



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Digital Signal Processing Report

By Matthew Taylor

TABLE OF CONTENTS

Abstract	3
Introduction.....	3
Theory	3
Simulation	4
Discussion	5
Conclusions.....	6
References.....	6
Appendix.....	7
BPSK Model with no hamming code	7
BPSK Model with hamming code	7
QPSK Model with no hamming code	7
QPSK Model with hamming code	7
.M file.....	8

ABSTRACT

The aim of this report is to investigate the performance of Binary Phase Key Shifting and Quadrative Phase Key Shifting in the presence of naturally occurring environmental noise. Furthermore, to investigate how Hamming Codes can be deployed to reduce the number of errors.

INTRODUCTION

Firstly, my report covers the theory necessary to investigate the stated aim. Secondly, I discussed the models I designed for my simulation, going into detail of their operation, and how I used MATLAB to manage the workflow and data management. Finally, I considered the real-world application of the presented information, and the limitations of the scope. This report is an important building block in my further understanding in wireless communication.

THEORY

An important part of sending data is the journey it will take. It is essential to consider how the data will be affected by the external conditions of the environment as it is traveling to the receiver. Through Simulink many types of environmental factors can be controlled and simulated, and their effects measured. For my analysis, all the data went through the “Additive White Gaussian Noise” (AWGN) block which mimics the noise from many random processes occurring in nature. (wirelesspi, n.d.)

For wireless transmission, there will be modulation of the signal. There are multiple techniques for modulation, like amplitude and frequency modulation, where the amplitude or frequency of the carrier signal is modified to convey the data in the original signal. (tutorialspoint, n.d.)

A reason for modulation is that it reduces the required size of the antenna, as a high frequency will be chosen for the carrier signal. The equation below shows the relationship between the antenna length and the frequency of the carrier signal. (electronics-notes, n.d.)

$$\text{Antenna Length} = \frac{\text{Speed of Light}}{\text{frequency}}$$

For modulation of digital signals there are techniques that involve phase shifting the carrier signal. In Binary Phase Key Shifting (BPSK) the carrier signal is phase shifted by 180 degrees to send a binary bit, in Quadrative Phase Key Shifting (QPSK), the carrier signal is phase shifted by 90 degrees which allows for 2 binary bits to be sent in one operation. These are the two techniques I looked at for my models, but the phase can be shifted by smaller number of degrees, allowing for more data to be including in each operation. (allaboutcircuits, n.d.)

Parity is a very simple method of error detection, but importantly it can not error correct, nor detect an even number of errors. The method is to add a ‘parity’ bit at the end of a byte, that indicates if the data prior should have been even or odd. If an odd number of bits change value, including the parity bit, the section of data won’t compute correctly on receival, and the data will need to be sent again. This method is appropriate for data paths with limited interference otherwise lots of data will have to be dumped and re-transmitted.

A Hamming code is an expansion on the idea of parity. Richard Hamming designed a code that included additional bits of data per section, that would facilitate the receiver to be

able to know if there was an error but also correct the error. Where parity only included one extra bit of data per byte, the Hamming code adds three bits for error detection and correction, for 4 bits of data. The Hamming code has been iterated on further, allowing for more errors per section to be correctly found and corrected, while reducing the number of additional data that needs to be transmitted. The newer methods can be used when longer sections of data are used. Hamming developed a nomenclature to describe an error detection system, using two numbers, one to describe the total number of bits in a block, and another to describe the number of data bits in the block, for example in his 7,4 code, there a 7 bits of data of which 4 bits are data and 3 bits to error correct. (Macleod, 1993)

The issue with this Hamming code that for every 4 bits of data to be sent there are an additional 3 bits to send, this results in a large overhead. The rate of a code describes the ratio between the useful data and the total data sent. Hamming's 7,4 code has a rate of 0.571 so only 57% of the bits being sent is the data. Hamming's later codes increase the rate while still being able to error correct. Hamming 15,11 and 31,26 codes have a respective 0.839 and 0.905 rate.

SIMULATION

I used MATLAB and Simulink to design a BPSK and QPSK system. I transmitted binary data through an AWGN channel using a range of signal to noise ratios to get a control, so that I could go on to compare the effectiveness of different Hamming Codes at error correction.

