connects the $M \times 1$ input vector $\mathbf{x}$ and $N \times 1$ output vector $\mathbf{y}$ such that

$$\mathbf{y} = \boldsymbol{H} \cdot \mathbf{x} + \mathbf{w} \tag{13.95}$$

where $\mathbf{w}$ is the $N \times 1$ additive white Gaussian noise vector with zero mean and covariance matrix $\boldsymbol{C}_{\mathbf{w}}$. As shown in Fig. 13.13, a MIMO system consists of M transmit antennas at the transmitter end and N receive antennas at the receiver end. Each transmit antenna can transmit to all N receive antennas. Given a fixed channel $\boldsymbol{H}$ of dimensions $N \times M$ (i.e., M transmit antennas and N receive antennas), the mutual information between the channel input and output vectors is

$$I(\mathbf{x}, \mathbf{y}) = H(\mathbf{y}) - H(\mathbf{y}|\mathbf{x}) \tag{13.96a}$$

$$= H(\mathbf{y}) - H(\mathbf{H} \cdot \mathbf{x} + \mathbf{w}|\mathbf{x}) \tag{13.96b}$$

Recall that under the condition that $\mathbf{x}$ is known, $\mathbf{H} \cdot \mathbf{x}$ is a constant mean. Hence, the conditional entropy of $\mathbf{y}$ given $\mathbf{x}$ is

$$H(\mathbf{y}|\mathbf{x}) = H(\boldsymbol{H} \cdot \mathbf{x} + \mathbf{w}|\mathbf{x}) = H(\mathbf{w}) \tag{13.97}$$

and

$$I(\mathbf{x}, \mathbf{y}) = H(\mathbf{y}) - H(\mathbf{w}) \tag{13.98a}$$

$$= H(\mathbf{y}) - \frac{1}{2}\left[N \cdot \log_2(2\pi e) + \log\det(\mathbf{C}_{\mathbf{w}})\right] \tag{13.98b}$$

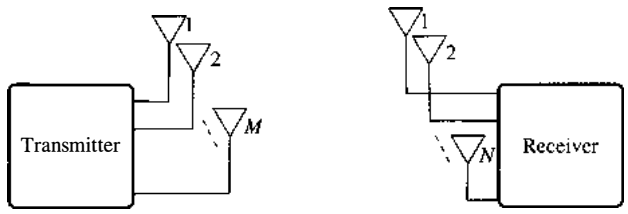


Figure 13.13 MIMO system with M transmit antennas and N receive antennas.

As a result, we can use the result of Eq. (13.94) to obtain

$$\max I(\mathbf{x}, \mathbf{y}) = \max H(\mathbf{y}) - \frac{1}{2}\left[N \cdot \log_2(2\pi e) + \log\det(\mathbf{C}_\mathbf{w})\right] \tag{13.99a}$$

$$= \frac{1}{2}\left[N \cdot \log_2(2\pi e) + \log\det(\boldsymbol{C}_\mathbf{y})\right] - \frac{1}{2}\left[N \cdot \log_2(2\pi e) + \log\det(\boldsymbol{C}_\mathbf{w})\right] \tag{13.99b}$$

$$= \frac{1}{2}\left[\log\det(\boldsymbol{C}_\mathbf{y}) - \log\det(\boldsymbol{C}_\mathbf{w})\right]$$

$$= \frac{1}{2}\left[\log\det(\boldsymbol{C}_\mathbf{y} \cdot \boldsymbol{C}_\mathbf{w}^{-1})\right] \tag{13.99c}$$

Since the channel input $\mathbf{x}$ is independent of the noise vector $\mathbf{w}$, we have

$$\boldsymbol{C}_\mathbf{y} = \mathrm{Cov}(\mathbf{y}, \mathbf{y}) = \boldsymbol{H} \cdot \boldsymbol{C}_\mathbf{x}\boldsymbol{H}^T + \boldsymbol{C}_\mathbf{w}$$

Thus, the capacity of the channel per vector transmission is

$$C_s = \max_{p(\boldsymbol{x})} I(\mathbf{x}, \mathbf{y})$$

$$= \frac{1}{2}\log\det\left(\boldsymbol{I}_N + \boldsymbol{H}\boldsymbol{C}_\mathbf{x}\boldsymbol{H}^T\boldsymbol{C}_\mathbf{w}^{-1}\right) \tag{13.100}$$

Given a symmetric low-pass channel with B Hz bandwidth, $2B$ samples of $\mathbf{x}$ can be transmitted to yield provide channel capacity of

$$C(\boldsymbol{H}) = B\log\det\left(I + \boldsymbol{H}\boldsymbol{C}_\mathbf{x}\boldsymbol{H}^T\boldsymbol{C}_\mathbf{w}^{-1}\right)$$

$$= B\log\det\left(I + \boldsymbol{C}_\mathbf{x}\boldsymbol{H}^T\boldsymbol{C}_\mathbf{w}^{-1}\boldsymbol{H}\right) \tag{13.101}$$

where we have invoked the equality that for matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of appropriate dimensions, $\det(\boldsymbol{I} + \boldsymbol{A} \cdot \boldsymbol{B}) = \det(\boldsymbol{I} + \boldsymbol{B} \cdot \boldsymbol{A})$. We clearly can see from Eq. (13.101) that the channel capacity depends on the covariance matrix $C_\mathbf{x}$ of the Gaussian input signal vector. This result shows that, given the knowledge of the MIMO channel ($\boldsymbol{H}^T\boldsymbol{C}_\mathbf{w}^{-1}\boldsymbol{H}$) at the transmitter, an optimum input signal can be determined by designing $C_\mathbf{x}$ to maximize the overall channel capacity $C(\boldsymbol{H})$.

We now are left with two scenarios to consider: (1) MIMO transmitters without the MIMO channel knowledge and (2) MIMO transmitters with channel knowledge that allows $\boldsymbol{C}_\mathbf{x}$ to be optimized. We shall discuss the MIMO channel capacity in these two separate cases.

13.8.2 Transmitter without Channel Knowledge

For transmitters without channel knowledge, the input covariance matrix $\boldsymbol{C}_\mathbf{x}$ should be chosen without showing any preference. As a result, the default $\boldsymbol{C}_\mathbf{x} = \sigma_\mathbf{x}^2\boldsymbol{I}$ should be selected. In this case, the MIMO system capacity is simply

$$C = B\log\det\left(\boldsymbol{I} + \sigma_\mathbf{x}^2\boldsymbol{H}^T\boldsymbol{C}_\mathbf{w}^{-1}\boldsymbol{H}\right) \tag{13.102}$$

Consider the eigendecomposition of

$$\boldsymbol{H}^T\boldsymbol{C}_{\mathrm{w}}^{-1}\boldsymbol{H} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^H$$

where $\boldsymbol{U}$ is a $N \times N$ square unitary matrix such that $\boldsymbol{U} \cdot \boldsymbol{U}^H = \boldsymbol{I}_N$ and $\boldsymbol{D}$ is a diagonal matrix with nonnegative diagonal elements in descending order:

$$\boldsymbol{D} = \mathrm{Diag}\,(d_1, d_2, \ldots, d_r, 0, \cdot 0)$$

Notice that $d_r > 0$ is the smallest nonzero eigenvalue of $\boldsymbol{H}^T\boldsymbol{C}_{\mathrm{w}}^{-1}\boldsymbol{H}$ whose rank is bounded by $r \le \min(N, M)$. Because $\det(\boldsymbol{I} + \boldsymbol{A}\boldsymbol{B}) = \det(\boldsymbol{I} + \boldsymbol{B}\boldsymbol{A})$ and $\boldsymbol{U}^H\boldsymbol{U} = \boldsymbol{I}$, we have

$$C = B\log\det\left(\boldsymbol{I} + \sigma_{\mathrm{x}}^2 \cdot \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^H\right) \tag{13.103a}$$

$$= B\log\det\left(\boldsymbol{I} + \sigma_{\mathrm{x}}^2 \cdot \boldsymbol{D}\boldsymbol{U}^H\boldsymbol{U}\right)$$

$$= B\log\det\left(\boldsymbol{I} + \sigma_{\mathrm{x}}^2\boldsymbol{D}\right)$$

$$= B\log\prod_{i=1}^{r}(1 + \sigma_{\mathrm{x}}^2 d_i) \tag{13.103b}$$

$$= B\sum_{i=1}^{r}\log(1 + \sigma_{\mathrm{x}}^2 d_i) \tag{13.103c}$$

In the special case of channel noise that is additive, white, and Gaussian, then $\boldsymbol{C}_{\mathrm{w}} = \sigma_w^2\boldsymbol{I}$ and

$$\boldsymbol{H}^T\boldsymbol{C}_{\mathrm{w}}^{-1}\boldsymbol{H} = \frac{1}{\sigma_{\mathrm{w}}^2}\boldsymbol{H}^T\boldsymbol{H} = \frac{1}{\sigma_{\mathrm{w}}^2}\boldsymbol{U}\begin{bmatrix}\gamma_1 & & & & & & \\ & \gamma_2 & & & & & \\ & & \ddots & & & & \\ & & & \gamma_r & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0\end{bmatrix}\boldsymbol{U}^H \tag{13.104}$$

where γ_i is the ith largest eigenvalue of $\boldsymbol{H}^T\boldsymbol{H}$, which is assumed to have rank r. Consequently, the channel capacity for this MIMO system is

$$C = B\sum_{i=1}^{r}\log\left(1 + \frac{\sigma_{\mathrm{x}}^2}{\sigma_w^2}\gamma_i\right) \tag{13.105}$$

In short, this channel capacity is the sum of the capacity of r parallel AWGN channels. Each subchannel SNR is $\sigma_{\mathrm{x}}^2 \cdot \gamma_i/\sigma_w^2$. Figure 13.14 demonstrates the equivalent system that consists of r parallel AWGN channels with r active input signals $\mathrm{x}_1, \ldots, \mathrm{x}_r$.

In the special case when the MIMO channel is so well conditioned that all its nonzero eigenvalues are identical $\gamma_i = \gamma$, the channel capacity is

$$C_{\mathrm{MIMO}} = r \cdot B\log\left(1 + \frac{\sigma_{\mathrm{x}}^2}{\sigma_w^2}\gamma\right) \tag{13.106}$$

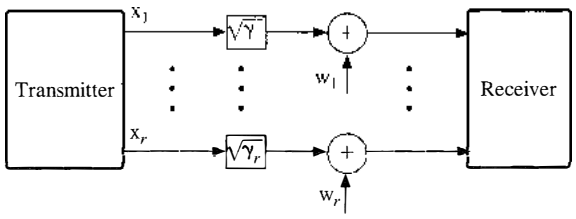


Figure 13.14 r-Channel communication system equivalent to a MIMO system without channel knowledge at the transmitter.

Compared with the single-input–single-output channel for which $\boldsymbol{H}$ is a scalar such that $r = 1$, the SISO channel capacity is simply

$$C_{\text{SISO}} = B \log \left(1 + \frac{\sigma_x^2}{\sigma_w^2}\gamma\right) \tag{13.107}$$

Therefore, by applying MIMO transceivers, the channel capacity is increased to r times the capacity of the original single-input–single-output channel. This result strongly demonstrates the significant advantages of MIMO technology in providing much-needed capacity improvement for wireless communications.

13.8.3 Transmitter with Channel Knowledge

In a number of wireless communication systems, the transmitter may acquire the knowledge of the MIMO channel $\boldsymbol{H}^T\boldsymbol{C}_{\mathbf{w}}^{-1}\boldsymbol{H}$ through a feedback mechanism. In this case, the transmitter can optimize the input signal covariance matrix $\boldsymbol{C}_{\mathbf{x}}$ to maximize the MIMO system capacity.[12]

First, we observe that the channel capacity of Eq. (13.101) can be increased simply by scaling the matrix $\boldsymbol{C}_{\mathbf{x}}$ with a large constant k. Of course, doing so would be effectively increasing the transmission power k times and would be unfair. This means that to be fair, the design of optimum covariance matrix $\boldsymbol{C}_{\mathbf{x}}$ must be based on some practical constraint. In a typical communication system, we know that a transmitter with higher signal power will lead to higher SNR and, hence, larger capacity. Therefore, similar to the water-pouring PSD design for frequency-selective channels, we should constrain the total transmission power of the MIMO transmitter by the transmitter power threshold P.

To show how this power constraint would affect the input covariance matrix $\boldsymbol{C}_{\mathbf{x}}$, we first need to introduce the "trace" (Tr) operator of square matrices. Consider an $M \times M$ square matrix $\boldsymbol{F}$ whose element on the ith row and the jth column is denoted by $F_{i,j}$. Then the trace of the matrix $\boldsymbol{F}$ is the sum of its diagonal elements

$$\text{Tr}\,(\boldsymbol{F}) = \sum_{i=1}^{M} F_{i,i} \tag{13.108}$$

Since the trace operator is linear, it follows from the property of the expectation operator $E\{\cdot\}$ [Eq. (8.59)] that

$$E\{\text{Tr}\,(\boldsymbol{F})\} = \text{Tr}\,(E\{\boldsymbol{F}\}) \tag{13.109}$$

We now introduce a very useful property of the trace operator. If matrix products $\boldsymbol{AB}$ and $\boldsymbol{BA}$ are both square matrices of appropriate sizes, then they both have the same trace, that is,

$$\text{Tr}\,(\boldsymbol{AB}) = \text{Tr}\,(\boldsymbol{BA}) \tag{13.110}$$

This equality turns out to be very important. By applying Eq. (13.110), we know that for vector $\mathbf{x}$

$$\mathbf{x}^T\mathbf{x} = \mathrm{Tr}\left[\mathbf{x}^T\mathbf{x}\right] \tag{13.111a}$$

$$= \mathrm{Tr}\left[\mathbf{x}\mathbf{x}^T\right] \tag{13.111b}$$

For the signal vector $\mathbf{x} = (x_1, x_2, \ldots, x_M)$, we can apply Eqs. (13.109) and (13.111) to show that the average sum power of the signal vector $\mathbf{x}$ is

$$\sum_{i=1}^{M} E\left\{x_i^2\right\} = E\left\{\sum_{i=1}^{M} x_i^2\right\} \tag{13.112a}$$

$$= E\left\{\mathbf{x}^T\mathbf{x}\right\}$$

$$= E\left\{\mathrm{Tr}\left[\mathbf{x}\mathbf{x}^T\right]\right\}$$

$$= \mathrm{Tr}\left[E\left\{\mathbf{x}\mathbf{x}^T\right\}\right]$$

$$= \mathrm{Tr}\left[\boldsymbol{C}_{\mathbf{x}}\right] \tag{13.112b}$$

As a result, we have established that the power constraint translates into the trace constraint

$$\mathrm{Tr}\left(\boldsymbol{C}_{\mathbf{x}}\right) \leq P$$

Therefore, given the knowledge of $\boldsymbol{H}^T\boldsymbol{C}_{\mathbf{w}}^{-1}\boldsymbol{H}$ at the transmitter, the optimum input signal covariance matrix to maximize the channel capacity is defined by

$$\max_{\boldsymbol{C}_{\mathbf{x}}:\ \mathrm{Tr}(\boldsymbol{C}_{\mathbf{x}})\leq P} B \log\det\left(\boldsymbol{I} + \boldsymbol{C}_{\mathbf{x}}\boldsymbol{H}^T\boldsymbol{C}_{\mathbf{w}}^{-1}\boldsymbol{H}\right) \tag{13.113}$$

This optimization problem is henceforth well defined.

To find the optimum $\boldsymbol{C}_{\mathbf{x}}$, recall the eigendecomposition

$$\boldsymbol{H}^T\boldsymbol{C}_{\mathbf{w}}^{-1}\boldsymbol{H} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^H$$

By applying the trace property of Eq. (13.110), we can rewrite the optimum covariance design problem into

$$\max_{\mathrm{Tr}(\boldsymbol{C}_{\mathbf{x}})\leq P} B \log\det\left(\boldsymbol{I} + \boldsymbol{C}_{\mathbf{x}}\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^H\right) = \max_{\mathrm{Tr}(\boldsymbol{C}_{\mathbf{x}})\leq P} B \log\det\left(\boldsymbol{I} + \boldsymbol{U}^H\boldsymbol{C}_{\mathbf{x}}\boldsymbol{U}\boldsymbol{D}\right) \tag{13.114}$$

Because covariance matrices are positive semidefinite (Appendix D.7), we can define a new positive semidefinite matrix

$$\overline{\boldsymbol{C}}_{\mathbf{x}} = \boldsymbol{U}^H\boldsymbol{C}_{\mathbf{x}}\boldsymbol{U} \tag{13.115}$$

According to Eq. (13.110), we know that

$$\begin{aligned}\mathrm{Tr}\left[\overline{C}_{\mathbf{x}}\right] &= \mathrm{Tr}\left[\boldsymbol{U}^H \boldsymbol{C}_{\mathbf{x}} \boldsymbol{U}\right] \\ &= \mathrm{Tr}\left[\boldsymbol{C}_{\mathbf{x}} \boldsymbol{U}\boldsymbol{U}^H\right] \\ &= \mathrm{Tr}\left[\boldsymbol{C}_{\mathbf{x}} \boldsymbol{I}\right] \\ &= \mathrm{Tr}\left[\boldsymbol{C}_{\mathbf{x}}\right] \end{aligned} \tag{13.116}$$

In fact, Eq. (13.116) states that the traces of $\boldsymbol{C}_{\mathbf{x}}$ and $\overline{C}_{\mathbf{x}}$ are identical. This equality allows us to simplify the capacity maximization problem into

$$\begin{aligned} C &= \max_{\boldsymbol{C}_{\mathbf{x}}:\, \mathrm{Tr}(\boldsymbol{C}_{\mathbf{x}}) \le P} B \log \det\left(\boldsymbol{I} + \boldsymbol{U}^H \boldsymbol{C}_{\mathbf{x}} \boldsymbol{U} \boldsymbol{D}\right) \\ &= \max_{\overline{C}_{\mathbf{x}}:\, \mathrm{Tr}(\overline{C}_{\mathbf{x}}) \le P} B \log \det\left(\boldsymbol{I} + \overline{C}_{\mathbf{x}} \boldsymbol{D}\right) \qquad (13.117a) \\ &= \max_{\overline{C}_{\mathbf{x}}:\, \mathrm{Tr}(\overline{C}_{\mathbf{x}}) \le P} B \log \det\left(\boldsymbol{I} + \boldsymbol{D}^{1/2} \overline{C}_{\mathbf{x}} \boldsymbol{D}^{1/2}\right) \qquad (13.117b) \end{aligned}$$

The problem of Eq. (13.117b) is simpler because $\boldsymbol{D}$ is a diagonal matrix. Furthermore, we can invoke the help of a very useful tool often used in matrix optimization known as the **Hadamard inequality**.

Hadamard Inequality: *Let a_{ij} be the element of complex $n \times n$ matrix A on the ith row and the jth column. A is positive semidefinite and Hermitian, that is, $(conj(A))^T = A$. Then the following inequality holds:*

$$det\,(A) \le \prod_{i=1}^{n} a_{ii}$$

with equality if and only if A is diagonal.

We can easily verify that $\boldsymbol{I} + \boldsymbol{D}^{1/2} \overline{C}_{\mathbf{x}} \boldsymbol{D}^{1/2}$ is positive semidefinite because $\overline{C}_{\mathbf{x}}$ is positive semidefinite (Prob. 13.8-3). By invoking Hadamard inequality in Eq. (13.117b), it is clear that, for maximum channel capacity we need

$$\boldsymbol{D}^{1/2} \overline{C}_{\mathbf{x}} \boldsymbol{D}^{1/2} = \text{diagonal}$$

In other words, the optimum channel input requires that

$$\overline{C}_{\mathbf{x}} = \boldsymbol{D}^{-1/2} \cdot \text{diagonal} \cdot \boldsymbol{D}^{-1/2} = \text{diagonal} \tag{13.118}$$

Equation (13.118) establishes that the optimum structure of $\overline{C}_{\mathbf{x}}$ is diagonal. This result greatly simplifies the capacity maximization problem. Denote the optimum structure covariance matrix as

$$\overline{C}_{\mathbf{x}} = \text{diagonal}\,(c_1,\ c_2,\ \ldots,\ c_M)$$

Then the capacity is maximized by a positive semidefinite matrix $\overline{C}_{\mathbf{x}}$ according to

$$C = \max_{\overline{C}_{\mathbf{x}}:\, \mathrm{Tr}(\overline{C}_{\mathbf{x}}) \le P} B \log \det \left(I + \boldsymbol{D}^{1/2} \overline{C}_{\mathbf{x}} \boldsymbol{D}^{1/2} \right) \tag{13.119a}$$

$$= \max_{\sum_{i=1}^{M} c_i \le P,\, c_i \ge 0} B \sum_{i=1}^{M} \log (1 + c_i d_i) \tag{13.119b}$$

In other words, our job is to find the optimum positive elements $\{c_i\}$ to maximize Eq. (13.119b) subject to the constraint $\sum_i c_i \le P$.

Taking the Lagrangian approach, we define a modified objective function

$$g(c_1, c_2, \ldots, c_M) = B \sum_{i=1}^{M} \log (1 + c_i d_i) + \lambda \left(P - \sum_{i=1}^{M} c_i \right) \tag{13.120}$$

Taking derivative of the modified objective function with respect to c_j ($j = 1, 2, \ldots, M$) and setting them to zero, we have

$$B \frac{\log e \cdot d_j}{1 + c_j d_j} - \lambda = 0 \qquad j = 1, 2, \ldots, M$$

or

$$c_j = \left[\frac{B}{\lambda \ln 2} - \frac{1}{d_j} \right] \qquad j = 1, 2, \ldots, M$$

The optimum diagonal elements $\{c_i\}$ are subject to the constraints

$$\sum_{i=1}^{M} c_i = P$$

$$c_j \ge 0 \qquad j = 1, \ldots, M$$

Similar to the problem of colored Gaussian noise channel power loading, we can define a water level $W = B/(\lambda \ln 2)$. By applying the same iterative water-pouring procedure, we can find the optimum power loading (on each eigenvector) to be

$$c_i = \max \left(W - \frac{1}{d_i}, 0 \right) \qquad i = 1, 2, \ldots, M \tag{13.121a}$$

with the total power constraint that

$$\sum_{i=1}^{M} c_i = P \tag{13.121b}$$

The water-filling interpretation of the optimum power loading at a MIMO transmitter given channel knowledge can be illustrated (Fig. 13.15).

The optimum input signal covariance matrix is therefore determined by

$$\boldsymbol{C}_{\mathbf{x}} = \boldsymbol{U} \cdot \mathrm{Diag}\,(c_1, c_2, \ldots, c_m, 0, \ldots, 0) \cdot \boldsymbol{U}^H$$

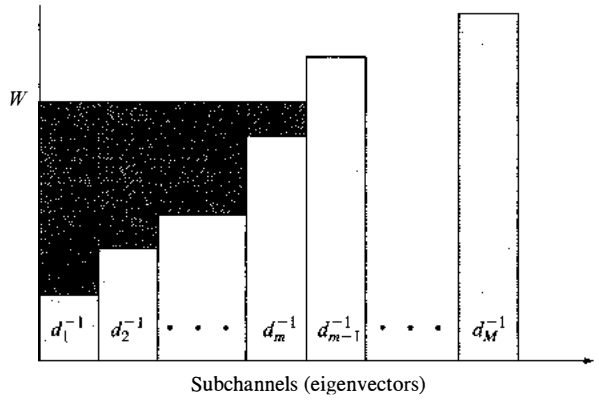


Figure 13.15 Water-filling interpretation of MIMO transmission power loading based on channel knowledge.

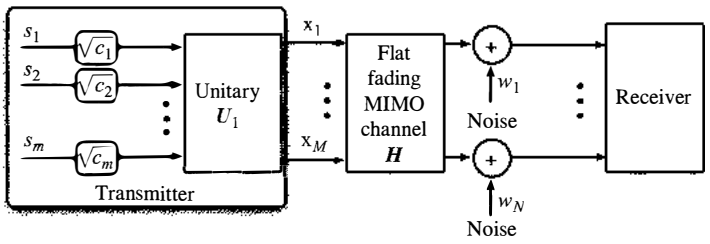


Figure 13.16 Water-pouring interpretation of the optimum MIMO transmission power loading based on channel knowledge.

In other words, the input signal vector can be formed by a unitary transformation $\boldsymbol{U}$ after we have found c_i based on water pouring. In effect, c_i is the amount of power loaded on the ith column of $\boldsymbol{U}$, that is, the ith eigenvector of $\boldsymbol{H}^T\boldsymbol{C}_{\mathbf{w}}^{-1}\boldsymbol{H}$.

Suppose we would like to transmit m independent signal streams $\{s_1, s_2, \ldots, s_m\}$ of zero mean and unit variance. Then the optimum MIMO channel input can be formed via

$$\mathbf{x} = \boldsymbol{U}_1 \operatorname{diag}(\sqrt{c_1}, \sqrt{c_2}, \ldots, \sqrt{c_m}) \cdot \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{bmatrix} \tag{13.122}$$

where $\boldsymbol{U}_1$ are the first m columns of $\boldsymbol{U}$. Figure 13.16 is the block diagram of this optimum MIMO transmitter, which will maximize channel capacity based on knowledge of the MIMO channel. The matrix multiplier $\boldsymbol{U}_1 \operatorname{diag}(\sqrt{c_1}, \sqrt{c_2}, \ldots, \sqrt{c_m})$ at the transmitter is known as the **optimum linear precoder**.

13.9 MATLAB EXERCISES

In this section, we provide MATLAB exercises to reinforce the concepts of source coding and channel capacity in this chapter.

COMPUTER EXERCISE 13.1: HUFFMAN CODE

The first program, `huffmancode.m`, is a Huffman encoder function. The user need only supply a probability vector that consists of all the source symbol probabilities. The probability entries do not need to be ordered.

```
function [huffcode,n]=huffmancode(p);
% input p is a probability vector consisting of
% probabilities of source symbols x_i
if min(p)<0,
    error('Negative element cannot be in a probability vector')
    return
else if abs(sum(p)-1)>1.e-12,
    error('Sum of input probability is not 1')
    return
end

[psort,pord]=sort(p);

n=length(p);
q=p;

for i=1:n-1
    [q,l]=sort(q);
    m(i,:)=[l(1:n-i+1),zeros(1,i-1)];
    q=[q(1)+q(2),q(3:end),1];
end

Cword=blanks(n^2);

Cword(n)='0';
Cword(2*n)='1';

for i1=1:n-2
    Ctemp=Cword;
    idx0=find(m(n-i1,:)==1)*n;
    Cword(1:n)=[Ctemp(idx0-n+2:idx0) '0'];
    Cword(n+1:2*n)=[Cword(1:n-1) '1'];
    for i2=2:i1+1
        idx2=find(m(n-i1,:)==i2);
        Cword(i2*n+1:(i2+1)*n)=Ctemp(n*(idx2-1)+1:n*idx2);
    end
end

for i=1:n
    idx1=find(m(1,:)==i);
    huffcode(i,1:n)=Cword(n*(idx1-1)+1:idx1*n);
end

end
```

The second program, `huffmanEx.m`, generates a very simple example of Huffman encoding. In this exercise, we provide an input probability vector of length 8. The MATLAB program `huffmanEx.m` will generate the list of codewords for all the input symbols. The entropy of this source $H(x)$ is computed and compared against the average Huffman codeword length. Their ratio shows the efficiency of the code.

```
% Matlab Program  <huffmanEx.m>
% This exercise requires the input of a
% probability vector p that list all the
```

```
% probabilities of each source input symbol
clear;
p=[0.2 0.05 0.03 0.1 0.3 0.02 0.22 0.08];   %Symbol probability vector
[huffcode,n]=huffmancode(p);                %Encode Huffman code
entropy=sum(-log(p)*p')/log(2);             %Find entropy of the source
% Display the results of Huffman encoder
display(['symbol',' --> ',' codeword','  Probability'])
for i=1:n
codeword_Length(i)=n-length(find(abs(huffcode(i,:))==32));
display(['x',num2str(i),'     --> ',huffcode(i,:),'     ',num2str(p(i))]);
end
codeword_Length
avg_length=codeword_Length*p';
display(['Entropy =  ', num2str(entropy)])
display(['Average codeword length =  ', num2str(avg_length)])
```

By executing the program `huffmanEx.m`, we can obtain the following results.

```
huffmanEx
symbol -->  codeword  Probability
x1     -->       00              0.2
x2     -->    10111    0.05
x3     -->   101101    0.03
x4     -->      100    0.1
x5     -->       11    0.3
x6     -->   101100    0.02
x7     -->       01    0.22
x8     -->     1010    0.08

codeword_Length =

     2     5     6     3     2     6     2     4

Entropy =  2.5705
Average codeword length =  2.61
```

COMPUTER EXERCISE 13.2: CHANNEL CAPACITY AND MUTUAL INFORMATION

This exercise provides an opportunity to compute the single-input–single-output channel capacity under additive white Gaussian noise.

MATLAB program `mutualinfo.m` contains a function that can compute the average mutual information between two data sequences x and y of equal length. We use a histogram to estimate the joint probability density function $p(x, y)$ before calculating the mutual information according to the definition of Eq. (13.45a).

```
function muinfo_bit=mutualinfo(x,y)
%mutualinfo Computes the mutual information of two
%           vectors x and y in bits
%   muinfo_bit = mutualinfo(X,Y)
%
%   output       : mutual information
%   X,Y          : The 1-D vectors to be analyzed
%
```

```
  minx=min(x);
  maxx=max(x);
  deltax=(maxx-minx)/(length(x)-1);
  lowerx=minx-deltax/2;
  upperx=maxx+deltax/2;
  ncellx=ceil(length(x)^(1/3));
  miny=min(y);
  maxy=max(y);
  deltay=(maxy-miny)/(length(y)-1);
  lowery=miny-deltay/2;
  uppery=maxy+deltay/2;
  ncelly=ncellx;

rout(1:ncellx,1:ncelly)=0;

xx=round( (x-lowerx)/(upperx-lowerx)*ncellx + 1/2 );
yy=round( (y-lowery)/(uppery-lowery)*ncelly + 1/2 );

for n=1:length(x)
  indexx=xx(n);
  indexy=yy(n);
  if indexx >= 1 & indexx <= ncellx & indexy >= 1 & indexy <= ncelly
    rout(indexx,indexy)=rout(indexx,indexy)+1;
  end;
end;

h=rout;

estimate=0;
sigma=0;
count=0;

% determine row and column sums

hy=sum(h);
hx=sum(h');

for nx=1:ncellx
  for ny=1:ncelly
    if h(nx,ny)~=0
      logf=log(h(nx,ny)/hx(nx)/hy(ny));
    else
      logf=0;
    end;
    count=count+h(nx,ny);
    estimate=estimate+h(nx,ny)*logf;
    sigma=sigma+h(nx,ny)*logf^2;
  end;
end;

% biased estimate

estimate=estimate/count;
sigma   =sqrt( (sigma/count-estimate^2)/(count-1) );
```

```
estimate=estimate+log(count);
nbias    =(ncellx-1)*(ncelly-1)/(2*count);

% remove bias
muinfo_bit=(estimate-nbias)/log(2);
```

In the main MATLAB program, `capacity_plot.m`, we calculate AWGN channel capacity for S/N ratio of 0, 5, 10, 15, and 20 dB. The channel capacity under different SNRs is plotted in Fig. 13.17. In addition, we can test the mutual information I(x, y) between the channel input x and the corresponding channel output y under the same SNR levels.

In this program, we estimate I(x, y) for five different zero-mean input signals of unit variance:

- Gaussian input
- Binary input of equal probability
- PAM-4 input of equal probability
- PAM-8 input of equal probability
- Uniform input in interval $[-\sqrt{3}, \sqrt{3}]$

The corresponding mutual information I(x, y) is estimated by averaging over 1,000,000 data samples.

```
% Matlab program <capacity_plot.m>
clear;clf;
Channel_gain=1;
H=Channel_gain; % AWGN Channel gain
SNRdb=0:5:20;       % SNR in dB
L=1000000;

SNR=10.^(SNRdb/10);
% Compute the analytical channel capacity
Capacity=1/2*log(1+H*SNR)/log(2);
% Now to estimate the mutual information between the input
% and the output signals of AWGN channels

for kk=1:length(SNRdb),
noise=randn(L,1)/sqrt(SNR(kk));
```

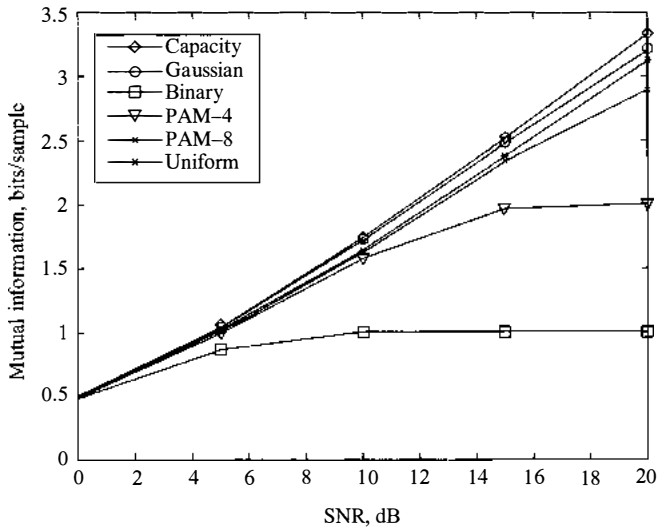


Figure 13.17 Channel capacity compared with mutual information between channel output and different input signals.

```
x=randn(L,1);
x1=sign(x);
x2=(floor(rand(L,1)*4-4.e-10)*2-3)/sqrt(5);
x3=(floor(rand(L,1)*8-4.e-10)*2-7)/sqrt(21);
x4=(rand(L,1)-0.5)*sqrt(12);

muinfovec(kk,1)=mutualinfo(x,x+noise);      %Gaussian input
muinfovec(kk,2)=mutualinfo(x1,x1+noise);    % Binary input (-1,+1)
muinfovec(kk,3)=mutualinfo(x2,x2+noise);    % 4-PAM input (-3,-1,1,3)
muinfovec(kk,4)=mutualinfo(x3,x3+noise);    % 8-PAM input (-7,-5,-3,-1,1,3,5
muinfovec(kk,5)=mutualinfo(x4,x4+noise);    % Uniform input(-0.5,0.5)
end
plot(SNRdb,Capacity,'k-d');hold on
plot(SNRdb,muinfovec(:,1),'k-o')
plot(SNRdb,muinfovec(:,2),'k-s')
plot(SNRdb,muinfovec(:,3),'k-v')
plot(SNRdb,muinfovec(:,4),'k-x')
plot(SNRdb,muinfovec(:,5),'k-*')
xlabel('SNR (dB)');ylabel('mutual information (bits/sample)')
legend('Capacity','Gaussian','binary','PAM-4','PAM-8',...
    'uniform','Location','NorthWest')
hold off
```

The estimated mutual information is plotted against the channel capacity under different SNR for the five different input distributions: (1) Gaussian; (2) binary (± 1); (3) 4-level PAM (or PAM-4); (4) 8-level PAM (or PAM-8); (5) uniform. All five symmetric distributions are scaled to have the same zero mean and unit power (variance). As shown in Figure 13.17, the mutual information achieved by Gaussian input closely matches the theoretical channel capacity. This result confirms the conclusion of Sec. 13.5 that Gaussian channel input achieves channel capacity. Fig. 13.17 shows that the mutual information for all other channel inputs falls below the mutual information achieved by the Gaussian input. Among the five different distributions, binary input achieves the least mutual information, whereas the mutual information of PAM-8 input is very close to the channel capacity for the SNR below 20 dB. This observation indicates that higher mutual information can be achieved when the distribution of the channel input is closer to Gaussian.

COMPUTER EXERCISE 13.3: MIMO CHANNEL CAPACITY

We show in this exercise how MIMO channel capacity varies for different numbers of transmit antennas and receive antennas. The MATLAB program `mimocap.m` will calculate the theoretical MIMO capacity of 200 random MIMO channels of different sizes at an SNR of 3 dB. We consider the case of a transmitter that does not have the MIMO channel knowledge. Hence, each transmit antenna is allocated the same signal power σ_x^2. Additionally, the channel noises are assumed to be independent additive white Gaussian with variance σ_w^2.

The entries in the MIMO channel matrix $\boldsymbol{H}$ are randomly generated from Gaussian distribution of zero mean and unit variance. Because the channels are random, for M transmit antennas and N receive antennas, the MIMO capacity per transmission is

$$C = \frac{1}{2}\log\left[\boldsymbol{I}_N + \frac{\sigma_x^2}{\sigma_w^2}\boldsymbol{H}\boldsymbol{H}^T\right]$$

Because the entries in the MIMO channel matrix $\boldsymbol{H}$ are randomly generated, its corresponding capacity is also random. From the 200 channels, each $N \times M$ MIMO configuration should generate 200 different capacity values.

```
% Matlab Program <mimocap.m>
% This program calculates the capacity of random MIMO (mxn) channels
% and plots the cumulative distribution (CDF) of the resulting
% capacity;
% Number of random channels:   K=200
% Signal to noise ratio:       SNRdb=3dB
clear
hold off
clf
K=200;
SNRdb=3;
SNR=10^(SNRdb/10);
m=1; n=1;   % 1x1 channels
for kk=1:K
    H=randn([m n]);     %Random MIMO Channel
    cap11(kk)=log(det(eye(n,n)+SNR*H'*H))/(2*log(2));
end
[N11,C11]=hist(cap11,K/10);     %CDF of MIMO capacity

m=2;n=2;    % 2x2 channels
for kk=1:K
    H=randn([m n]);     %Random MIMO Channel
    cap22(kk)=log(det(eye(n,n)+SNR*H'*H))/(2*log(2));
end
[N22,C22]=hist(cap22,K/10);     %CDF of MIMO capacity

m=4;n=2;    % 4x2 channels
for kk=1:K
    H=randn([m n]);     %Random MIMO Channel
    cap42(kk)=log(det(eye(n,n)+SNR*H'*H))/(2*log(2));
end
[N42,C42]=hist(cap42,K/10);     %CDF of MIMO capacity

m=2;n=4;    % 4x2 channels
for kk=1:K
    H=randn([m n]);     %Random MIMO Channel
    cap24(kk)=log(det(eye(n,n)+SNR*H'*H))/(2*log(2));
end
[N24,C24]=hist(cap24,K/10);     %CDF of MIMO capacity

m=4;n=4;    % 4x2 channels
for kk=1:K
    H=randn([m n]);     %Random MIMO Channel
    cap44(kk)=log(det(eye(n,n)+SNR*H'*H))/(2*log(2));
end
[N44,C44]=hist(cap44,K/10);     %CDF of MIMO capacity

m=8;n=4;    % 4x2 channels
for kk=1:K
    H=randn([m n]);     %Random MIMO Channel
    cap84(kk)=log(det(eye(n,n)+SNR*H'*H))/(2*log(2));
end
[N84,C84]=hist(cap84,K/10);     %CDF of MIMO capacity
```

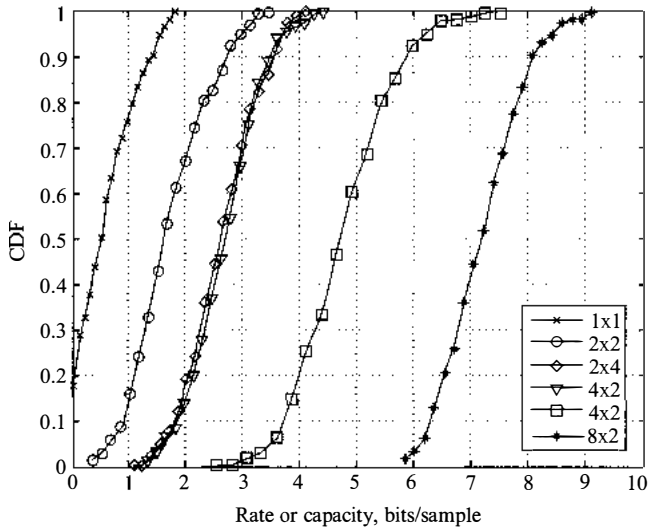


Figure 13.18 Cumulative distribution function (CDF) of different MIMO configurations.

```
m=8;n=8;     % 4x2 channels
for kk=1:K
    H=randn([m n]);      %Random MIMO Channel
    cap88(kk)=log(det(eye(n,n)+SNR*H'*H))/(2*log(2));
end
[N88,C88]=hist(cap88,K/10);     %CDF of MIMO capacity
% Now ready to plot the CDF of the capacity distribution
plot(C11,cumsum(N11)/K,'k-x',C22,cumsum(N22)/K,'k-o',...
    C24,cumsum(N24)/K,'k-d',C42,cumsum(N42)/K,'k-v',...
    C44,cumsum(N44)/K,'k-s',C84,cumsum(N84)/K,'k-*');
legend('1x1','2x2','2x4','4x2','4x4','8x4','Location','SouthEast');
grid
xlabel('Rate or Capacity (bits/sample) for SNR=3dB');ylabel('CDF');
% End of the plot
```

In Fig. 13.18, we illustrate the cumulative distribution function (CDF) of the channel capacity C_{MIMO}

$$\text{Prob}(C_{MIMO} \leq r)$$

of each MIMO configuration estimated from the 200 random channels. We computed the CDF of channel capacity for six different MIMO configurations: 1×1, 2×2, 2×4, 4×2, 4×4, and 8×4. The results clearly show that MIMO systems with more transmit and receive antennas will have CDF distributions concentrated at higher capacity or rate. For example, 2×2 MIMO systems will have capacity below 4 bits/sample with a probability of 1. However, for 4×4 MIMO systems, the probability drops to only 0.2. When considering 8×4 MIMO systems, the probability falls below 0.05. These numerical examples clearly demonstrate the higher capacity achieved by MIMO technologies.

REFERENCES

1. C. E. Shannon, "Mathematical Theory of Communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, July 1948; pp. 623–656, Oct. 1948.
2. R. V. L. Hartley, "Transmission of Information," *Bell Syst. Tech. J.*, vol. 7, pp. 535–563, July 1928.

3. H. Nyquist, "Certain Factors Affecting Telegraph Speed," *Bell Syst. Tech. J.*, vol. 3, pp. 324–346, April 1924.
4. N. Abramson, *Information Theory and Coding*, McGraw-Hill, New York, 1963.
5. R. G. Gallager, *Information Theory and Reliable Communication*, Wiley, New York, 1968.
6. D. A. Huffman, "A Method for Construction of Minimum Redundancy Codes," *Proc. IRE*, vol. 40, pp. 1098–1101, Sept. 1952.
7. R. W. Hamming, *Coding and Information Theory*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1986.
8. C. E. Shannon, "Communication in the Presence of Noise," *Proc. IRE*, vol. 37, pp. 10–21, Jan. 1949.
9. J. M. Wozencraft and I. A. Jacobs, *Principles of Communication Engineering*, Wiley, New York, 1965, Chapter 5.
10. A. J. Viterbi, *Principles of Coherent Communication*, McGraw-Hill, New York, 1966.
11. A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*, Cambridge University Press, Cambridge, 2003.
12. T. M. Cover and J. A Thomas, *Elements of Information Theory*, Wiley-Interscience, New York, 1991.

PROBLEMS

13.1-1 A message source generates one of four messages randomly every microsecond. The probabilities of these messages are 0.4, 0.3, 0.2, and 0.1. Each emitted message is independent of the other messages in the sequence.

(a) What is the source entropy?

(b) What is the rate of information generated by this source (in bits per second)?

13.1-2 A standard television picture is composed of approximately 300,000 basic picture elements (about 600 picture elements in a horizontal line and 500 horizontal lines per frame). Each of these elements can assume 10 distinguishable brightness levels (such as black and shades of gray) with equal probability. Find the information content of a television picture frame.

13.1-3 A radio announcer describes a television picture orally in 1000 words from his vocabulary of 10,000 words. Assume that each of the 10,000 words in the announcer's vocabulary is equally likely to occur in the description of this picture (a crude approximation, but good enough to give an idea). Determine the amount of information broadcast by the announcer in describing the picture. Would you say the announcer can do justice to the picture in 1000 words? Is the old adage "A picture is worth a thousand words" an exaggeration or an understatement of the reality? Use data in Prob. 13.1-2 to estimate the information of a picture.

13.1-4 From the town of the Old North Church in Boston, Paul Revere's friend was to show one lantern if the British army began advancing overland and two lanterns if they had chosen to cross the bay in boats.

(a) Assume that Revere had no way of guessing ahead of time what route the British might choose. How much information did he receive when he saw two lanterns?

(b) What if Revere were 90% sure the British would march overland? Then, how much information would the two lanterns have conveyed?

13.1-5 Estimate the information per letter in the English language by various methods, assuming that each character is independent of the others. (This is not true, but is good enough to get a rough idea.)

(a) In the first method, assume that all 27 characters (26 letters and a space) are equiprobable. This is a gross approximation, but good for a quick answer.

(b) In the second method, use the table of probabilities of various characters (Table P13.1-5).

TABLE P.13.1-5
Probability of Occurrence of Letters in the English Language

Letter	Probability	$-\log P_i$
Space	0.187	2.46
E	0.1073	3.22
T	0.0856	3.84
A	0.0668	3.90
O	0.0654	3.94
N	0.0581	4.11
R	0.0559	4.16
I	0.0519	4.27
S	0.0499	4.33
H	0.04305	4.54
D	0.03100	5.02
L	0.02775	5.17
F	0.02395	5.38
C	0.02260	5.45
M	0.02075	5.60
U	0.02010	5.64
G	0.01633	5.94
Y	0.01623	5.95
P	0.01623	5.95
W	0.01620	6.32
B	0.01179	6.42
V	0.00752	7.06
K	0.00344	8.20
X	0.00136	9.54
J	0.00108	9.85
Q	0.00099	9.98
Z	0.00063	10.63

(c) Use Zipf's law relating the word rank to its probability. In English prose, if we order words according to the frequency of usage so that the most frequently used word (*the*) is word number 1 (rank 1), the next most probable word (*of*) is number 2 (rank 2), and so on, then empirically it is found that $P(r)$, the probability of the rth word (rank r) is very nearly

$$P(r) = \frac{0.1}{r}$$

Now use Zipf's law to compute the entropy per word. Assume that there are 8727 words. The reason for this number is that the probabilities $P(r)$ sum to 1 for r from 1 to 8727. Zipf's law, surprisingly, gives reasonably good results. Assuming there are 5.5 letters (including space) per word on the average, determine the entropy or information per letter.

13.2-1 A source emits seven messages with probabilities 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, and 1/64, respectively. Find the entropy of the source. Obtain the compact binary code and find the average length of the codeword. Determine the efficiency and the redundancy of the code.

13.2-2 A source emits seven messages with probabilities 1/3, 1/3, 1/9, 1/9, 1/27, 1/27, and 1/27, respectively. Find the entropy of the source. Obtain the compact 3-ary code and find the average length of the codeword. Determine the efficiency and the redundancy of the code.

13.2-3 A source emits one of four messages randomly every microsecond. The probabilities of these messages are 0.5, 0.3, 0.1, and 0.1. Messages are generated independently.

(a) What is the source entropy?

(b) Obtain a compact binary code and determine the average length of the codeword, the efficiency, and the redundancy of the code.

(c) Repeat part **(b)** for a compact ternary code.

13.2-4 For the messages in Prob. 13.2-1, obtain the compact 3-ary code and find the average length of the codeword. Determine the efficiency and the redundancy of this code.

13.2-5 For the messages in Prob. 13.2-2, obtain the compact binary code and find the average length of the codeword. Determine the efficiency and the redundancy of this code.

13.2-6 A source emits three equiprobable messages randomly and independently.

(a) Find the source entropy.

(b) Find a compact ternary code, the average length of the codeword, the code efficiency, and the redundancy.

(c) Repeat part **(b)** for a binary code.

(d) To improve the efficiency of a binary code, we now code the second extension of the source. Find a compact binary code, the average length of the codeword, the code efficiency, and the redundancy.

13.4-1 A binary channel matrix is given by

$$\begin{array}{cc} & \text{Outputs} \\ & \begin{array}{cc} y_1 & y_2 \end{array} \\ \text{Inputs} \begin{array}{c} x_1 \\ x_2 \end{array} & \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{10} & \frac{9}{10} \end{pmatrix} \end{array}$$

This means $P_{y|x}(y_1|x_1) = 2/3$, $P_{y|x}(y_2|x_1) = 1/3$, etc. You are also given that $P_x(x_1) = 1/3$ and $P_x(x_2) = 2/3$. Determine $H(x)$, $H(x|y)$, $H(y)$, $H(y|x)$, and $I(x; y)$.

13.4-2 For the ternary channel in Fig. P13.4-2, $P_x(x_1) = P$, $P_x(x_2) = P_x(x_3) = Q$. (*Note*: $P+2Q = 1$.)

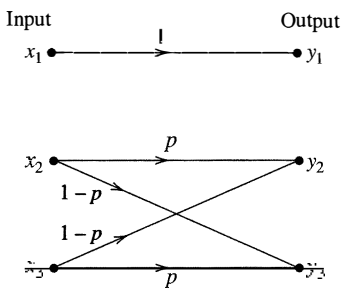


Figure P.13.4-2

(a) Determine $H(x)$, $H(x|y)$, $H(y)$, and $I(x; y)$.

(b) Show that the channel capacity C_s is given by

$$C_s = \log\left(\frac{\beta + 2}{\beta}\right) \tag{13.123}$$

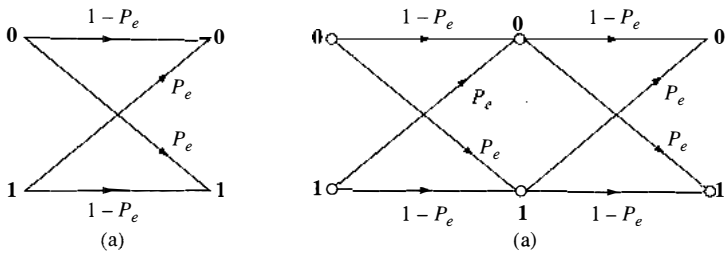
where $\beta = 2^{-[p \log p + (1-p) \log(1-p)]}$.

13.4-3 Consider the binary symmetric channel shown in Fig. P13.4-3a. The channel matrix is given by

$$M = \begin{bmatrix} 1 - P_e & P_e \\ P_e & 1 - P_e \end{bmatrix}$$

Figure P13.4-3b shows a cascade of two such BSCs.

Figure P.13.4-3



(a) Determine the channel matrix for the cascaded channel in Fig. P13.4-3b. Show that this matrix is M^2.

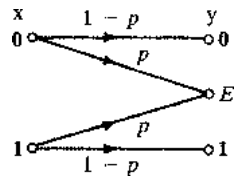
(b) If the two BSC channels in Fig. P13.4-3b have error probabilities P_{e1} and P_{e2}, with channel matrices M_1 and M_2, respectively, show that the channel matrix of the cascade of these two channels is $M_1 M_2$.

(c) Use the results in part **(b)** to show that the channel matrix for the cascade of k identical BSCs each with channel matrix M is M^k. Verify your answer for $n = 3$ by confirming the results in Example 8.7.

(d) Use the result in part **(c)** to determine the channel capacity for a cascade of k identical BSC channels each with error probability P_e.

13.4-4 In data communication using error detection code, as soon as an error is detected, an automatic request for retransmission (ARQ) enables retransmission of the data in error. In such a channel, the data in error is erased. Hence, there is an erase probability p, but the probability of error is zero. Such a channel, known as a **binary erasure channel (BEC)**, can be modeled as shown in Fig. P13.4-4. Determine $H(x)$, $H(x|y)$, and $I(x; y)$ assuming the two transmitted messages equiprobable.

Figure P.13.4-4



13.4-5 A cascade of two channels is shown in Fig. P13.4-5. The symbols at the source, at the output of the first channel, and at the output of the second channel are denoted by x, y, and z. Show that

$$H(\mathrm{x}|\mathrm{z}) \geq H(\mathrm{x}|\mathrm{y})$$

and

$$I(\mathrm{x};\mathrm{y}) \geq I(\mathrm{x};\mathrm{z})$$

This shows that the information that can be transmitted over a cascaded channel can be no greater than that transmitted over one link. In effect, information channels tend to leak information.

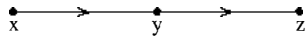
Hint: For a cascaded channel, observe that

$$P(z_k|\mathrm{y}_j, \mathrm{x}_i) = P(z_k|\mathrm{y}_j)$$

Hence, by Bayes' rule,

$$P(\mathrm{x}_k|\mathrm{y}_j, z_k) = P(\mathrm{x}_i|\mathrm{y}_j)$$

Figure P.13.4-5



13.5-1 For a continuous random variable x constrained to a peak magnitude M ($-M < x < M$), show that the entropy is maximum when x is uniformly distributed in the range $(-M, M)$ and has zero probability density outside this range. Show that the maximum entropy is given by $\log 2M$.

13.5-2 For a continuous random variable x constrained to only positive values $0 < x < \infty$ and a mean value A, show that the entropy is maximum when

$$P_{\mathrm{x}}(x) = \frac{1}{A} e^{-x/A} u(x)$$

Show that the corresponding entropy is

$$H(\mathrm{x}) = \log eA$$

13.5-3 A television transmission requires 30 frames of 300,000 picture elements each to be transmitted per second. Use the data in Prob. 13.1-2 to estimate the theoretical bandwidth of the AWGN channel if the SNR at the receiver is required to be at least 50 dB.

13.7-1 In a communication system over a frequency-selective channel with transfer function

$$H(f) = \frac{2}{1 + j\pi(f/200)}$$

the input signal PSD is

$$S_{\mathrm{x}}(f) = \Pi\left(\frac{f}{400}\right)$$

The channel noise is AWGN with spectrum $S_{\mathrm{n}}(f) = 10^{-2}$. Find the mutual information between the channel input and the channel output.

14 ERROR CORRECTING CODES

As seen from the discussion in Chapter 13, the key to achieving error-free digital communication in the presence of distortion, noise, and interference is the addition of appropriate redundancy to the original data bits. The addition of a single parity check digit to detect an odd number of errors is a good example. Since Shannon's pioneering paper,[1] a great deal of work has been carried out in the area of forward error correcting (FEC) codes. In this chapter, we will provide an introduction; readers can find much more in-depth coverage of this topic from the classic textbook by Lin and Costello.[2]

14.1 OVERVIEW

Generally, there are two important classes of FEC codes: block codes and convolutional codes. In **block codes**, every block of k data digits is encoded into a longer codeword of n digits ($n > k$). Every unique sequence of k data digits fully determines a unique codeword of n digits. In **convolutional codes**, the coded sequence of n digits depends not only on the k data digits but also on the previous $N - 1$ data digits ($N > 1$). Hence, the coded sequence for a certain k data digits is not unique but depends also on $N - 1$ earlier data digits. In short, the encoder has memory. In block codes, k data digits are accumulated and then encoded into an n-digit codeword. In convolutional codes, the encoding is done on a continuous running basis rather than by blocks of k data digits.

Shannon's pioneer work[1] on the capacity of noisy channels has yielded a famous result known as the **noisy channel coding theorem**. This result states that for a noisy channel with a capacity C, there exist codes of rate $R < C$ such that maximum likelihood decoding can lead to error probability

$$P_e \leq 2^{-nE_b(R)} \tag{14.1}$$

where $E_b(R)$ is the energy per information bit defined as a function of code rate R. This remarkable result shows that **arbitrarily small** error probability can be achieved by increasing the block code length n while keeping the code rate constant. A similar result for convolutional codes was also shown in Ref. 1. Note that this result establishes the existence of **good** codes. It does not, however, tell us how to find such codes. In fact, it is not simply a question of designing good codes. Indeed, this result also requires large n to reduce error probability and requires decoders to use large storage and high complexity for large codewords of size n. Thus, the key problem in code design is the dual task of searching for good error correction codes with large

length n to reduce error probability, as well as decoders that are simple to implement. The best results thus far are the recent discovery of turbo codes and the rediscovery of low-density parity check (LDPC) codes, to be discussed later. The former are derived from convolutional codes, whereas the latter are a form of block code.

Error correction coding requires a strong mathematical background. To provide a sufficiently detailed introduction of various important topics on this subject, we organize this chapter according to the level of mathematical background necessary for understanding. We begin by covering the simpler and more intuitive block codes that require the least amount of probability analysis. We then introduce the concepts and principles of convolutional codes and their decoding. Finally, we focus on the more sophisticated soft-decoding concept, which lays the foundation for the subsequent coverage of recent progresses on high-performance turbo codes and low-density parity check codes.

14.2 REDUNDANCY FOR ERROR CORRECTION

In FEC codes, a codeword is a unit of bits that can be decoded independently. The number of bits in codeword is known as the code length. If k data digits are transmitted by a codeword of n digits, the number of check digits is $m = n - k$. The **code rate** is $R = k/n$. Such a code is known as an (n, k) code. Data digits $(d_1, d_2, \ldots, d_k)$ are a k-tuple, and, hence, this is a k-dimensional vector $\boldsymbol{d}$. Similarly, a codeword $(c_1, c_2, \ldots, c_n)$ is an n-dimensional vector $\boldsymbol{c}$. As a preliminary, we shall determine the minimum number of check digits required to detect or correct t number of errors in an (n, k) code.

If the binary code length is n, then a total of 2^n code words (or vertices of an n-dimensional hypercube) is available to assign to 2^k data words. Suppose we wish to find a code that will correct up to t wrong digits. In this case, if we transmit a data word $\boldsymbol{d}_j$ by using one of the codewords (or vertices) $\boldsymbol{c}_j$, then because of channel errors the received word will not be $\boldsymbol{c}_j$ but will be $\boldsymbol{c}'_j$. If the channel noise causes errors in t or fewer digits, then $\boldsymbol{c}'_j$ will lie somewhere inside the Hamming sphere* of radius t centered at $\boldsymbol{c}_j$. If the code is to correct up to t errors, then the code must have the property that all the Hamming spheres of radius t centered at the codewords are nonoverlapping. This means that we must not use vertices as codewords that are within a Hamming distance t from any codeword. If a received word lies within a Hamming sphere of radius t centered at $\boldsymbol{c}_j$, then we decide that the true transmitted codeword was $\boldsymbol{c}_j$. This scheme is capable of correcting up to t errors, and $d_{\min}$, the minimum distance between t error correcting codewords without overlapping, is

$$d_{\min} = 2t + 1 \tag{14.2}$$

Next, to find a relationship between n and k, we observe that 2^n vertices, or words, are available for 2^k data words. Thus, there are $2^n - 2^k$ redundant vertices. How many vertices, or words, can lie within a Hamming sphere of radius t? The number of sequences (of n digits) that differ from a given sequence by j digits is the number of possible combinations of n things taken j at a time and is given by $\binom{n}{j}$ [Eq. (8.16)]. Hence, the number of ways in which up to t errors can occur is given by

$$\sum_{j=1}^{t} \binom{n}{j}$$

* See Chapter 13 for definitions of Hamming distance and Hamming sphere.

Thus for each codeword, we must leave

$$\sum_{j=1}^{t} \binom{n}{j}$$

vertices (words) unused. Because we have 2^k codewords, we must leave a total of

$$2^k \sum_{j=1}^{t} \binom{n}{j}$$

words unused. Therefore, the total number of words must at least be

$$2^k + 2^k \sum_{j=1}^{t} \binom{n}{j} = 2^k \sum_{j=0}^{t} \binom{n}{j}$$

But the total number of words, or vertices, available is 2^n. Thus, we require,

$$2^n \geq 2^k \sum_{j=0}^{t} \binom{n}{j}$$

or

$$2^{n-k} \geq \sum_{j=0}^{t} \binom{n}{j} \tag{14.3a}$$

Observe that $n - k = m$ is the number of check digits. Hence, Eq. (14.3a) can be expressed as

$$2^m \geq \sum_{j=0}^{t} \binom{n}{j} \tag{14.3b}$$

This is known as the **Hamming bound**. It should also be remembered that the Hamming bound is a necessary but not a sufficient condition in general. However, for single-error correcting codes, it is a necessary and sufficient condition. If some m satisfies the Hamming bound, it does not necessarily mean that a t-error correcting code of n digits can be constructed. Table 14.1 shows some examples of error correction codes and their rates.

A code for which the inequalities in Eqs. (14.3) become equalities is known as a **perfect code**. In such a code the Hamming spheres (about all the codewords) not only are nonoverlapping but they exhaust all the 2^n vertices, leaving no vertex outside some sphere. An e-error correcting perfect code satisfies the condition that every possible (received) sequence is at a distance at most e from some codeword. Perfect codes exist in only a comparatively few cases. Binary, single-error correcting, perfect codes are called **Hamming codes.** For a Hamming code, $t = 1$ and $d_{\min} = 3$, and from Eq. (14.3b) we have

$$2^m = \sum_{j=0}^{1} \binom{n}{j} = 1 + n \tag{14.4}$$

TABLE 14.1
Some Examples of Error Correcting Codes

	n	k	Code	Code Efficiency (or Code Rate)
Single-error correcting, $t = 1$	3	1	(3, 1)	0.33
Minimum code separation 3	4	1	(4, 1)	0.25
	5	2	(5, 2)	0.4
	6	3	(6, 3)	0.5
	7	4	(7, 4)	0.57
	15	11	(15, 11)	0.73
	31	26	(31, 26)	0.838
Double-error correcting, $t = 2$	10	4	(10, 4)	0.4
Minimum code separation 5	15	8	(15, 8)	0.533
Triple-error correcting, $t = 3$	10	2	(10, 2)	0.2
Minimum code separation 7	15	5	(15, 5)	0.33
	23	12	(23, 12)	0.52

and

$$n = 2^m - 1$$

Thus, Hamming codes are (n, k) codes with $n = 2^m - 1$ and $k = 2^m - 1 - m$ and minimum distance $d_{\min} = m$. In general, we often write Hamming code as $(2^m - 1, 2^m - 1 - m, m)$ code. One of the most well-known Hamming codes is the (7, 4, 3) code.

Another way of correcting errors is to design a code to detect (not to correct) up to t errors. When the receiver detects an error, it can request retransmission. This mechanism is known as **a**utomatic **r**epeat re**q**uest (or ARQ). Because error detection requires fewer check digits, these codes operate at a higher rate (efficiency).

To detect t errors, codewords need to be separated by a Hamming distance of at least $t + 1$. Otherwise, an erroneously received bit string with up to t error bits could be another transmitted codeword. Suppose a transmitted codeword $\boldsymbol{c}_j$ contains α bit errors ($\alpha \leq t$). Then the received codeword $\boldsymbol{c}_j'$ is at a distance of α from $\boldsymbol{c}_j$. Because $\alpha \leq t$, however, $\boldsymbol{c}_j'$ can never be any other valid codeword, since all codewords are separated by at least $t + 1$. Thus, the reception of $\boldsymbol{c}_j'$ immediately indicates that an error has been made. Thus, the minimum distance $d_{\min}$ between t error detecting codewords is

$$d_{\min} = t + 1$$

In presenting coding theory, we shall use modulo-2 addition, defined by

$$\mathbf{1} \oplus \mathbf{1} = \mathbf{0} \oplus \mathbf{0} = \mathbf{0}$$
$$\mathbf{0} \oplus \mathbf{1} = \mathbf{1} \oplus \mathbf{0} = \mathbf{1}$$

This is also known as the exclusive OR (XOR) operation in digital logic. Note that the modulo-2 sum of any binary digit with itself is always zero. All the additions in the mathematical development of binary codes presented henceforth are modulo-2.

14.3 LINEAR BLOCK CODES

A codeword consists of n digits $c_1, c_2, \ldots, c_n$, and a data word consists of k digits $d_1, d_2, \ldots, d_k$. Because the codeword and the data word are an n-tuple and a k-tuple, respectively, they are n- and k-dimensional vectors. We shall use row vectors to represent these words:

$$\boldsymbol{c} = (c_1, c_2, \ldots, c_n)$$
$$\boldsymbol{d} = (d_1, d_2, \ldots, d_k)$$

For the general case of linear block codes, all the n digits of $\boldsymbol{c}$ are formed by linear combinations (modulo-2 additions) of k data digits. A special case in which $c_1 = d_1, c_2 = d_2, \ldots, c_k = d_k$ and the remaining digits from c_{k+1} to c_n are linear combinations of $d_1, d_2, \ldots, d_k$, is known as a **systematic code.** In a systematic code, the leading k digits of a codeword are the data (or information) digits and the remaining $m = n - k$ digits are the **parity check digits**, formed by linear combinations of data digits $d_1, d_2, \ldots, d_k$:

$$\begin{aligned}
c_1 &= d_1 \\
c_2 &= d_2 \\
&\vdots \\
c_k &= d_k \\
c_{k+1} &= h_{11}d_1 \oplus h_{12}d_2 \oplus \cdots \oplus h_{1k}d_k \\
c_{k+2} &= h_{21}d_1 \oplus h_{22}d_2 \oplus \cdots \oplus h_{2k}d_k \\
&\vdots \\
c_n &= h_{m1}d_1 \oplus h_{m2}d_2 \oplus \cdots \oplus h_{mk}d_k
\end{aligned} \tag{14.5a}$$

or

$$\boldsymbol{c} = \boldsymbol{dG} \tag{14.5b}$$

where

$$\boldsymbol{G} = \left[\begin{array}{ccccc|ccccc}
1 & 0 & 0 & \cdots & 0 & h_{11} & h_{21} & \cdots & h_{m1} \\
0 & 1 & 0 & \cdots & 0 & h_{12} & h_{22} & \cdots & h_{m2} \\
 & & & \vdots & 0 & & & \vdots & 0 \\
0 & 0 & 0 & \cdots & 1 & h_{1k} & h_{2k} & \cdots & h_{mk}
\end{array}\right] \tag{14.6}$$

$\underbrace{\qquad\qquad\qquad\qquad\qquad}_{\boldsymbol{I}_k(k \times k)}$ $\underbrace{\qquad\qquad\qquad\qquad}_{\boldsymbol{P}(k \times m)}$

The $k \times n$ matrix $\boldsymbol{G}$ is called the **generator matrix**. For systematic codes, $\boldsymbol{G}$ can be partitioned into a $k \times k$ identity matrix $\boldsymbol{I}_k$ and a $k \times m$ matrix $\boldsymbol{P}$. The elements of $\boldsymbol{P}$ are either 0 or 1. The

codeword can be expressed as

$$\begin{aligned} \boldsymbol{c} &= \boldsymbol{dG} \\ &= \boldsymbol{d}[\boldsymbol{I}_k \quad \boldsymbol{P}] \\ &= [\boldsymbol{d} \quad \boldsymbol{dP}] \\ &= [\boldsymbol{d} \quad \boldsymbol{c}_p] \end{aligned} \tag{14.7}$$

where the check digits, also known as the checksum bits or parity bits, are

$$\boldsymbol{c}_p = \boldsymbol{dP} \tag{14.8}$$

Thus, knowing the data digits, we can calculate the check digits from Eq. (14.8) and consequently the codeword $\boldsymbol{c}_p$. The **weight** of the codeword $\boldsymbol{c}$ is the number of **1**s in the codeword. The **Hamming distance** between two codewords $\boldsymbol{c}_a$ and $\boldsymbol{c}_b$ is the number of elements by which they differ, or

$$d(\boldsymbol{c}_a, \boldsymbol{c}_b) = \textbf{weight of } (\boldsymbol{c}_a \oplus \boldsymbol{c}_b)$$

Example 14.1 For a (6, 3) code, the generator matrix $\boldsymbol{G}$ is

$$\boldsymbol{G} = \left[\underbrace{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}}_{I_k} \quad \underbrace{\begin{matrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{matrix}}_{P}\right]$$

For all eight possible data words, find the corresponding codewords, and verify that this code is a single-error correcting code.

Table 14.2 shows the eight data words and the corresponding codewords formed from $\boldsymbol{c} = \boldsymbol{dG}$.

TABLE 14.2

Data Word d	Codeword c
111	111000
110	110110
101	101011
100	100101
011	011101
010	010011
001	001110
000	000000

Note that the distance between any two codewords is at least 3. Hence, the code can correct at least one error. The possible encoder for this code shown in Fig. 14.1 uses a three-digit shift register and three modulo-2 adders.

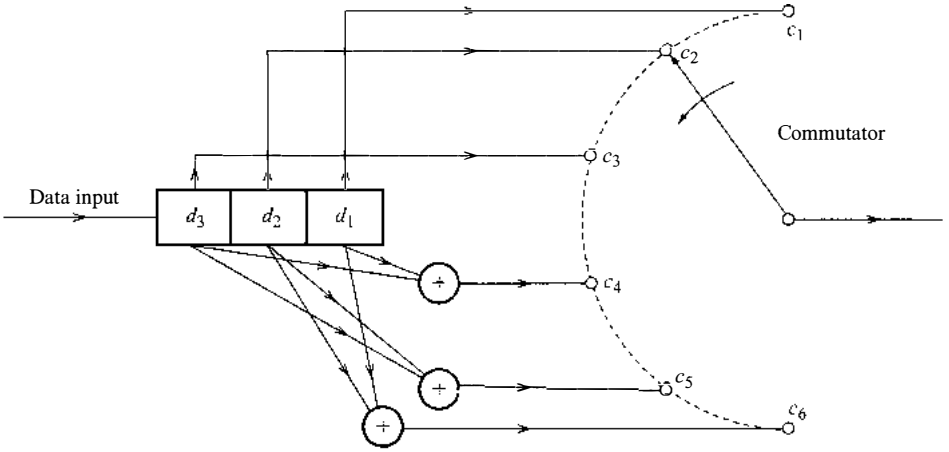


Figure 14.1 Encoder for linear block codes.

Linear Codes

A block code is a **linear block code** if for every pair of codewords $\mathbf{c}_a$ and $\mathbf{c}_b$ from the block code,

$$\mathbf{c}_a \oplus \mathbf{c}_b$$

is also a codeword. For this reason, linear codes must have an all-zero codeword **000 · · · 00**. For linear codes, the **minimum distance** equals the **minimum weight**.

Decoding

Let us consider some codeword properties that could be used for the purpose of decoding. From Eq. (14.8) and the fact that the modulo-2 sum of any sequence with itself is zero, we get

$$\boldsymbol{d} \cdot \boldsymbol{P} \oplus \boldsymbol{c}_p = \underbrace{[\boldsymbol{d} \quad \boldsymbol{c}_p]}_{\boldsymbol{c}} \begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{I}_m \end{bmatrix} = 0 \tag{14.9}$$

where $\boldsymbol{I}_m$ is the identity matrix of order $m \times m$ ($m = n - k$). Thus,

$$\boldsymbol{c}\boldsymbol{H}^T = 0 \tag{14.10a}$$

where

$$\boldsymbol{H}^T = \begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{I}_m \end{bmatrix} \tag{14.10b}$$

and its transpose

$$\boldsymbol{H} = [\boldsymbol{P}^T \quad \boldsymbol{I}_m] \tag{14.10c}$$

is called the **parity check matrix**. Every codeword must satisfy Eq. (14.10a). This is our clue to decoding. Consider the received word $\boldsymbol{r}$. Because of possible errors caused by channel noise, $\boldsymbol{r}$ in general differs from the transmitted codeword $\boldsymbol{c}$,

$$\boldsymbol{r} = \boldsymbol{c} \oplus \boldsymbol{e}$$

where the error word (or error vector) $\boldsymbol{e}$, is also a row vector of n elements. For example, if the data word **100** in Example 14.1 is transmitted as a codeword **100101** (see Table 14.2), and the channel noise causes a detection error in the third digit, then

$$\boldsymbol{r} = \mathbf{101101}$$
$$\boldsymbol{c} = \mathbf{100101}$$

and

$$\boldsymbol{e} = \mathbf{001000}$$

Thus, an element **1** in $\boldsymbol{e}$ indicates an error in the corresponding position, and **0** indicates no error. The Hamming distance between $\boldsymbol{r}$ and $\boldsymbol{c}$ is simply the number of **1**s in $\boldsymbol{e}$.

Suppose the transmitted codeword is $\boldsymbol{c}_i$ and the channel noise causes an error $\boldsymbol{e}_i$, making the received word $\boldsymbol{r} = \boldsymbol{c}_i + \boldsymbol{e}_i$. If there were no errors, that is, if $\boldsymbol{e}_i$ were **000000**, then we would have $\boldsymbol{r}\boldsymbol{H}^T = 0$. But because of possible channel errors, $\boldsymbol{r}\boldsymbol{H}^T$ is in general a nonzero row vector $\boldsymbol{s}$, called the **syndrome**:

$$\begin{aligned} \boldsymbol{s} &= \boldsymbol{r}\boldsymbol{H}^T && \text{(14.11a)} \\ &= (\boldsymbol{c}_i \oplus \boldsymbol{e}_i)\boldsymbol{H}^T \\ &= \boldsymbol{c}_i H^T \oplus \boldsymbol{e}_i \boldsymbol{H}^T \\ &= \boldsymbol{e}_i \boldsymbol{H}^T && \text{(14.11b)} \end{aligned}$$

Receiving $\boldsymbol{r}$, we can compute the syndrome $\boldsymbol{s}$ [Eq. (14.11a)] and presumably we can compute $\boldsymbol{e}_i$ from Eq. (14.11b). Unfortunately, knowledge of $\boldsymbol{s}$ does not allow us to solve $\boldsymbol{e}_i$ uniquely. This is because $\boldsymbol{r}$ can also be expressed in terms of codewords other than $\boldsymbol{c}_i$. Thus,

$$\boldsymbol{r} = \boldsymbol{c}_j \oplus \boldsymbol{e}_j \qquad j \neq i$$

Hence,

$$\boldsymbol{s} = (\boldsymbol{c}_j \oplus \boldsymbol{e}_j)\boldsymbol{H}^T = \boldsymbol{e}_j \boldsymbol{H}^T$$

Because there are 2^k possible codewords,

$$\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^T$$

is satisfied by 2^k error vectors. In other words, the syndrome $\boldsymbol{s}$ by itself cannot define a unique error vector. For example, if a data word $\boldsymbol{d} = \mathbf{100}$ is transmitted by a codeword **100101** in Example 14.1, and if a detection error is caused in the third digit, then the received word is **101101**. In this case we have $\boldsymbol{c} = \mathbf{100101}$ and $\boldsymbol{e} = \mathbf{001000}$. But the same word could have been received if $\boldsymbol{c} = \mathbf{101011}$ and $\boldsymbol{e} = \mathbf{000110}$, or if $\boldsymbol{c} = \mathbf{010011}$ and $\boldsymbol{e} = \mathbf{111110}$, and so on. Thus, there are eight possible error vectors (2^k error vectors) that all satisfy Eq. (14.11b). Which vector shall we choose? For this, we must define our decision criterion. One reasonable criterion is the maximum likelihood rule according to which, if we receive $\boldsymbol{r}$, then we decide in favor of that $\boldsymbol{c}$ for which $\boldsymbol{r}$ is most likely to be received. In other words, we decide "$\boldsymbol{c}_i$ transmitted" if

$$P(\boldsymbol{r}|\boldsymbol{c}_i) > P(\boldsymbol{r}|\boldsymbol{c}_k) \qquad \text{all } k \neq i$$

For a binary symmetric channel (BSC), this rule gives a very simple answer. Suppose the Hamming distance between $\boldsymbol{r}$ and $\boldsymbol{c}_i$ is d; that is, the channel noise causes errors in d digits.

Then if P_e is the digit error probability of a BSC,

$$P(\boldsymbol{r}|\boldsymbol{c}_i) = P_e^d(1-P_e)^{n-d} = (1-P_e)^n\left(\frac{P_e}{1-P_e}\right)^d$$

If $P_e < 0.5$ holds for a reasonable channel, then $P(\boldsymbol{r}|\boldsymbol{c}_i)$ is a monotonically decreasing function of d because $P_e/(1-P_e) < 1$. Hence, to maximize $P(\boldsymbol{r}|\boldsymbol{c}_i)$, we must choose that $\boldsymbol{c}_i$ which is closest to $\boldsymbol{r}$; that is, we must choose the error vector $\boldsymbol{e}$ with the smallest number of **1**s. A vector with $\boldsymbol{e}$ the smallest number of **1**s is called the **minimum weight vector**. This minimum weight error vector $\boldsymbol{e}_{min}$ will be used to correct the error in $\boldsymbol{r}$ via

$$\boldsymbol{c} = \boldsymbol{r} \oplus \boldsymbol{e}_{\min}$$

Example 14.2 A linear (6, 3) code is generated according to the generating matrix in Example 14.1. The receiver receives $\boldsymbol{r} = \mathbf{100011}$. Determine the corresponding data word if the channel is a BSC and the maximum likelihood decision is used.

We have

$$\begin{aligned}\boldsymbol{s} &= \boldsymbol{r}\boldsymbol{H}^T \\ &= [\mathbf{1}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{1}\quad \mathbf{1}]\cdot\begin{bmatrix}1&0&1\\0&1&1\\1&1&0\\1&0&0\\0&1&0\\0&0&1\end{bmatrix} \\ &= [\mathbf{1}\quad \mathbf{1}\quad \mathbf{0}]\end{aligned}$$

Because for modulo-2 operation, subtraction is the same as addition, the correct transmitted codeword $\boldsymbol{c}$ is given by

$$\boldsymbol{c} = \boldsymbol{r} \oplus \boldsymbol{e}$$

where $\boldsymbol{e}$ satisfies

$$\begin{aligned}\boldsymbol{s} &= [\mathbf{1}\quad \mathbf{1}\quad \mathbf{0}] = \boldsymbol{e}\boldsymbol{H}^T \\ &= [e_1\quad e_2\quad e_3\quad e_4\quad e_5\quad e_6]\cdot\begin{bmatrix}1&0&1\\0&1&1\\1&1&0\\1&0&0\\0&1&0\\0&0&1\end{bmatrix}\end{aligned}$$

We see that $\boldsymbol{e} = \mathbf{001000}$ satisfies this equation. But so does $\boldsymbol{e} = \mathbf{000110}$, or **010101**, or **011011**, or **111110**, or **110000**, or **101101**, or **100011**. The suitable choice, the minimum weight $\boldsymbol{e}_{\min}$, is **001000**. Hence,

$$\boldsymbol{c} = \mathbf{100011} \oplus \mathbf{001000} = \mathbf{101011}$$

TABLE 14.3
Decoding Table for Code in Table 14.2

e	s
000000	**000**
100000	**101**
010000	**011**
001000	**110**
000100	**100**
000010	**010**
000001	**001**
100010	**111**

The decoding procedure just described is quite disorganized. To make it systematic, we would consider all possible syndromes and for each syndrome associate a minimum weight error vector. For instance, the single-error correcting code in Example 14.1 has a syndrome with three digits. Hence, there are eight possible syndromes. We prepare a table of minimum weight error vectors corresponding to each syndrome (see Table 14.3). This table can be prepared by considering all possible minimum weight error vectors and using Eq. (14.11b) to compute $\boldsymbol{s}$ for each of them. The first minimum weight error vector **000000** is a trivial case that has the syndrome **000**. Next, we consider all possible unit weight error vectors. There are six such vectors: **100000**, **010000**, **001000**, **000100**, **000010**, **000001**. Syndromes for these can readily be calculated from Eq. (14.11b) and tabulated (Table 14.3). This still leaves one syndrome, **111**, that is not matched with an error vector. Since all unit weight error vectors are exhausted, we must look for error vectors of weight 2.

We find that for the first seven syndromes (Table 14.3), there is a unique minimum weight vector $\boldsymbol{e}$. But for $\boldsymbol{s} = \mathbf{111}$, the error vector $\boldsymbol{e}$ has a minimum weight of 2, and it is not unique. For example, $\boldsymbol{e} = \mathbf{100010}$ or **010100** or **001001** all have $\boldsymbol{s} = \mathbf{111}$, and all three $\boldsymbol{e}$ are minimum weight (weight 2). In such a case, we can pick any one $\boldsymbol{e}$ as a **correctable** error pattern. In Table 14.3, we have picked $\boldsymbol{e} = \mathbf{100010}$ as the double-error correctable pattern. This means the present code can correct all six single-error patterns and one double-error pattern (**100010**). For instance, if $\boldsymbol{c} = \mathbf{101011}$ is transmitted and the channel noise causes the double error **100010**, the received vector $\boldsymbol{r} = \mathbf{001001}$, and

$$\boldsymbol{s} = \boldsymbol{r}\boldsymbol{H}^T = [1\,1\,1]$$

From Table 14.3 we see that corresponding to $\boldsymbol{s} = \mathbf{111}$ is $\boldsymbol{e} = \mathbf{100010}$, and we immediately decide $\boldsymbol{c} = \boldsymbol{r} \oplus \boldsymbol{e} = \mathbf{101011}$. Note, however, that this code will not correct double-error patterns except for **100010**. Thus, this code corrects not only all single errors but one double-error pattern as well. This extra bonus of one double-error correction occurs because n and k oversatisfy the Hamming bound [Eq. (14.3b)]. In case n and k were to satisfy the bound exactly, we would have only single-error correction ability. This is the case for the (7, 4) code, which can correct all single-error patterns only.

Thus for systematic decoding, we prepare a table of all correctable error patterns and the corresponding syndromes. For decoding, we need only calculate $\boldsymbol{s} = \boldsymbol{r}\boldsymbol{H}^T$ and, from the decoding table, find the corresponding $\boldsymbol{e}$. The decision is $\boldsymbol{c} = \boldsymbol{r} \oplus \boldsymbol{e}$. Because $\boldsymbol{s}$ has $m = n - k$ digits, there is a total of 2^{n-k} syndromes, each consisting of $n - k$ digits. There is the same number of correctable error vectors, each of n digits. Hence, for the purpose of decoding, we need a storage of $(2n - k)2^{n-k} = (2n - k)2^m$ bits. This storage requirement grows exponentially

with m, and the number of parity check digits can be enormous, even for moderately complex codes.