Due to the large number of models and the even larger data set it generates, I had to set up a robust workflow, that managed all the variables and recorded the results into a single table. I did this by using a MATLAB live script to call the Simulink systems in turn, using the variables that the script placed in the workspace. I used a different Simulink file for the BPSK and QPSK system, and one of each with Hamming included and excluded, for a total of four Simulink files. I configured the Hamming encoder and decoder with variables that were written to the workspace by the script as required. I controlled the signal to noise ratio with a 'for' loop, iterating though the chosen ratios, placing the variable in the workspace.

A way of speeding up the simulation is to increase the data in each frame, but the frame size is constrained when using the Hamming encoder as Simulink required the date to be in a matrix of a specific size. To prevent an error, I set the frame size to the size of the message length, which allowed for me to use larger frames as I simulated the later Hamming Codes.

While making the control data, I ran over a million data bits through the system at each of the signal to noise ratios I was testing, which was 1 to 15. On my table of the simulation results, the first column is the signal to noise ratio, and the second and sixth is the percentage of data that was incorrect after passing through the noisy channel.

The BPSK system outperformed the QPSK, in terms of error rate, as expected from the discussion of the theory.

Once I had collected the control data, I wanted to simulate the BPSK and QPSK systems with various Hamming codes to investigate their effects on the accuracy of the data. I was going to use three different codes, 7/4, 15/11, 31/26 (codeword length/message length), as part of the models. When I ran the code, the BPSK model and the BPSK models ran without error and returned promising results, showing the expected improvement.

I ran the simulations of the BPSK systems with the Hamming encoder and decoder included, in succession, at all the signal to noise ratios used in the control data. There is a marked improvement in the accuracy of the data transmitted from any Hamming encoded system compared to the control, as seen in the table. Comparing the BPSK systems with Hamming encoding, when the codeword length is increased, the accuracy of the transmitted data improved, which is what was expected after discussing the theory.

The initial results from the QPSK model were clearly incorrect, with an error rate of 50%, there was a clear sign that the data was being mis-handled. In an attempt to design a model that correctly modelled QPSK modulation with a Hamming code, I used function blocks to control the sizes of the matrices as they pass through the model. With this new model I had what initially looked like reasonable results, so I added the Simulink file to my MATLAB workflow. When I ran the script and was able to see all the data together, I recognized that there is likely still an issue with the QPSK data that went through a Hamming Code. As can be seen in the data below, the model performed worse than when a Hamming Code was not included in the model.

index	BPSK no hamming	BPSK hamming 7/4	BPSK hamming 15/1	BPSK hamming 31/26	QPSK no hamming	QPSK hamming 7/4
1	5.624%	1.589%	1.313%	1.322%	10.949%	11.675%
2	3.748%	0.667%	0.453%	0.415%	7.398%	8.227%
3	2.296%	0.232%	0.124%	0.105%	4.534%	5.228%
4	1.246%	0.057%	0.025%	0.012%	2.486%	3.329%
5	0.596%	0.011%	0.003%	0.001%	1.180%	1.369%
6	0.239%	0.001%	0.000%	0.000%	0.476%	0.440%
7	0.080%	0.000%	0.000%	0.000%	0.157%	0.160%
8	0.019%	0.000%	0.000%	0.000%	0.041%	0.000%
9	0.004%	0.000%	0.000%	0.000%	0.007%	0.000%
10	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%
11	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%
12	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%

DISCUSSION

Following the completion of the report, I have identified three areas of discussion. Firstly, the limitation of the Simulink software, where the QPSK block does not support the output matrix of the Hamming Encoder. My initial workaround involved including additional ‘function blocks’ to modify the size of the matrices, but as discussed in my results, this did not resolve the issue. Fortunately, as I included control data, I was able to identify that the model was not suitable, so I did not go on to draw any conclusions from that part of the data.

The scope of this report has some real-world limitations. The scope only covered Additive White Gaussian Noise, it did not include Rayleigh fading and the Doppler effect. These are important effects to consider in real-world applications, as Rayleigh fading describes the impact of using wireless communication in urban environments. It describes the delay of the reflected data from surfaces such as, glass buildings, water bodies, troposphere, and ionosphere (Proakis, 1995). The Doppler effect describes the change in a wave’s

waveform due to either the receiver or transmitter moving, effectively squeezing or stretching the waveform, changing its frequency.