Constructing Hamming Codes

It is still not clear how to design or choose coefficients of the generator or parity check matrix. Unfortunately, there is no general systematic way to design codes, except for the few special cases such as cyclic codes and the class of single-error correcting codes known as *Hamming codes*. Let us consider a single-error correcting (7, 4) code. This code satisfies the Hamming bound exactly, and we shall see that a proper code can be constructed. In this case $m = 3$, and there are seven nonzero syndromes, and because $n = 7$, there are exactly seven single-error patterns. Hence, we can correct all single-error patterns and no more. Consider the single-error pattern $\boldsymbol{e} = \mathbf{1000000}$. Because

$$\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^T$$

$\boldsymbol{e}\boldsymbol{H}^T$ will be simply the first row of $\boldsymbol{H}^T$. Similarly, for $\boldsymbol{e} = \mathbf{0100000}$, $\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^T$ will be the second row of $\boldsymbol{H}^T$, and so on. Now for unique decodability, we require that all seven syndromes corresponding to the seven single-error patterns be distinct. Conversely, if all seven syndromes are distinct, we can decode all the single-error patterns. This means that the only requirement on $\boldsymbol{H}^T$ is that all seven of its rows be distinct and nonzero. Note that $\boldsymbol{H}^T$ is an $(n \times n - k)$ matrix (i.e., 7×3 in this case). Because there exist seven nonzero patterns of three digits, it is possible to find seven nonzero rows of three digits each. There are many ways in which these rows can be ordered. But we emphasize that the three bottom rows must form the identity matrix I_m [see Eq. (14.10b)].

One possible form of $\boldsymbol{H}^T$ is

$$\boldsymbol{H}^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{P} \\ \\ \boldsymbol{I}_m \end{bmatrix}$$

The corresponding generator matrix $\boldsymbol{G}$ is

$$\boldsymbol{G} = [\boldsymbol{I}_k \quad \boldsymbol{P}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Thus when $\boldsymbol{d} = \mathbf{1011}$, the corresponding code word $\boldsymbol{c} = \mathbf{1011001}$, and so forth.

A general linear (n, k) code has m-dimensional syndrome vectors ($m = n - k$). Hence, there are $2^m - 1$ distinct nonzero syndrome vectors that can correct $2^m - 1$ single-error patterns. Because in an (n, k) code there are exactly n single-error patterns, all these single errors can be corrected if

$$2^m - 1 \geq n$$

This is precisely the condition in Eq. (14.4) for $t = 1$. Thus, for any (n, k) satisfying this condition, it is possible to construct a single-error correcting code by the procedure discussed. To summarize, a $(2^m - 1,\ 2^m - 1 - m,\ m)$ Hamming code has the following attributes:

Number of parity bits	$m \geq 3$
Code length	$n = 2^m - 1$
Number of message bits	$k = 2^m - m - 1$
Minimum distance	$d_{\min} = 3$
Error correcting capability	$t = 1$

For more discussion on block coding, the readers should consult the books by Peterson and Weldon[3] and by Lin and Costello.[2]

14.4 CYCLIC CODES

Cyclic codes are a subclass of linear block codes. As seen before, a procedure for selecting a generator matrix is relatively easy for single-error correcting codes. This procedure, however, cannot carry us very far in constructing higher order error correcting codes. Cyclic codes satisfy a nice mathematical structure that permits the design of higher order correcting codes. Second, for cyclic codes, encoding and syndrome calculations can be easily implemented by using simple shift registers.

In cyclic codes, the codewords are simple lateral cyclic shifts of one another. For example, if $\boldsymbol{c} = (c_1, c_2, \ldots, c_{n-1}, c_n)$ is a codeword, then so are $(c_2, c_3, \ldots, c_n, c_1)$, $(c_3, c_4, \ldots, c_n, c_1, c_2)$, and so on. We shall use the following notation. If

$$\boldsymbol{c} = (c_1, c_2, \ldots, c_n) \tag{14.12a}$$

is a code vector of a code C, then $\boldsymbol{c}^{(i)}$ denotes $\boldsymbol{c}$ shifted cyclically i places to the left, that is,

$$\boldsymbol{c}^{(i)} = (c_{i+1}, c_{i+2}, \ldots, c_n, c_1, c_2, \ldots, c_i) \tag{14.12b}$$

Cyclic codes can be described in a polynomial form. This property is extremely useful in the analysis and implementation of these codes. The code vector $\boldsymbol{c}$ in Eq. (14.12a) can be expressed as the $(n-1)$-degree polynomial

$$c(x) = c_1 x^{n-1} + c_2 x^{n-2} + \cdots + c_n \tag{14.13a}$$

The coefficients of the polynomial are either **0** or **1**, and they obey the following properties:

$$\begin{aligned} &\mathbf{0} + \mathbf{0} = \mathbf{0} && \mathbf{0} \times \mathbf{0} = \mathbf{0} \\ &\mathbf{0} + \mathbf{1} = \mathbf{1} + \mathbf{0} = \mathbf{1} && \mathbf{0} \times \mathbf{1} = \mathbf{1} \times \mathbf{0} = \mathbf{0} \\ &\mathbf{1} + \mathbf{1} = \mathbf{0} && \mathbf{1} \times \mathbf{1} = \mathbf{1} \end{aligned}$$

The code polynomial $c^{(i)}(x)$ for the code vector $\boldsymbol{c}^{(i)}$ in Eq. (14.12b) is

$$c^{(i)}(x) = c_{i+1} x^{n-1} + c_{i+2} x^{n-2} + \cdots + c_n x^i + c_1 x^{i-1} + \cdots + c_i \tag{14.13b}$$

One of the interesting properties of code polynomials is that when $x^i c(x)$ is divided by $x^n + 1$, the remainder is $c^{(i)}(x)$. We can verify this property as follows:

$$
\begin{array}{l}
xc(x) = c_1 x^n + c_2 x^{n-1} + \cdots + c_n x \\
\qquad\quad c_1 \\
\underline{x^n + 1}\,\overline{c_1 x^n + c_2 x^{n-1} + \cdots + c_n x} \\
\qquad\quad \underline{c_1 x^n + \qquad\qquad\qquad\qquad\qquad c_1} \\
\qquad\quad \underbrace{c_2 x^{n-1} + c_3 x^{n-2} + \cdots + c_n x + c_1}_{\text{remainder}}
\end{array}
$$

The remainder is clearly $c^{(1)}(x)$. In deriving this result, we have used the fact that subtraction amounts to summation when modulo-2 operations are involved. Continuing in this fashion, we can show that the remainder of $x^i c(x)$ divided by $x^n + 1$ is $c^{(i)}(x)$.

We now introduce the concept of code generator polynomial $g(x)$. Since each (n, k) codeword can be represented by a code polynomial

$$c(x) = c_1 x^{n-1} + c_2 x^{n-2} + \cdots + c_n$$

$g(x)$ is a code generator polynomial (of degree $n - k$), if for a data polynomial $d(x)$ of degree $k - 1$

$$d(x) = d_1 x^{k-1} + d_2 x^{k-2} + \cdots + d_k$$

we can generate code polynomial via

$$c(x) = d(x)\, g(x) \tag{14.14}$$

Notice that there are 2^k distinct code polynomials (or codewords). For cyclic code, a codeword after cyclic shift is still a codeword.

We shall now prove an important theorem in cyclic codes:

Cyclic Linear Block Code Theorem: *If $g(x)$ is a polynomial of degree $n - k$ and is a factor of $x^n + 1$ (modulo-2), then $g(x)$ is a generator polynomial that generates an (n, k) linear cyclic block code.*

Proof: For a data vector $(d_1, d_2, \ldots, d_k)$, the data polynomial is

$$d(x) = d_1 x^{k-1} + d_2 x^{k-2} + \cdots + d_k \tag{14.15}$$

Consider k polynomials

$$g(x), \quad xg(x), \quad \ldots, \quad x^{k-1} g(x)$$

which have degrees $n - k, n - k + 1, \ldots, n - 1$, respectively. Hence, a linear combination of these polynomials equals

$$d_1 x^{k-1} g(x) + d_2 x^{k-2} g(x) + \cdots + d_k g(x) = d(x)\, g(x) \tag{14.16}$$

Regardless of the data values $\{d_i\}$, $d(x)\,g(x)$ still has degree $n-1$ or less while being a multiple of $g(x)$. Hence, a codeword is formed by using Eq. (14.16). There are a total of 2^k such distinct polynomials (codewords) of the data polynomial $d(x)$, corresponding to 2^k data vectors. Thus, we have a linear (n, k) code generated by Eq. (14.14). To prove that this code is cyclic, let

$$c(x) = c_1 x^{n-1} + c_2 x^{n-2} + \cdots + c_n$$

be a code polynomial in this code [Eq. (14.16)]. Then,

$$\begin{aligned} xc(x) &= c_1 x^n + c_2 x^{n-1} + \cdots + c_n x \\ &= c_1(x^n + 1) + (c_2 x^{n-1} + c_3 x^{n-2} + \cdots + c_n x + c_1) \\ &= c_1(x^n + 1) + c^{(1)}(x) \end{aligned}$$

Because $xc(x)$ is $xd(x)g(x)$, and $g(x)$ is a factor of $x^n + 1, c^{(1)}(x)$ must also be a multiple of $g(x)$ and can also be expressed as $d(x)g(x)$ for some data vector $\boldsymbol{d}$. Therefore, $c^{(1)}(x)$ is also a code polynomial. Continuing this way, we see that $c^{(2)}(x), c^{(3)}(x), \ldots$ are all code polynomials generated by Eq. (14.16). Thus, the linear (n, k) code generated by $d(x)g(x)$ is indeed cyclic. ■

Example 14.3 Find a generator polynomial $g(x)$ for a (7, 4) cyclic code, and find code vectors for the following data vectors: **1010**, **1111**, **0001**, and **1000**.

In this case $n = 7$ and $n - k = 3$, and

$$x^7 + 1 = (x+1)(x^3 + x + 1)(x^3 + x^2 + 1)$$

For a (7, 4) code, the generator polynomial must be of the order $n - k = 3$. In this case, there are two possible choices for $g(x)$: $x^3 + x + 1$ or $x^3 + x^2 + 1$. Let us pick the latter, that is,

$$g(x) = x^3 + x^2 + 1$$

as a possible generator polynomial. For $\boldsymbol{d} = [\mathbf{1}\quad \mathbf{0}\quad \mathbf{1}\quad \mathbf{0}]$,

$$d(x) = x^3 + x$$

and the code polynomial is

$$\begin{aligned} c(x) &= d(x)g(x) \\ &= (x^3 + x)(x^3 + x^2 + 1) \\ &= x^6 + x^5 + x^4 + x \end{aligned}$$

Hence,

$$\boldsymbol{c} = \mathbf{1110010}$$

TABLE 14.4

d	*c*
1010	1110010
1111	1001011
0001	0001101
1000	1101000

Similarly, codewords for other data words can be found (Table 14.4). Note the structure of the codewords. The first k digits are not necessarily the data bits. Hence, this is not a systematic code.

In a systematic code, the first k digits are data bits, and the last $m = n - k$ digits are the parity check bits. Systematic codes are a special case of general codes. Our discussion thus far applies to general cyclic codes, of which systematic cyclic codes are a special case. We shall now develop a method of generating systematic cyclic codes.

Systematic Cyclic Codes

We shall show that for a systematic code, the codeword polynomial $c(x)$ corresponding to the data polynomial $d(x)$ is given by

$$c(x) = x^{n-k}d(x) + \rho(x) \tag{14.17a}$$

where $\rho(x)$ is the remainder from dividing $x^{n-k}d(x)$ by $g(x)$,

$$\rho(x) = \text{Rem}\,\frac{x^{n-k}d(x)}{g(x)} \tag{14.17b}$$

To prove this we observe that

$$\frac{x^{n-k}d(x)}{g(x)} = q(x) + \frac{\rho(x)}{g(x)} \tag{14.18a}$$

where $q(x)$ is of degree $k - 1$ or less. We add $\rho(x)/g(x)$ to both sides of Eq. (14.18a), and because $f(x) + f(x) = 0$ under modulo-2 operation, we have

$$\frac{x^{n-k}d(x) + \rho(x)}{g(x)} = q(x) \tag{14.18b}$$

or

$$q(x)g(x) = x^{n-k}d(x) + \rho(x) \tag{14.18c}$$

Because $q(x)$ is on the order of $k - 1$ or less, $q(x)g(x)$ is a code polynomial. As $x^{n-k}d(x)$ represents $d(x)$ shifted to the left by $n - k$ digits, the first k digits of this codeword are precisely $\boldsymbol{d}$, and the last $n - k$ digits corresponding to $\rho(x)$ must be parity check digits. This will become clear by considering a specific example.

Example 14.4 Construct a systematic (7, 4) cyclic code using a generator polynomial (see Example 14.3).

We use

$$g(x) = x^3 + x^2 + 1$$

Consider a data vector $\boldsymbol{d} = \mathbf{1010}$,

$$d(x) = x^3 + x$$

and

$$x^{n-k}d(x) = x^6 + x^4$$

Hence,

$$\begin{array}{r} x^3 + x^2 + 1 \qquad\qquad\qquad\qquad \leftarrow q(x) \\ x^3 + x^2 + 1 \overline{)\, x^6 + x^4} \\ \underline{x^6 + x^5 + x^3} \\ x^5 + x^4 + x^3 \\ \underline{x^5 + x^4 + x^2} \\ x^3 + x^2 \\ \underline{x^3 + x^2 + 1} \\ 1 \qquad\qquad\qquad\qquad \leftarrow \rho(x) \end{array}$$

Hence, from Eq. (14.17a),

$$\begin{aligned} c(x) &= x^3 d(x) + \rho(x) \\ &= x^3(x^3 + x) + 1 \\ &= x^6 + x^4 + 1 \end{aligned}$$

and

$$\boldsymbol{c} = \mathbf{1010001}$$

We could also have found the codeword $\boldsymbol{c}$ directly by using Eq. (14.18c). Thus, $c(x) = q(x)g(x) = (x^3+x^2+1)(x^3+x^2+1) = x^6+x^4+1$. We construct the entire code table in this manner (Table 14.5). This is quite a tedious procedure. There is, however, a shortcut, by means of the code generating matrix $\boldsymbol{G}$. We can use the earlier procedure to compute the codewords corresponding to the data words **1000**, **0100**, **0010**, **0001**. These are **1000110**, **0100011**, **0010111**, **0001101**. Now recognize that these four codewords are the four rows of $\boldsymbol{G}$. This is because $\boldsymbol{c} = \boldsymbol{d}\,\boldsymbol{G}$, and when $\boldsymbol{d} = \mathbf{1000}$, $\boldsymbol{d} \cdot \boldsymbol{G}$ is the first row of $\boldsymbol{G}$, and

TABLE 14.5

d	*c*
1111	**1111111**
1110	**1110010**
1101	**1101000**
1100	**1100101**
1011	**1011100**
1010	**1010001**
1001	**1001011**
1000	**1000110**
0111	**0111001**
0110	**0110100**
0101	**0101110**
0100	**0100011**
0011	**0011010**
0010	**0010111**
0001	**0001101**
0000	**0000000**

so on. Hence,

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Now, we can use $\boldsymbol{c} = \boldsymbol{dG}$ to construct the rest of the code table. This is an efficient method because it allows us to construct the entire code table from the knowledge of only k codewords.

Table 14.5 shows the complete code. Note that $d_{\min}$, the minimum distance between two codewords, is 3. Hence, this is a single-error correcting code, and 14 of these codewords can be obtained by successive cyclic shifts of the two codewords **1110010** and **1101000**. The remaining two codewords, **1111111** and **0000000**, remain unchanged under cyclic shift.

Generator Polynomial and Generator Matrix of Cyclic Codes

Cyclic codes can also be described by a generator matrix $\boldsymbol{G}$ (Probs. 14.3-6 and 14.3-7). It can be shown that Hamming codes are cyclic codes. Once the generator polynomial $g(x)$ has been given, it is simple to find the systematic code generator matrix $\boldsymbol{G} = [\boldsymbol{I} \ \ \boldsymbol{P}]$ by determining the parity submatrix $\boldsymbol{P}$:

$$\begin{aligned} &\text{1st row of } \boldsymbol{P}\text{: } \operatorname{Rem}\frac{x^{n-1}}{g(x)} \\ &\text{2nd row of } \boldsymbol{P}\text{: } \operatorname{Rem}\frac{x^{n-2}}{g(x)} \\ &\qquad\qquad \vdots \\ &k\text{th row of } \boldsymbol{P}\text{: } \operatorname{Rem}\frac{x^{n-k}}{g(x)} \end{aligned} \tag{14.19}$$

Consider a Hamming (7, 4, 3) code with generator polynomial

$$g(x) = x^3 + x + 1. \tag{14.20}$$

$$\text{Rem}\frac{x^6}{g(x)} = x^2 + 1$$

$$\text{Rem}\frac{x^5}{g(x)} = x^2 + x + 1$$

$$\text{Rem}\frac{x^4}{g(x)} = x^2 + x$$

$$\text{Rem}\frac{x^3}{g(x)} = x + 1$$

Therefore, the cyclic code generator matrix is

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{14.21}$$

Correspondingly, one form of its parity check matrix is

$$\boldsymbol{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{14.22}$$

Cyclic Code Generation

One of the advantages of cyclic codes is that their encoding and decoding can be implemented by means of such simple elements as shift registers and modulo-2 adders. A systematically generated code is described in Eqs. (14.17). It involves a division of $x^{n-k}d(x)$ by $g(x)$ that can be implemented by a dividing circuit consisting of a shift register with feedback connections according to the generator polynomial* $g(x) = x^{n-k} + g_1x^{n-k-1} + \cdots + g_{n-k-1}x + 1$. The gain g_k are either 0 or 1. An encoding circuit with $n-k$ shift registers is shown in Fig. 14.2. An understanding of this dividing circuit requires some background in linear sequential networks. An explanation of its functioning can be found in Peterson and Weldon.[3] The k data digits

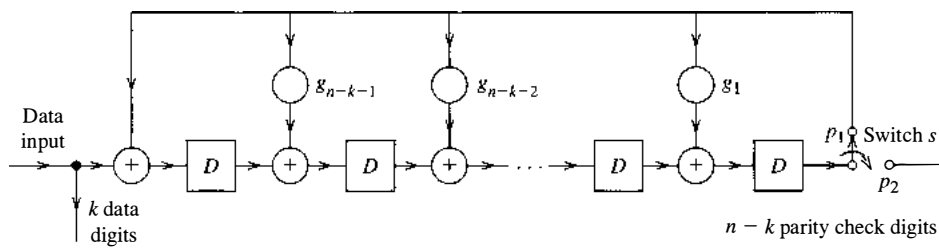


Figure 14.2 Encoder for systematic cyclic code.

* It can be shown that for cyclic codes, the generator polynomial must be of this form.

are shifted in one at a time at the input with the switch s held at position p_1. The symbol D represents a one-digit delay. As the data digits move through the encoder, they are also shifted out onto the output line, because the first k digits of the codeword are the data digits themselves. As soon as the last (or kth) data digit clears the last [or $(n-k)$th] register, all the registers contain the parity check digits. The switch s is now thrown to position p_2, and the $n-k$ parity check digits are shifted out one at a time onto the line.

Decoding

Every valid code polynomial $c(x)$ is a multiple of $g(x)$. In other words, $c(x)$ is divisible by $g(x)$. When an error occurs during the transmission, the received word polynomial $r(x)$ will not be a multiple of $g(x)$ if the number of errors in $\mathbf{r}$ is correctable. Thus,

$$\frac{r(x)}{g(x)} = m_1(x) + \frac{s(x)}{g(x)} \tag{14.23}$$

and

$$s(x) = \text{Rem}\,\frac{r(x)}{g(x)} \tag{14.24}$$

where the syndrome polynomial $s(x)$ has a degree $n-k-1$ or less.

If $e(x)$ is the error polynomial, then

$$r(x) = c(x) + e(x)$$

Remembering that $c(x)$ is a multiple of $g(x)$,

$$\begin{aligned} s(x) &= \text{Rem}\,\frac{r(x)}{g(x)} \\ &= \text{Rem}\,\frac{c(x)+e(x)}{g(x)} \\ &= \text{Rem}\,\frac{e(x)}{g(x)} \end{aligned} \tag{14.25}$$

Again, as before, a received word $\boldsymbol{r}$ could result from any one of the 2^k codewords and a suitable error. For example, for the code in Table 14.5, if $\boldsymbol{r} = \mathbf{0110010}$, this could mean $\boldsymbol{c} = \mathbf{1110010}$ and $\boldsymbol{e} = \mathbf{1000000}$, or $\boldsymbol{c} = \mathbf{1101000}$ and $\boldsymbol{e} = \mathbf{1011010}$, or 14 more such combinations. As seen earlier, the most likely error pattern is the one with the minimum weight (or minimum number of **1**s). Hence, here $\boldsymbol{c} = \mathbf{1110010}$ and $\boldsymbol{e} = \mathbf{1000000}$ is the correct decision.

It is convenient to prepare a decoding table, that is, to list the syndromes for all correctable errors. For any $\boldsymbol{r}$, we compute the syndrome from Eq. (14.24), and from the table we find the corresponding correctable error $\boldsymbol{e}$. Then we determine $\boldsymbol{c} = \boldsymbol{r} \oplus \boldsymbol{e}$. Note that computation of $s(x)$ [Eq. (14.24)] involves exactly the same operation as that required to compute $\rho(x)$ during encoding [Eq. (14.18a)]. Hence, the circuit in Fig. 14.2 can also be used to compute $s(x)$.

Example 14.5 Construct the decoding table for the single-error correcting (7, 4) code in Table 14.5. Determine the data vectors transmitted for the following received vectors $\boldsymbol{r}$: (a) **1101101**; (b) **0101000**; (c) **0001100**.

TABLE 14.6

e	s
1000000	110
0100000	011
0010000	111
0001000	101
0000100	100
0000010	010
0000001	001

The first step is to construct the decoding table. Because $n - k - 1 = 2$, the syndrome polynomial is of the second order, and there are seven possible nonzero syndromes. There are also seven possible correctable single-error patterns because $n = 7$. We can use

$$\boldsymbol{s} = \boldsymbol{e} \cdot \boldsymbol{H}^T$$

to compute the syndrome for each of the seven correctable error patterns. Note that (Example 14.4)

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

We can now compute the syndromes based on $\boldsymbol{H}$. For example, for $\boldsymbol{e} = \mathbf{1000000}$,

$$\begin{aligned} \boldsymbol{s} &= [\mathbf{1000000}] \cdot \boldsymbol{H}^T \\ &= \mathbf{110} \end{aligned}$$

In a similar way, we compute the syndromes for the remaining error patterns (see Table 14.6).

When the received word $\boldsymbol{r}$ is **1101101**, we can compute $s(x)$, either according to Eq. (14.24) or by simply applying the matrix product

$$\begin{aligned} \boldsymbol{s} &= \boldsymbol{r} \cdot \boldsymbol{H}^T \\ &= [\mathbf{1101101}] \cdot \boldsymbol{H}^T \\ &= \mathbf{101} \end{aligned}$$

Hence, From Table 14.6, this gives $\boldsymbol{e} = \mathbf{0001000}$, and

$$\boldsymbol{c} = \boldsymbol{r} \oplus \boldsymbol{e} = \mathbf{1101101} \oplus \mathbf{0001000} = \mathbf{1100101}$$

Because this code is systematic,

$$\boldsymbol{d} = \mathbf{1100}$$

In a similar way, we determine for $\boldsymbol{r} = \mathbf{0101000}$, $\boldsymbol{s} = \mathbf{110}$ and $\boldsymbol{e} = \mathbf{1000000}$; hence $\boldsymbol{c} = \boldsymbol{r} \oplus \boldsymbol{e} = \mathbf{1101000}$, and $\boldsymbol{d} = \mathbf{1101}$. For $\boldsymbol{r} = \mathbf{0001100}$, $\boldsymbol{s} = \mathbf{001}$ and $\boldsymbol{e} = \mathbf{0000001}$; hence $\boldsymbol{c} = \boldsymbol{r} \oplus \boldsymbol{e} = \mathbf{0001101}$, and $\boldsymbol{d} = \mathbf{0001}$.

Bose-Chaudhuri-Hocquenghen (BCH) Codes and Reed-Solomon Codes

The BCH codes are perhaps the best studied class of random error correcting cyclic codes. Moreover, their decoding procedure can be implemented simply. The Hamming code is a special case of BCH codes. These codes are described as follows: for any positive integers m and t ($t < 2^{m-1}$), there exists a t-error correcting (n, k) code with $n = 2^m - 1$ and $n - k \leq mt$. The minimum distance $d_{\min}$ between codewords is bounded by the inequality $2t+1 \leq d_{\min} \leq 2t+2$.

As a special case of *nonbinary* BCH codes, Reed-Solomon codes are by far the most successful forward error correction (FEC) codes in practice today. Reed-Solomon codes have found broad applications in digital storage (DVD, CD-ROM), high-speed modems, broadband wireless systems, and HDTV, among numerous others. The detailed treatment of BCH codes and Reed-Solomon codes requires extensive use of modern algebra and is beyond the scope of this introductory chapter. For in-depth discussion of BCH codes and Reed-Solomon codes, the reader is referred to the classic text by Lin and Costello.[2]

Cyclic Redundancy Check (CRC) Codes for Error Detection

One of the most widely used cyclic codes is the cyclic redundancy check codes for detection of data transmission errors. CRC codes are cyclic, designed to detect erroneous data packets at the receivers (often after error correction). To verify the integrity of the payload data block (packet), each data packet is encoded by CRC codes of length $n \leq 2^m - 1$. The most common CRC codes have m =12, 16, or 32 with code generator polynomial of the form

$$g(x) = (1 + x)g_c(x)$$

$$g_c(x) = \text{generator polynomial of a cyclic Hamming code}$$

To select a code generator matrix, the design criterion is to control the probability of undetected error events. In other words, the CRC codes must be able to detect the most likely error patterns such that the probability of undetected errors

$$P\left(\boldsymbol{e}\,\boldsymbol{H}^T = \mathbf{0} \middle| \boldsymbol{e} \neq \mathbf{0}\right) < \epsilon \tag{14.26}$$

where ϵ is set by the user according to its quality requirement. The most common CRC codes are given in Table 14.7 along with their generator matrices. For each frame of data bits at the transmitter, the CRC encoder generates a frame-checking sequence (FCS) of length 8, 12, 16, or 32 bits for error detection. For example, the IEEE 802.11 and IEEE 802.11b packets are checked by the 16-bit sequence of CRC-CCITT, whereas the IEEE 802.11a packets are checked by the CRC-32 sequence.

14.5 THE EFFECTS OF ERROR CORRECTION

Comparison of Coded and Uncoded Systems

It is instructive to compare the bit error probabilities (or bit error rate) when coded and uncoded schemes are under similar constraints of power and information rate.

Let us consider a t-error correcting (n, k) code. In this case, k information digits are coded into n digits. For a proper comparison, we shall assume that k information digits are transmitted in the same time interval over both systems and that the transmitted power S_i is also maintained

TABLE 14.7
Commonly Used CRC Codes and Corresponding Generator Polynomials

Code	Number of Bits in FCS	Generator Polynomial $g(x)$
CRC-8	8	$x^8 + x^7 + x^6 + x^4 + x^2 + 1$
CRC-12	12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$
CRC-16	16	$x^{16} + x^{15} + x^2 + 1$
CRC-CCITT	16	$x^{16} + x^{12} + x^5 + 1$
CRC-32	32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

the same for both systems. Because only k digits are required to be transmitted in the uncoded system (versus n over the coded one), the bit rate R_b is lower for the uncoded system than the coded one by a factor of k/n. This means that the bandwidth ratio of the uncoded system over the coded system is k/n. Clearly, the coded system sacrifices bandwidth for better reliability. On the other hand, the coded system sends n code bits for k information bits. To be fair, the total energy used by the n code bits must equal to the total energy used by the uncoded system for the k information bits. Thus, in the coded system, each code bit has E_b that is k/n times that of the uncoded system bit. We need to illustrate how error correction can reduce the originally higher bit error rate (BER) despite this loss of code bit energy.

Let P_{bu} and P_{bc} represent the raw data bit error probabilities in the uncoded and coded cases, respectively. For the uncoded case, the raw bit error rate is the final bit error rate P_{eu}.

For a t-error correcting (n, k) code, the raw BER can be reduced because the decoder can correct up to t bit errors in every n bits. We consider the ideal case that the decoder will not attempt to correct the codeword when there are more than t errors in n bits. This action of the ideal error correction decoder can reduce the average BER. Let $P(j, n)$ denote the probability of j errors in n digits. Then the average number of bit errors in each codeword after error correction is

$$\begin{aligned} \overline{n}_e &= E\{j \text{ bit errors in } n \text{ bits}\} \\ &= \sum_{j=t+1}^{n} j \cdot P(j, n) \end{aligned} \tag{14.27a}$$

Therefore the average BER after error correction should be

$$P_{\text{ec}} = \frac{\overline{n}_e}{n} \tag{14.27b}$$

Because there are $\binom{n}{j}$ ways in which j errors can occur in n digits (Example 8.6), we have

$$P(j, n) = \binom{n}{j} (P_{\text{bc}})^j (1 - P_{\text{bc}})^{n-j}$$

Based on Eq. (14.27a)

$$\bar{n}_e = \sum_{j=t+1}^{n} j\binom{n}{j}(P_{bc})^j(1-P_{bc})^{n-j} \tag{14.28a}$$

$$P_{ec} = \sum_{j=t+1}^{n} \frac{j}{n}\binom{n}{j}(P_{bc})^j(1-P_{bc})^{n-j}$$

$$= \sum_{j=t+1}^{n} \binom{n-1}{j-1}(P_{bc})^j(1-P_{bc})^{n-j} \tag{14.28b}$$

For $P_{bc} \ll 1$, the first term in the summation of Eq. (14.28b) dominates all the other terms, and we are justified in ignoring all but the first term. Hence,

$$P_{ec} = \binom{n-1}{t}(P_{bc})^{t+1}(1-P_{bc})^{n-(t+1)} \tag{14.29a}$$

$$\simeq \binom{n-1}{t}(P_{bc})^{t+1} \qquad \text{for} \quad P_{bc} \ll 1 \tag{14.29b}$$

For further comparison, we must assume some specific transmission scheme. Let us consider a coherent PSK scheme. In this case, for an additive white Gaussian noise (AWGN) channel,

$$P_{eu} = Q\left(\sqrt{\frac{2E_b}{\mathcal{N}}}\right) \tag{14.30a}$$

and because E_b for the coded case is k/n times that for the uncoded case,

$$P_{bc} = Q\left(\sqrt{\frac{2kE_b}{n\mathcal{N}}}\right) \tag{14.30b}$$

Hence,

$$P_{ec} = \binom{n-1}{t}\left[Q\left(\sqrt{\frac{2kE_b}{\mathcal{N}}}\right)\right]^{t+1} \tag{14.31a}$$

$$P_{eu} = Q\left(\sqrt{\frac{2E_b}{\mathcal{N}}}\right) \tag{14.31b}$$

To compare coded and uncoded systems, we could plot P_{eu} and P_{ec} as functions of the raw $E_b/\mathcal{N}$ (for the uncoded system). Because Eqs. (14.31) involve parameters t, n, and k, a proper comparison requires families of plots. For the case of a (7, 4) single-error correcting code ($t = 1, n = 7$, and $k = 4$), P_{ec} and P_{eu} in Eqs. (14.31) are plotted in Fig. 14.3 as a function of $E_b/\mathcal{N}$. Observe that the coded scheme is superior to the uncoded scheme at higher $E_b/\mathcal{N}$, but the improvement (about 1 dB) is not too significant. For large n and k, however, the coded scheme can become significantly superior to the uncoded one. For practical channels plagued by fading and impulse noise, stronger codes can yield substantial gains, as shown in our next example.

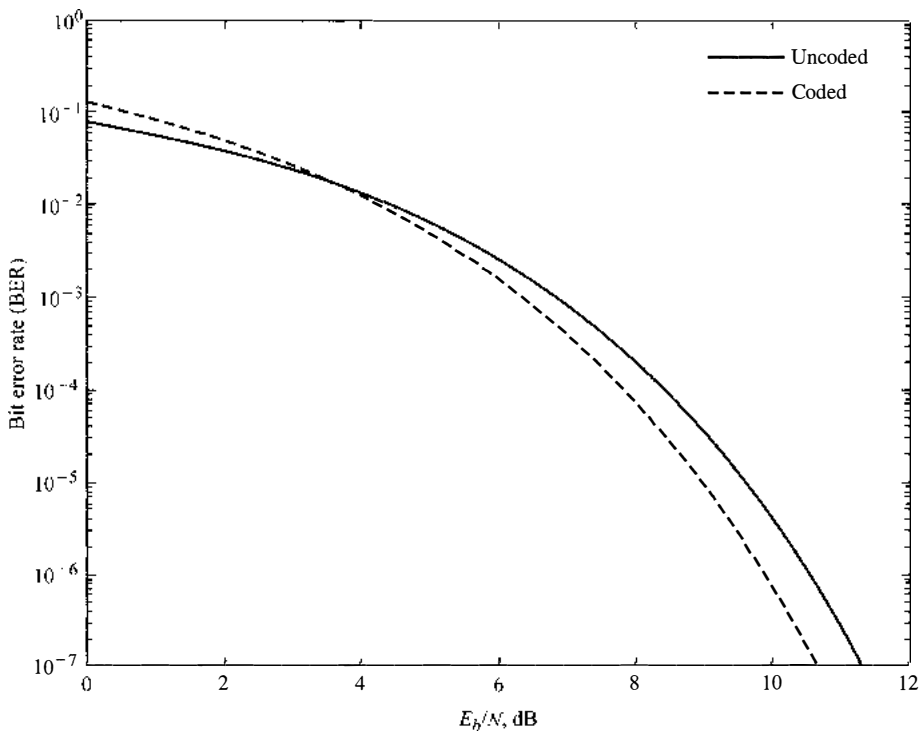


Figure 14.3 Performance comparison of coded (doshed) and uncoded (solid) systems.

It should be noted that the coded system performance of Fig. 14.3 is in fact a slightly optimistic approximation. The reason is that in analyzing its bit error rate, we assumed that the decoder will not take any action when the number of errors in each codeword exceeds t. In practice, the decoder never knows how many errors are in a codeword. Thus, the decoder will always attempt to correct the codeword, assuming that there are no more than t bit errors. This means that when there are more than t bit errors, the decoding process may even increase the number of errors. This counterproductive decoding effect is more likely when P_e is high at low $E_b/\mathcal{N}$. This effect will be shown later in Sec. 14.13 as a MATLAB exercise.

Example 14.6 Compare the performance of an AWGN BSC using a single-error correcting (15, 11) code with that of the same system using uncoded transmission, given that $E_b/\mathcal{N} = 9.0946$ for the uncoded scheme and coherent PSK is used to transmit the data.

From Eq. (14.31b),

$$P_{\text{eu}} = Q(\sqrt{18.1892}) = 1.0 \times 10^{-5}$$

and from Eq. (14.31a),

$$P_{\text{ec}} = 14\left[Q\left(\sqrt{\frac{11}{15}(18.1892)}\right)\right]^2$$
$$= 14(1.3 \times 10^{-4})^2 = 2.03 \times 10^{-7}$$

Note that the word error probability of the coded system is reduced by a factor of 50. On the other hand, if we wish to achieve the error probability of the coded transmission (2.03×10^{-7}) by means of the uncoded system, we must increase the transmitted power. If E_b' is the new value of E_b to achieve $P_{\text{eu}} = 2.03 \times 10^{-7}$,

$$P_{\text{eu}} = Q\left(\sqrt{\frac{2E_b'}{\mathcal{N}}}\right) = 2.03 \times 10^{-7}$$

This gives $E_b'/\mathcal{N} = 13.5022$. This is an increase over the old value of 9.0946 by a factor of 1.4846, or 1.716 dB.

Burst Error Detecting and Correcting Codes

Thus far we have considered detecting or correcting errors that occur independently, or randomly, in digit positions. On some channels, disturbances can wipe out an entire block of digits. For instance, a stroke of lightning or a human-made electrical disturbance can affect several adjacent transmitted digits. On magnetic storage systems, magnetic material defects usually affect a block of digits. Burst errors are those that wipe out some or all of a sequential set of digits. In general, random error correcting codes are not efficient for correcting burst errors. Hence, special **burst error correcting codes** are used for this purpose.

A burst of length b is defined as a sequence of digits in which the first digit and the bth digit are definitely in error, with the $b-2$ digits in between either in error or correct. For example, an error vector $\boldsymbol{e} = \mathbf{0010010100}$ has a burst length of 6.

It can be shown that for detecting all burst errors of length b or less with a linear block code of length n, b parity check bits are necessary and sufficient.[3] We shall prove the sufficiency part of this theorem by constructing a code of length n with b parity check digits that will detect a burst of length b.

To construct such a code, let us group k data digits into segments of b digits in length (Fig. 14.4). To this we add a last segment of b parity check digits, which are determined as follows. The modulo-2 sum of the ith digits from each segment (including the parity check segment) must be zero. For example, the first digits in the five data segments are **1**, **0**, **1**, **1**, and **1**. Hence, to obtain a modulo-2 sum zero, we must have **0** as the first parity check digit. We continue in this way with the second digit, the third digit, and so on, to the bth digit. Because parity check digits are a linear combination of data digits, this is a linear block code. Moreover, it is a systematic code.

It is easy to see that if a digit sequence of length b or less is in error, parity will be violated and the error will be detected (but not corrected), and the receiver can request retransmission of the digits lost. One of the interesting properties of this code is that b, the number of parity

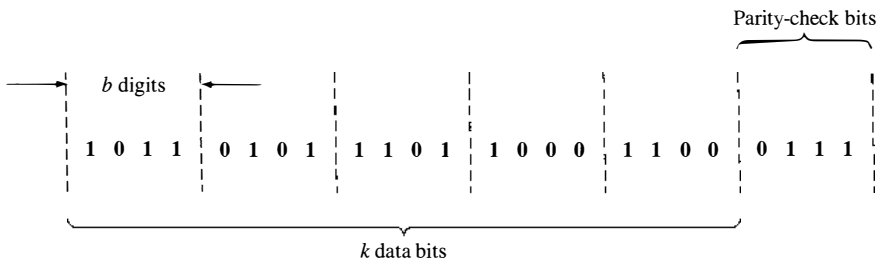


Figure 14.4 Burst error detection.

check digits, is independent of k (or n), which makes it a very useful code for such systems as packet switching, where the data digits may vary from packet to packet. It can be shown that a linear code with b parity bits detects not only all bursts of length b or less, but also a high percentage of longer bursts.[3]

If we are interested in correcting rather than detecting burst errors, we require twice as many parity check digits. According to the Hamming sphere reasoning: to correct all burst errors of length b or less, a linear block code must have at least $2b$ parity-check digits.[3]

14.6 CONVOLUTIONAL CODES

Convolutional (or recurrent) codes, introduced in 1955,[4] differ from block codes as follows. In a block code, the block of n code digits generated by the encoder in any particular time unit depends only on the block of k input data digits within that time unit. In a convolutional code, on the other hand, the block of n code digits generated by the encoder in a particular time unit depends not only on the block of k message digits within that time unit but also on the data digits within a previous span of $N - 1$ time units ($N > 1$). For convolutional codes, k and n are usually small. Convolutional codes can be devised for correcting random errors, burst errors, or both. Encoding is easily implemented by shift registers. As a class, convolutional codes are easier to encode.

14.6.1 Convolutional Encoder

A convolutional encoder with **constraint length** N consists of an N-stage shift register and ℓ modulo-2 adders. Figure 14.5 shows such an encoder for the case of $N = 3$ and $\ell = 2$. The message digits are applied at the input of the shift register. The coded digit stream is obtained at the commutator output. The commutator samples the ℓ modulo-2 adders in sequence, once during each input-bit interval. We shall explain this operation with reference to the input digits **11010**.

Initially, all the contents of the register are **0**. At time $k = 1$, the first data digit **1** enters the register. The content d_k shows **1** and all the other contents $d_{k-1} = 0$ and $d_{k-2} = 0$ are still unchanged. The two modulo-2 adders show encoder output $v_{k,1} = \mathbf{1}$ and $v_{k,2} = \mathbf{1}$ for this data input. The commutator samples this output. Hence, the encoder output is **11**. At $k = 2$,

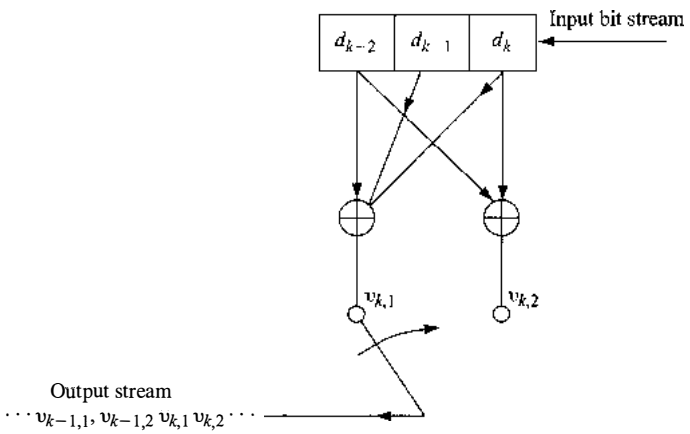


Figure 14.5 Convolutional encoder.

the second message bit **1** enters the register. It enters the register stage d_k, and the previous **1** in d_1 is now shifted to d_{k-1}, where as d_{k-2} is still **0**. The modulo-2 adders now show $v_{k,1} = \mathbf{0}$ and $v_{k,2} = \mathbf{1}$. Hence, the encoder output is **01**. In the same way, when the new digit **0** enters the register, we have $d_k = \mathbf{0}$, $d_{k-1} = \mathbf{1}$, and $d_{k-2} = \mathbf{1}$, and the encoder output is **01**.

Observe that each data digit influences N groups of ℓ digits in the output (in this case three groups of two digits). The process continues until the last data digit enters the stage d_k.* We cannot stop here, however. We add $N - 1$ number of **0**s to the input stream (dummy or augmented data) to make sure that the last data digit (**0** in this case) proceeds all the way through the shift register, to influence the N groups of v digits. Hence, when the input digits are **11010**, we actually apply (from left to right) **1101000**, which contains $N - 1$ augmented zeros to the input of the shift register. It can be seen that when the last digit of the augmented message stream enters d_k, the last digit of the message stream has passed through all the N stages of the register. The reader can verify that the encoder output is given by **11010100101100**. Thus, there are in all $n = (N + k - 1)\ell$ digits in the coded output for every k data digits. In practice, $k \gg N$, and, hence, there are approximately $k\ell$ coded output digits for every k data digits, giving an rate $\eta \simeq 1/\ell$.†

It can be seen that unlike the block encoder, the convolutional encoder operates on a continuous basis, and each data digit influences N groups of ℓ digits in the output.

Code Tree

The process of coding and decoding is considerably facilitated by what is known as the **code tree**, which shows the coded output for any possible sequence of data digits. The code tree for the encoder in Fig. 14.5 with $k = 5$ is shown in Fig. 14.6. When the first digit is **0**, the encoder output is **00**, and when it is **1**, the output is **11**. This is shown by the two tree branches that start at the initial node. The upper branch represents **0**, and the lower branch represents **1**. This convention will be followed throughout. At the terminal node of each of the two branches, we follow a similar procedure, corresponding to the second data digit. Hence, two branches initiate from each node, the upper one for **0** and the lower one for **1**. This continues until the kth data digit. From there on, all the input digits are **0** (augmented digit), and we have only one branch until the end. Hence, in all there are 32 (or 2^k) outputs corresponding to 2^k possible data vectors. The coded output for input **11010** can be easily read from this tree (the path shown dashed in Fig. 14.6).

Figure 14.6 shows that the code tree becomes repetitive after the third branch. This can be seen by noting that the two blocks enclosed in the dashed lines are identical. It means that the output from the fourth input digit is the same whether the first digit was **1** or **0**. This is not surprising, since when the fourth input digit enters the shift register, the first input digit is shifted out of the register and ceases to influence the output digits. In other words, the data vector $\mathbf{1}x_1x_2x_3x_4\cdots$ and the data vector $\mathbf{0}x_1x_2x_3x_4\cdots$ generate the same output after the third group of output digits. It is convenient to label the four third-level nodes (the nodes appearing at the beginning of the third branch) as nodes a, b, c, and d (Fig. 14.6). The repetitive structure begins at the fourth-level nodes and continues at the fifth-level nodes, whose behavior is similar to that of nodes a, b, c, and d at the third level. Hence, we label the fourth- and fifth-level nodes also as either a, b, c, or d. What this means is that at the fifth-level nodes, the first two data digits have become irrelevant; that is, any of the four combinations (**11**, **10**, **01**, or **00**) for the first two data digits will give the same output after the fifth node.

---

* For a systematic code, one of the output digits must be the data digit itself.

† In general, instead of shifting one digit at a time, b digits may be shifted at a time. In this case $\eta \simeq b/\ell$.

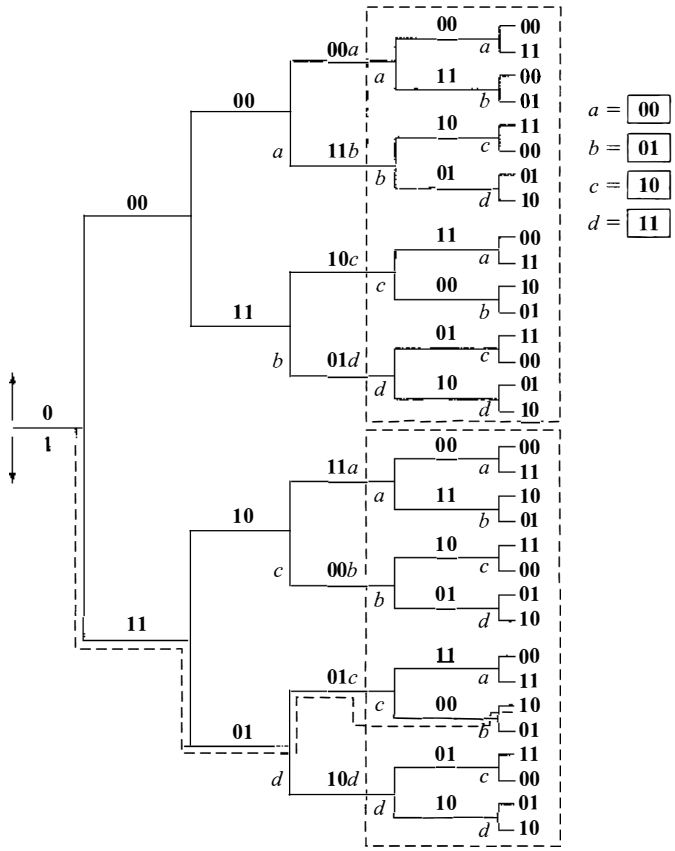


Figure 14.6
Code tree for the encoder in Fig. 14.5.

State Transition Diagram Representation

The encoder behavior can be seen from the perspective of a finite state machine with its state transition diagram. When a data bit enters the shift register (in d_k), the output bits are determined not only by the data bit in d_k, but by the two previous data bits already in stages d_{k-2} and d_{k-1}. There are four possible combinations of the two previous bits (in d_{k-2} and d_{k-1}): **00**, **01**, **10**, and **11**. We shall label these four states a, b, c, and d, respectively, as shown in Fig. 14.7a. Thus, when the previous two bits are **01** ($d_{k-2} = \mathbf{0}$, $d_{k-1} = \mathbf{1}$), the state is b, and so on. The number of states is equal to 2^{N-1}.

A data bit **0** or **1** generates four different outputs, depending on the encoder state. If the data bit is **0**, the encoder output is **00**, **10**, **11**, or **01**, depending on whether the encoder state is a, b, c, or d. Similarly if the data bit is **1**, the encoder output is **11**, **01**, **00**, or **10**, depending on whether the encoder state is a, b, c, or d. This entire behavior can be concisely expressed by the state transition diagram (Fig. 14.7b), a four-state directed graph that uniquely represents the input-output relation of this encoder. We label each transition path with a label of input bit over output bits:

$$\{d_k\}/\{v_{k,1}\ v_{k,2}\}$$

This way, we know exactly the input information bit d_k for each state transition and its corresponding encoder output bits. $\{v_{k,1}\ v_{k,2}\}$.

For instance, when the encoder is in state a, and we input **1**, the encoder output is **11**. Thus the transition path is labeled with $1/11$. The encoder now goes to state b for the next data bit because at this point the previous two bits become $d_{k-2} = \mathbf{0}$ and $d_{k-1} = \mathbf{1}$. Similarly, when

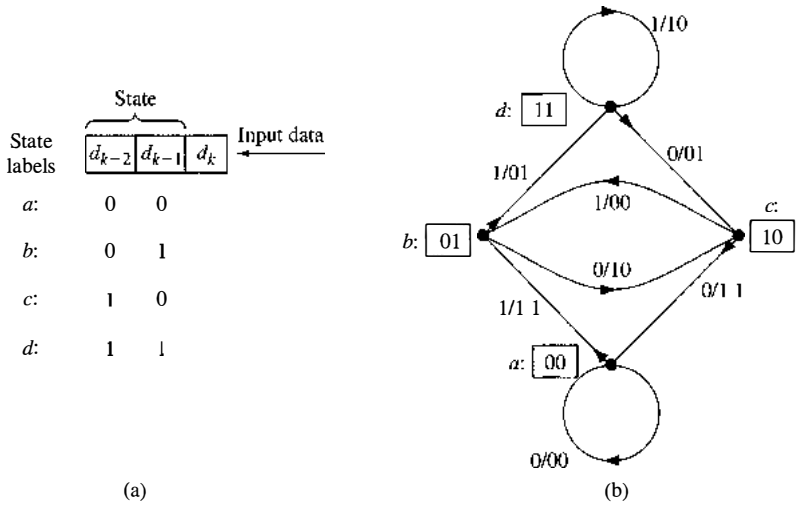


Figure 14.7
(a) State and (b) state transition diagram of the encoder in Fig. 14.5.

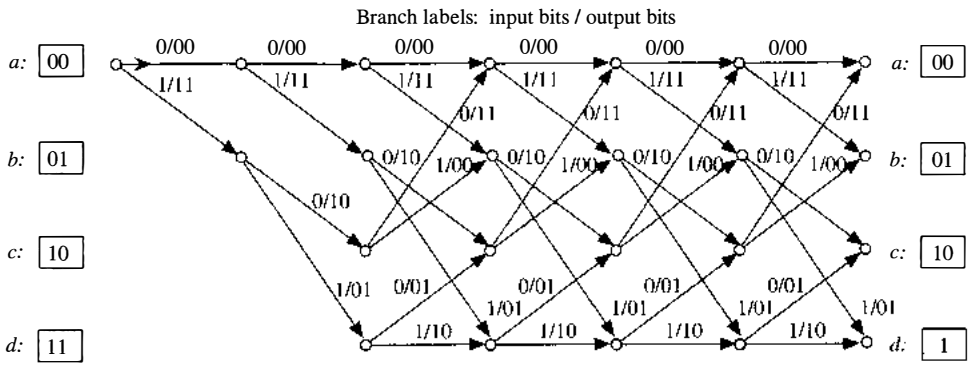


Figure 14.8
Trellis diagram for the encoder in Fig. 14.5.

the encoder is in state a and the input is **0**, the output is **00** (solid line), and the encoder remains in state a. Note that the encoder cannot go directly from state a to states c or d. From any given state, the encoder can go to only two states directly by inputting a single data bit. This is an extremely important observation, which will be used later. The encoder goes from state a to state b (when the input is **1**), or to state a (when the input is **0**), and so on. The encoder cannot go from a to c in one step. It must go from a to b to c, or from a to b to d to c, and so on. We can also verify these facts from the code tree. Figure 14.7b contains the complete information of the code tree.

Trellis Diagram

Another useful way of representing the code tree is the trellis diagram (Fig. 14.8). The diagram starts from scratch (all **0**s in the shift register, i.e., state a) and makes transitions corresponding to each input data digit. These transition **branches** are labeled just as we labeled the state transition diagram. Thus, when the first input digit is **0**, the encoder output is **00**, and the trellis branch is labeled **0/00**. This is readily seen from Fig. 14.7b. We continue this way with the second input digit. After the first two input digits, the encoder is in one of the four states a, b, c, or d, as shown in Fig. 14.8. If the encoder is in state a (previous two data digits **00**), it goes to state b if the next input bit is **1** or remains in state a if the next input bit is **0**. In so doing, the encoder output is **11** (a to b) or **00** (a to a). Note that the structure of the trellis diagram is completely repetitive, as expected, and can be readily drawn by using the state diagram in Fig. 14.7b.

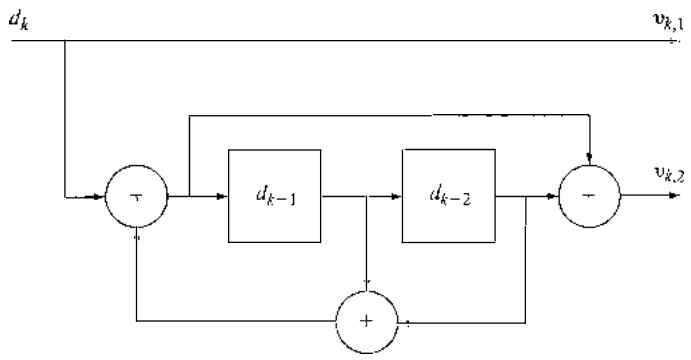


Figure 14.9 A recursive systematic convolutional (RSC) encoder.

It should be noted that the convolutional encoder can have feedback branches. In fact, feedback in the convolutional encoder generates the so-called recursive code. As shown in Fig. 14.9, the data bit can have a direct path to the output bit. The bits from the top branch will be the information bits from the input directly. This code is therefore systematic. This encoder leads to a recursive **systematic** convolutional (RSC) code. It can be shown (see Prob. 14.5-3) that the RSC encoder can also be represented by a similar state transition diagram and a trellis diagram. Consequently, recursive convolutional code can be decoded by using the methods described next for nonrecursive convolutional codes.

14.6.2 Decoding Convolutional Codes

We shall consider two important techniques: (1) maximum likelihood decoding (Viterbi algorithm) and (2) sequential decoding. Although both are known as hard-decision decoders, the Viterbi algorithm (VA) is much more flexible and can be easily adapted to allow soft input and to generate soft output decisions, to be described later in this chapter.

Maximum Likelihood Decoding: The Viterbi Algorithm

Among various decoding methods for convolutional codes, Viterbi's maximum likelihood algorithm[5] is one of the best techniques for digital communications when computational complexity dominates in importance. It permits major equipment simplification while obtaining the full performance benefits of maximum likelihood decoding. The decoder structure is relatively simple for short constraint length N, making decoding feasible at relatively high rates of up to 10 Gbit/s.

In AWGN channels, the maximum likelihood receiver requires selecting a codeword closest to the received word. For a long sequence of received data representing k message bits and 2^k codewords, direct implementation of maximum likelihood decision (MLD) involves storage of 2^k words and their comparison to the received sequence. This computational need places a severe burden on MLD receivers for large values of k for convolutionally encoded data frames, typically in the order of hundreds or thousands of bits!

Viterbi made a major simplification for MLD. We shall use the convolutional code example of Figs. 14.5 and 14.7 to illustrate the fundamental operations of the VA. First, we stress that each path that traverses through the trellis represents a valid codeword. The objective of MLD is to find the best path through the trellis that is **closest** to the received data bit sequence. To understand this, consider again the trellis diagram in Fig. 14.8. Our problem is as follows: given a received sequence of bits, we need to find a path in the trellis diagram with the output digit sequence that agrees best with the received sequence. The minimum (Hamming) distance path represents the most likely sequence up to stage i.

As shown in Fig. 14.8, each codeword is a trellis path that should start from state a (**00**). Because every path at stage i must grow out of the paths at stage $i-1$, the optimum path to each state at stage i must contain one of the best paths to each of the four states at stage $i-1$. In short, the optimum path to each state at stage i is a descendant of the predecessors at stage $i-1$. All optimum paths at any stage $i+i_0$ are descendants of the optimum path at stage i. Hence, only the **best path** to each state need be stored at a given stage. There is no reason to store anything but the optimum path to each state at every stage because nonoptimum paths would only increase the metric of path distance to the received data sequence.

In the special example of Fig. 14.7, its trellis diagram (Fig. 14.8) shows that each of the four states (a, b, c, and d) has only two predecessors; that is, each state can be reached only through two previous states. More importantly, since only the four best surviving paths (one for each state) exist at stage $i-1$, there are only two possible paths for each state at stage i. Hence, by comparing the total Hamming distances (from the received sequence) of the two paths, we can find the optimum path with the minimum Hamming distance for every state at stage i that corresponds to a codeword which is closest to the received sequence up to stage i. The optimum path to each state is known as the **survivor** or the **surviving path**.

Example 14.7 We now study a decoding example of the Viterbi algorithm for maximum likelihood decoding of the convolutional code generated by the encoder of Fig. 14.5. Let the first 12 received digits be **01 10 11 00 00 00**, as shown in Fig. 14.10a–e. Showing the received digits along with the branch output bits makes it easier to computer the branch Hamming distance in each stage.

We start from the initial state of a (**00**). Every stage of the decoding process is to find the optimum path to the four states given the 2 received bits during the stage. There are two possible states leading to each state in any given state. The survivor with the minimum Hamming distance is retained (solid line), whereas the other path with larger distance is discarded (dashed line). The Hamming distance of each surviving path is labeled at the end of a stage to each of the four states.

- After two stages, there is exactly one optimum (surviving) path to each state (Fig. 14.10a). The Hamming distances of the surviving paths are labeled as 2, 2, 1, and 3, respectively.
- Each state at stage 3 has two possible paths (Fig. 14.10b). We keep the optimum path with the minimum distance (solid line). The distances of the two possible paths (from top to bottom) arriving at each state are given in the minimization label. For example, for state a, the first path (dashed line from a) has Hamming distance of $2+2=4$, whereas the second path (solid line from c) has the distance of $1+0=1$.
- Repeat the same step for stages 4, 5, and 6, as illustrated in Fig. 14.10c–e.
- The **final** optimum path after stage 6 is identified as the shaded solid path with minimum distance of 2 ending in state a, as shown in Fig. 14.10(e). Thus, the MLD output should be

$$\text{Codeword: } \mathbf{11\ 10\ 11\ 00\ 00\ 00} \tag{14.32a}$$

$$\text{Information bits: } \mathbf{1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0} \tag{14.32b}$$

Note that there are only four contending paths (the four survivors at states a, b, c, and d) until the end of stage 6. All four paths merged up till stage 3. This means that the first three branch selections are the most reliable. In fact, continuing the VA when given additional received bits will **not** change the first three branches and their associated decoder outputs.

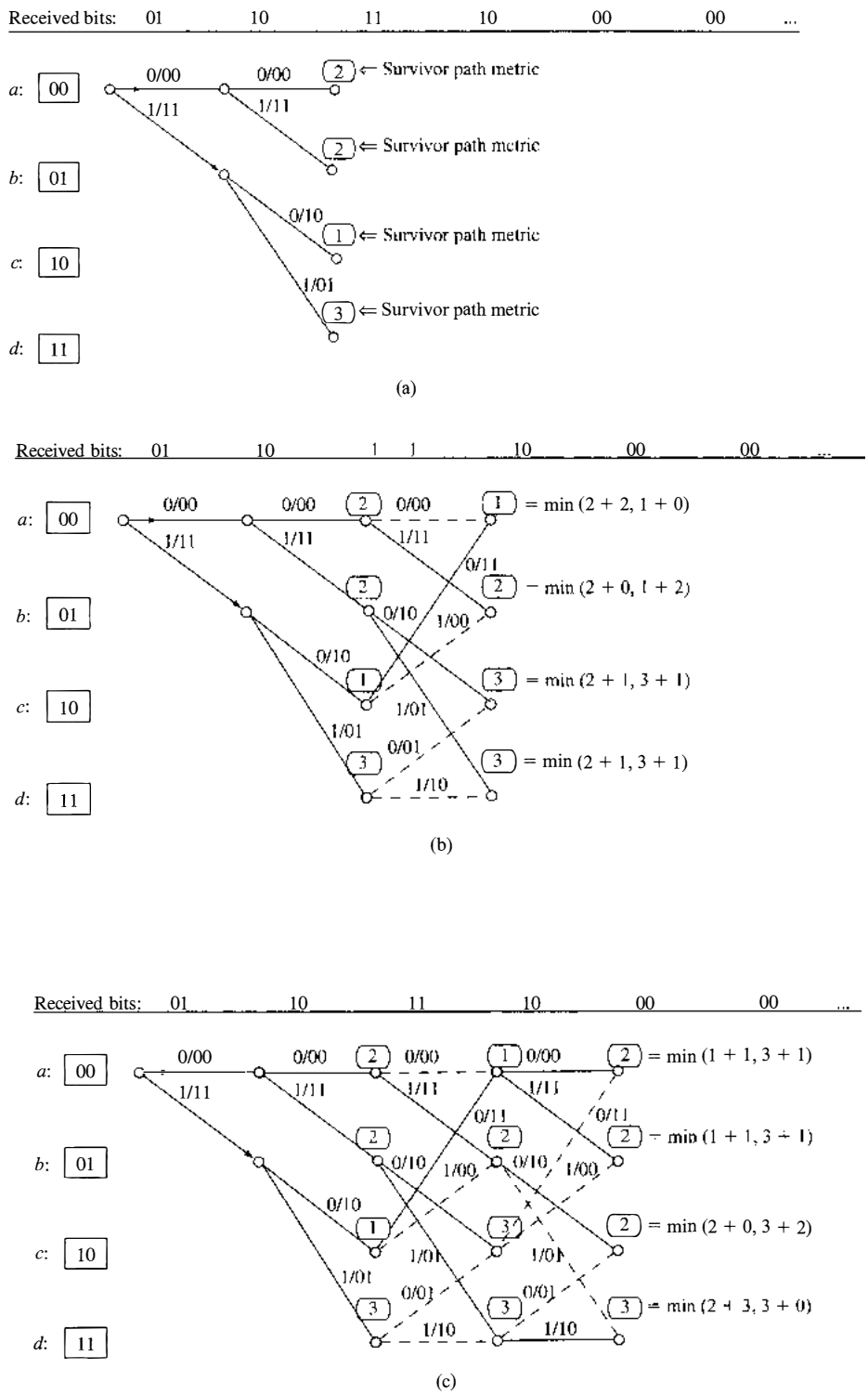
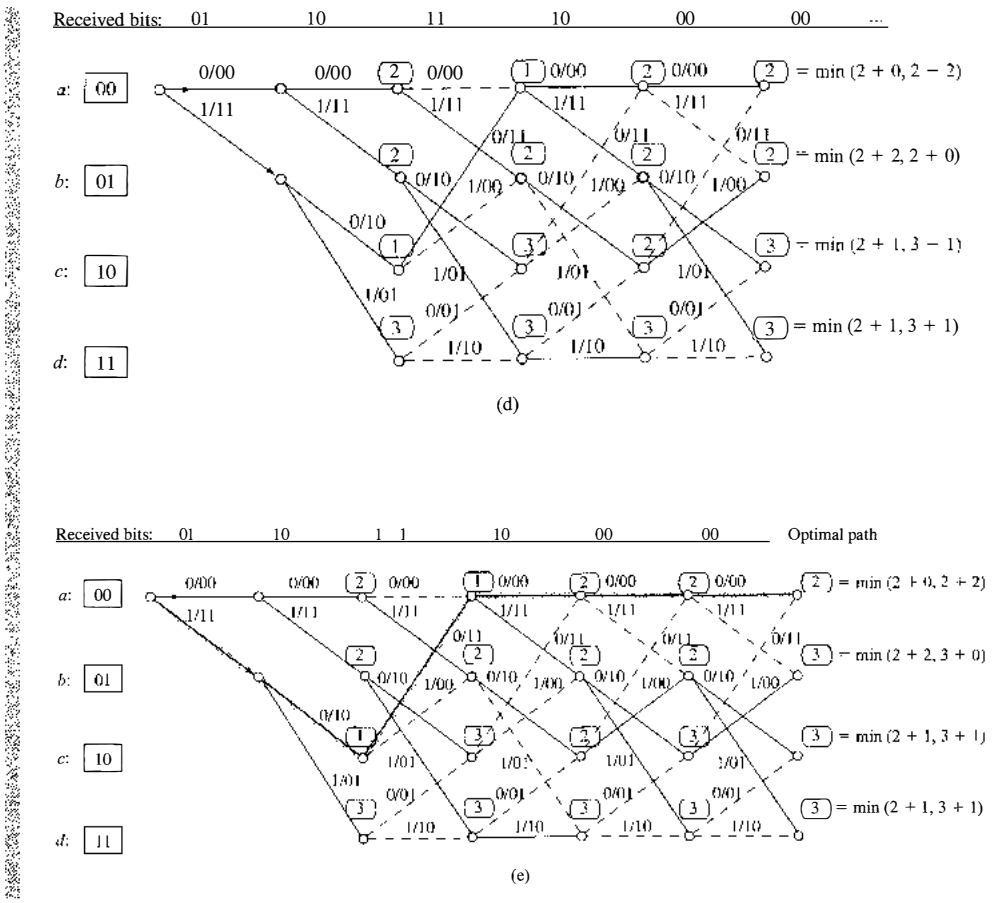


Figure 14.10 Viterbi decoding example in Fig. 14.5: (a) stages 1 and 2; (b) stage 3; (c) stage 4; (d) stage 5; (e) stage 6.

Figure 14.10 Continued



In the preceding example, we have illustrated how to progress from one stage to the next by determining the optimum path (survivor) leading to each of the state. When these survivors do merge, the merged branches represent the **most reliable** MLD outputs. For the later stages that do not exhibit a merged path, we are ready to make a maximum likelihood decision based on the received data bits up to that stage. This process, known as truncation, is designed to a force a decision on one path among all the survivors without leading to a long decoding delay. One way to make a truncated decision is to take the minimum distance path as in Eq. (14.32). Another alternative is to rely on extra codeword information. In Fig. 14.10e, if the encoder always forces the last two data digits to be **00**, then we can consider only the survivor ending at state a.

With the Viterbi algorithm, storage and computational complexity are proportional to 2^{N-1} and are very attractive for constraint length $N < 10$. To achieve very low error probabilities, longer constraint lengths are required, and sequential decoding (to be discussed next) may become attractive.

Sequential Decoding

In sequential decoding, a technique proposed by Wozencraft, the complexity of the decoder increases linearly rather than exponentially. To explain this technique, let us consider an encoder

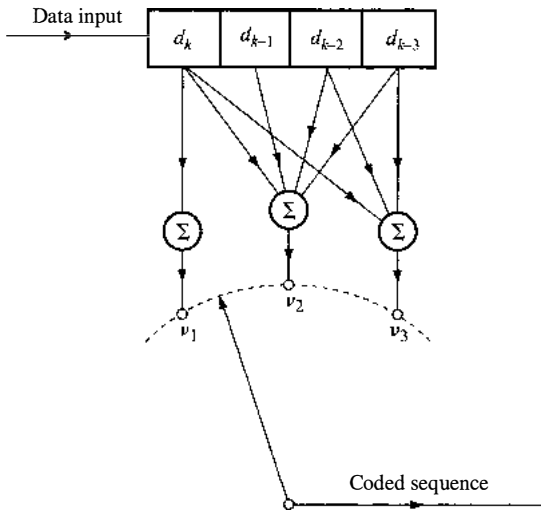


Figure 14.11
Convolutional encoder.

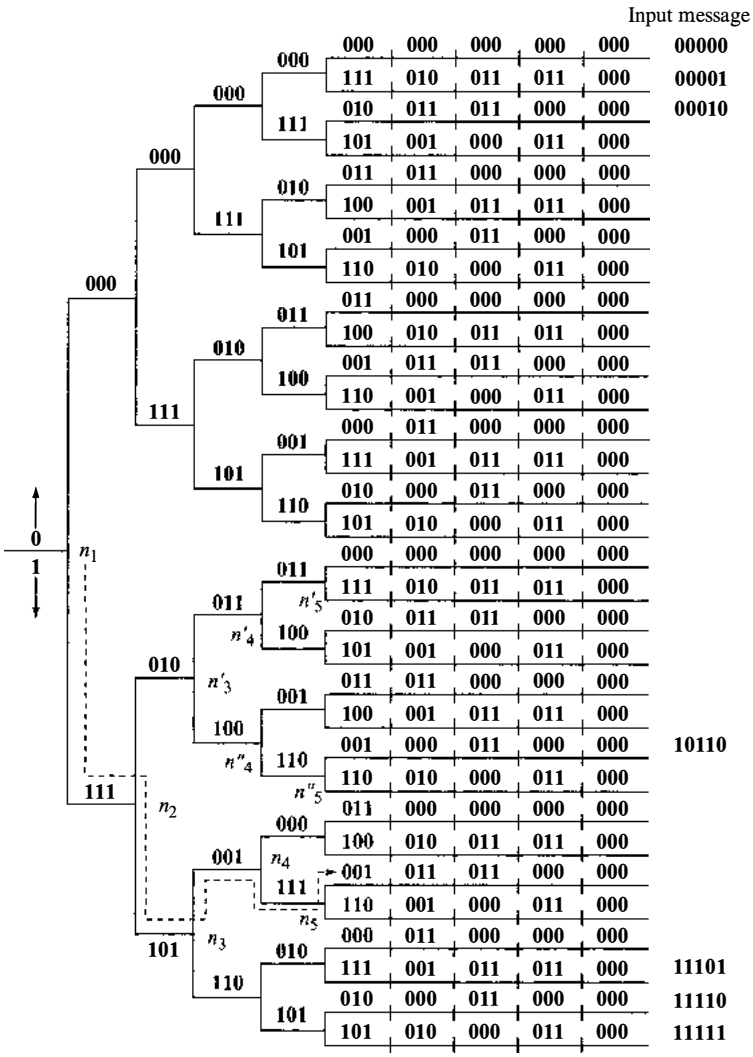


Figure 14.12
Code tree for the encoder in Fig. 14.11.

with $N = 4$ and $\ell = 3$ (Fig. 14.11). The code tree for this encoder is shown in Fig. 14.12. Each data digit generates three ($\ell = 3$) output digits but affects four groups of three digits (12 digits) in all.

In this decoding scheme, we observe only three (or ℓ) digits at a time to make a tentative decision, with readiness to change our decision if it creates difficulties later. A sequential detector acts much like a driver who occasionally makes a wrong choice at a fork in the road, but quickly discovers the error (because of road signs), goes back, and takes the other path.

Applying this insight to our decoding problem, the analogous procedure would be as follows. We look at the first three received digits. There are only two paths of three digits from the initial node n_1. We choose that path whose sequence is at the shortest Hamming distance from the first three received digits. We thus progress to the most likely node. From this node there are two paths of three digits. We look at the second group of the three received digits and choose that path whose sequence is closest to these received digits. We progress this way until the fourth node. If we were unlucky enough to have a large number of errors in a certain received group of ℓ digits, we will take a wrong turn, and from there on we will find it more difficult to match the received digits with those along the paths available from the wrong node. This is the clue to the realization that an error has been made. Let us explain this by an example.

Suppose a data sequence **11010** is encoded by the encoder in Fig. 14.11. Because $N = 4$, we add three dummy **0**s to this sequence so that the augmented data sequence is **11010000**. The coded sequence will be (see the code tree in Fig. 14.12)

111 101 001 111 001 011 011 000

Let the received sequence be

101 011 001 111 001 011 011 000

There are three bit errors: one in the first group and two in the second group. We start at the initial node n_1. The first received group **101** (one error) being closer to **111**, we make a correct decision to go to node n_2. But the second group **001** (two errors) is closer to **010** than to **101** and will lead us to the wrong node n_3' rather than to n_3. From here on we are on the wrong track, and, hence, the received digits will not match any path starting from n_3'. The third received group is **001** and does not match any sequence starting at n_3' (viz., **001** and **100**). But it is closer to **011**. Hence, we go to node n_4'. Here again the fourth received group **111** does not match any group starting at n_4' (viz., **011** and **100**). But it is closer to **011**. This takes us to node n_5'. It can be seen that the Hamming distance between the sequence of 12 digits along the path $n_1n_2n_3'n_4'n_5'$ and the first 12 received digits is 4, indicating four errors in 12 digits (if our path is correct). Such a high number of errors should immediately make us suspicious. If P_e is the digit error probability, then the expected number of errors n_e in d digits is $P_e d$. Because P_e is on the order of 10^{-4} to 10^{-6}, four errors in 12 digits is unreasonable. Hence, we go back to node n_3' and try the lower branch, leading to n_5''. This path, $n_1n_2n_3'n_4''n_5''$, is even worse than the previous one: it gives five errors in 12 digits. Hence, we go back even farther to node n_2 and try the path leading to n_3 and farther. We find the path $n_1n_2n_3n_4n_5$, giving three errors. If we go back still farther to n_1 and try alternate paths, we find that none yields less than five errors. Thus, the correct path is taken as $n_1n_2n_3n_4n_5$, giving three errors. This enables us to decode the first transmitted digit as **1**. Next, we start at node n_2, discard the first three received digits, and repeat the procedure to decode the second transmitted digit. We repeat this until all the digits have been decoded.

The next important question concerns the criterion for deciding when the wrong path is chosen. The plot of the expected number of errors n_e as a function of the number of decoded

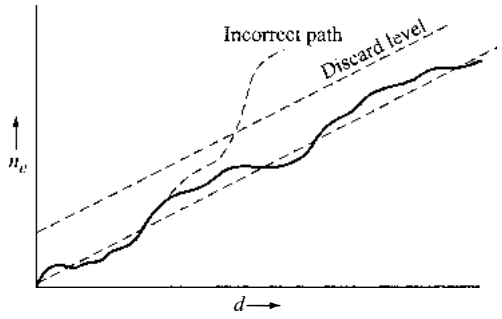


Figure 14.13 Setting the threshold in sequential decoding.

digits d is a straight line ($n_e = P_e d$) with slope P_e, as shown in Fig. 14.13. The actual number of errors along the path is also plotted. If the errors remain within a limit (the discard level), the decoding continues. If at some point the errors exceed the discard level, we go back to the nearest decision node and try an alternate path. If errors still increase beyond the discard level, we then go back one more node along the path and try an alternate path. The process continues until the errors are within the set limit. By making the discard level very stringent (close to the expected error curve), we reduce the average number of computations. On the other hand, if the discard level is made too stringent, the decoder will discard all possible paths in some extremely rare cases of an unusually large number of errors due to noise. This difficulty is usually resolved by starting with a stringent discard level. If on rare occasions the decoder rejects all paths, the discard level can be relaxed little by little until one of the paths is acceptable.

It can be shown that the error probability in this scheme decreases exponentially as N, whereas the system complexity grows only linearly with k. The code rate is $\eta \simeq 1/\ell$. It can be shown that for $\eta < \eta_o$, the average number of incorrect branches searched per decoded digit is bounded, whereas for $\eta > \eta_o$ it is not; hence η_o is called the computational cutoff rate.

There are several disadvantages to sequential decoding:

1. The number of incorrect path branches, and consequently the computation complexity, is a random variable depending on the channel noise.
2. To make storage requirements easier, the decoding speed has to be maintained at 10 to 20 times faster than the incoming data rate. This limits the maximum data rate capability.
3. The average number of branches can occasionally become very large and may result in a storage overflow, causing relatively long sequences to be erased.

A third technique for decoding convolutional codes is **feedback decoding**, with threshold decoding[6] as a subclass. Threshold decoders are easily implemented. Their performance, however, does not compare favorably with the previous two methods.

14.7 TRELLIS DIAGRAM OF BLOCK CODES

Whereas a trellis diagram is connected with convolutional code in a direct and simple way, a **syndrome** trellis can also be constructed for a binary linear (n, k) block code according to its parity check matrix[7] $\boldsymbol{H}$. The construction can be stated as follows:

- Let $(c_1, c_2, \ldots, c_n)$ be a codeword of the block code.
- Let $\boldsymbol{H} = \left[\vec{h}_1\ \vec{h}_2\ \cdots\ \vec{h}_n\right]$ be the $(n-k) \times n$ parity check matrix with columns $\{\vec{h}_i\}$.

- There are $\min(2^k, 2^{n-k})$ possible states in the trellis.
- The state of a codeword at instant i is determined by the codeword and the parity check matrix by syndrome from the first codeword bit to the ith codeword bit:

$$z_i = c_1\vec{h}_i \oplus c_2\vec{h}_2 \oplus \cdots \oplus c_i\vec{h}_i. \tag{14.33}$$

Note that this syndrome trellis, unlike the state transition trellis of convolutional code, is typically nonrepeating. In fact, it always starts from the "zero" state and ends in "zero" state. Indeed, this trellis is a time-varying trellis. We use an example to illustrate the construction of a syndrome trellis.

Example 14.8 Consider a Hamming (7, 4, 3) code with parity check matrix

$$\boldsymbol{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{14.34}$$

Sketch the trellis diagram for this block code.

For this code, there are 3 error syndrome bits defining a total of $2^3 = 8$ states. Denote the eight states as $(S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7)$. There are $2^k = 2^4 = 16$ total codewords with 7 code bits that are in the null-space of the parity check matrix H. By enumerating all 16 codewords, we can follow Eq. (14.33) to determine all the paths through the trellis.

The corresponding time-varying trellis diagram is shown in Fig. 14.14. Notice that each path corresponds to a codeword. We always start from state S_0 initially and end at the state S_0. Unlike the case of convolutional code, it is not necessary to label the trellis branches in this case. Whenever there is a state transition between different states, the branch automatically corresponds to a "**1**" code bit. When a state stays the same, then the transition branch corresponds to a "**0**" code bit.

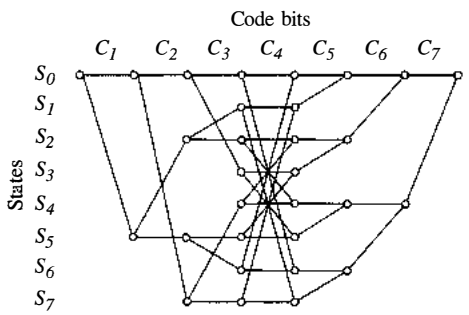


Figure 14.14 Trellis diagram of a Hamming (7, 4, 3) code with parity check matrix of Eq. (14.34).

Once we have a trellis diagram, the Viterbi decoding algorithm can be implemented for the MLD of the block code at reduced complexity. Maximum likelihood detection of block codes can perform better than a syndrome-based decoder. Keep in mind that the example we show is a very short code that does not benefit from Viterbi decoding. Clearly, the Viterbi algorithm makes more sense when one is decoding a long code.

14.8 CODE COMBINING AND INTERLEAVING

Simple and short codes can be combined in various ways to generate longer or more powerful codes. Certainly there are many possible ways of combining multiple codes. In this section, we briefly describe several of the most common methods of code construction through code combining.

Interleaving Codes for Correcting Burst and Random Errors

One of the simplest and yet most effective tools for code combining is **interleaving**, the process of reordering or shuffling (multiple) codewords generated by the encoder. Thus, a burst of bit errors will be affecting multiple codewords instead of one. The purpose of interleaving is to disperse a large burst of errors over multiple codewords such that each codeword needs to correct only a fraction of the error burst. This is because, in general, random error correcting codes are designed to tackle sporadic errors in each codeword. Unfortunately, in most practical systems, we have errors of both kinds. Among methods proposed to simultaneously correct random and burst errors, interleaving is simple and effective.

For an (n, k) code, if we interleave λ codewords, we have what is known as a $(\lambda n, \lambda k)$ **interleaved code**. Instead of transmitting codewords one by one, we group λ codewords and interleave them. Consider, for example, the case of $\lambda = 3$ and a two-error correcting (15, 8) code. Each codeword has 15 digits. We group codewords to be transmitted in groups of three. Suppose the first three code words to be transmitted are $\boldsymbol{x} = (x_1, x_2, \ldots, x_{15})$, $\boldsymbol{y} = (y_1, y_2, \ldots, y_{15})$, and $\boldsymbol{z} = (z_1, z_2, \ldots, z_{15})$, respectively. Then instead of transmitting $\boldsymbol{xyz}$ in sequence as $x_1, x_2, \ldots, x_{15}, y_1, y_2, \ldots, y_{15}, z_1, z_2, \ldots, z_{15}$, we transmit $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, \ldots, x_{15}, y_{15}, z_{15}$. This can be explained graphically by Fig. 14.15, where λ codewords (three in this case) are arranged in rows. In normal transmission, we transmit one row after another. In the interleaved case, we transmit columns (of λ elements) in sequence. When all the 15 (n) columns are transmitted, we repeat the procedure for the next λ codewords to be transmitted.

To explain the error correcting capabilities of this interleaved code, we observe that the decoder will first remove the interleaving and regroup the received digits as $x_1, x_2, \ldots, x_{15}, y_1, y_2, \ldots, y_{15}, z_1, z_2, \ldots, z_{15}$. Suppose the digits in the shaded boxes in Fig. 14.15 were in error. Because the code is a two-error correcting code, up to two errors in each row will be corrected. Hence, all the errors in Fig. 14.15 are correctable. We see that there are two random, or independent, errors and one burst of length 4 in all the 45 digits transmitted. In general, if the original (n, k) code is t-error correcting, the interleaved code can correct any combination of t bursts of length λ or less.

Because of the interleaver described in Fig. 14.15 takes a block bits and generates output sequence in a fixed orderly way, interleavers of this kind are known as **block interleavers**. The total memory length of the interleaver is known as the **interleaving depth**. Interleavers

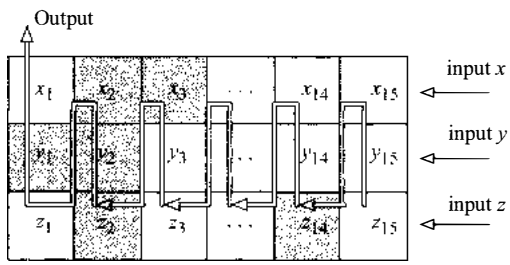


Figure 14.15 A block (nonrandom) interleaver for correcting random and burst errors.

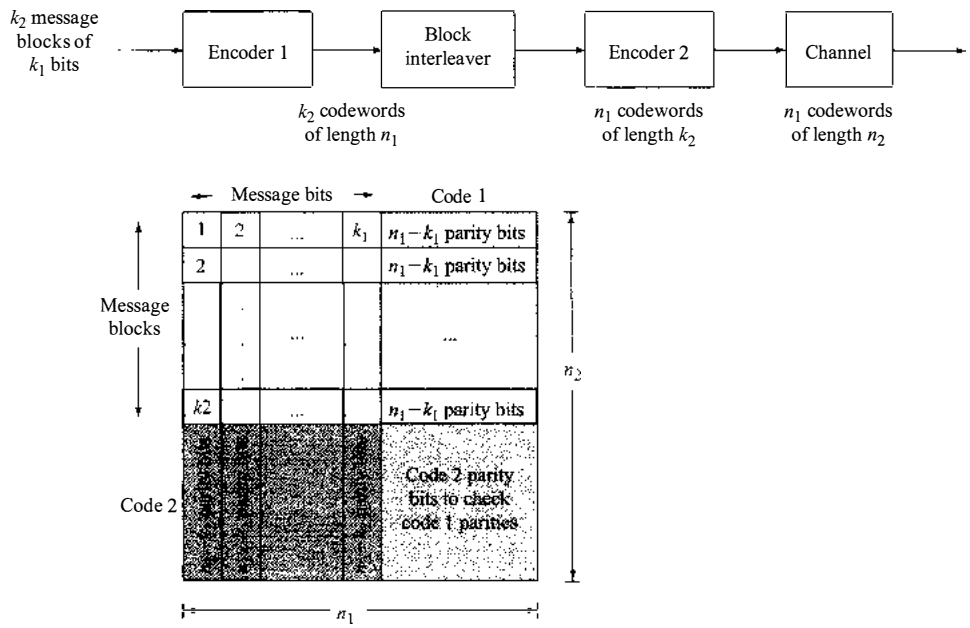


Figure 14.16 Product code formed by two encoders separated by a block interleaver.

with larger depths can better handle longer bursts of errors, at the cost of larger memory and longer encoding and decoding delays. A more general interleaver can pseudorandomly reorder the data bits inside the interleaver and output the bits in an order known to both the transmitter and the receiver. Such an interleaver is known as a **random interleaver**. Random interleavers are generally more effective in combating both random and burst errors. Because they do not generate outputs following a fixed order, there is a much smaller probability of receiving a burst of error bits in a codeword because of certain random error patterns.

Product Code

Interleaved code can be generalized by further encoding the interleaved codewords. The resulting code can be viewed as a large codeword that must satisfy two parity checks (or constraints). Figure 14.16 illustrates how to form a product code from two systematic block codes that are known as component codes. The first is an (n_1, k_1) code and the second is an (n_2, k_2) code. More specifically, a rectangular block of $k_1 \times k_2$ message bits is encoded by two encoders. First, k_2 blocks of k_1 message bits is encoded by the first encoder into k_2 codewords of the (n_1, k_1) code. Then an $n_1 \times k_2$ block interleaver sends n_1 blocks of k_2 bits into the second encoder. The second (n_2, k_2) encoder adds $n_2 - k_2$ parity bits for each of the n_1 blocks, generating n_1 codewords of the (n_2, k_2) code for the channel to transmit.

The use of a product code is a simple way to combine two block codes into a single more powerful code. In a product code, every code bit is constrained by two sets of parities, one from each of the two codes.

Concatenated Codes

Note from the block diagram of the product code that a block interleaver connects the two component codes. More generally, as shown in Fig. 14.17, the two component codes need not

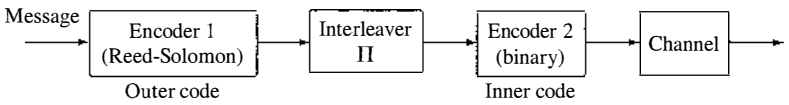


Figure 14.17 Concatenated code with a nonbinary outer code and a binary inner code.

be limited to binary block codes, and a more general interleaver Π can be used. The resulting code is known as a concatenated code. Indeed, Forney[8] proposed concatenating one binary and one nonbinary code to construct a much more powerful code. It is clear that product codes are a special class of concatenated codes with binary component codes and a block interleaver.

In this serial concatenation, encoder 2 is known as the inner code whereas encoder 1 is known as the outer code. The most successful concatenation as proposed by Forney[8] uses a Reed-Solomon outer code and a binary convolutional inner code. The concatenated code can be decoded separately by first decoding the inner code before de-interleaving and decoding the outer code. More complex ways of iterative decoding are also possible to potentially achieve better performance.

14.9 SOFT DECODING

Thus far, we have focused on decoding methods that generate hard decisions based on either maximum likelihood or syndrome-based algebraic decoding. Hard-decision decoding refer to the fact that the decoder generates **only** the most likely codeword without providing the relative confidence of this decoded codeword with respect to other possibilities. In other words, the hard-decision decoded codeword does not indicate how confident the decoder is about this decision. A stand-alone decoder can function as a hard-decision decoder because its goal is to provide the best candidate as the decoded codeword. It does not have to indicate how much confidence can be placed in this decision.

In practice, however, a decoder is often operating in conjunction with other decoders and other receiver units. This means that the decoded codeword not only must meet the constraint of the current parity check matrix, its output must also satisfy other constraints such as those imposed by the parities of different component codes in a concatenated error correction code. By providing more than just one hard decision, a soft-decision decoder can output multiple possible codewords, each with an associated reliability (likelihood) metric. This kind of soft decoding can allow other units in the receiver to jointly select the best codeword by utilizing the "soft" (reliability) information from the decoder along with other relevant constraints that the codeword must satisfy.

It is more convenient to illustrate the soft decoding concept by means of a BPSK modulation example. Let us revisit the optimum receiver of Sec. 10.6. We will focus on the special case of binary modulation with modulated data symbol represented by $b_i = \pm 1$ under additive white Gaussian noise channel. Let $c_{j,i}$ denote the ith code bit of the jth codeword $\boldsymbol{c}_j$. Because the modulation is BPSK, the relationship between the code bit $c_{j,i}$ and its corresponding modulated symbol $b_{j,i}$ is simply

$$b_{j,i} = 2 \cdot c_{j,i} - 1$$

Assuming that the receiver filter output signal is ISI free, then the received signal samples r_i corresponding to the transmission of the n-bit (n, k) codeword $[c_{j,1}\ c_{j,2}\ \cdots\ c_{j,n}]$ can be

written as

$$r_i = \sqrt{E_b} b_{j,i} + w_i \qquad i = 1, 2, \ldots, n \tag{14.35}$$

Here w_i is an AWGN sample. We use C to denote the collection of all valid codewords. Based on the optimum receiver of Sec. 10.6 [Eq.(10.91) and Fig. 10.18], the maximum likelihood decoder (MLD) of the received signal under coding corresponds to

$$\begin{aligned} \boldsymbol{c} &= \arg\max_{\boldsymbol{c}_j \in C} \sum r_i b_{j,i} \\ &= \arg\max_{\boldsymbol{c}_j \in C} \sum r_i (2c_{j,i} - 1) \\ &= \arg\max_{\boldsymbol{c}_j \in C} 2 \sum r_i c_{j,i} - \sum r_i \\ &= \arg\max_{\boldsymbol{c}_j \in C} \sum r_i c_{j,i} \end{aligned} \tag{14.36}$$

Among all the 2^k codewords, the soft MLD not only can determine the most likely codeword as the output, it should also preserve the metric

$$M_j = \sum r_i b_{j,i}$$

as the **relative likelihood** of the codeword $\mathbf{c}_j$ during the decoding process. Although equivalent to the distance measure, this (correlation) metric should be maximized for MLD. Unlike distance, the correlation metric can be both positive and negative.

Although the soft MLD appears to be a straightforward algorithm to implement, its computational complexity is affected by the size of the code. Indeed, when the code is long with a very large k, the computational complexity grows exponentially because 2^k metrics must be calculated. For many practical block codes, this requirement becomes unmanageable when the code length exceeds several hundred bits.

To simplify this optimum decoder, Chase proposed several types of suboptimum soft decoding algorithms[9] that are effective at significantly reduced computational cost. The first step of the Chase algorithms is to derive temporary hard bit decisions based on the received samples r_i. These temporary bits do not necessarily form a codeword. In other words, find

$$\bar{\boldsymbol{y}} = [y_1 \quad y_2 \quad \cdots \quad y_n] \tag{14.37a}$$

where

$$y_i = \text{sign}\,(r_i) \qquad i = 1, 2, \ldots, n \tag{14.37b}$$

Each individual bit decision has reliability $|r_i|$. These temporary bits $\{y_i\}$ are sent to an algebraic decoder based on, for example, error syndromes. The result is an initial codeword $\bar{c}_0 = [\bar{c}_{0,1} \quad \bar{c}_{0,2} \quad \cdots \quad \bar{c}_{0,n}]$. This step is exactly the same as a conventional hard-decision decoder. However, Chase algorithms allow additional modifications to the hard decoder input $\boldsymbol{y}$ by flipping the least reliable bits. Flipping means changing a code bit from **1** to **0** and from **0** to **1**.

The idea of soft decoding is to provide multiple candidate codewords, each with an associated reliability measure. Chase algorithms generate most likely flip patterns to be used to

modify the hard decoder input $\boldsymbol{y}$. Each flip pattern $\boldsymbol{e}_i$ consists of **1**s in bit positions to be flipped and **0**s in the remaining bit positions. For each flip pattern $\boldsymbol{e}_i$, construct

$$\bar{\boldsymbol{c}}_i = \text{hard decision}\,(\boldsymbol{y}_i) \oplus \boldsymbol{e}_i \tag{14.38a}$$

and compute the corresponding reliability metric

$$M_i = \sum_{j=1}^{n} r_j \cdot (2\bar{c}_{i,j} - 1) \tag{15.38b}$$

The codeword with the maximum M_i is the decoded output.

There are three types of Chase algorithm. First, we sort the bit reliability from low to high:

$$|r_{i_1}| \le |r_{i_2}| \le \cdots \le |r_{i_n}| \tag{14.39}$$

Type 1 Test all flipping patterns of weight less than or equal to $(d_{\min} - 1)$.

Type 2 Identify the $\lfloor d_{\min}/2 \rfloor$ least reliable bit positions $\{i_1\ i_2\ \cdots\ i_{\lfloor d_{\min}/2 \rfloor}\}$. Test all flipping patterns of weight less than or equal to $\lfloor d_{\min}/2 - 1 \rfloor$.*

Type 3 Test flipping patterns of weight $w = 1, 3, \ldots, d_{\min} - 1$ by placing **1**s in the w least reliable bit positions.

The block diagram of Chase algorithms is shown in Fig. 14.18. The three Chase algorithms differ only in how the flipping patterns are generated. In addition, we should note that Chase decoders can exchange reliability and likelihood information with other receiver units in a joint effort to improve the decoding performance. From the input end, the set of flipping patterns can take additional suggestions from other receiver units. From the output end, multiple codeword candidates, along with their reliability metrics, can be sent to additional decoding units for further processing and eventual elimination.

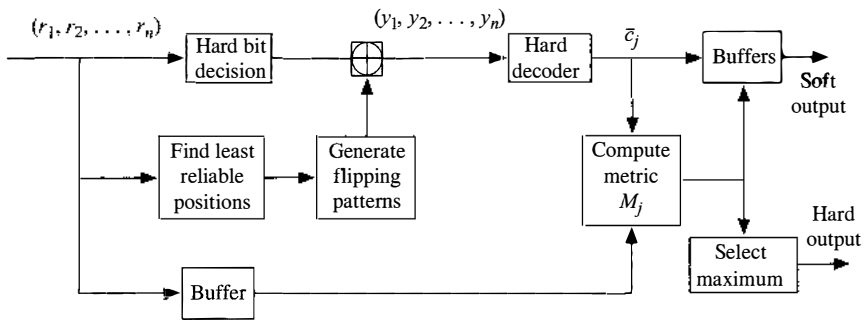


Figure 14.18 Block diagram of Chase soft-decoding algorithms.

* The operation $\lfloor \cdot \rfloor$ is often known as the "floor." In particular, $\lfloor x \rfloor$ represents the largest integer less than or equal to x.

14.10 SOFT-OUTPUT VITERBI ALGORITHM (SOVA)

Chase algorithms can generate multiple candidate codewords and the associated reliability metrics. The metric information can be exploited by other receiver processing units to determine the final decoded codeword. If the decoder can produce soft reliability information on every decoded bit, then it can be much better utilized jointly with other soft-output decoders and processors. Unlike Chase algorithms, soft-output Viterbi algorithms (SOVA)[10] and the *maximum a posterior*; (MAP) algorithms are two most general soft decoding methods to produce bit reliability information. We first describe the principles of SOVA here.

The most reliable and informative soft bit information is the log-likelihood ratio (LLR) of a particular code bit c_i based on the received signal vector

$$\boldsymbol{r} = (r_1, \ r_2, \ \ldots, \ r_n)$$

In other words, the LLR[11] as defined by

$$\Lambda(c_i) = \log \frac{P\,[c_i = 1 | \mathbf{r} = \boldsymbol{r}]}{P\,[c_i = 0 | \mathbf{r} = \boldsymbol{r}]} \tag{14.40}$$

indicates the degree of certainty by the decoder on the decision of $c_i = 1$. The degree of certainty various from $-\infty$ when $P\,[c_i = 0|\boldsymbol{r}] = 1$ to $+\infty$ when $P\,[c_i = 0|\boldsymbol{r}] = 0$

Once again, we consider the BPSK case in which $(2c_i - 1) = \pm 1$ is the transmitted data and

$$\mathrm{r}_i = (2c_i - 1) + \mathrm{w}_i \qquad i = 1, \ 2, \ \ldots, \ n \tag{14.41}$$

where w_i is the AWGN sample. Similar to the Chase algorithms, the path metric is computed by the correlation between $\{\mathrm{r}_i\}$ and the BPSK signal $\{c_i\}$. In other words, based on the received data samples $\{r_i\}$, we can estimate

$$\text{path metric between stages } n_1 \text{ and } n_2 = \sum_{j=n_1+1}^{n_2} r_j \cdot (2\bar{c}_j - 1) \tag{14.42}$$

Like the traditional Viterbi algorithm, the SOVA decoder operates on the corresponding trellis of the (convolutional or block) code. SOVA consists of a forward step and a backward step. During the **forward step**, as in the conventional Viterbi algorithm, SOVA first finds the most likely sequence (survivor path). Unlike conventional VA, which stores only the surviving path metrics at the states in the current stage, SOVA stores the metric of every surviving path leading to a state for all stages.

To formulate the idea formally, denote

$$S_\ell(i) = \text{ state } \ell \text{ at stage (time) } i$$

For each survivor at state S_ℓ in stage i, we will determine the forward path metric leading to this state. These forward metrics ending in state ℓ at time i are denoted as $M_\ell^f(i)$. The maximum total path metric at the final state of the forward VA, denoted $M_{\max}$, corresponds to the optimum forward path. During the backward step, SOVA then applies VA backward from the terminal

(final) state at stage K and ends at the initial state at stage 0, also storing the backward metrics ending in state ℓ at stage i as $M_\ell^b(i)$.

Since the likely value of the information bit $d_i = 0,\ 1$ that leads to the transition between state $S_{\ell_a}(i-1)$ and state $S_{\ell_b}(i)$ has been identified by VA during the forward step, the metric of information bit $M_i(d_i)$ can be fixed as total path metric

$$M_i(d_i) = M_{\max}$$

Our next task is to determine the best path and the corresponding maximum path metric $M_i(1-d_i)$ **if** the opposite information bit value of $1-d_i$ is chosen instead at stage i

$$M(1-d_i) = \max_{\ell_a \xrightarrow{1-d_i} \ell_b} \left[M_{\ell_a}^f \left(S_{\ell_a}(i-1) \right) + B_{\ell_a,\,\ell_b} + M_{\ell_b}^b \left(S_{\ell_b}(i) \right) \right] \tag{14.43}$$

where $B_{\ell_a,\,\ell_b}$ is the path distance from state transition ℓ_a to ℓ_b with respect to the received sample r_i. The maximization is over all state transitions denoted by $(\ell_a \xrightarrow{1-d_i} \ell_b)$ that can be caused by the information bit value of $1-d_i$ at stage i.

This step allows us to find the best alternative path through the trellis if the alternative bit value $1-d_i$ is selected. Now that we have both $M_i(d_i)$ and $M_i(1-d_i)$ for every stage i, likelihood of every information bit is proportional to the metric difference

$$\begin{aligned} \Lambda_i = M_i(1) - M_i(0) &= (2d_i - 1)\left[M_i(d_i) - M_i(1-d_i)\right] \\ &= (2d_i - 1)\left[M_{\max} - M_i(1-d_i)\right] \end{aligned} \tag{14.44}$$

Hence, the log-likelihood ratio Λ_i can generated by SOVA for every information bit d_i. We now can use the survivor path to determine the LLR [Eq. (14.40] for every bit in this most likely sequence. The basic concept of finding the best alternative surviving path caused by an information bit value of $1-d_i$ is illustrated in Fig. 14.19.

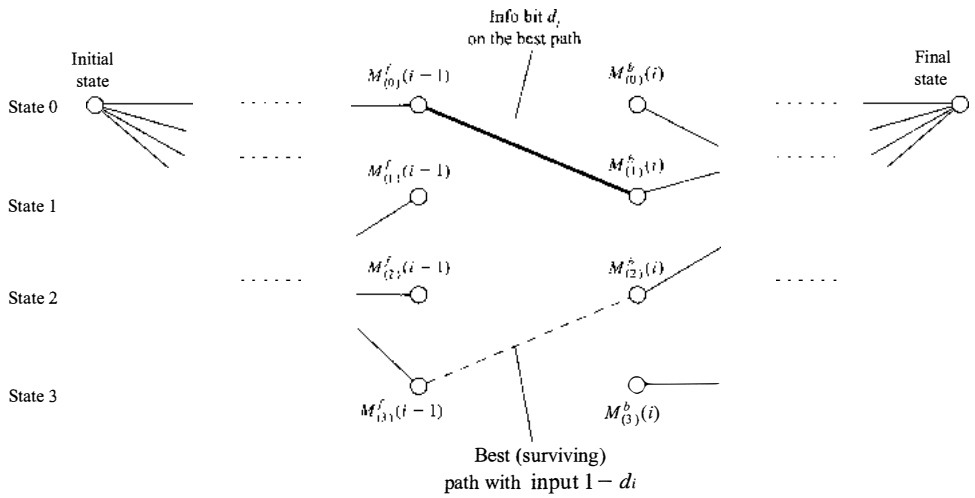


Figure 14.19
Block diagram of Chase soft decoding algorithms.

14.11 TURBO CODES

As we briefly mentioned in Sec. 14.1, turbo codes represent one of the major breakthroughs[12] in coding theory over the past several decades. The mechanism that made turbo codes possible is its simplified decoder. Turbo codes would not have been possible without a soft decoding algorithm. In fact, a short paper published more than 30 years earlier by Bahl, Cocke, Jelinek, and Raviv[13] played a major role. Their maximum a posterior (MAP) soft decoding algorithm is known as the BCJR algorithm. Before describing the essence of turbo codes, we introduce the fundamentals of BCJR algorithm.

BCJR Algorithm for MAP Detection

Our description of the BCJR MAP algorithm is based on the presentation by Bahl, Cocke, Jelinek, and Raviv.[13] We first assume that a sequence of information data bits denoted by

$$d_1\ d_2\ \cdots\ d_N \tag{14.45}$$

The information bit $\{d_i\}$ are encoded into codeword bits $\{v_i\}$, which are further modulated into (complex) modulated data symbols $\{b_i\}$. In the general case, we simply note that there is a mapping of

$$\{d_i\} \longrightarrow \{b_i\} \tag{14.46}$$

In the special case of BPSK, $b_i = \pm 1$.

The modulated data symbols are transmitted in an i.i.d. noise channel, and the received signal samples are

$$\mathrm{r}_i = b_i + \mathrm{w}_i \tag{14.47}$$

in which w_i are i.i.d. noise samples. Following MATLAB notation, we denote the received data

$$\vec{r}_{k_1:k_2} = (r_{k_1}, r_{k_1+1}, \ldots, r_{k_2})$$
$$\vec{r} = (r_1, r_2, \ldots, r_N)$$

Because the data symbols and the channel noise are i.i.d., we conclude that the conditional probability depends only on the current modulated symbol

$$p(r_i|b_i, \vec{r}_{1:i-1}) = p(r_i|b_i) \tag{14.48}$$

The (convolutional or block) code is represented by a trellis diagram in which $S_i = m$ denotes the event that the trellis state is m at time i. The transition probability between state m' and m from stage $i-1$ to stage i is represented by

$$P[S_i = m|S_{i-1} = m']$$

The definition of the state trellis means that S_i is a Markov process.* Based on the properties of Markov processes, and the knowledge that $\vec{r}_{i+1:N}$ and $\vec{r}_{1:i}$ are independent, we have the

* A random process x_k is a Markov process if its conditional probability satisfies

$$p_{\mathrm{x}_k|\mathrm{x}_{k-1},\ldots}(x_k|x_{k-1},\cdots) = p_{\mathrm{x}_k|\mathrm{x}_{k-1}}(x_k|x_{k-1})$$

In other words, a Markov process has a very short memory. All the information relevant to x_k from entire its history is available in its immediate past x_{k-1}.

following simplifications of the conditional probabilities:

$$p\left(\vec{r}_{i+1:N}\middle|S_i = m,\ S_{i-1} = m',\ \vec{r}_{1:i}\right) = p\left(\vec{r}_{i+1:N}\middle|S_i = m\right) \tag{14.49a}$$

$$p\left(r_i,\ S_i = m\middle|S_{i-1} = m',\ \vec{r}_{1:i-1}\right) = p\left(r_i,\ S_i = m\middle|S_{i-1} = m'\right) \tag{14.49b}$$

The MAP detector needs to determine the log-likelihood ratio

$$\Lambda(d_i) \triangleq \log \frac{P[d_i = 1|\vec{r}\,]}{P[d_i = 0|\vec{r}\,]} = \log \frac{p\,(d_i = 1,\ \vec{r}\,)}{p\,(d_i = 0,\ \vec{r}\,)} \tag{14.50}$$

We are now ready to explain the operations of the BCJR algorithm. First, let $\Omega_i(u)$ denote the set of all possible state transitions from $S_{i-1} = m'$ to $S_i = m$ when $d_i = u$ ($u = 0, 1$). There are only two such sets for $d_i = 1$ and $d_i = 0$. We can see that

$$\begin{aligned} p\,(d_i = 1,\ \vec{r}\,) &= \sum_{(m',\,m)\in\Omega_i(1)} p\left(S_{i-1} = m',\ S_i = m,\ \vec{r}\right) \\ &= \sum_{(m',\,m)\in\Omega_i(1)} p\left(S_{i-1} = m',\ \vec{r}_{1:i},\ S_i = m,\ \vec{r}_{i+1:N}\right) \\ &= \sum_{(m',\,m)\in\Omega_i(1)} p\left(S_{i-1} = m',\ \vec{r}_{1:i-1},\ S_i = m,\ r_i\right) \\ &\quad \cdot p\left(\vec{r}_{i+1:N}\middle|S_{i-1} = m',\ \vec{r}_{1:i},\ S_i = m\right) \end{aligned} \tag{14.51}$$

Applying Eqs. (14.49a) and (14.49b) to the last equality, we have

$$\begin{aligned} p\,(d_i = 1,\ \vec{r}\,) &= \sum_{(m',\,m)\in\Omega_i(1)} p\left(S_{i-1} = m',\ \vec{r}_{1:i-1},\ S_i = m,\ r_i\right) \cdot p\left(\vec{r}_{i+1:N}\middle|S_i = m\right) \\ &= \sum_{(m',\,m)\in\Omega_i(1)} p\left(S_{i-1} = m',\ \vec{r}_{1:i-1}\right) \cdot p\left(S_i = m,\ r_i\middle|S_{i-1} = m'\right) \\ &\quad \cdot p\left(\vec{r}_{i+1:N}\middle|S_i = m\right) \end{aligned} \tag{14.52}$$

Applying the notations used by Bahl et al.,[13] we define

$$\alpha_{i-1}(m') \triangleq p\left(S_{i-1} = m',\ \vec{r}_{1:i-1}\right) \tag{14.53a}$$

$$\beta_i(m) \triangleq p\left(\vec{r}_{i+1:N}\middle|S_i = m\right) \tag{14.53b}$$

$$\gamma_i(m',\ m) \triangleq p\left(S_i = m,\ r_i\middle|S_{i-1} = m'\right) \tag{14.53c}$$

Given the notations in Eq. (14.53), we can use Eqs. (14.50) to (14.52) to write the LLR of each information bit d_i as

$$\Lambda(d_i) = \log \frac{\sum_{(m',\,m)\in\Omega_i(1)} \alpha_{i-1}(m')\gamma_i(m',\ m)\beta_i(m)}{\sum_{(m',\,m)\in\Omega_i(0)} \alpha_{i-1}(m')\gamma_i(m',\ m)\beta_i(m)} \tag{14.54}$$

This provides the soft decoding information for the ith information bit d_i. The MAP decoding can generate a hard decision simply by taking the sign of the LLR

$$u = \text{sign}\left[\Lambda(d_i)\right]$$

To implement the BCJR algorithm, we apply a forward recursion to obtain $\alpha_i(m)$, that is,

$$\begin{aligned}
\alpha_i(m) &\triangleq p\left(S_i = m, \vec{r}_{1:i}\right) \\
&= \sum_{m'} p\left(S_i = m, S_{i-1} = m', \vec{r}_{1:i-1}, r_i\right) \\
&= \sum_{m'} p\left(S_i = m, r_i \,|S_{i-1} = m', \vec{r}_{1:i-1}\right) \cdot p\left(S_{i-1} = m', \vec{r}_{1:i-1}\right) \\
&= \sum_{m'} \gamma_i(m', m) \cdot \alpha_{i-1}(m') \qquad (14.55)
\end{aligned}$$

The last equality comes from Eq. (14.49b). The initial state of the encoder should be $S_0 = 0$. In other words,

$$\alpha_0(m) = P[S_0 = m] = \delta[m] = \begin{cases} 1 & m = 0 \\ 0 & m \neq 0 \end{cases}$$

from which the forward recursion can proceed. The backward recursion is for computing $\beta_{i-1}(m')$ from $\beta_i(m)$:

$$\begin{aligned}
\beta_{i-1}(m') &= p\left(\vec{r}_{i:N} \,|S_{i-1} = m'\right) \\
&= \sum_{m} p\left(S_i = m, r_i, \vec{r}_{i+1:N} \,|S_{i-1} = m'\right) \\
&= \sum_{m} p\left(\vec{r}_{i+1:N} \,|S_{i-1} = m', S_i = m, r_i, \right) \cdot p\left(S_i = m, r_i | S_{i-1} = m'\right) \\
&= \sum_{m} p\left(\vec{r}_{i+1:N} \,|S_i = m, \right) \cdot \gamma_i(m', m) \\
&= \sum_{m} \beta_i(m) \cdot \gamma_i(m', m) \qquad (14.56)
\end{aligned}$$

For an encoder with a known terminal state of $S_N = 0$, we can start the backward recursion from

$$\beta_N(m) = \delta[m]$$

from which the backward recursion can be initialized.

Notice that both the forward and backward recursions depends on the function $\gamma_i(m', m)$. In fact, $\gamma_i(m', m)$ is already in a simple matrix form. The entry $\gamma_i(m', m)$ can be simplified and

derived from the basic modulation and channel information:

$$\begin{aligned}\gamma_i(m', m) &\triangleq p\left(S_i = m, r_i \,|S_{i-1} = m'\right) \\ &= p\left(r_i \,|S_{i-1} = m', S_i = m\right) \cdot P[S_i = m \,|S_{i-1} = m'\,] \\ &= p\left(r_i \,|c_i[m', m]\right) \cdot P[d_i = u\,] \end{aligned} \tag{14.57}$$

where $c_i[m', m]$ is the codeword from the encoder output corresponding to the state transition from m' to m, where as $d_i = u$ is the corresponding input bit. To determine $\gamma_i(m', m)$ for $d_i = u$ according to Eq. (14.57), $P[r_i \,|c_i[m', m]]$ is determined by the mapping from encoder output $c_i[m', m]$ to the modulated symbol b_i and the the channel noise distribution w_i.

In the special case of the convolutional code in Fig. 14.5, for every data symbol d_i, the convolutional encoder generates two coded bits $\{v_{i,1}, v_{i,2}\}$. The mapping from the coded bits $\{v_{i,1}, v_{i,2}\}$ to modulated symbol(s) b_i depends on the modulations. In BPSK, then each coded bit is mapped to ± 1 and b_i has two entries

$$b_i = \begin{bmatrix} 2v_{i,1} - 1 \\ 2v_{i,2} - 1 \end{bmatrix}$$

If QPSK modulation is applied, then we can use a Gray mapping

$$b_i = e^{j\phi_i}$$

where

$$\phi_i = \begin{cases} 0, & \{v_{i,1}, v_{i,2}\} = \{0, 0\} \\ \pi/2, & \{v_{i,1}, v_{i,2}\} = \{0, 1\} \\ \pi, & \{v_{i,1}, v_{i,2}\} = \{1, 1\} \\ -\pi/2, & \{v_{i,1}, v_{i,2}\} = \{1, 0\} \end{cases}$$

Hence, in a baseband AWGN channel, the received signal sample under QPSK is

$$\mathrm{r}_i = \sqrt{E_s} e^{j\phi_i} + \mathrm{w}_i \tag{14.58}$$

in which w_i is the complex, i.i.d. channel noise with probability density function

$$p_w(x) = \frac{1}{\pi \mathcal{N}} \exp\left(-\frac{|x|^2}{\mathcal{N}}\right)$$

As a result, in this case

$$\begin{aligned} p\left(r_i \,|c_i[m', m]\right) &= p\left(r_i \,|d_i = u\right) \\ &= p\left(r_i \,|b_i = e^{j\phi_i}\right) \\ &= p_w\left(r_i - \sqrt{E_s} e^{j\phi_i}\right) \\ &= \frac{1}{\pi \mathcal{N}} \exp\left(-\frac{|r_i - \sqrt{E_s} e^{j\phi_i}|^2}{\mathcal{N}}\right) \end{aligned} \tag{14.59}$$

The BCJR MAP algorithm can compute the LLR of each information bit according to

$$\Lambda(d_i) = \log \frac{\sum_{(m',m)\in\Omega_i(1)} \alpha_{i-1}(m')p\left(r_i \,|c_i[m',m]\right) P[d_i=1]\beta_i(m)}{\sum_{(m',m)\in\Omega_i(0)} \alpha_{i-1}(m')p\left(r_i \,|c_i[m',m]\right) P[d_i=0]\beta_i(m)}$$

$$= \underbrace{\log \frac{P[d_i=1]}{P[d_i=0]}}_{\Lambda^{(a)}(d_i)} + \underbrace{\log \frac{\sum_{(m',m)\in\Omega_i(1)} \alpha_{i-1}(m')c_i[m',m]\beta_i(m)}{\sum_{(m',m)\in\Omega_i(0)} \alpha_{i-1}(m')c_i[m',m]\beta_i(m)}}_{\Lambda^{(\ell)}(d_i)} \tag{14.60}$$

Equation (14.60) shows that the LLR of a given information symbol d_i consists of two parts:

- The a priori information $\Lambda_i^{(a)}$ from the prior probability of the data symbol d_i, which may be provided a priori or externally by another decoder.
- The local information $\Lambda_i^{(\ell)}$ that is specified by the received signals and the code trellis (or state transition) constraints.

With this decomposition view of the LLR, we are now ready to explain the concept of turbo codes, or more appropriately, turbo decoding.

Turbo Codes

The concept of turbo codes was first proposed by Berrou, Glavieux, and Thitimajshima[12] in 1993 at the annual IEEE International Conference on Communications. The authors' claim of near-Shannon-limit error correcting performance was initially met with great skepticism. This reaction was natural because the proposed turbo code exhibited BER performance within 1 dB of the Shannon limit that had been considered to be extremely challenging, if not impossible to achieve under reasonable computational complexity. Moreover, the construction of the so-called turbo codes does not take a particularly structured form. It took nearly two years for the coding community to become convinced of the extraordinary BER performance of turbo codes and to begin to understand their principles. Today, turbo codes have permeated many aspects of digital communications, often taking specially evolved forms. In this part of the section, we provide a brief introduction to the basic principles of turbo codes.

A block diagram of the first turbo encoder is shown in Fig. 14.20a. This turbo consists of two recursive systematic convolutional RSC codes. Representing a unit delay as D, the 1×2 generator matrix of the rate 1/2 RSC code is of the form

$$G(D) = \left[1 \quad \frac{g_2(D)}{g_1(D)}\right]$$

In particular, the example turbo code of Berrou et al.[12] was specified by

$$G(D) = \left[1 \quad \frac{1+D^2+D^3+D^4}{1+D+D^4}\right]$$

The simple implementation of the encoder is shown in Fig. 14.20b.

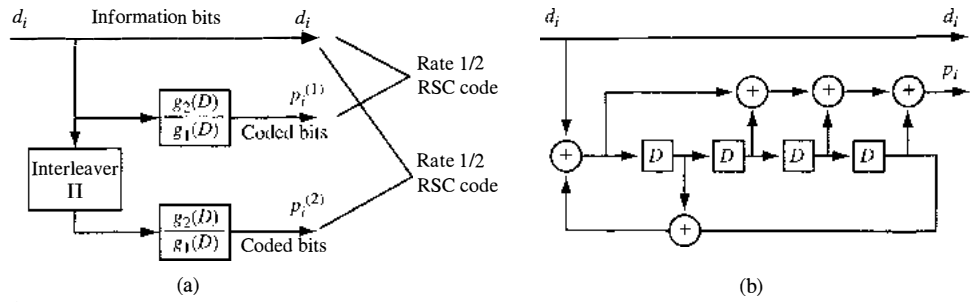


Figure 14.20 Parallel concatenated turbo code: (a) rate 1/3 turbo encoder; (b) implementation of recursive systematic convolutional (RSC) encoder $g_1(D)=1+D+D^4$, $g_2(D)=1+D^2+D^3+D^4$.

In this example, a frame of information bits d_i is sent through two RSC encoders. Both convolutional codes have rate 1/2 and are systematic. Thus, the first RSC encoder generates a frame of coded bits $p_i^{(1)}$ of length equal to the information frame. Before entering the second RSC encoder, the information bits are interleaved by a random block interleaver Π. As a result, even with the same encoder structure as the first encoder, the second encoder will generate a different coded bit frame $p_i^{(2)}$. The overall turbo code consists of the information bits and the two coded (parity) bit streams. The code rate is 1/3, as the turbo code has two coded frames for the same information frame. Then $\{d_i, p_i^{(1)}, p_i^{(2)}\}$ are modulated and transmitted over communication channels. Additional interleavers and RSC encoders can be added to obtain codes that have lower rates and are more powerful.

To construct turbo codes that have higher rates, the two convolutional encoder outputs $p_i^{(1)}$ and $p_i^{(2)}$ can be selectively but systematically discarded (e.g., by keeping only half the bits in $p_i^{(1)}$ and $p_i^{(2)}$). This process, commonly referred to as puncturing, creates two RSC codes that are more efficient, each of rate 2/3. The total turbo code rate is therefore 1/2, since for every information bit, there are two coded bits (one information bit and one parity bit).

Thus, the essence of turbo code is simply a combination of two component RSC codes. Although each component code has very few states and can be routinely decoded via decoding algorithms such as VA, SOVA, and BJCR, the random interleaver makes the overall code much more challenging to decode exactly because it consists too many states to be decoded by means of traditional MAP or VA decoders. Since each component code can be decoded by using simple decoders, the true merit of turbo codes in fact lies in iterative decoding, the concept of allowing the two component decoders to exchange information iteratively.

Iterative Decoding for Turbo Codes

It is important to note that naive iteration between two (hard) decoders cannot guarantee convergence to the result of the highly complex but exact turbo decoder. Turbo decoding is made possible and powerful by utilizing the previously discussed BCJR decoding algorithm (or its variations). Each component code can be decoded by using a BCJR soft decoding algorithm. BCJR soft decoding makes it possible for iterative turbo decoding to exchange soft information between the two soft decoders.

The idea of iterative decoding can be simply described as follows. Given the channel output, both decoders can generate the soft information $\Lambda(d_i)$ according to Eq. (14.60):

$$\Lambda_1(d_i) = \Lambda_1^{(a)}(d_i) + \Lambda_1^{(\ell)}(d_i) \tag{14.61a}$$

$$\Lambda_2(d_i) = \Lambda_2^{(a)}(d_i) + \Lambda_2^{(\ell)}(d_i) \tag{14.61b}$$

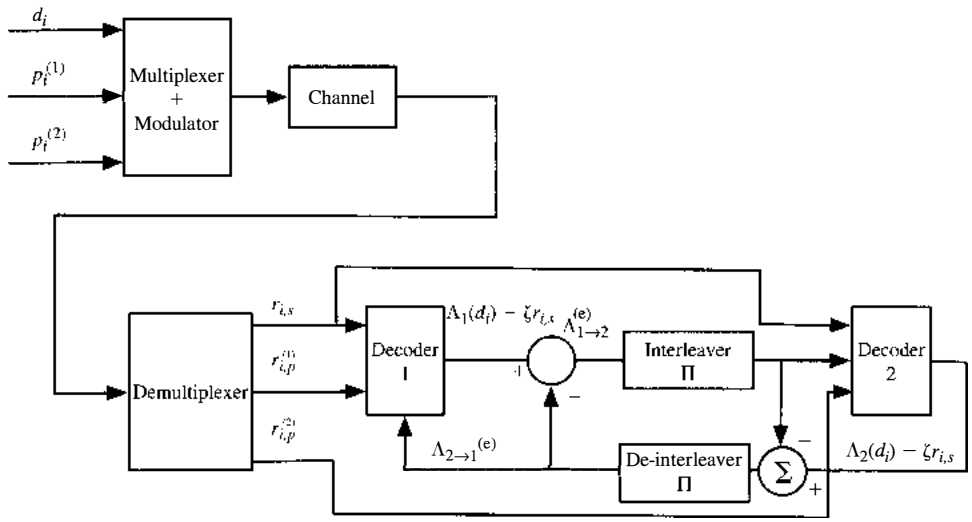


Figure 14.21 Exchange of extrinsic information between two component BCJR decoders for iterative turbo decoding.

Note that $\Lambda_1^{(a)}(d_i)$ and $\Lambda_2^{(a)}(d_i)$ are the a priori information on the information bit d_i at decoder 1 and decoder 2, respectively. Without any prior knowledge, the decoders should just treat them as 0 because $d_i = \pm 1$ are equally likely.

Iterative decoding must allow the two low complexity decoders to exchange information. To accomplish this, decoder 1 can apply BCJR algorithm to find the LLR information about d_k. It can then pass this learned information to decoder 2 as the a priori LLR. Note that this learned information must be unavailable to decoder 2 from its own decoder and other input signals. To provide innovative information, decoder 1 should remove any redundant information to generate its **extrinsic** information $\Lambda_{1\to2}^{(e)}(d_i)$ to pass to decoder 2. Similarly, decoder 2 will find out its extrinsic information $\Lambda_{2\to1}^{(e)}(d_i)$ (previously unavailable to decoder 1) and pass it back to decoder 1 as a priori information for decoder 1 to **refresh/update** its LLR on d_k. This closed-loop iteration will repeat multiple iterations until satisfactory convergence. The conceptual block diagram of this iterative turbo decoder appears in Fig. 14.21.

We now use the example given by Bahl et al.[13] to explain how to update the extrinsic information for exchange between two soft decoders. Figure 14.21 illustrates the basic signal flow of the iterative turbo decoder. There are two interconnected BCJR MAP decoders. Let us now focus on one decoder (decoder 1) and its BCJR implementation. For the first systematic RSC code, the output code bits corresponding to the information bit d_i are

$$c_i[m', m] = (d_i,\ p_i^{(1)})$$

To determine $p(r_i|c_i[m', m])$, it is necessary to specify the modulation format and the channel model.

We consider the special and simple case of BPSK modulation under channel noise that is additive, white, and Gaussian. In this case, there are two received signal samples as a result of the coded bits $c_i[m', m] = (d_i,\ p_i^{(1)})$. More specifically, from encoder 1, the channel output consists of two signal sequences

$$\mathrm{r}_{i,s} = \sqrt{E_b}(2d_i - 1) + \mathrm{w}_i \tag{14.62a}$$

$$\mathrm{r}_{i,p}^{(1)} = \sqrt{E_b}(2p_i^{(1)} - 1) + \mathrm{w}_{i,1} \tag{14.62b}$$

whereas from encoder 2, the channel outputs are

$$\mathrm{r}_{i,s} = \sqrt{E_b}(2d_i - 1) + \mathrm{w}_i \tag{14.63a}$$

$$\mathrm{r}_{i,p}^{(2)} = \sqrt{E_b}(2p_i^{(2)} - 1) + \mathrm{w}_{i,2} \tag{14.63b}$$

Note that the Gaussian noises w_i, $\mathrm{w}_{i,1}$, and $\mathrm{w}_{i,2}$ are all independent with identical Gaussian distribution of zero mean and variance $\mathcal{N}/2$. The first BCJR decoder is given signals $r_{i,s}$ and $r_{i,p}^{(1)}$ to decode, whereas the second BCJR decoder is given signals $r_{i,s}$ and $r_{i,p}^{(2)}$ to decode.

Let's first denote $p_i[m', m]$ as the ith parity bit at a decoder corresponding to message bit d_i. It naturally corresponds to the transition from state m' to state m. For each decoder, the received channel output signals $\boldsymbol{r}_i = [r_{i,s}, r_{i,p}]$ specifies $\gamma_i(m', m)$ via

$$\begin{aligned}
\gamma_i(m', m) &= p(\mathbf{r}_i | c_i[m', m]) P(d_i) \\
&= p\left(r_{i,s}, r_{i,p} | d_i, p_i[m', m]\right) P(d_i) \\
&= \frac{1}{\pi\mathcal{N}} \exp\left[-\frac{|r_{i,s} - \sqrt{E_b}(2d_i - 1)|^2 + |r_{i,p} - \sqrt{E_b}(2p_i[m', m] - 1)|^2}{\mathcal{N}}\right] P(d_i) \\
&= \frac{1}{\pi\mathcal{N}} \exp\left[-\frac{r_{i,s}^2 + r_{i,p}^2 + 2E_b|^2}{\mathcal{N}}\right] \exp\left\{\frac{2\sqrt{E_b}}{\mathcal{N}}\left[r_{i,s}(2d_i - 1) + r_{i,p}(2p_i[m', m] - 1)\right]\right\} \\
&\quad \times P(d_i)
\end{aligned} \tag{14.64}$$

Notice that the first term in Eq. (14.64) is independent of the codeword or the transition from m' to m. Thus, the LLR at this decoder becomes

$$\Lambda(d_i) = \log \frac{\sum_{(m', m) \in \Omega_i(1)} \alpha_{i-1}(m') p\left(\boldsymbol{r}_i | c_i[m', m]\right) P[d_i = 1] \beta_i(m)}{\sum_{(m', m) \in \Omega_i(0)} \alpha_{i-1}(m') p\left(\boldsymbol{r}_i | c_i[m', m]\right) P[d_i = 0] \beta_i(m)} \tag{14.65}$$

$$= \log \frac{P[d_i = 1]}{P[d_i = 0]} + \log \frac{\sum_{(m', m) \in \Omega_i(1)} \alpha_{i-1}(m') \exp\left\{\frac{2\sqrt{E_b}}{\mathcal{N}}\left[r_{i,s} + 2r_{i,p} p_i[m', m]\right]\right\} \beta_i(m)}{\sum_{(m', m) \in \Omega_i(0)} \alpha_{i-1}(m') \exp\left\{\frac{2\sqrt{E_b}}{\mathcal{N}}\left[-r_{i,s} + 2r_{i,p} p_i[m', m]\right]\right\} \beta_i(m)}$$

By defining the gain parameter $\zeta = 4\sqrt{E_b}/\mathcal{N}$, we can simplify the LLR into

$$\Lambda(d_i) = \underbrace{\log \frac{P[d_i = 1]}{P[d_i = 0]}}_{\Lambda^{(a)}} + \underbrace{\zeta \cdot r_{i,s}}_{\Lambda^{(c)}} + \underbrace{\log \frac{\sum_{(m', m) \in \Omega_i(1)} \alpha_{i-1}(m') \exp\left(\zeta \cdot r_{i,p} p_i[m', m]\right) \beta_i(m)}{\sum_{(m', m) \in \Omega_i(0)} \alpha_{i-1}(m') \exp\left(\zeta \cdot r_{i,p} p_i[m', m]\right) \beta_i(m)}}_{\Lambda^{(e)}} \tag{14.66}$$

In other words, for every information bit d_i, the LLR of both decoders can be decomposed into three parts as in

$$\Lambda_j(d_i) = \Lambda_j^{(a)}(d_i) + \Lambda_j^{(c)}(d_i) + \Lambda_j^{(e)}(d_i) \qquad j = 1, 2$$

where $\Lambda_j^{(a)}(d_i)$ is the prior information provided by the other decoder, $\Lambda_j^{(c)}(d_i)$ is the channel output information shared by both decoders, and $\Lambda_j^{(e)}(d_i)$ is the **extrinsic information** uniquely obtained by the jth decoder that is used by the other decoder as prior information. This means that at any given iteration, decoder 1 needs to compute

$$\Lambda_1(d_i) = \Lambda_{2\to 1}^{(e)}(d_i) + \zeta \cdot r_{i,s} + \Lambda_{1\to 2}^{(e)}(d_i) \tag{14.67a}$$

in which $\Lambda_{2\to 1}^{(e)}(d_i)$ is the extrinsic information passed from decoder 2, whereas $\Lambda_{1\to 2}^{(e)}(d_i)$ is the new extrinsic information to be sent to decoder 2 to refresh or update its LLR via

$$\Lambda_2(d_i) = \Lambda_{1\to 2}^{(e)}(d_i) + \zeta \cdot r_{i,s} + \Lambda_{2\to 1}^{(e)}(d_i) \tag{14.67b}$$

At both decoders, the updating of the extrinsic information requires the updating of $\alpha_i(m)$ and $\beta_i(m)$ before the computation of extrinsic information

$$\Lambda^{(e)} = \log \frac{\sum_{(m',m)\in\Omega_i(1)} \alpha_{i-1}(m') \exp\left(\zeta \cdot r_{i,p}\, p_i[m', m]\right) \beta_i(m)}{\sum_{(m',m)\in\Omega_i(0)} \alpha_{i-1}(m') \exp\left(\zeta \cdot r_{i,p}\, p_i[m', m]\right) \beta_i(m)} \tag{14.68}$$

To refresh $\alpha_i(m)$ and $\beta_i(m)$ based on the extrinsic information $\Lambda^{(e)}$, we need to recompute at each decoder

$$\gamma_i(m', m) = p(\boldsymbol{r}_i | d_i) P(d_i) \tag{14.69}$$

$$\sim \left\{(1 - d_i) + d_i \exp[\Lambda^{(e)}]\right\} \exp(0.5\zeta \cdot r_{i,s}) \exp\left(\zeta \cdot r_{i,p}\, p_i[m', m]\right) \tag{14.70}$$

Once decoder 1 has finished its BCJR decoding, it can provide its soft output as the prior information about d_i to decoder 2. When decoder 2 finishes its BCJR decoding, utilizing the prior information from decoder 1, it should provide its new soft information about d_i back to decoder 1. To ascertain that decoder 2 does not feed back the "stale" information that originally came from decoder 1, we must subtract the stale information before feedback, thereby providing only the extrinsic information $\Lambda_{1\to 2}^{(e)}(d_i)$ back to decoder 1 as "priors" for decoder 1 in the next iteration. Similarly, in the next iteration, decoder 1 will update its soft output and subtract the stale information that originally came from decoder 2 to provide refreshed extrinsic information $\Lambda_{2\to 1}^{(e)}(d_i)$ as priors for decoder 2. This exchange of extrinsic information is illustrated in Fig. 14.21.

As an illustrative example, the original decoding performance of the turbo code proposed by Berrou et al.[12] is reproduced in Fig. 14.22. The results demonstrate the progressive performance improvement of successive iterations during iterative soft decoding. After 18 iterations, the bit error rate performance is only 0.7 dB away from the theoretical limit.

14.12 LOW-DENSITY PARITY CHECK (LDPC) CODES

Following the discovery of turbo codes, researchers carried out a flurry of activitity aimed at finding equally powerful, if not more powerful, error correcting codes that are suitable for soft iterative decoding. Shortly thereafter, another class of near-capacity codes known as low-density parity check (LDPC) codes, originally introduced by Gallager[15] in 1963, was rediscovered by MacKay and Neal[16, 17] in 1995. Since then, LDPC code designs and efficient

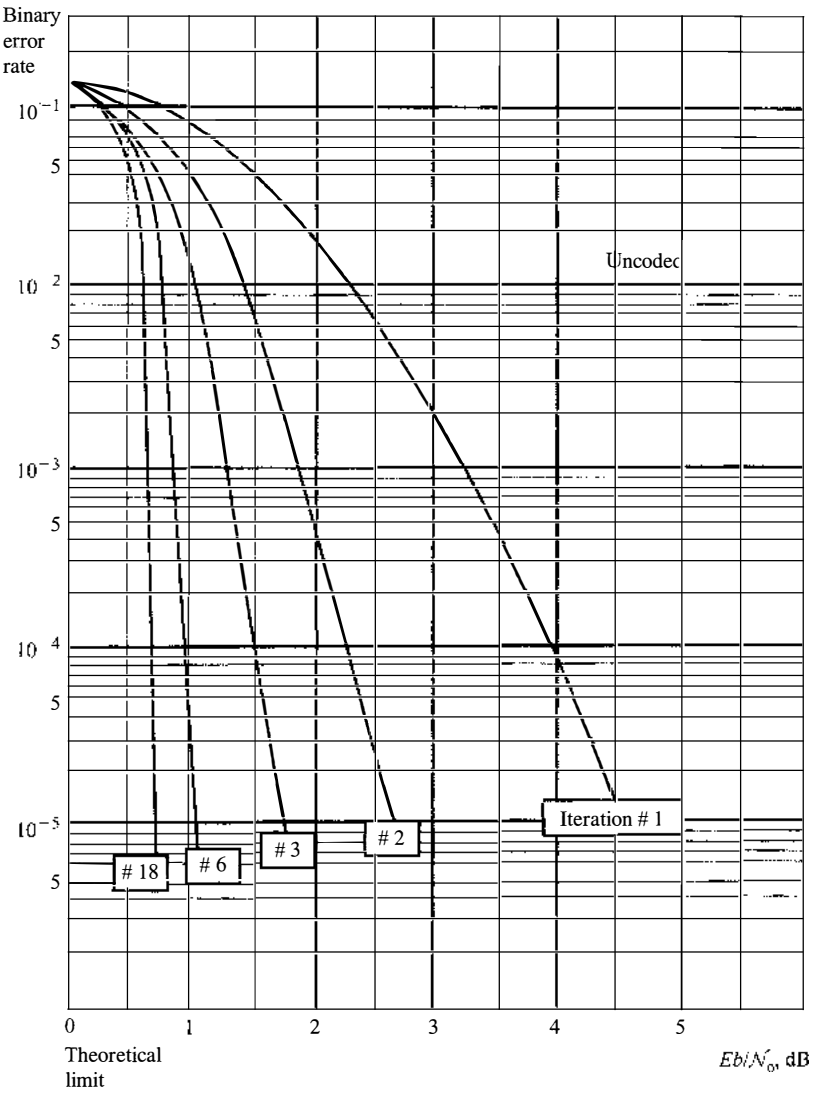


Figure 14.22 The decoding performance of a rate 1/2 Turbo code is shown to be very close to the theoretical limit. (Reproduced with copyright permission from IEEE from Ref. 14.)

means of LDPC decoding have been topics of intensive research in the coding community. A large number of LDPC codes have been proposed as strong competitors to turbo codes, often achieving better performance with comparable code lengths, code rates, and decoding complexity.

LDPC codes are linear block codes with sparse parity check matrices. In essence, the parity check matrix $\boldsymbol{H}$ consists of mostly **0**s and very few **1**s, forming a **low-density** parity check matrix. LDPC codes are typically quite long (normally longer than 1000 bits) and noncyclic. Thus, an exact implementation of the BCJR MAP decoding algorithm is quite complex and mostly impractical. Fortunately, there are several well-established methods for decoding LDPC codes that can achieve near-optimum performance.

The design of LDPC code is equivalent to the design of a sparse parity matrix $\boldsymbol{H}$. Once $\boldsymbol{H}$ has been defined, the LDPC code is the null-space of the parity matrix $\boldsymbol{H}$. The number of **1**s in the ith row of $\boldsymbol{H}$ is known as the row weight ρ_i, whereas the number of **1**s in the jth column

is known as the column weight γ_j. In LDPC codes, both row and column weights are much smaller than the code length n, that is,

$$\rho_i \ll n \qquad \gamma_j \ll n$$

For **regular** LDPC codes, all rows have equal weight $\rho_i = \rho$ and all columns have equal weight $\gamma_i = \gamma$. For **irregular** LDPC codes, the row weights and column weights do vary and typically exhibit certain weight distributions. Regular codes are easier to generate, whereas irregular codes with large code length may have better performance.

Bipartite (Tanner) Graph

A Tanner graph is a graphic representation that can conveniently describe a linear block code; This bipartite graph with incidence matrix $\boldsymbol{H}$ was introduced by R. M Tanner in 1981.[18] Consider an (n, k) linear block code. There are n coded bits and $n - k$ parity bits. The Tanner graph of this linear block code has n variable nodes corresponding to the n code bits. These n variable nodes are connected to their respective parity nodes (or check nodes) according to the **1**s in the parity check matrix $\boldsymbol{H}$. A variable node (a column) and a check node (a row) are connected if the corresponding element in $\boldsymbol{H}$ is a **1**. Because $\boldsymbol{H}$ is sparse, there are only a few connections to each variable node or check node. These connections are known as edges. Each row represents the connection of a check node, and each column represents the connection of a variable node. For LDPC codes, if the ith row of $\boldsymbol{H}$ has row weight of ρ_i, then the check node has ρ_i edges. If column j has column weight of γ_i, then the variable node has γ_i edges. We use an example to illustrate the relationship between $\boldsymbol{H}$ and the Tanner graph.

Example 14.9 Consider a Hamming (7, 4, 3) code with parity check matrix

$$\boldsymbol{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{14.71}$$

Determine its Tanner graph.

This code has 7 variable nodes and 3 check nodes. Based on the entries in $\boldsymbol{H}$, each check node is connected to 4 variable nodes. The first row of $\boldsymbol{H}$ corresponding the connection to check node 1. The nonzero entries of $\boldsymbol{H}$ mark the connected variable nodes. The resulting Tanner graph is shown in Fig. 14.23.

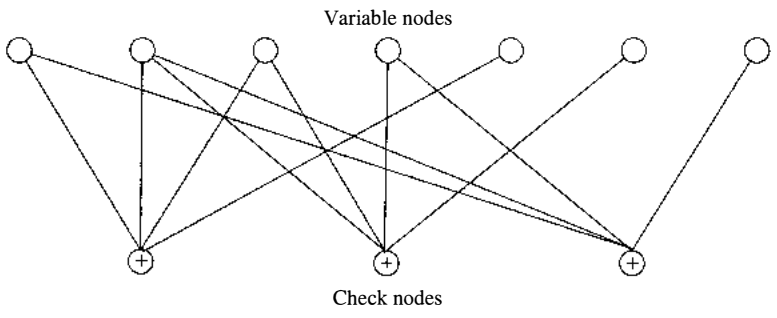


Figure 14.23 Tanner graph of the (7, 4, 3) Hamming code.

Because LDPC codes are typically of length greater than 1000, their Tanner graphs are normally too large to illustrate in practice. However, the basic Tanner graph concept is very helpful to understanding LDPC codes and its iterative decoding.

A **cycle** in the Tanner graph is marked by a closed loop of connected edges. The loop originates from and ends at the same variable (or check) node. The length of a cycle is defined by the number of its edges. In Example 14.9, there exist several cycles of length 4 and length 6. Cycles of lengths 4 and 6 are considered to be **short** cycles. Short cycles are known to be undesirable in some iterative decoding algorithms for LDPC codes. When a Tanner graph is free of short cycles, iterative decoding of LDPC codes based on the so-called sum-product algorithm can converge and generate results close to the full-scale MAP decoder that is too complex to implement in practice.

To prevent a cycle of length 4, LDPC code design usually imposes an additional constraint on the parity matrix $\boldsymbol{H}$: **No two rows or columns may have more than one component in common.** This property, known as the "row-column (RC) constraint," is sufficient and necessary to avoid cycles of length 4. The presence of cycles is often unavoidable in most LDPC code designs based on computer searches. A significant number of researchers have been studying the challenging problem of either reducing the number of, or eliminating short cycles of length 4, 6, and possibly 8. Interested readers should consult the book by Lin and Costello.[2]

We now describe two decoding methods for LDPC codes.

Bit-Flipping LDPC Decoding

The large code length of LDPC codes makes decoding a highly challenging problem. Two of the most common decoding algorithms are the hard decision bit-flipping (BF) algorithm and the soft-decision sum-product algorithm (SPA).

The bit-flipping (BF) algorithm operates on a sequence of hard-decision bits $\boldsymbol{r} = \mathbf{011010\cdots010}$. Parity checks on $\boldsymbol{r}$ generate the syndrome vector

$$\boldsymbol{s} = \boldsymbol{r}\boldsymbol{H}^T$$

Those syndrome bits of value **1** indicate parity failure. The BF algorithm tries to change a bit (by flipping) in $\boldsymbol{r}$ based on how the flip would affect the syndrome bits.

When a code bit participates in only a single failed parity check, flipping this bit at best will correct 1 failed parity check but will cause $\gamma - 1$ new parity failures. For this reason, BF only flips bits that affect a large number of failed parity checks. A simple BF algorithm consists of the following steps:[2]

Step 1: Calculate the parity checks $\boldsymbol{s} = \boldsymbol{r}\boldsymbol{H}^T$. If all syndromes are zero, stop decoding.

Step 2: Determine the number of failed parity checks for every bit:

$$f_i \qquad i = 1, 2, \ldots, n$$

Step 3: Identify the set of bits $F_{\max}$ with the largest f_i and flip the bits in $F_{\max}$ to generate a new codeword $\boldsymbol{r}'$.

Step 4. Let $\boldsymbol{r} = \boldsymbol{r}'$ and repeat steps 1 to 3 until the maximum number of iterations has been reached.

Sum-Product Algorithm for LDPC Decoding

The sum-product algorithm (SPA) is the most commonly used LDPC decoding method. It is an efficient soft-input, soft-output decoding algorithm based on iterative belief propagation. SPA can be better interpreted via the Tanner graph. SPA is similar to a see-saw game. In one step, every variable node passes information via its edges to its connected check nodes in the top-down pass-flow. In the next step, every check node passes back information to all the variable nodes it is connected to in a bottom-up pass-flow.

To understand SPA, let the parity matrix be $\boldsymbol{H}$ of size $J \times n$ where $J = n - k$ for an (n, k) LDPC block code. Let the codeword be represented by variable node bits $\{v_j, \quad j = 1, \ldots, n\}$. For the jth variable node v_j, let

$$\boldsymbol{\mu}_j = \{i : h_{ij} = 1, \ 1 \leq i \leq J\}$$

denote the set of variable nodes connected to v_j. For the ith check node z_i, let

$$\boldsymbol{\sigma}_i = \{j : h_{ij} = 1, \ 1 \leq j \leq n\}$$

denote the set of variable nodes connected to z_i.

First, define the probability of satisfying check node $z_i = 0$ when $v_j = u$ as

$$R_{i,j}(u) = P\left[z_i = 0 \middle| v_j = u\right] \qquad u = 0, 1 \tag{14.72}$$

Let us denote the vector of variable bits as $\boldsymbol{v}$. We can use the Bayes theorem on conditional probability (Sec. 8.1) to show that

$$\begin{aligned} R_{i,j}(u) &= \sum_{\boldsymbol{v}:\, v_j = u} P\left[z_i = 0 \middle| \boldsymbol{v}\right] \cdot P\left[\boldsymbol{v} \middle| v_j = u\right] \\ &= \sum_{v_\ell:\, \ell \in \boldsymbol{\sigma}_i,\, \ell \neq j} P\left[z_i = 0 \middle| v_j = u, \{v_\ell : \ell \in \boldsymbol{\sigma}_i, \ell \neq j\}\right] \cdot P\left[\{v_\ell : \ell \in \boldsymbol{\sigma}_i, \ell \neq j\} \middle| v_j = u\right] \end{aligned} \tag{14.73}$$

This is message passing in the bottom-up direction.

For the check node z_i to estimate the probability $P\left[\{v_\ell : \ell \in \boldsymbol{\sigma}_i, \ell \neq j\} \middle| v_j = u\right]$, the check node must collect information from the variable node set $\boldsymbol{\sigma}_i$. Define the probability of $v_\ell = x$ obtained from its check nodes except for the ith one as

$$Q_{i,\ell}(x) = P\left[v_\ell = x \middle| \{z_m = 0 : m \in \boldsymbol{\mu}_\ell, m \neq i\}\right] \qquad x = 0, 1 \tag{14.74}$$

Furthermore, assume that the variable node probabilities are approximately independent. We can estimate

$$P\left[\{v_\ell : \ell \in \boldsymbol{\sigma}_i, \ell \neq j\} \middle| v_j = u\right] = \prod_{\ell \in \boldsymbol{\sigma}_i, \ell \neq j} Q_{i,\ell}(v_\ell) \tag{14.75}$$

This means that the check nodes can update the message through

$$R_{i,j}(u) = \sum_{v_\ell:\, \ell \in \boldsymbol{\sigma}_i, \ell \neq j} P\left[z_i = 0 \middle| v_j = u, \{v_\ell : \ell \in \boldsymbol{\sigma}_i, \ell \neq j\}\right] \cdot \prod_{\ell \in \boldsymbol{\sigma}_i, \ell \neq j} Q_{i,\ell}(v_\ell) \tag{14.76}$$

Note that the probability of $P\left[z_i = 0 | v_j = u, \{v_\ell : \ell \in \sigma_i, \ell \neq j\}\right]$ is either 0 or 1; that is, the check node $z_i = 0$ either succeeds or fails. The relationship Eq. (14.76) allows $R_{i,j}(u)$ to be updated when the ith check node receives $Q_{i,\ell}(v_\ell)$.

Once $R_{i,j}(u)$ have been updated, they can be passed to the variable nodes in the bottom-up direction to update $Q_{i,\ell}(x)$. Again using Bayes' theorem (Sec. 8.1), we have

$$Q_{i,\ell}(x) = \frac{P[v_\ell = x] P\left[\{z_m = 0 : m \in \boldsymbol{\mu}_\ell, m \neq i\} | v_\ell = x\right]}{P[\{z_m = 0 : m \in \boldsymbol{\mu}_\ell, m \neq i\}]} \tag{14.77}$$

Once again assuming that each parity check is independent, we then write

$$P\left[\{z_m = 0 : m \in \boldsymbol{\mu}_\ell, m \neq i\} | v_\ell = x\right] = \prod_{m \in \boldsymbol{\mu}_\ell, m \neq i} R_{m,\ell}(x) \tag{14.78}$$

Now define the prior variable bit probability as $p_\ell(x) = P(v_\ell = x)$. Let $\alpha_{i,\ell}$ be the normalization factor such that $Q_{i,\ell}(1) + Q_{i,\ell}(0) = 1$. We can update $Q_{i,\ell}(x)$ at the variable nodes based on Eq. (14.76):

$$Q_{i,\ell}(x) = \alpha_{i,\ell} \cdot p_\ell(x) \prod_{m \in \boldsymbol{\mu}_\ell, m \neq i} R_{m,\ell}(x) \tag{14.79}$$

This message will then be passed back in the top-down direction to the check nodes. Figure 14.24 illustrates the basic operation of message passing in the bottom-up and the top-down directions in the sum-product algorithm. The SPA can be summarized as follows.

Initialization: Let $m = 0$ and let m_{max} be the maximum number of iterations. For every $h_{i,\ell} = 1$ in $\boldsymbol{H}$, use prior probabilities to set

$$Q_{i,\ell}^{(0)}(1) = p_\ell(1), \quad \text{and} \quad Q_{i,\ell}^{(0)}(0) = p_\ell(0)$$

Step 1: Let the check node i update its information

$$R_{i,j}^{(m)}(1) = \sum_{v_\ell : \ell \in \sigma_i, \ell \neq j} P\left[z_i = 0 | v_j = 1, \{v_\ell\}\right] \cdot \prod_{\ell \in \sigma_i, \ell \neq j} Q_{i,\ell}^{(m)}(v_\ell). \tag{14.80a}$$

$$R_{i,j}^{(m)}(0) = \sum_{v_\ell : \ell \in \sigma_i, \ell \neq j} P\left[z_i = 0 | v_j = 0, \{v_\ell\}\right] \cdot \prod_{\ell \in \sigma_i, \ell \neq j} Q_{i,\ell}^{(m)}(v_\ell) \tag{14.80b}$$

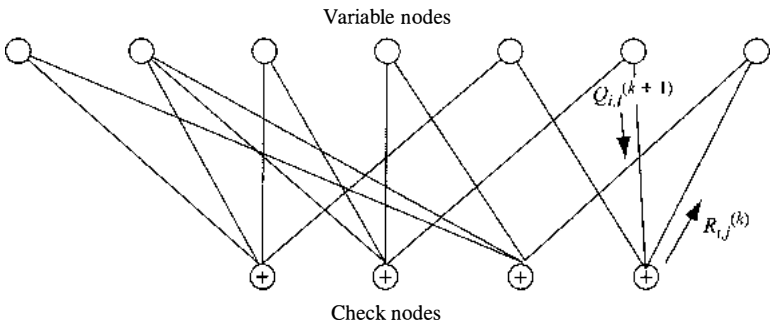


Figure 14.24 Message passing in the sum-product algorithm.

Step 2: At every variable node (indexed by ℓ), update

$$Q_{i,\ell}^{(m+1)}(0) = \alpha_{i,\ell}^{(m+1)} \cdot p_\ell(0) \prod_{m \in \boldsymbol{\mu}_\ell, m \neq i} R_{m,\ell}^{(m)}(0) \tag{14.81a}$$

$$Q_{i,\ell}^{(m+1)}(1) = \alpha_{i,\ell}^{(m+1)} \cdot p_\ell(1) \prod_{m \in \boldsymbol{\mu}_\ell, m \neq i} R_{m,\ell}^{(m)}(1) \tag{14.81b}$$

where the normalization factor $\alpha_{i,\ell}^{(m+1)}$ is selected such that

$$Q_{i,\ell}^{(m+1)}(0) + Q_{i,\ell}^{(m+1)}(1) = 1$$

Step 3: At the variable nodes, also estimate the a posteriori probabilities

$$P^{(m+1)}\left[v_\ell = 0 | \boldsymbol{r}\right] = \alpha_\ell^{(m+1)} \cdot p_\ell(0) \prod_{m \in \boldsymbol{\mu}_\ell} R_{m,\ell}^{(m)}(0) \tag{14.82a}$$

$$P^{(m+1)}\left[v_\ell = 1 | \boldsymbol{r}\right] = \alpha_\ell^{(m+1)} \cdot p_\ell(1) \prod_{m \in \boldsymbol{\mu}_\ell} R_{m,\ell}^{(m)}(1) \tag{14.82b}$$

where the normalization factor $\alpha_\ell^{(m+1)}$ is selected such that

$$P^{(m+1)}\left[v_\ell = 0 | \boldsymbol{r}\right] + P^{(m+1)}\left[v_\ell = 1 | \boldsymbol{r}\right] = 1$$

Step 4: Make hard decisions of each code bit

$$\hat{v}_\ell = \text{sign}\left\{\log \frac{P^{(m+1)}\left[v_\ell = 1 | \boldsymbol{r}\right]}{P^{(m+1)}\left[v_\ell = 0 | \boldsymbol{r}\right]}\right\}$$

If the decode codeword satisfies all parity checks, stop decoding. Otherwise, go back to step 1 for another iteration.

Notice that external input signals $\{r_i\}$ are involved only during the estimation of a priori probabilities $p_\ell(1)$ and $p_\ell(0)$. SPA uses the a priori probabilities as follows:

$$p_\ell(1) = \frac{p(\boldsymbol{r}|v_\ell = 1)}{p(\boldsymbol{r}|v_\ell = 1) + p(\boldsymbol{r}|v_\ell = 0)} \quad \text{and} \quad p_\ell(0) = \frac{p(\boldsymbol{r}|v_\ell = 0)}{p(\boldsymbol{r}|v_\ell = 1) + p(\boldsymbol{r}|v_\ell = 0)}$$

For a more concrete example, consider the example of an AWGN channel with BPSK modulation. For v_ℓ, the received signal sample is

$$\mathrm{r}_\ell = \sqrt{E_b}(2v_\ell - 1) + \mathrm{w}_\ell$$

where w_ℓ is Gaussian with zero mean and variance $\mathcal{N}/2$. Because $\{\mathrm{r}_\ell\}$ are independent, when we receive $\mathrm{r}_\ell = r_\ell$, we can simply use

$$p_\ell(1) = \frac{1}{1 + \exp\left(-4\frac{\sqrt{E_b}}{\mathcal{N}} r_\ell\right)} \quad \text{and} \quad p_\ell(0) = \frac{1}{1 + \exp\left(4\frac{\sqrt{E_b}}{\mathcal{N}} r_\ell\right)}$$

This completes the introduction of sum-product algorithm for the decoding of LDPC codes.

14.13 MATLAB EXERCISES

In this section, we provide MATLAB programs to illustrate simple examples of block encoders and decoders. We focus on the simpler case of hard-decision decoding based on syndromes.

COMPUTER EXERCISE 14.1

In the first experiment, we provide a program to decode the (6, 3) linear block code of Example 14.1.

```
% Matlab Program <Ex14_1.m>
% to illustrate encoding and decoding of (6,3) block code
% in Example 14.1
%
G=[1 0 0 1 0 1                          %Code Generator
   0 1 0 0 1 1
   0 0 1 1 1 0];
H=[1 0 1                                %Parity Check Matrix
   0 1 1
   1 1 0
   1 0 0
   0 1 0
   0 0 1]';
E=[0 0 0 0 0 0                          %List of correctable errors
   1 0 0 0 0 0
   0 1 0 0 0 0
   0 0 1 0 0 0
   0 0 0 1 0 0
   0 0 0 0 1 0
   0 0 0 0 0 1
   1 0 0 0 1 0];
K=size(E,1);
Syndrome=mod(mtimes(E,H'),2);           %Find Syndrome List
r=[1     1     1     0     1     1]     %Received codeword
display(['Syndrome ','Error Pattern'])
display(num2str([Syndrome  E]))
x=mod(r*H',2);                          %Compute syndrome
for kk=1:K,
    if Syndrome(kk,:)==x,
        idxe=kk;                        %Find the syndrome index
    end
end
syndrome=Syndrome(idxe,:)               %display the syndrome
error=E(idxe,:)
cword=xor(r,error)                  %Error correction
```

The execution of this MATLAB program will generate the following results, which include the erroneous codeword, the syndrome, the error pattern, and the corrected codeword.

```
Ex14_2
Syndrome Error Pattern
0  0  0  0  0  0  0  0  0
1  0  1  1  0  0  0  0  0
0  1  1  0  1  0  0  0  0
1  1  0  0  0  1  0  0  0
```

```
1  0  0  0  0  0  1  0  0
0  1  0  0  0  0  0  1  0
0  0  1  0  0  0  0  0  1
1  1  1  1  0  0  0  1  0

syndrome =

     0     1     1


error =

     0     1     0     0     0     0


cword =

     1     0     1     0     1     1
```

In our next exercise, we provide a program to decode the (7, 4) Hamming code of Example 14.3.

```
% Matlab Program <Ex14_3.m>
% to illustrate encoding and decoding of Hamming (7,4) code
%
G=[1 0 0 0 1 0 1                          % Code Generating Matrix
   0 1 0 0 1 1 1
   0 0 1 0 1 1 0
   0 0 0 1 0 1 1];
H=[G(:,5:7)', eye(3,3)];                  %Parity Check Matrix
E=[1 0 0 0 0 0 0                          %List of correctable errors
   0 1 0 0 0 0 0
   0 0 1 0 0 0 0
   0 0 0 1 0 0 0
   0 0 0 0 1 0 0
   0 0 0 0 0 1 0
   0 0 0 0 0 0 1];
K=size(E,1);
Syndrome=mod(mtimes(E,H'),2);             %Find Syndrome List
r=[1  0  1  0  1  1  1]                   %Received codeword
display(['Syndrome ','Error Pattern'])
display(num2str([Syndrome  E]))
x=mod(r*H',2);                            %Compute syndrome
for kk=1:K,
    if Syndrome(kk,:)==x,
        idxe=kk;                          %Find the syndrome index
    end
end
syndrome=Syndrome(idxe,:)                 %display the syndrome
error=E(idxe,:)
cword=xor(r,error)                    %Error correction
```

Executing MATLAB program `Ex14_3.m` will generate for a erroneous codeword `r` its syndrome, the error pattern, and the corrected codeword:

```
Ex14_3

r =

     1     0     1     0     1     1     1

Syndrome Error Pattern
1  0  1  1  0  0  0  0  0  0
1  1  1  0  1  0  0  0  0  0
1  1  0  0  0  1  0  0  0  0
0  1  1  0  0  0  1  0  0  0
1  0  0  0  0  0  0  1  0  0
0  1  0  0  0  0  0  0  1  0
0  0  1  0  0  0  0  0  0  1

syndrome =

     1     0     0


error =

     0     0     0     0     1     0     0


cword =

     1     0     1     0     0     1     1
```

COMPUTER EXERCISE 14.2

In a more realistic example, we will use the Hamming (7,4) code to encode a long binary message bit sequence. The coded bits will be transmitted in polar signaling over an additive white Gaussian noise (AWGN) channel. The channel outputs will be detected using a hard-decision function `sgn`. The channel noise will lead to hard-decision errors. The detector outputs will be decoded using the Hamming (7,4) decoder that is capable of correcting 1 bit error in each codeword of length 7.

This result is compared against the uncoded polar transmission. To be fair, the average $E_b/\mathcal{N}$ ratio for every information bit is made equal for both cases. MATLAB program `Sim74Hamming.m` is given; the resulting BER comparison is shown in Fig. 14.25.

```
% Matlab Program <Sim74Hamming.m>
% Simulation of the Hamming (7,4) code performance
% under polar signaling in AWGN channel and performance
% comparison  with uncoded polar signaling
clf;clear sigcw BER_uncode BER_coded
G=[1 0 0 0 1 0 1                    % Code Generator
   0 1 0 0 1 1 1
   0 0 1 0 1 1 0
   0 0 0 1 0 1 1];
H=[1 1 1 0 1 0 0                    % Parity Check Matrix
   0 1 1 1 0 1 0
```

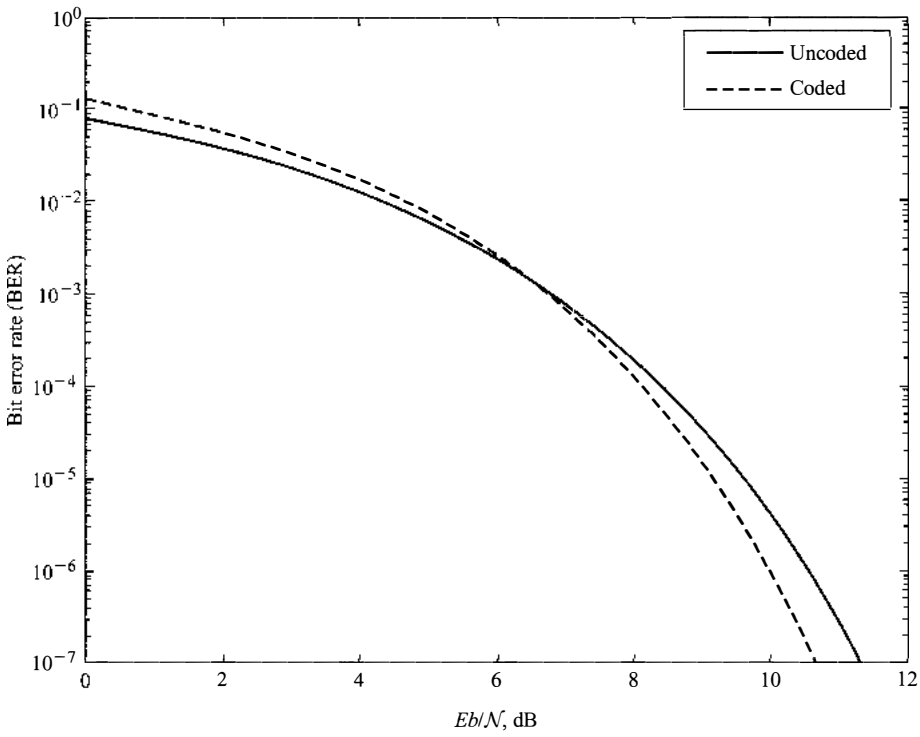


Figure 14.25 Comparison of bit error rates of uncoded polar signaling transmission and polar signaling transmission of Hamming (7, 4) encoded [(dashed) and uncoded (solid) message bits.

```
   1 1 0 1 0 0 1];
E=[1 0 0 0 0 0 0                    % Error patterns
   0 1 0 0 0 0 0
   0 0 1 0 0 0 0
   0 0 0 1 0 0 0
   0 0 0 0 1 0 0
   0 0 0 0 0 1 0
   0 0 0 0 0 0 1
   0 0 0 0 0 0 0];
K2=size(E,1);
Syndrome=mod(mtimes(E,H'),2);       % Syndrome list

L1=25000;K=4*L1;                        %Decide how many codewords

sig_b=round(rand(1,K));                 %Generate message bits
sig_2=reshape(sig_b,4,L1);              %4 per column for FEC
xig_1=mod(G'*sig_2,2);                  %Encode column by column
xig_2=2*reshape(xig_1,1,7*L1)-1;        %P/S conversion
AWnoise1=randn(1,7*L1);                 %Generate AWGN for coded Tx
AWnoise2=randn(1,4*L1);                 %Generate AWGN for uncoded Tx
% Change SNR and compute BER's
for ii=1:14
    SNRdb=ii;
    SNR=10^(SNRdb*0.1);
    xig_n=sqrt(SNR*4/7)*xig_2+AWnoise1;    %Add AWGN and adjust SNR
```

```
    rig_1=(1+sign(xig_n))/2;                %Hard decisions
    r=reshape(rig_1,7,L1)';                 %S/P to form 7 bit codewords
    x=mod(r*H',2);                          % generate error syndromes
    for k1=1:L1,
        for k2=1:K2,
            if Syndrome(k2,:)==x(k1,:),
            idxe=k2;                         %find the Syndrome index
            end
        end
        error=E(idxe,:);                     %look up the error pattern
        cword=xor(r(k1,:),error);           %error correction
        sigcw(:,k1)=cword(1:4);              %keep the message bits
    end
    cw=reshape(sigcw,1,K);
    BER_coded(ii)=sum(abs(cw-sig_b))/K;     % Coded BER on info bits

    % Uncoded Simulation Without Hamming code
    xig_3=2*sig_b-1;                         % Polar signaling
    xig_m=sqrt(SNR)*xig_3+AWnoise2;          % Add AWGN and adjust SNR
    rig_1=(1+sign(xig_m))/2;                 % Hard decision
    BER_uncode(ii)=sum(abs(rig_1-sig_b))/K; % Compute BER
end
EboverN=[1:14]-3;     % Need to note that SNR = 2 Eb/N
```

Naturally, when the $E_b/\mathcal{N}$ is low, there tends to be more than 1 error bit per codeword. Thus, when there is more than 1 bit error, the decoder will still consider the codeword to be corrupted by only 1 bit error. Its attempt to correct 1-bit error may in fact add an error bit. When the $E_b/\mathcal{N}$ is high, it is more likely that a codeword has at most 1 bit error. This explains why the coded BER is worse at lower $E_b/\mathcal{N}$ and better at higher $E_b/\mathcal{N}$. On other other hand, Fig. 14.3 gives an optimistic approximation by assuming a cognitive decoder that will take no action when the number of bit errors in each codeword exceeds 1. Its performance is marginally better at low $E_b/\mathcal{N}$ ratio.

REFERENCES

1. C. E. Shannon, "A Mathematical Theory of Communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, 1948.
2. S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 2004.
3. W. W. Peterson and E. J. Weldon, Jr., *Error Correcting Codes*, 2nd ed., Wiley, New York, 1972.
4. P. Elias, "Coding for Noisy Channels," *IRE Natl. Convention. Rec.*, vol. 3, part 4, pp. 37–46, 1955.
5. A. J. Viterbi, "Convolutional Codes and Their Performance in Communications Systems," *IEEE Trans. Commun. Technol.*, vol. CT-19, pp. 751–771, Oct. 1971.
6. J. L. Massey, *Threshold Decoding*, MIT Press, Cambridge, MA, 1963.
7. J. K. Wolf, "Efficient Maximum-Likelihood Decoding of Linear Block Codes Using a Trellis," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 76–80, Jan. 1978.
8. G. D. Forney, Jr., *Concatenated Codes*, MIT Press, Cambridge, MA, 1966.
9. D. Chase, "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information," *IEEE Trans. Inform. Theory*, IT-18, pp. 170–182, 1972.
10. J. Hagenauer and P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and Its Applications," *Proc. of IEEE Globecom*, pp. 1680–1686, Nov. 1989.

11. H. L. Van Trees, *Detection, Estimation, and Modulation Theory*, Part I, Wiley InterScience, 2001 (reprint), Hoboken, NJ.
12. C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error Correcting Coding and Decoding: Turbo Codes," *Proc. 1993 IEEE International Conference on Communications*, pp. 1064–1070, Geneva, Switzerland, May 1993.
13. L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimum Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, March 1974.
14. C. Berrou and A. Glavieux, "Near Optimum Error Correcting Coding and Decoding: Turbo-Codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.
15. R. G. Gallager, *Low Density Parity Check Codes*, Monograph, MIT Press, cambridge, MA, 1963.
16. David J. C. MacKay and R. M. Neal, "Good Codes Based on Very Sparse Matrices," *Fifth IMA Conference on Cryptography and Coding, Lecture Notes on Computer Science no. 1025*, Colin Boyd, Ed., Springer, Berlin, 1995 pp. 100–111.
17. D. J. C. MacKay and R. M. Neal, "Near Shannon Limit Performance of Low Density Parity Check Codes," *Electron. Lett.*, vol. 33, March 13, 1997.
18. R. M. Tanner, "A Recursive Approach to Low Complexity Codes," *IEEE Trans. Inform. Theory*, IT-27, pp. 533–547, Sept. 1981.

PROBLEMS

14.1-1 Golay's (23, 12) codes are three-error correcting codes. Verify that $n = 23$ and $k = 12$ satisfies the Hamming bound exactly for $t = 3$.

14.1-2 **(a)** Determine the Hamming bound for a ternary code (whose three code symbols are 0, 1, 2).

(b) A ternary (11, 6) code exists that can correct up to two errors. Verify that this code satisfies the Hamming bound exactly.

14.1-3 Confirm the possibility of a (18, 7) binary code that can correct up to three errors. Can this code correct up to four errors?

14.2-1 If $\boldsymbol{G}$ and $\boldsymbol{H}$ are the generator and parity check matrices, respectively, then show that

$$\boldsymbol{G}\boldsymbol{H}^T = 0$$

14.2-2 Given a generator matrix

$$\boldsymbol{G} = [1 \quad 1 \quad 1]$$

construct a (3, 1) code. How many errors can this code correct? Find the codeword for data vectors $\boldsymbol{d} = \mathbf{0}$ and $\boldsymbol{d} = \mathbf{1}$. Comment.

14.2-3 Repeat Prob. 14.2-2 for

$$\boldsymbol{G} = [1 \quad 1 \quad 1 \quad 1 \quad 1]$$

This gives a (5, 1) code.

14.2-4 A generator matrix

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

generates a (4,2) code.

(a) Is this a systematic code?

(b) What is the parity check matrix of this code?

(c) Find the codewords for all possible input bits.

(d) Determine the minimum distance of the code and the number of bit errors this code can correct.

14.2-5 Consider the following $(k+1, k)$ systematic linear block code with the parity check digit c_{k+1} given by

$$c_{k+1} = d_1 + d_2 + \cdots + d_k \tag{14.83}$$

(a) Construct the appropriate generator matrix for this code.

(b) Construct the code generated by this matrix for $k = 3$.

(c) Determine the error detecting or correcting capabilities of this code.

(d) Show that

$$\boldsymbol{c}\boldsymbol{H}^T = 0$$

and

$$\boldsymbol{r}\boldsymbol{H}^T = \begin{cases} \mathbf{0} & \text{if no error occurs} \\ \mathbf{1} & \text{if single error occurs} \end{cases}$$

14.2-6 Consider a generator matrix $\boldsymbol{G}$ for a nonsystematic (6, 3) code:

$$\boldsymbol{G} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Construct the code for this $\boldsymbol{G}$, and show that $d_{\min}$, the minimum distance between codewords, is 3. Consequently, this code can correct at least one error.

14.2-7 Repeat Prob. 14.2-6 if

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

14.2-8 Find a generator matrix $\boldsymbol{G}$ for a (15, 11) single-error correcting linear block code. Find the codeword for the data vector **10111010101**.

14.2-9 For a (6, 3) systematic linear block code, the three parity-check digits c_4, c_5, and c_6 are

$$c_4 = d_1 + d_2 + d_3$$
$$c_5 = d_1 + d_2$$
$$c_6 = d_1 + d_3$$

(a) Construct the appropriate generator matrix for this code.

(b) Construct the code generated by this matrix.

(c) Determine the error correcting capabilities of this code.

(d) Prepare a suitable decoding table.

(e) Decode the following received words: **101100**, **000110**, **101010**.

14.2-10 **(a)** Construct a code table for the (6, 3) code generated by the matrix $\boldsymbol{G}$ in Prob. 14.2-6.

(b) Prepare a suitable decoding table.

14.2-11 Construct a single-error correcting (7, 4) linear block code (Hamming code) and the corresponding decoding table.

14.2-12 For the (6, 3) code in Example 14.1, the decoding table is Table 14.3. Show that if we use this decoding table, and a two-error pattern **010100** or **001001** occurs, it will not be corrected. If it is desired to correct a single two-error pattern **010100** (along with six single-error patterns), construct the appropriate decoding table and verify that it does indeed correct one two-error pattern **010100** and that it cannot correct any other two-error patterns.

14.2-13 **(a)** Given $k = 8$, find the minimum value of n for a code that can correct at least one error.

(b) Choose a generator matrix $\boldsymbol{G}$ for this code.

(c) How many double errors can this code correct?

(d) Construct a decoding table (syndromes and corresponding correctable error patterns).

14.2-14 Consider a (6, 2) code generated by the matrix

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

(a) Construct the code table for this code and determine the minimum distance between codewords.

(b) Prepare a suitable decoding table.

Hint: This code can correct all single-error patterns, seven double-error patterns, and two triple-error patterns. Choose the desired seven double-error patterns and the two triple-error patterns.

14.3-1 **(a)** Use the generator polynomial $g(x) = x^3 + x + 1$ to construct a systematic (7, 4) cyclic code.

(b) What are the error correcting capabilities of this code?

(c) Construct the decoding table.

(d) If the received word is **1101100**, determine the transmitted data word.

14.3-2 A three-error correcting (23, 12) Golay code is a cyclic code with a generator polynomial

$$g(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$$

Determine the codewords for the data vectors **000011110000**, **101010101010**, and **11000101011110**.

14.3-3 Factorize the polynomial

$$x^3 + x^2 + x + 1$$

Hint: A third-order polynomial must have one factor of first order. The only first-order polynomials that are prime (not factorizable) are x and $x+1$. Since x is not a factor of the given polynomial, try $x+1$. Divide x^3+x^2+x+1 by $x+1$.

14.3-4 The concept explained in Prob. 14.3-3 can be extended to factorize any higher order polynomial. Using this technique, factorize

$$x^5+x^4+x^2+1$$

Hint: There must be at least one first-order factor. Try dividing by the two first-order prime polynomials x and $x+1$. The given fifth-order polynomial can now be expressed as $\phi_1(x)\phi_4(x)$, where $\phi_1(x)$ is a first-order polynomial and $\phi_4(x)$ is a fourth-order polynomial that may or may not contain a first-order factor. Try dividing $\phi_4(x)$ by x and $x+1$. If it does not work, it must have two second-order polynomials both of which are prime. The possible second-order polynomials are x^2, x^2+1, x^2+x, and x^2+x+1. Determine which of these are prime (not divisible by x or $x+1$). Now try dividing $\phi_4(x)$ by these prime polynomials of the second order. If neither divides, $\phi_4(x)$ must be a prime polynomial of the fourth order and the factors are $\phi_1(x)$ and $\phi_4(x)$.

14.3-5 Use the concept explained in Prob. 14.3-4 to factorize a seventh-order polynomial x^7+1.

Hint: Determine prime factors of first-, second-, and third-order polynomials. The possible third-order polynomials are $x^3, x^3+1, x^3+x, x^3+x+1, x^3+x^2, x^3+x^2+1, x^3+x^2+x$, and x^3+x^2+x+1. See hint in Prob. 14.3-4.

14.3-6 Equation (14.16) suggests a method of constructing a generator matrix $\boldsymbol{G}'$ for a cyclic code,

$$\boldsymbol{G}' = \begin{bmatrix} x^{k-1}g(x) \\ x^{k-2}g(x) \\ \cdots \\ g(x) \end{bmatrix} = \begin{bmatrix} g_1 & g_2 & \cdots & g_{n-k+1} & 0 & 0 & \cdots & 0 \\ 0 & g_1 & g_2 & \cdots & g_{n-k+1} & 0 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & g_1 & g_2 & \cdots & g_{n-k+1} \end{bmatrix}$$

where $g(x) = g_1x^{n-k} + g_2x^{n-k-1} + \cdots + g_{n-k+1}$ is the generator polynomial. This is, in general, a nonsystematic cyclic code.

(a) For a single-error correcting (7, 4) cyclic code with a generator polynomial $g(x) = x^3 + x^2 + 1$, find $\boldsymbol{G}'$ and construct the code.

(b) Verify that this code is identical to that derived in Example 14.3 (Table 14.4).

14.3-7 The generator matrix $\boldsymbol{G}$ for a systematic cyclic code (see Prob. 14.3-6) can be obtained by realizing that adding any row of a generator matrix to any other row yields another valid generator matrix, since the codeword is formed by linear combinations of data digits. Also, a generator matrix for a systematic code must have an identity matrix $\boldsymbol{I}_k$ in the first k columns. Such a matrix is formed step by step as follows. Observe that each row in $\boldsymbol{G}'$ in Prob. 14.3-6 is a left shift of the row below it, with the last row being $g(x)$. Start with the kth (last) row g(x). Because g(x) is of the order $n-k$, this row has the element 1 in the kth column, as required. For the $(k-1)$th row, use the last row with one left shift. We require a 0 in the kth column of the $(k-1)$th row to form $\boldsymbol{I}_k$. If there is a 0 in the kth column of this $(k-1)$th row, we accept it as a valid $(k-1)$th row. If not, then we add the kth row to the $(k-1)$th row to obtain 0 in its kth column. The resulting row is the final $(k-1)$th row. This row with a single left shift serves as the $(k-2)$th row. But if this newly formed $(k-2)$th row does not have a 0 in its kth column, we add the kth (last) row to it to get the desired 0. We continue this way until all k rows have been formed. This gives the generator matrix for a systematic (n, k) cyclic code.

(a) For a single-error correcting (7, 4) systematic cyclic code with a generator polynomial $g(x) = x^3 + x^2 + 1$, find G and construct the code.

(b) Verify that this code is identical to that in Table 14.5 (Example 14.4).

14.3-8 **(a)** Use the generator polynomial $g(x) = x^3 + x + 1$ to find the generator matrix $\boldsymbol{G}'$ for a nonsystematic (7, 4) cyclic code .

(b) Find the code generated by this matrix $\boldsymbol{G}'$.

(c) Determine the error correcting capabilities of this code.

14.3-9 Use the generator polynomial $g(x) = x^3 + x + 1$ (see Prob. 14.3-8) to find the generator matrix $\boldsymbol{G}$ for a systematic (7, 4) cyclic code.

14.3-10 Discuss the error correcting capabilities of an interleaved $(\lambda n, \lambda k)$ cyclic code with $\lambda = 10$ and using a three-error correcting (31, 16) BCH code.

14.3-11 The generator polynomial

$$g(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$$

generates a cyclic BCH (15, 5) code.

(a) Determine the (cyclic) code generating matrix.

(b) For encoder input data $\boldsymbol{d} = \mathbf{10110}$, find the corresponding codeword.

(c) Show how many errors this code can correct.

14.4-1 Uncoded data is transmitted by using PSK over an AWGN channel with $E_b/\mathcal{N} = 9$. This data is now coded using a three-error correcting (23, 12) Golay code (Prob. 14.1-1) and transmitted over the same channel at the same data rate and with the same transmitted power.

(a) Determine the corrected error probability P_{eu} and P_{ec} for the coded and the uncoded systems.

(b) If it is decided to achieve the error probability P_{ec} computed in part **(a)**, using the uncoded system by increasing the transmitted power, determine the required value of $E_b/\mathcal{N}$.

14.4-2 The simple code for detecting burst errors (Fig. 14.4) can also be used as a single-error correcting code with a slight modification. The k data digits are divided into groups of b digits in length, as in Fig. 14.4. To each group we add one parity check digit, so that each segment now has $b + 1$ digits (b data digits and one parity check digit). The parity check digit is chosen to ensure that the total number of **1**s in each segment of $b + 1$ digits is even. Now we consider these digits as our new data and augment them with the last segment of $b + 1$ parity check digits, as was done in Fig. 14.4. The data in Fig. 14.4 will be transmitted thus:

10111 01010 11011 10001 11000 01111

Show that this (30, 20) code is capable of single error correction as well as the detection of a single burst of length 5.

14.5-1 For the convolutional encoder in Fig. 14.5, the received bits are **01 00 01 00 10 11 11 00**. Use Viterbi's algorithm and the trellis diagram in Fig. 14.8 to decode this sequence.

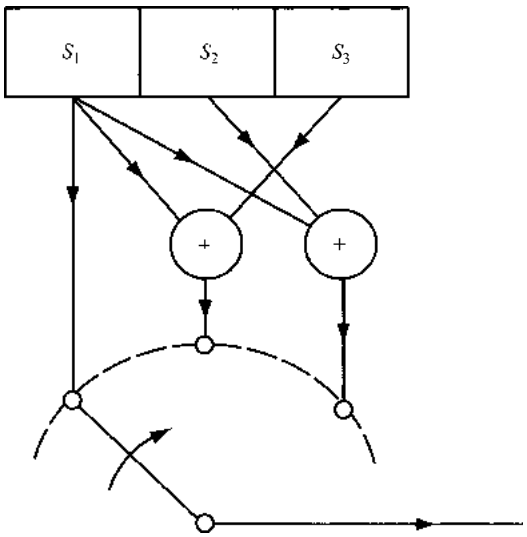
14.5-2 For the convolutional encoder shown in Fig. P14.5-2:

(a) Draw the state and trellis diagrams and determine the output digit sequence for the data digits **11010100**.

(b) Use Viterbi's algorithm to decode the following received sequences:

(i) 100 110 111 101 001 101 001 010
(ii) 010 110 111 101 101 101 001 010
(iii) 111 110 111 111 001 101 001 101

Figure P.14.5-2



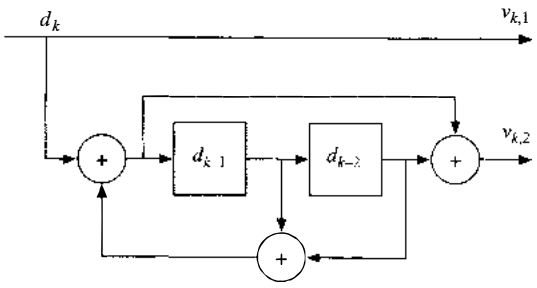
14.5-3 A systematic recursive convolution encoder (Fig. P14.5-3) generates a rate 1/2 code. Unlike earlier examples, this encoder is recursive with feedback branches. It turns out that we can still use a simple trellis and state transition diagram to represent this encoder. The maximum likelihood Viterbi decoder also applies. Denote the state value as (d_{k-1}, d_{k-2}).

(a) Illustrate the state transition diagram of this encoder.

(b) Find the corresponding trellis diagram.

(c) For an input data sequence of **0100110100**, determine the corresponding codeword.

Figure P.14.5-3



14.6-1 A block code has parity check matrix

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

(a) Find the code-generating matrix of this code.

(b) Find the minimum distance.

(c) Find the trellis diagram.

14.6-2 For the block code in Prob. 14.2-9,

(a) Find the code-generating matrix.

(b) Find the minimum distance.

(c) Find the trellis diagram.

APPENDIX A

ORTHOGONALITY OF SOME SIGNAL SETS

A.1 Orthogonality of the Trigonometric and Exponential Signal Set

Consider an integral I defined by

$$I = \int_{T_0} \cos n\omega_0 t \cos m\omega_0 t \, dt \tag{A.1a}$$

where $\int_{T_0}$ stands for integration over any contiguous interval of $T_0 = 2\pi/\omega_0$ seconds. By using a trigonometric identity (Appendix E), Eq. (A.1a) can be expressed as

$$I = \frac{1}{2}\left[\int_{T_0} \cos (n+m)\omega_0 t \, dt + \int_{T_0} \cos (n-m)\omega_0 t \, dt\right] \tag{A.1b}$$

Since $\cos \omega_0 t$ executes one complete cycle during any interval of T_0 seconds, $\cos (n+m)\omega_0 t$ executes $(n+m)$ complete cycles during any interval of duration T_0. Therefore, the first integral in Eq. (A.1b), which represents the area under $(n+m)$ complete cycles of a sinusoid, equals zero. The same argument shows that the second integral in Eq. (A.1b) is also zero, except when $n = m$. Hence, I in Eq. (A.1b) is zero for all $n \neq m$. When $n = m$, the first integral in Eq. (A.1b) is still zero, but the second integral yields

$$I = \frac{1}{2}\int_{T_0} dt = \frac{T_0}{2}$$

Thus,

$$\int_{T_0} \cos n\omega_0 t \cos m\omega_0 t \, dt = \begin{cases} 0 & n \neq m \\ \frac{T_0}{2} & m = n \neq 0 \end{cases} \tag{A.2a}$$

We can use similar arguments to show that

$$\int_{T_0} \sin n\omega_0 t \sin m\omega_0 t \, dt = \begin{cases} 0, & n \neq m \\ \frac{T_0}{2} & n = m \neq 0 \end{cases} \tag{A.2b}$$

and

$$\int_{T_0} \sin n\omega_0 t \cos m\omega_0 t \, dt = 0 \qquad \text{all } n \text{ and } m \tag{A.2c}$$

A.2 Orthogonality of the Exponential Signal Set

The set of exponentials $e^{jn\omega_0 t}$ $(n = 0, \pm 1, \pm 2, \ldots)$ is orthogonal over any interval of duration T_0, that is,

$$\int_{T_0} e^{jm\omega_0 t}(e^{jn\omega_0 t})^* \, dt = \int_{T_0} e^{j(m-n)\omega_0 t} \, dt = \begin{cases} 0 & m \neq n \\ T_0 & m = n \end{cases} \tag{A.3}$$

Let the integral on the left-hand side of Eq. (A.3) be I, where

$$\begin{aligned} I &= \int_{T_0} e^{jm\omega_0 t}(e^{jn\omega_0 t})^* \, dt \\ &= \int_{T_0} e^{j(m-n)\omega_0 t} \, dt \end{aligned} \tag{A.4}$$

The case of $m = n$ is trivial: the integrand is unity, and $I = T_0$. When $m \neq n$, however,

$$\begin{aligned} I &= \frac{1}{j(m-n)\omega_0} e^{j(m-n)\omega_0 t}\Big|_{t_1}^{t_1+T_0} \\ &= \frac{1}{j(m-n)\omega_0} e^{j(m-n)\omega_0 t_1}[e^{j(m-n)\omega_0 T_0} - 1] = 0 \end{aligned}$$

The last result follows from the fact that $\omega_0 T_0 = 2\pi$, and $e^{j2\pi k} = 1$ for all integral values of k.

APPENDIX B

CAUCHY-SCHWARZ INEQUALITY

Prove the following Cauchy-Schwarz inequality for a pair of real finite energy signals $f(t)$ and $g(t)$:

$$\left[\int_a^b f(t)g(t)\,dt\right]^2 \leq \left[\int_a^b f^2(t)\,dt\right]\left[\int_a^b g^2(t)\,dt\right] \tag{B.1}$$

with equality only if $g(t) = cf(t)$, where c is an arbitrary constant.

The Cauchy-Schwarz inequality for finite energy, complex-valued functions $X(\omega)$ and $Y(\omega)$ is given by

$$\left|\int_{-\infty}^{\infty} X(\omega)Y(\omega)\,d\omega\right|^2 \leq \int_{-\infty}^{\infty} |X(\omega)|^2 d\omega \int_{-\infty}^{\infty} |Y(\omega)|^2\,d\omega \tag{B.2}$$

with equality only if $Y(\omega) = cX^*(\omega)$, where c is an arbitrary constant.

We can prove Eq. (B.1) as follows: for any real value of λ, we know that

$$\int_a^b [\lambda f(t) - g(t)]^2\,dt \geq 0 \tag{B.3}$$

or

$$\lambda^2 \int_a^b f^2(t)\,dt - 2\lambda \int_a^b f(t)g(t)\,dt + \int_a^b g^2(t)\,dt \geq 0 \tag{B.4}$$

Because this quadratic equation in λ is nonnegative for any value of λ, its discriminant must be nonpositive, and Eq. (B.1) follows. If the discriminant is zero, then for some value of $\lambda = c$, the quadratic equals zero. This is possible only if $cf(t) - g(t) = 0$, and the result follows.

To prove Eq. (B.2), we observe that $|X(\omega)|$ and $|Y(\omega)|$ are real functions and inequality Eq. (B.1) applies. Hence,

$$\left[\int_a^b |X(\omega)Y(\omega)|\,d\omega\right]^2 \leq \int_a^b |X(\omega)|^2 d\omega \int_a^b |Y(\omega)|^2\,d\omega \tag{B.5}$$

with equality only if $|Y(\omega)| = c|X(\omega)|$, where c is an arbitrary constant. Now recall that

$$\left| \int_a^b X(\omega)Y(\omega)\, d\omega \right| \leq \int_a^b |X(\omega)||Y(\omega)|\, d\omega = \int_a^b |X(\omega)Y(\omega)|\, d\omega \tag{B.6}$$

with equality if and only if $Y(\omega) = cX^*(\omega)$, where c is an arbitrary constant. Equation (B.2) immediately follows from Eqs. (B.5) and (B.6).

APPENDIX C

GRAM-SCHMIDT ORTHOGONALIZATION OF A VECTOR SET

We have defined the dimensionality of a vector space as the maximum number of independent vectors in the space. Thus in an N-dimensional space, there can be no more than N vectors that are independent. Alternatively, it is always possible to find a set of N vectors that are independent. Once such a set has been chosen, any vector in this space can be expressed in terms of (as a linear combination of) the vectors in this set. This set forms what we commonly refer to as a basis set, which forms the coordinate system. This set of N independent vectors is by no means unique. The reader is familiar with this propery in the physical space of three dimensions, where one can find an infinite number of independent sets of three vectors. This is clear from the fact that we have an infinite number of possible coordinate systems. An orthogonal set, however, is of special interest because it is easier to deal with than a nonorthogonal set. If we are given a set of N independent vectors, it is possible to obtain from this set another set of N independent vectors that is orthogonal. This is done by the Gram-Schmidt process of orthogonalization.

In the following derivation, we use the result [derived in Eq. (2.27)] that the projection (or component) of a vector $\mathbf{x}_2$ upon another vector $\mathbf{x}_1$ (see Fig. C.1) is $c_{12}\mathbf{x}_1$, where

$$c_{12} = \frac{< \mathbf{x}_1, \mathbf{x}_2 >}{\|\mathbf{x}_1\|^2}\mathbf{y}_1 \tag{C.1}$$

The error in this approximation is the vector $\mathbf{x}_2 - c_{12}\mathbf{x}_1$, that is,

$$\text{error vector} = \mathbf{x}_2 - \frac{< \mathbf{x}_1, \mathbf{x}_2 >}{\|\mathbf{x}_1\|^2}\mathbf{x}_1 \tag{C.2}$$

The error vector, shown dashed in Fig. C.1 is orthogonal to vector $\mathbf{x}_1$.

To get physical insight into this procedure, we shall consider a simple case of two-dimensional space. Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be two independent vectors in a two-dimensional space (Fig. C.1). We wish to generate a new set of two orthogonal vectors $\mathbf{y}_1$ and $\mathbf{y}_2$ from $\mathbf{x}_1$ and $\mathbf{x}_2$. For convenience, we shall choose

$$\mathbf{y}_1 = \mathbf{x}_1 \tag{C.3}$$

We now find another vector $\mathbf{y}_2$ that is orthogonal to $\mathbf{y}_1$ (and $\mathbf{x}_1$. Figure C.1 shows that the error vector in approximation of $\mathbf{x}_2$ by $\mathbf{y}_1$ (dashed lines) is orthogonal to $\mathbf{y}_1$, and can be taken as $\mathbf{y}_2$.

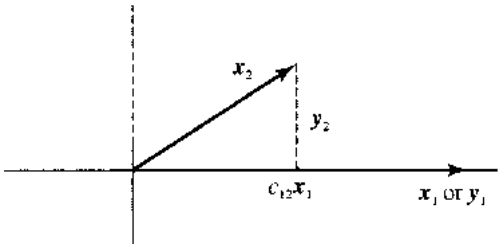


Figure C.1 Gram-Schmidt process for a two-dimensional case.

Hence,

$$\begin{aligned}\mathbf{y}_2 &= \mathbf{x}_2 - \frac{< \mathbf{x}_1, \mathbf{x}_2 >}{\|\mathbf{x}_1\|^2}\mathbf{x}_1 \\ &= \mathbf{x}_2 - \frac{< \mathbf{y}_1, \mathbf{x}_2 >}{\|\mathbf{y}_1\|^2}\mathbf{y}_1 \end{aligned} \tag{C.4}$$

Equations (C.3) and (C.4) yield the desired orthogonal set. Note that this set is not unique. There is an infinite number of possible orthogonal vector sets $(\mathbf{y}_1, \mathbf{y}_2)$ that can be generated from $(\mathbf{x}_1, \mathbf{x}_2)$. In our derivation, we could as well have started with $\mathbf{y} = \mathbf{x}_2$ instead of $\mathbf{y}_1 = \mathbf{x}_1$. This starting point would have yielded an entirely different set.

The reader can extend these results to a three-dimensional case. If vectors $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ form an independent set in this space, then we form vectors $\mathbf{y}_1$ and $\mathbf{y}_2$ as in Eqs. (C.3) and (C.4). To determine $\mathbf{y}_3$, we approximate $\mathbf{x}_3$ in terms of vectors $\mathbf{y}_1$ and $\mathbf{y}_2$. The error in this approximation must be orthogonal to both $\mathbf{y}_1$ and $\mathbf{y}_2$ and, hence, can be taken as the third orthogonal vector $\mathbf{y}_3$. Hence,

$$\begin{aligned}\mathbf{y}_3 &= \mathbf{x}_3 - \text{sum of projections of } \mathbf{x}_3 \text{ on } \mathbf{y}_1 \text{ and } \mathbf{y}_2 \\ &= \mathbf{x}_3 - \frac{< \mathbf{y}_1, \mathbf{x}_3 >}{\|\mathbf{y}_1\|^2}\mathbf{y}_1 - \frac{< \mathbf{y}_2, \mathbf{x}_3 >}{\|\mathbf{y}_2\|^2}\mathbf{y}_2 \end{aligned} \tag{C.5}$$

These results can be extended to an N-dimensional space. In general, given N independent vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$, if we proceed along similar lines, we can obtain an orthogonal set $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N$, where

$$\mathbf{y}_1 = \mathbf{x}_1 \tag{C.6}$$

and

$$\mathbf{y}_j = \mathbf{x}_j - \sum_{k=1}^{j-1} \frac{< \mathbf{y}_k, \mathbf{x}_j >}{\|\mathbf{y}_k\|^2}\mathbf{y}_k \qquad j = 2, 3, \ldots, N \tag{C.7}$$

Note that this is one of the infinitely many orthogonal sets that can be formed from the set $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$. Moreover, this set is not an orthonormal set. The orthonormal set $\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_N$ can be obtained by normalizing the lengths of the respective vectors,

$$\hat{\mathbf{y}}_k = \frac{\mathbf{y}_k}{\|\mathbf{y}_k\|}$$

We can apply these concepts to signal space because one-to-one correspondence exists between signals and vectors. If we have N independent signals $x_1(t), x_2(t), \ldots, x_N(t)$, we can form a

set of N orthogonal signals $y_1(t), y_2(t), \ldots, y_N(t)$ as

$$y_1(t) = x(t)$$

$$y_j(t) = x_j(t) - \sum_{k=1}^{j-1} c_{kj} y_k(t) \qquad j = 2, 3, \ldots, N \tag{C.8}$$

where

$$c_{kj} = \frac{\int y_k(t) x_j(t)\, dt}{\int y_k^2(t)\, dt} \tag{C.9}$$

Note that this is one of the infinitely many possible orthogonal sets that can be formed from the set $x_1(t), x_2(t), \ldots, x_N(t)$. The set can be normalized by dividing each signal $y_j(t)$ by its energy.

Example C.1 The exponential signals

$$g_1(t) = e^{-pt} u(t)$$

$$g_2(t) = e^{-2pt} u(t)$$

$$\vdots$$

$$g_N(t) = e^{-Npt} u(t)$$

form an independent set of signals in N-dimensional space, where N may be any integer. This set, however, is not orthogonal. We can use the Gram-Schmidt process to obtain an orthogonal set for this space. If $y_1(t), y_2(t), \ldots, y_N(t)$ is the desired orthogonal basis set, we choose

$$y_1(t) = g_1(t) = e^{-pt} u(t)$$

From Eqs. (C.8) and (C.9) we have

$$y_2(t) = x_2(t) - c_{12} y_1(t)$$

where

$$c_{12} = \frac{\int_{-\infty}^{\infty} y_1(t) x_2(t)\, dt}{\int_{-\infty}^{\infty} y_1^2(t)\, dt}$$

$$= \frac{\int_0^{\infty} e^{-pt} e^{-2pt}\, dt}{\int_0^{\infty} e^{-2pt}\, dt}$$

$$= \frac{2}{3}$$

Hence,

$$y_2(t) = (e^{-2pt} - \tfrac{2}{3} e^{-pt}) u(t) \tag{C.10}$$

Similarly, we can proceed to find the remaining functions $y_3(t), \ldots, y_N(t)$, and so on. The reader can verify that all this represents a mutually orthogonal set.

APPENDIX D

BASIC MATRIX PROPERTIES AND OPERATIONS

D.1 Notation

An $n \times 1$ column vector $\boldsymbol{x}$ consists of n entries and is formed by

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{D.1a}$$

The transpose of $\boldsymbol{x}$ is a row vector represented by

$$\boldsymbol{x}^T = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \tag{D.1b}$$

The conjugate transpose of $\boldsymbol{x}$ is also a row vector written as

$$\boldsymbol{x}^H = (\boldsymbol{x}^*)^T = \begin{bmatrix} x_1^* & x_2^* & \cdots & x_n^* \end{bmatrix} \tag{D.1c}$$

$\boldsymbol{x}^H$ is also known as the Hermitian of $\boldsymbol{x}$.

An $m \times n$ matrix consists of n column vectors

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_n \end{bmatrix} \tag{D.2a}$$

$$= \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \tag{D.2b}$$

We also define its transpose and Hermitian, respectively, as

$$\boldsymbol{A}^T = \begin{bmatrix} a_{1,1} & a_{2,1} & \cdots & a_{m,1} \\ a_{1,2} & a_{2,2} & \cdots & a_{m,2} \\ \vdots & \vdots & \cdots & \vdots \\ a_{1,n} & a_{2,n} & \cdots & a_{m,n} \end{bmatrix} \qquad \boldsymbol{A}^H = \begin{bmatrix} a_{1,1}^* & a_{2,1}^* & \cdots & a_{m,1}^* \\ a_{1,2}^* & a_{2,2}^* & \cdots & a_{m,2}^* \\ \vdots & \vdots & \cdots & \vdots \\ a_{1,n}^* & a_{2,n}^* & \cdots & a_{m,n}^* \end{bmatrix} \tag{D.2c}$$

- If $\boldsymbol{A}^T = \boldsymbol{A}$, then we say that $\boldsymbol{A}$ is a symmetric matrix.
- If $\boldsymbol{A}^H = \boldsymbol{A}$, then we say that $\boldsymbol{A}$ is a Hermitian matrix.
- If $\boldsymbol{A}$ consists of only real entries, then it is both Hermitian and symmetric.

D.2 Matrix Product and Properties

For an $m \times n$ matrix $\boldsymbol{A}$ and an $n \times \ell$ matrix $\boldsymbol{B}$ with

$$\boldsymbol{B} = \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,\ell} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,\ell} \\ \vdots & \vdots & \cdots & \vdots \\ b_{n,1} & b_{n,2} & \cdots & b_{n,\ell} \end{bmatrix} \tag{D.3}$$

the matrix product $\boldsymbol{C} = \boldsymbol{A} \cdot \boldsymbol{B}$ has dimension $m \times \ell$ and equals

$$\boldsymbol{C} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & a_{1,\ell} \\ c_{2,1} & c_{2,2} & \cdots & a_{2,\ell} \\ \vdots & \vdots & \cdots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & a_{m,\ell} \end{bmatrix} \quad \text{where} \quad c_{i,j} = \sum_{k=1}^{n} a_{i,k} b_{k,j} \tag{D.4}$$

In general $\boldsymbol{AB} \neq \boldsymbol{BA}$. In fact, the products may not even be well defined. To be able to multiply $\boldsymbol{A}$ and $\boldsymbol{B}$, the number of columns of $\boldsymbol{A}$ must equal the number of rows of $\boldsymbol{B}$.

In particular, the product of a row vector and a column vector is

$$\boldsymbol{y}^H \boldsymbol{x} = \sum_{k=1}^{n} y_k^* x_k \tag{D.5a}$$

$$= < \boldsymbol{x}, \boldsymbol{y} > \tag{D.5b}$$

Therefore, $\boldsymbol{x}^H \boldsymbol{x} = \|\boldsymbol{x}\|^2$.

Two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are orthogonal if $\boldsymbol{y}^H \boldsymbol{x} = \boldsymbol{x}^H \boldsymbol{y} = 0$.

There are several commonly used properties of matrix products:

$$\boldsymbol{A}(\boldsymbol{B} + \boldsymbol{C}) = \boldsymbol{AB} + \boldsymbol{AC} \tag{D.6a}$$

$$\boldsymbol{A}(\boldsymbol{BC}) = (\boldsymbol{AB})\boldsymbol{C} \tag{D.6b}$$

$$(\boldsymbol{AB})^* = \boldsymbol{A}^* \boldsymbol{B}^* \tag{D.6c}$$

$$(\boldsymbol{AB})^T = \boldsymbol{B}^T \boldsymbol{A}^T \tag{D.6d}$$

$$(\boldsymbol{AB})^H = \boldsymbol{B}^H \boldsymbol{A}^H \tag{D.6e}$$

D.3 Identity and Diagonal Matrices

An $n \times n$ square matrix is diagonal if all its off-diagonal entries are zero, that is,

$$D = \text{diag}\,(d_1, d_2, \ldots, d_n) \tag{D.7a}$$

$$= \begin{bmatrix} d_1 & 0 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & d_{n-1} & 0 \\ 0 & 0 & \cdots & 0 & d_n \end{bmatrix} \tag{D.7b}$$

An identity matrix $\boldsymbol{I}_n$ has unit diagonal entries

$$\boldsymbol{I}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & 1 \end{bmatrix}_{n \times n} \tag{D.8}$$

For an $n \times n$ square matrix $\boldsymbol{A}$, if there exists a $n \times n$ square matrix $\boldsymbol{B}$ such that

$$\boldsymbol{BA} = \boldsymbol{AB} = \boldsymbol{I}_n$$

then

$$\boldsymbol{B} = \boldsymbol{A}^{-1} \tag{D.9}$$

is the inverse matrix of $\boldsymbol{A}$. For example, given a diagonal matrix

$$D = \text{diag}\,(d_1, d_2, \ldots, d_n)$$

$$D^{-1} = \text{diag}\left(\frac{1}{d_1}, \frac{1}{d_2}, \ldots, \frac{1}{d_n}\right)$$

D.4 Determinant of Square Matrices

The **determinant** of $n \times n$ square matrix $\boldsymbol{A}$ is defined recursively by

$$\det(\boldsymbol{A}) = \sum_{i=1}^{n} a_{ij}(-1)^{i+j}\boldsymbol{M}_{ij} \tag{D.10}$$

where $\boldsymbol{M}_{ij}$ is an $(n-1) \times (n-1)$ matrix known as the **minor** of $\boldsymbol{A}$ by eliminating its ith row and its jth column. Specifically, for a 2×2 matrix,

$$\det\begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc$$

Based on the definition of determinant, for a scalar α,

$$\det(\alpha \boldsymbol{A}) = \alpha^n \det(\boldsymbol{A}) \tag{D.11a}$$

$$\det\left(\boldsymbol{A}^T\right) = \det(\boldsymbol{A}) \tag{D.11b}$$

For an identity matrix

$$\det(\boldsymbol{I}) = 1 \tag{D.11c}$$

Also, for two square matrices $\boldsymbol{A}$ and $\boldsymbol{B}$,

$$\det(\boldsymbol{AB}) = \det(\boldsymbol{A}) \det(\boldsymbol{B}) \tag{D.11d}$$

Therefore,

$$\det\left(\boldsymbol{A}\boldsymbol{A}^{-1}\right) = \det(\boldsymbol{A}) \det\left(\boldsymbol{A}^{-1}\right) = 1 \tag{D.11e}$$

For an $m \times n$ matrix $\boldsymbol{A}$ and an $n \times m$ matrix $\boldsymbol{B}$, we have

$$\det(\boldsymbol{I}_m + \boldsymbol{AB}) = \det(\boldsymbol{I}_n + \boldsymbol{BA}) \tag{D.12}$$

D.5 Trace

The trace of square matrix $\boldsymbol{A}$ is the sum of its diagonal entries

$$\mathrm{Tr}(\boldsymbol{A}) = \sum_{i=1}^{n} a_{i,i} \tag{D.13}$$

For an $m \times n$ matrix $\boldsymbol{A}$ and an $n \times m$ matrix $\boldsymbol{B}$, we have

$$\mathrm{Tr}(\boldsymbol{AB}) = \mathrm{Tr}(\boldsymbol{BA}) \tag{D.14}$$

D.6 Eigendecomposition

If the $n \times n$ square matrix $\boldsymbol{A}$ is Hermitian, then the equation

$$\boldsymbol{Au} = \lambda \boldsymbol{u} \tag{D.15}$$

specifies an eigenvalue λ and the associated eigenvector $\boldsymbol{u}$.

When $\boldsymbol{A}$ is Hermitian, its eigenvalues are real-valued. Furthermore, $\boldsymbol{A}$ can be decomposed into

$$\boldsymbol{A} = \boldsymbol{U} \Lambda \boldsymbol{U}^H \tag{D.16}$$

in which the matrix

$$\boldsymbol{U} = [\boldsymbol{u}_1 \ \boldsymbol{u}_2 \ \cdots \ \boldsymbol{u}_n] \tag{D.17}$$

consists of orthogonal eigenvectors such that

$$\boldsymbol{U}\boldsymbol{U}^H = \boldsymbol{I}_n \tag{D.18}$$

Matrices satisfying this property are called unitary.

Furthermore, the diagonal matrix

$$\Lambda = \text{diag}\,(\lambda_1\ \lambda_2\ \cdots\ \lambda_n) \tag{D.19}$$

consists of the corresponding eigenvalues of $\boldsymbol{A}$.

Because

$$\boldsymbol{U}^H\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^H = \boldsymbol{I}_n \tag{D.20}$$

we can also write

$$\boldsymbol{U}^H\boldsymbol{A}\boldsymbol{U} = \Lambda \tag{D.21}$$

The eigenvalues of $\boldsymbol{A}$ are very useful characteristics. In particular,

$$\det(\boldsymbol{A}) = \prod_{i=1}^{n} \lambda_i \tag{D.22a}$$

$$\text{Trace}\,(\boldsymbol{A}) = \sum_{i=1}^{n} \lambda_i \tag{D.22b}$$

D.7 Special Hermitian Square Matrices

Let an $n \times n$ matrix $\boldsymbol{A}$ be Hermitian. $\boldsymbol{A}$ is **positive definite** if for any $n \times 1$ vector $\boldsymbol{x} \neq 0$, we have

$$\boldsymbol{x}^H\boldsymbol{A}\boldsymbol{x} > 0 \tag{D.23}$$

$\boldsymbol{A}$ is **semipositive definite** if for any $n \times 1$ vector $\boldsymbol{x}$, we have

$$\boldsymbol{x}^H\boldsymbol{A}\boldsymbol{x} \geq 0 \tag{D.24}$$

$\boldsymbol{A}$ is **negative definite** if for any $n \times 1$ vector $\boldsymbol{x} \neq 0$, we have

$$\boldsymbol{x}^H\boldsymbol{A}\boldsymbol{x} < 0 \tag{D.25}$$

$\boldsymbol{A}$ is positive definite if and only if all its eigenvalues are positive.

APPENDIX E

MISCELLANEOUS

E.1 L'Hôpital's Rule

If $\lim f(x)/g(x)$ results in the indeterministic form $0/0$ or ∞/∞, then

$$\lim \frac{f(x)}{g(x)} = \lim \frac{\dot{f}(x)}{\dot{g}(x)} \tag{E.1}$$

E.2 Taylor and Maclaurin Series

$$f(x) = f(a) + \frac{(x-a)}{1!}\dot{f}(a) + \frac{(x-a)^2}{2!}\ddot{f}(a) + \cdots$$

$$f(x) = f(0) + \frac{x}{1!}\dot{f}(0) + \frac{x^2}{2!}\ddot{f}(0) + \cdots$$

E.3 Power Series

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + \cdots$$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \cdots$$

$$\tan x = x + \frac{x^3}{3} + \frac{2x^5}{15} + \frac{17x^7}{315} + \cdots \qquad x^2 < \frac{\pi^2}{4}$$

$$Q(x) = \frac{e^{-x^2/2}}{x\sqrt{2\pi}}\left(1 - \frac{1}{x^2} + \frac{1\cdot 3}{x^4} - \frac{1\cdot 3\cdot 5}{x^6} + \cdots\right)$$

$$(1+x)^n = 1 + nx + \frac{n(n-1)}{2!}x^2 + \frac{n(n-1)(n-2)}{3!}x^3 + \cdots + \binom{n}{k}x^k + \cdots + x^n$$

$$\approx 1 + nx \qquad |x| \ll 1$$

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \cdots \qquad |x| < 1$$

E.4 Sums

$$\sum_{m=0}^{k} r^m = \frac{r^{k+1} - 1}{r - 1} \qquad r \neq 1$$

$$\sum_{m=M}^{N} r^m = \frac{r^{N+1} - r^M}{r - 1} \qquad r \neq 1$$

$$\sum_{m=0}^{k} \left(\frac{a}{b}\right)^m = \frac{a^{k+1} - b^{k+1}}{b^k(a - b)} \qquad a \neq b$$

E.5 Complex Numbers

$$e^{\pm j\pi/2} = \pm j$$

$$e^{\pm jn\pi} = \begin{cases} 1 & n \text{ even} \\ -1 & n \text{ odd} \end{cases}$$

$$e^{\pm j\theta} = \cos\theta \pm j \sin\theta$$

$$a + jb = re^{j\theta} \qquad r = \sqrt{a^2 + b^2}, \quad \theta = \tan^{-1}\left(\frac{b}{a}\right)$$

$$(re^{j\theta})^k = r^k e^{jk\theta}$$

$$(r_1 e^{j\theta_1})(r_2 e^{j\theta_2}) = r_1 r_2 e^{j(\theta_1 + \theta_2)}$$

E.6 Trigonometric Identities

$$e^{\pm jx} = \cos x \pm j \sin x$$

$$\cos x = \frac{1}{2}(e^{jx} + e^{-jx})$$

$$\sin x = \frac{1}{2j}(e^{jx} - e^{-jx})$$

$$\cos\left(x \pm \frac{\pi}{2}\right) = \mp \sin x$$

$$\sin\left(x \pm \frac{\pi}{2}\right) = \pm \cos x$$

$$2 \sin x \cos x = \sin 2x$$

$$\sin^2 x + \cos^2 x = 1$$

$$\cos^2 x - \sin^2 x = \cos 2x$$

$$\cos^2 x = \frac{1}{2}(1 + \cos 2x)$$

$$\sin^2 x = \frac{1}{2}(1 - \cos 2x)$$

$$\cos^3 x = \frac{1}{4}(3\cos x + \cos 3x)$$

$$\sin^3 x = \frac{1}{4}(3\sin x - \sin 3x)$$

$$\sin(x \pm y) = \sin x \cos y \pm \cos x \sin y$$

$$\cos(x \pm y) = \cos x \cos y \mp \sin x \sin y$$

$$\tan(x \pm y) = \frac{\tan x \pm \tan y}{1 \mp \tan x \tan y}$$

$$\sin x \sin y = \frac{1}{2}[\cos(x - y) - \cos(x + y)]$$

$$\cos x \cos y = \frac{1}{2}[\cos(x - y) + \cos(x + y)]$$

$$\sin x \cos y = \frac{1}{2}[\sin(x - y) + \sin(x + y)]$$

$$a\cos x + b\sin x = C\cos(x + \theta)$$

$$\text{in which } C = \sqrt{a^2 + b^2} \quad \text{and} \quad \theta = \tan^{-1}\left(\frac{-b}{a}\right)$$

E.7 Indefinite Integrals

$$\int u\,dv = uv - \int v\,du$$

$$\int f(x)\dot{g}(x)\,dx = f(x)g(x) - \int \dot{f}(x)g(x)\,dx$$

$$\int \sin ax\,dx = -\frac{1}{a}\cos ax \qquad \int \cos ax\,dx = \frac{1}{a}\sin ax$$

$$\int \sin^2 ax\,dx = \frac{x}{2} - \frac{\sin 2ax}{4a} \qquad \int \cos^2 ax\,dx = \frac{x}{2} + \frac{\sin 2ax}{4a}$$

$$\int x\sin ax\,dx = \frac{1}{a^2}(\sin ax - ax\cos ax)$$

$$\int x\cos ax\,dx = \frac{1}{a^2}(\cos ax + ax\sin ax)$$

$$\int x^2\sin ax\,dx = \frac{1}{a^3}(2ax\sin ax + 2\cos ax - a^2x^2\cos ax)$$

$$\int x^2\cos ax\,dx = \frac{1}{a^3}(2ax\cos ax - 2\sin ax + a^2x^2\sin ax)$$

$$\int \sin ax\sin bx\,dx = \frac{\sin(a-b)x}{2(a-b)} - \frac{\sin(a+b)x}{2(a+b)} \qquad a^2 \neq b^2$$

$$\int \sin ax\cos bx\,dx = -\left[\frac{\cos(a-b)x}{2(a-b)} + \frac{\cos(a+b)x}{2(a+b)}\right] \qquad a^2 \neq b^2$$

$$\int \cos ax \cos bx \, dx = \frac{\sin (a-b)x}{2(a-b)} + \frac{\sin (a+b)x}{2(a+b)} \qquad a^2 \neq b^2$$

$$\int e^{ax} \, dx = \frac{1}{a} e^{ax}$$

$$\int x e^{ax} \, dx = \frac{e^{ax}}{a^2}(ax - 1)$$

$$\int x^2 e^{ax} \, dx = \frac{e^{ax}}{a^3}(a^2x^2 - 2ax + 2)$$

$$\int e^{ax} \sin bx \, dx = \frac{e^{ax}}{a^2 + b^2}(a \sin bx - b \cos bx)$$

$$\int e^{ax} \cos bx \, dx = \frac{e^{ax}}{a^2 + b^2}(a \cos bx + b \sin bx)$$

$$\int \frac{1}{x^2 + a^2} \, dx = \frac{1}{a} \tan^{-1} \frac{x}{a}$$

$$\int \frac{x}{x^2 + a^2} \, dx = \frac{1}{2} \ln (x^2 + a^2)$$

INDEX

Adaptive delta modulation (ADM), 300
Adaptive differential pulse code modulation (ADPCM), 294–295
Additive white Gaussian noise (AWGN), 10, 536
Aliasing error (spectral folding), 121, 259–261
All-pass vs. distortionless system, 93–94
Alternate mark inversion (AMI), 327, 339–341
Amplitude modulation (AM), 11, 83–84, 140, 141–142, 151–158, 470–471
  bandwidth-efficient, 158–167
  demodulation of, 156–158
  double-sideband, 142–151
  generation of, 156
  pulse (PAM), 267
  single-sideband (SSB), 159–160
  vestigial sideband (VSB), 167–170
Amplitude shift keying (ASK), 373, 581–584
  and AM modulation, connection between, 376
  binary (BASK), 521
  detection, 377
Analog signals, 4, 23–24
Analog-to-digital (A/D) conversion, 6–7, 251
  maximum information rate, 262–263
  nonideal practical sampling analysis, 263–267
  signal reconstruction, 253–258
    aliasing error (spectral folding), 259–261
    antialiasing filter, 261
    filters, realizability of, 258–259
    ideal, 254–255
    practical, 255–258
Angle modulation, 141–142, 202, 204
  bandwidth of, 209–222
  features of, 229–231
  immunity of, 234–235
  narrowband, 210–211
  power of, 206–209
Antheil, George, 623
Antialiasing filter, 261
  vs. matched filter, 670–673
Antipodal signals, 574
Aperiodic signals, 24–25
  representation by Fourier integral, 62–69
    conjugate symmetry property, 66
    Fourier transform, existence of, 67–68
    Fourier transform, linearity of, 68
    Fourier transform, physical appreciation of, 68–69
A posteriori probability, 541, 749
Arbitrarily small error probability, 802
Armstrong, Edwin H., 220–222
  indirect methods of, 225–227
Asymmetric digital subscriber line (ADSL), 707–708
Asynchronous channels, 287–288
Autocorrelation function, 36, 109–111
  of power signals, 113–117
Automatic gain control (AGC), 103
Automatic repeat request (ARQ), 805

Balanced circuits, 147
Balanced discriminator, 233
Balanced modulators, 147
  double, 147
  single, 147
Band-limited white Gaussian noise, 761
Bandpass limiter, 223–224
Bandpass matched filter, as coherent receiver, 521–522
Bandpass random process, 491–499
Bandpass signals, 85–87
Bandwidth, 9
  of angle modulated waves, 209–222
  essential, 105–108, 122, 335
  of product of signals, 88
Bandwidth-efficient amplitude modulations, 158–167
Baseband analog systems, performance analysis of, 486–488
Baseband communications, 140–141
Baseband signal, 2, 11, 140
Basis functions, 39, 530–531
Basis vectors, 37, 526
Bayes' decision rule, 399, 424, 542, 578, 749
Bayes receiver, 578–579
Bayes' theorem, 406–407
BCJR MAP decoding algorithm, 846–850
Bernoulli trials, 400–404
Bessel function, modified zero-order, 498
Best filter, 509
Bezout identity, 686–687
Binary
  amplitude shift keying (BASK), 521
  with eight-zero substitution (B8ZS), 343
  message, 4
  with N zero substitution (BNZS) signaling, 343
  phase shift keying (BPSK), 520–521
  polar signaling, optimum linear detector for, 506–512
  signaling, 512–520
  with six-zero substitution (B6ZS), 343
  symmetric channel (BSC), 745, 809
  systems, 516–520
  with three-zero substitution (B3ZS), 343
  threshold detection, 507–508
Bipartite (Tanner) graph, 856–857
Bipolar (pseudoternary) signaling, 327, 339–341
Bit (binary digit), 269
Bit error rate (BER), 506, 515, 553, 823
  of orthogonal signaling, 566–567
Bit-flipping LDPC decoding, 857
Bit loading, 705–706
Bit stuffing, 287–288
Blind equalization, 711–712
Block codes, 802
  linear, 806, 808
Block interleaver, 839
Bluetooth, 621–622
Bose-Chaudhuri-Hocquenghen (BCH) codes, 822
Bounded-input–bounded-output (BIBO) linear system, 90
Broad OFDM applications, 709
Burst error correction codes, 366, 826
Butterworth filters, 97

Carrier, 11, 83
Carrier communications, 141
  OFDM, 692–702
Carrier power, 155–156
Carson's rule, 213
Cauchy-Schwartz inequality, 35n, 489, 509, 875
Causal signal, 27
CCITT (Comité Consultatif International Téléphonique et Télégraphique), 172
cdmaOne (IS-95), 637, 644, 645–646
Cellular networks, 643–644
Cellular systems, CDMA in, 644–645
Central limit theorem
  for sample mean, 446–448
  for sum of independent random variables, 448
Channel, 3
Channel bank, 289
Channel capacity, 10, 745, 751, 753, 754, 764
  of band-limited AWGN channel, 764–767
  of continuous memoryless channel, 756

of discrete memoryless channel, 748
of infinite bandwidth, 767
Channel diversity, 715
Channel equalization, 669
receiver, 670–676
Channel estimation, 688–689
Channel matrix, 749
Channel shortening, 706
Chase algorithms, 842–843
Chebychev filters, 97
Chebyshev's inequality, 435–436
Chrominance signals, 167
Cochannel interference, 167
Code-division multiple access (CDMA), 623
in cellular phone networks, 643–647
of DSSS, 630–637, 643–649
in GPS, 647–649
power control in, 636–637
Code efficiency, 742
Code generator polynomial, 814
Code rate, 802, 803
Code tree, 828
Coherent AM demodulation, 152
Coherent receivers, for digital carrier modulations, 520–525
Color burst, 167
Common channel interoffice signaling (CCIS), 284
Communication
baseband, 140–141
carrier, 141
systems, 1–4
Compact code, 741
Compact disc, 270
Compandor, 275, 276–277
Complement, 394
Complete orthogonal basis, 37, 38
Complete orthogonal set, 38, 527
Concatenated codes, 840–841
Conditional densities, 424
Conditional entropy, 749
Conditional probability, 398–400
multiplication rule for, 404
of random variables, 410–412
Conference on European Postal and Telegraph Administration (CEPT), 284
Constraint length, 827
Continuous phase frequency shift keying (CPFSK), 524
Continuous random variables, 409, 413–416
Continuous time signal, 23
Convolution
codes, 802, 827
frequency, 87
time, 87
Convolution theorem, 87–88
Coprime, 687
Correlation coefficient, 34–36, 436–439
Correlation functions, 35–36
Correlation receiver, 512
Costas loop, 180
Cross-correlation coefficients, 35, 561
Cross-correlation function, 479
Cross-power spectral density, 479–480
Cumulative distribution function (CDF), 412–413, 458
Cyclic codes, 813–822
Cyclic linear block code theorem, 814
Cyclic prefix, 694, 706
Cyclic Redundancy Check (CRC) codes, 822

D-dimensional sphere, 769–772
Decision feedback equalizer (DFE), 689–692
error propagation in, 691–692
Decision feedback MUD receiver, 642–643
Decision regions and error probability, 545–551
Decorrelator MUD receiver, 640
Deductive logic, 407
Delta modulation (DM), 141, 295–300
adaptive (ADM), 300
comparison with PCM, 296–297
overloading in, 297–299
sigma-delta modulation, 299–300
threshold of coding, 297–299
Demodulation, 11, 13, 140, 143–146, 202
of AM signals, 156–158
of DSB-SC signals, 151
of FM signals, 231–234

Destination, 3
Detection error probability, 365–366
Detection signal space dimensionality, 538–541
Deterministic signals, 25–26, 393
Differential encoding, 354–355, 379, 587
Differential GPS, 648
Differential phase shift keying (DPSK), 378–380
  error probability, 587–589
Differential pulse code modulation (DPCM), 290–293
  analysis of, 292–293
  SNR improvement, 293
Digital audio broadcasting (DAB), 709–711
Digital broadcasting, 709
Digital carrier systems
  analog and digital carrier modulations, connections between, 376–377
  binary carrier modulations, 372–374
  demodulation, 377–380
  PSD of, 374–376
Digital communication systems, 326–329, 666
  advantages of, 270–271
  line coder, 327–328
  multiplexer, 328
  performance analysis of, 506
  regenerative repeater, 328–329
  source, 326
Digital data system (DDS), 289
Digital data transmission, principles of, 326
Digital multiplexing
  asynchronous channels and bit stuffing, 287–288
  plesiochronous digital hierarchy, 288–290
  signal format, 285–287
Digital signals, 4–8, 23–24
  at level 0 (DS0), 288
  at level 1 (DS1), 282, 289
  distortionless regenerative repeaters and nodes, viability of, 5–6
  noise immunity of, 4–5
  pulse code modulation, 7–8
Digit interleaving, 285
Diode bridge modulator, 148
Direct FM generation, 227–229
Direct sequence spread spectrum (DSSS)
  against broadband jammers, 630
  CDMA of, 630–637, 643–651
  against narrowband jammers, 629–630
  PN sequence generation, 625–626
  PSK, optimum detection of, 624–625
  resilient features of, 628–630
  single-user, 626–628
Discrete Fourier transform (DFT)
  FFT algorithm in, 123
  numerical computation of, 118–123
  points of discontinuity, 122
Discrete multitone (DMT) modulation, 702–706
  real-life applications of, 707–711
Discrete random variables, 408–410
Discrete time signal, 23
Dispersion, 98
Distortion, 92, 97–103
  in audio and video signals, 94–95
  linear, 97–99
  due to multipaths, 101–102
  nonlinear, 99–101, 234–239
Distortionless transmission, 92–95
Distributed coordinator function (DCF), 651
Doppler effect, 712
Double balanced modulator, 147
Double-sideband, suppressed-carrier (DSB-SC) modulation, 142, 143
  carrier acquisition in, 179–180
  nonlinear, 146–147
  signals, demodulation of, 151
  switching, 147–151
Double-sideband amplitude modulation, 142–151
Downconversion, 151
Duality
  property, 77–79
  time-frequency, 76–77
Duobinary signaling, 351–352
  detection of, 353–355
  modified, 353

Elastic store, 287
Element, 394

Energy
of modulated signals, 108–109
signal, 20–21, 22, 25, 103–111
scaler product and, 528–529
of sum of orthogonal signals, 34
Energy spectral density (ESD), 104–105
of input and output, 111
time autocorrelation function and, 109–111
Ensemble, of random process, 456, 459
statistics, 459
Entropy, 14, 737
Envelope detection, 152–154
condition for, 153–155
Envelope detector, 157–158
Equalizers
decision feedback (DFE), 689–692
feedforward (FFW), 690
finite length MMSE, 681–682
fractionally spaced (FSE), 684, 686–687, 688
linear, 689–690
time domain (TEQ), 706
zero-forcing, 359–362, 677–679
Equivalent optimum binary receivers, 516
Equivalent signal sets, 569–577
Ergodic wide-sense stationary processes, 463–465
Error correction coding, 14–15
Error-free communication, 745–748, 754–755, 768–769
Error probability of optimum receivers, 561–569
Error propagation, in DFE, 691–692
Error vector, 29
Essential bandwidth, 105–108, 122, 335
Event, 394
Exclusive OR (XOR), 805
Experiment, 393–394
Exponential Fourier series, 39–46
Fourier spectra, 41–42
negative frequency mean, 42–45
Parseval's theorem in, 46
Exponential Fourier spectra, 41–42
Exponential modulation, 204
Extended superframe (ESF), 284
Eye diagrams, 366–369
in PAM, 372

Fading channels, 103
flat, 714–715
frequency-selective, 713–714
conversion to flat fading channel, 715
False alarm, 580
False dismissal, 580
Faraday, Michael, 16
Fast Fourier transform (FFT)
algorithm in DFT computations, 123
Feedback decoding, 837
Feedforward (FFW) equalizers, 690
Filtering, 128
Filters
antialiasing, 261
bandpass matched, 521–522
best, 509
Butterworth, 97
first-order-hold, 258
ideal vs. practical, 95–97
matched, 509–512
optimum receiver, 508–512
reconstruction filters, realizability of, 258–259
VSB, 168–169
Finite length MMSE equalizers, 681–682
First-order-hold filter, 258
Flat fading channels, 714–715
Folding frequency, 122
Forward error correcting (FEC) codes, 802
Fourier integral
aperiodic signal representation by, 62–69
Fourier series
exponential, 39–46
Parseval's theorem in, 46
Fourier transform
direct, 65
discrete
numerical computation of, 118–123
duality property, 77–79
existence of, 67–68
frequency-shifting property, 77, 83–87
inverse, 65
linearity of, 68

Fourier transform (*Continued*)
physical appreciation of, 68–69
time differentiation property, 88–90
time-frequency duality, 76–77
time integration property, 88–90
time-scaling property, 79–81
time-shifting property, 81–82
Fractionally spaced equalizers (FSE)
MMSE design, 688
SIMO model, 684–686
ZF design, 686–687
Frame, 283
Framing bit, 283
Frequency converter (mixer), 150–151
Frequency convolution, 87
Frequency counters, 233
Frequency demodulators, practical, 232–233
Frequency division multiplexing (FDM), 13, 141, 172–173
Frequency hopping spread spectrum (FHSS), 614–617
applications of, 621–624
asynchronous, 619–620
performance, with multiple user access, 618–619
Frequency modulation (FM), 11, 202, 204
broadcasting system, 241–242
narrowband (NBFM), 210, 211
and phase modulation, relationship between, 205–206
signals, demodulation of, 231–234
tone, 214–217
waves, generating, 222–231
wideband (WBFM), 211–213
Frequency multipliers, 225
Frequency resolution, 122
Frequency shift keying (FSK), 208, 373, 522–523, 584–586
detection, 377–378
and FM modulation, connection between, 376
Gaussian, 622
M-ary FSK and orthogonal signaling, 380–382
Frequency-selective channels, 669, 776–780
Frequency-selective fading channel, 102, 713–714
conversion to flat fading channel, 715
Frequency-shifting property, 77, 83–87
Full-cosine roll-off characteristics, 349

Gaussian approximation, of nonorthogonal MAI, 633–634
Gaussian random process properties, 534–536
Gaussian random variable, 416–422
sum of, 444–446
Generalized angle, 203
Generalized Fourier series, 39
Generator matrix, 806, 818–819
Generator polynomial, 818–819
code, 814
Geometric interpretation, in signal space, 546–551
Geometrical signal space, 525–527
Global Positioning System (GPS), 647
differential, 648
operation, 647–648
spread spectrum in, 648–649
Gram-Schmidt orthogonalization, 530, 531, 877–879
Gray code, 553
Group (envelope) delay, 94n
GSM cellular phones, 675

Hadamard inequality, 787
Hamming bound, 804
Hamming codes, 804, 812–813
Hamming distance, 746, 807
Hard-decision decoding, 841
Hermitian symmetry, 66n
High-definition television (HDTV), 309–310
High-density bipolar (HDB) signaling, 342
High-quality LP vocoders, 304
High-speed packet access (HSPA), 647
Hilbert transform, 160–161
Hold-in (lock) range, 178
Homodyne modulators, 151
Huffman code, 740–745, 789–791

Ideal vs. practical filters, 95–97
IEEE 802.11, 621–622, 649–651
Image stations, 240
Impulse noise, 366
Independence vs. uncorrelatedness, 439
Independent events, 400
Independent random variables, 425
  variance of sum of, 434–435
Indirect FM generation, 225–227
Inductive logic, 407
Information
  commonsense measure of, 734–735
  engineering measure of, 735–737
  entropy of a source, 737–739
In-phase component, 491
Input transducer, 2
Instantaneous frequency, 203–204
Instantaneous velocity, 203
Integrated services digital network (ISDN), 371
Interleaved code, 839–840
Interleaving depth, 839
Interleaving effect, 241
International Mobile Telecommunications-2000 standard (IMT-2000), 646
International Telecommunications Union known, 172
Interpolation, 253, 255–258
  ideal, 254–255
Intersection, 395
Intersymbol interference (ISI), 98, 343–344, 669

Jitter, 364, 365
Joint distribution, 422–424
Joint event, 395
Joint source-channel coding, 15
Justification, 287

Karhunen-Löeve expansion, 531, 577

Lamarr, Hedy, 623
L-ary digital signal, 269
Levinson-Durbin algorithm, 302
Linear distortion, 3, 97–99
Linear equalizers, 689–690
Linear mean square estimation, 440–443
Linear prediction coding (LPC) vocoders, 301–304
  high-quality LP vocoders, 304
  LPC-10 vocoder, 303
  models, 302–304
  voice models and model-based vocoders, 301–302
Linear system
  signal transmission through, 90–95
    distortion, in audio and video signals, 94–95
    distortionless transmission, 92–95
    signal distortion, 92
Linear time-invariant (LTI) continuous time system, 90, 111
  frequency response of, 91–92
Line coder, 327–328
Line coding, 327, 329–343
  bipolar (pseudoternary or AMI) signaling, 339–341
  BNZS signaling, 343
  HDB signaling, 342
  on-off signaling, 337–339
  polar signaling, 334–336
  power spectral density, 330–334
    constructing dc null in, 336–337
  properties of, 329–330
Line spectral pairs (LSP), 303
Local carrier synchronization, 170–172
Logarithmic units, 280–281
Low-density parity check (LDPC) codes, 854–860
  bipartite (tanner) graph, 856–857
  bit-flipping decoding, 857
  decoding, sum-product algorithm for, 858–860
LPC-10 vocoder, 303

Manchester (split-phase) signaling, 337
Marginal densities, 423
Marginal probabilities, 411

M-ary
ASK and noncoherent detection, 380
bandwidth and power trade offs, 554, 567–568
binary polar signaling, 551–554
FSK and orthogonal signaling, 380–382
message, 4, 5
PAM signaling, 369–372, 383–384, 385
QAM analysis, 384–385, 554–560
trading power and bandwidth, 385–386
Matched filter, 509–512
MATLAB exercises
for AM modulation and demodulation, 185–188
basic signals and signal graphing, 46–54
coefficients D_n, numerical computation of, 52–54
for delta modulation, 317–319
digital communication systems, 715
performance analysis of, 589
for DSB-SC modulation and demodulation, 181–185
for error correcting code, 861
for eye diagrams, 386–387
for FM modulation and demodulation, 242–246
for Fourier transform computation, 123–130
for information theory, 789
lowpass signals, sampling and reconstruction of, 310–313
for PCM, 313–317
periodic signals and signal power, 48–49
for QAM modulation and demodulation, 191–195
signal correlation, 49–52
for spread spectrum technologies, 651
for SSB-SC modulation and demodulation, 188–191
Matrix product and properties, 880
Maximum a posterior (MAP) detection, BCJR algorithm for, 846
Maximum a posteriori probability (MAP) detector, 542
Maximum capacity power loading, 777–779
Maximum information rate, in digital communication, 262–263
Maximum length shift register sequences, 625
Maximum likelihood decoding, 809, 831–834
Maximum likelihood receiver, 579–580, 639–640
Maximum likelihood sequence estimation (MLSE), 673–676
complexity and practical implementations, 675–676
Mean, 427–428
of function of random variable, 428–429
of product of two functions, 430
of sum, 429
Measure-zero set, 39n
Memoryless source, 737
Message signal, 2
Minimax receiver, 580–581
Minimum energy signal set, 572–575
Minimum mean square error (MMSE), 363, 679
finite data design, 683–684
finite length MMSE equalizers, 681–682
FSE design, 688
and optimum delay, 681
receiver, 640–642
vs. ZF, 682–683
Minimum shift keying (MSK), 523–525
Minimum weight vector, 810
Mobile telephone switching office (MTSO), 644
Modem, 372
Modern telecommunications, historical review of, 15–19
Modified duobinary signaling, 353
Modulated signal, 83
phase spectrum, shifting, 84–85
power spectral density of, 118
Modulating signal, 83
Modulation
amplitude (AM), 11, 83–84, 140, 141–142
angle, 141–142, 202, 204
application of, 84–85
delta (DM), 141, 295–300

discrete multitone (DMT), 702
double-sideband, suppressed-carrier (DSB-SC), 142, 143
double-sideband amplitude, 142–151
frequency (FM), 11, 202, 204, 205–206, 210–217
index, 213
nonlinear, 202–209
phase (PM), 11, 204, 205–206, 210, 213–214
pulse amplitude, 141
pulse code (PCM), 7–8, 141, 229, 402–403, 774–776
pulse position (PPM), 141, 267, 268
pulse width (PWM), 141, 267, 268
single-sideband (SSB), 159–160
tone, 144
vestigial sideband (VSB), 167–170
Modulators
balanced, 147
coherent, 151
diode bridge, 148
double balanced, 147
frequency, 232–233
homodyne, 151
multiplier, 146
nonlinear, 146–147
single balanced, 147
switching, 147–151
synchronous, 151
Moments of random variables, 430
central, 430
Morse code, 4, 14
Moving Picture Experts Group (MPEG), 301, 304–309
MPSK signals, 557–560
Multiamplitude signaling, 551–554
Multicarrier communication system, 699
Multipath transmission, 101–102
Multiple-input–multiple-output (MIMO), 715
channel capacity, 781–783, 794–796
transmitter with channel knowledge, 785–789
transmitter without channel knowledge, 783–785
Multiplexer, 328
Multiplexing, 12–13
digital, 285–290
frequency division (FDM), 13, 141, 172–173
time division (TDM), 13, 267, 268
T1 time division, 281–282
Multiplication rule, for conditional probabilities, 404
Multiplier modulators, 146
Multitone signaling (MFSK), 564–566
noncoherent, 586–587
Multiuser detecion (MUD), 637–643
decision feedback receiver, 642–643
decorrelator receiver, 640
MMSE receiver, 640–642
optimum, 639–640
vs. power control, 647
Mutual information, 750, 762–764
channel capacity and, 791–794
Mutually exclusive (disjoint) events, 395

Narrowband angle modulation, 210–211
Narrowband frequency modulation (NBFM), 210, 211
generation, 222–223
Narrowband modulation, 210–211
Narrowband phase modulation (NBPM), 210
Natural binary code (NBC), 269
Near-far problem, 635–636
Near-far resistance, 637
Nodes, 5–6
Noise, 3, 14
Noisy channel coding theorem, 802
Noncoherent detection, 581–589
Nonideal practical sampling analysis, 263–267
Nonlinear distortion, 3, 99–101
Nonlinear DSB-SC modulators, 146–147
Nonlinear modulation, 202–209
Non-return-to-zero (NRZ) pulses, 328
Nonwhite channel noise, 577
Null event, 394
Nyquist criteria for zero ISI, 344–349
Nyquist interval, 253
Nyquist sampling rate, 253

Offset QPSK (OQPSK), 645
On-off keying (OOK), 373
On-off signaling, 327, 337–339, 518–519
Optimum delay, 681
Optimum filter, 483–486
Optimum linear precoder, 789
Optimum linear receiver analysis, 512–516
Optimum MUD receiver, 639
Optimum power distribution, 704
Optimum power loading
  in OFDM/DMT, 780
  water-pouring interpretation of, 779–780
Optimum preemphasis-deemphasis systems, 488–491
Optimum receiver
  filter, 508–512
  for white Gaussian noise channels, 536
Optimum threshold, 513–515
Orthogonal frequency division modulation (OFDM)
  channel equalization and, 669, 701–702
  channel noise, 698–700
  cyclic prefix redundancy in, 701
  principles of, 692–698
  real-life applications of, 707–711
  zero-padded, 700–701
Orthogonality
  complex signal space and, 32–33
  of exponential signal set, 874
  of trigonometric signal set, 873
Orthogonal signaling, 519–520, 562–564, 776
  bandwidth and power trade-offs of M-ary, 567–568
  energy of sum of, 34
Orthogonal signal sets, 36–39
Orthogonal signal space, 38–39
Orthogonal vectors, 30, 526–527
Orthogonal vector space, 36–38
Orthonormal basis set, 38, 529–530
Orthonormal vectors, 527
Outcomes, 394
Output transducer, 3
Overhead bits, 285
Paley-Wiener criterion, 96, 259
Parity check digits, 806
Parity check matrix, 808
Parseval's theorem, 39, 103–104
  in Fourier series, 46
Partial reflection coefficients (PARCOR), 303
Partial response signaling, 350–351
Perfect code, 804
Periodic signals, 24–25
Phase coherent (in phase) lock, 178
Phase delay, 94n
Phase-locked loop (PLL), 172, 173–181, 233–234
  basic operation, 174–175
  first-order loop analysis, 177–178
  hold-in (lock) range, 178
  phase coherent (in phase) lock, 178
  pull-in (capture) range, 178
  small-error analysis, 175–177
Phase modulation (PM), 11, 204, 213–214
  and frequency modulation, relationship between, 205–206
  narrowband (NBPM), 210
Phase shift keying (PSK), 208, 373
  binary (BPSK), 520–521
  detection, 378
  differential (DPSK), 378–380
  differentially coherent, 587–589
  and QAM modulation, connection between, 376–377
Phase shift method, 163
Piconet, 621
Plain-old-telephone-service (POTS), 707
Plesiochronous digital hierarchy, 288–290
Polar signaling, 334–336, 516–518
Power
  of angle modulated wave, 206–209
  control, 636–637
    vs. MUD, 647
  loading, 704
  signal, 25, 111–118
Power spectral density (PSD), 111–112, 330–334, 465, 486
  of amplitude shift keying, 374–376
  constructing dc null in, 336–337

of digital carrier modulation, 374–376
of frequency shift keying, 375–376
input and output, 117–118
interpretation of, 114
of modulated signals, 118
of phase shift keying, 375–376
Prediction coefficients, 292
Probability, 393–408
axiomatic theory of, 407–408
Probability density function (PDF), 414, 458
Product code, 840
Progressive taxation, 274–278
Pseudonoise (PN) sequence generation, 625–626
Public switched telephone network (PSTN), 707
Pull-in (capture) range, 178
Pulse amplitude modulation (PAM), 141, 267, 268, 551–554
M-ary baseband signaling, for higher order rate, 369–372
Pulse code modulation (PCM), 7–8, 141, 229, 267, 268–281, 774–776
channel noise, mean square value of, 432–434
differential (DPCM), 290–293
encoder, 278
quantization error, mean square value of, 431–432
repeater error probability, 402–403
in T1 carrier systems, 281–284
total mean square error in, 435
transmission bandwidth and output SNR, 278–281
Pulse detection error, 271
Pulse generation, 355
Pulse position modulation (PPM), 141, 267, 268
Pulse shaping, 336–337, 343–355
controlled ISI or partial response signaling, 350–351
differential encoding, 354–355
duobinary pulse, 351–352, 353–355
intersymbol interferences, 343–344
Nyquist's criteria for zero ISI, 344–349
in PAM, 371
zero-ISI, duobinary, and modified duobinary, pulse relationship between, 352–353
Pulse stuffing, 287–288
Pulse width modulation (PWM), 141, 267, 268

QCELP (Qualcomm code-excited linear prediction) vocoder, 645
Quadrature, 491
nonuniqueness of representation, 495–496
Quadrature amplitude modulation (QAM), 159, 165–167
M-ary, 384–385, 554–560
and PSK, connection between, 376–377
Quadrature multiplexing, 165, 167
Quantization, 7, 269, 271–273
error, 271
noise, 272
nonuniform, 274–278

Radiation, 11–12
Raised cosine characteristics, 349
Random interleaver, 840
Randomness, 14
Random processes, 393, 456
autocorrelation function of, 459–461
bandpass, 491–499
baseband analog systems, performance analysis of, 486–488
basic functions determination for, 530–531
binary, 471–473
characterization of, 458–459
ergodic wide-sense stationary processes, 463–465
Gaussian properties, 534–536
independent process, 479
multiple, 479–480
optimum filter, 483–486
optimum preemphasis-deemphasis systems, 488–491
orthogonal process, 479
PAM pulse train, 473–478
power of, 468

Random processes (*Continued*)
power spectral density, 465, 486
stationary, 461
sum of, 481–483
transmission of, through linear systems, 480–483
uncorrelated process, 479
wide-sense stationary process, 461–463
Random variables
conditional probabilities of, 410–412
continuous, 409, 413–416
discrete, 408–410
Gaussian, 416–422
independent, 425, 434–435
sum of, 443–446
Ratio detector, 233
Rayleigh density, 425–427
Receiver, 3
Rectifier detector, 156–157
Recursive systematic convolutional (RSC) code, 831, 850–851
Redundancy, 13, 14, 742
Reed-Solomon codes, 822
Regenerative repeater, 5–6, 328–329, 358–359
Relative frequency, 395–398
Relative likelihood, 842
Resource exchange, 10–11
Return-to-zero (RZ) pulses, 328
Rice density, 499
Ring modulator, 148, 149
Robbed-bit signaling, 284
Roll-off factor, 348
Root-raised-cosine pulse, 674
Row-column (RC) constraint, 857

Sample function, of random process, 456
Sample space, 394
Sampling theorem, 6, 251–253
applications of, 267–268
Scalar product and signal energy, 528–529
Scatter diagram, 437, 599
Scrambling, 355–358
Selective-filtering method, 163–164
Sequential decoding, 834–837
Series bridge diode modulator, 148, 149
Shannon's equation, 10, 773
Shunt bridge diode modulator, 148, 149
Sideband, 155–156
Sigma-delta modulation, 299–300
Signal distortion, 92
in audio and video signals, 94–95
Signal energy, 20–21, 22, 103–111
scalar product and, 528–529
Signal power, 9–10, 111–118
time autocorrelation of, 113–117
Signal reconstruction, 253–258
aliasing error (spectral folding), 259–261
antialiasing filter, 261
filters, realizability of, 258–259
ideal, 254–255
practical, 255–258
Signals
bandpass, 85–87
energy, 20–21, 22
power, 21
size of, 20–22
vs. vectors, 28–34
Signals, classification of, 22–26
analog, 23–24
aperiodic, 24–25
continuous time, 23
deterministic, 25–26
digital, 23–24
discrete time, 23
energy, 25
periodic, 24–25
power, 25
probabilistic, 25
Signals, correlation of, 34–36
correlation functions, 35–36
Signal space
analysis of optimum detection, 525–530
and basis signals, 527–530
Signal-squaring method, 179–180
Signal-to-noise ratio (SNR), 9, 511
in DPCM, 293
exchange with bandwidth, 875
in PCM, 278–281

transmitter power loading for maximizing receiver, 703–704
Simplex signal set, 575–577
Simplified signal space and decision procedure, 541–545
Sinc function, 70–72
SINCGARS, 622–623
Single balanced modulator, 147
Single-input–multiple-output (SIMO) model, 684–685
Single-sideband, suppressed-carrier (SSB-SC) modulation
carrier acquisition in, 180–181
Single-sideband (SSB) modulation, 159–160
signals with carrier (SSB+C), 165
systems, 163–164
phase shift method, 163
selective-filtering method, 163–164
Weaver's method, 164
time domain representation of, 161–163
Sinusoidal carrier, 141
Sinusoidal signal, in noise, 498–499
Slope detection, 232, 233
Slope overload, 298
Slotted frequency hopping, 619
SNR improvement, 483
Soft decoding, 841–843
Soft-output Viterbi algorithm (SOVA), 844–845
Source, 2
Source coding, 13
randomness and, 14
redundancy and, 14
Source encoding, 739–745
Spectral density, 69
energy (ESD), 104–105, 109–111
power (PSD), 111–112, 114, 117–118, 336–337, 330–334, 374–376
Spectrum, 69
direct sequence spread (DSSS), 624–637, 643–651
frequency hopping spread (FHSS), 614–624
phase, 84–85
spread spectrum, in GPS, 614, 648–649
vestigial, 348
Standard deviation, 430
State transition diagram, 829–830
Stationary random process, 461
Stationary white noise, 531
Successive interference cancellation (SIC) receiver, 642
Sum-product algorithm, for LDPC decoding, 858–860
Superframe, 284
extended (ESF), 284
Superheterodyne receiver, 239–240
Superposition theorem, 68
Switching modulators, for DSB-SC, 147–151
Synchronization, 283–284
Synchronous detection, 144
Synchronous modulators, 151
Syndrome, 809, 837
Systematic code, 806
Systemic cyclic codes, 816–818
Systems, 20
BIBO linear, 90
binary, 516–520
cellular, 644–645
communication, 1–4
digital carrier, 372–380
digital communication, 326–329, 506, 666, 715
FM broadcasting, 241–242
global positioning (GPS), 647–649
linear, 90–95
multicarrier communication, 699
T1 carrier, 281–284

Thermal noise, 480–481
3G cellular services, 646–647
Threshold detection, 420–422
binary, 507–508
Time autocorrelation function, 109–111
of power signals, 113–117
Time convolution, 87
Time differentiation property, 88–90
Time division multiple-access (TDMA) systems, 287

Time division multiplexing (TDM), 13, 267, 268
Time domain equalizer (TEQ), 706
Time-frequency duality, 76–77
Time integration property, 88–90
Time scaling
  signal duration, reciprocity of, 80–81
  significance of, 79–80
Time shifting, 81–82
  linear phase, physical explanation of, 81–82
Time-varying channels, 712–715
Timing extraction, 363–364
Timing jitter, 364, 365
Toeplitz matrix, 361
T1 carrier systems, 281–284
  synchronizing and signaling, 283–284
  time division multiplexing, 281–282
Tone frequency modulation
  spectral analysis of, 214–217
Tone modulation, 144
T1 multiplexer, 289
T1 time division multiplexing, 281–282
Total probability theorem, 404–406
Transmission, 11–12, 90–95
  digital data, 326
  distortionless, 92–95
  multipath, 101–102
Transmitter, 2, 703–704
  MIMO, 783–789
Trellis diagram, 830, 837–838
T-spaced equalization (TSE), 671, 676–684
  based on MMSE, 679
  zero-forcing equalizer, 677–679
Turbo codes, 846–854

Uncorrelated variables, 438
  mean square of sum of, 439
Undetermined multipliers, 759
Union, 394–395
Unitary matrix, 883
Unit impulse function, 26
  multiplication of, 26–27
  sampling property of, 27
Unit impulse signal, 26–28
Unit rectangular function, 70
Unit step function, 27–28
Unit triangular function, 70
Upconversion, 151

Variance, 430
  of sum of independent random variables, 434–435
Vector decomposition of white noise random processes, 530–536
Vectors
  vs. signals, 28–34
    complex signal space and orthogonality, 32–33
    component of vector along another vector, 28–30
    decomposition of signal and signal components, 30–32
Vector space, 28
Vestigial sideband (VSB) modulation, 167–170
  in broadcast television, 169–170
  filter, 168–169
  signals with carrier (VSB+C), 168–169
Vestigial spectrum, 348
Video compression, 300, 304–309
Viterbi algorithm, 831–834
Vocoders, 300
  linear prediction coding, 301–304
Voltage-controlled oscillator (VCO), 174

Weaver's method of SSB generation, 164
Weiner-Hopf filter. *See* Optimum filter
White channel noise, 531–532
White Gaussian noise, 515–516, 533–534, 544
White noise, 531–532
  additive white Gaussian noise (AWGN), 536
  Gaussian, 515–516, 533–534, 536, 544
  geometrical representation of, 531–533
  stationary, 531

vector decomposition of random processes, 530–536
Wideband frequency modulation (WBFM), 211–213
Wide-sense stationary process, 461–463
Wiener-Khintchine theorem, 468
Wireless multipath channels, linear distortions of, 666–669
Word interleaving, 285

Zero-crossing detectors, 233
Zero-forcing (ZF) equalizer, 359–362, 677–679
FSE design, 686–687
Zero padding, 120, 700