A consideration in the use of any form of error detection or error correction, is that the importance of a low error rate needs to be weighed up against the need for more efficient use of the bandwidth available. Hamming codes are often used in EEC memory hardware which is deployed in every computer and embedded processor.

CONCLUSIONS

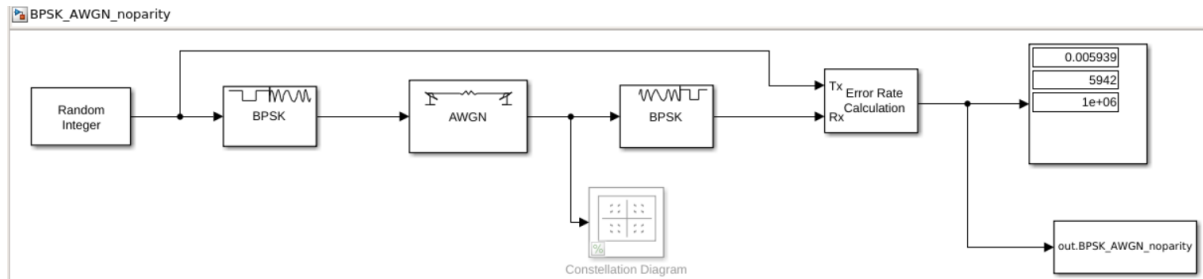
In conclusion, I have researched Hamming codes which are still prevalent today in wireless communication. The Hamming Codes have a strong limitation their efficiency in bandwidth usage, so their deployment must be weighed against other requirements. Because Hamming's Code can error correct as opposed to only detection, it will continue to be prevalent in industry, and a valuable topic for me to build my knowledge of.

REFERENCES

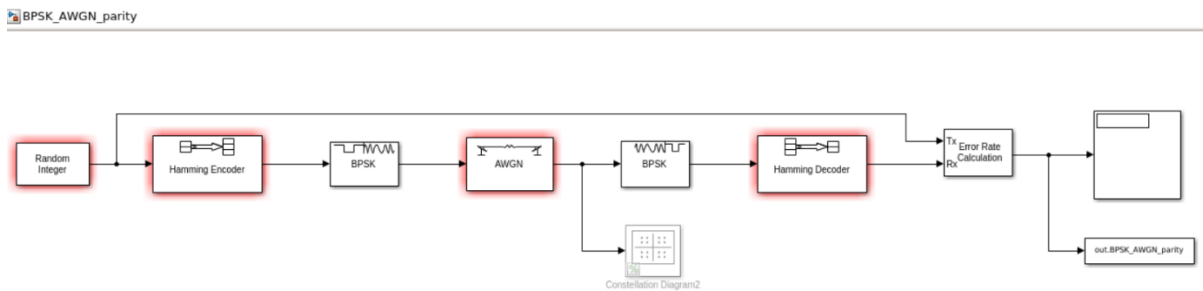
- allaboutcircuits, n.d. *Digital Phase Modulation: BPSK, QPSK, DQPSK*. [Online] Available at: <https://www.allaboutcircuits.com/textbook/radio-frequency-analysis-design/radio-frequency-modulation/digital-phase-modulation-bpsk-qpsk-dqpsk/> [Accessed 27 5 2021].
- electronics-notes, n.d. *Dipole Antenna Length: calculation & formula*. [Online] Available at: <https://www.electronics-notes.com/articles/antennas-propagation/dipole-antenna/length-calculations-equation-formula.php> [Accessed 27 5 2021].
- Macleod, D. M. D., 1993. *Telecommunications Engineer's Reference Book*. First ed. Cambridge: Butterworth-Heinemann,.
- Proakis, J. G., 1995. *Digital Communications*. Third ed. Singapore: McGraw–Hill Book Co..
- tutorialspoint, n.d. *Principles of Communication - Modulation*. [Online] Available at: https://www.tutorialspoint.com/principles_of_communication/principles_of_communication_modulation.htm [Accessed 27 5 2021].
- wirelesspi, n.d. *Additive White Gaussian Noise (AWGN)*. [Online] Available at: <https://wirelesspi.com/additive-white-gaussian-noise-awgn/> [Accessed 27 5 2021].

APPENDIX

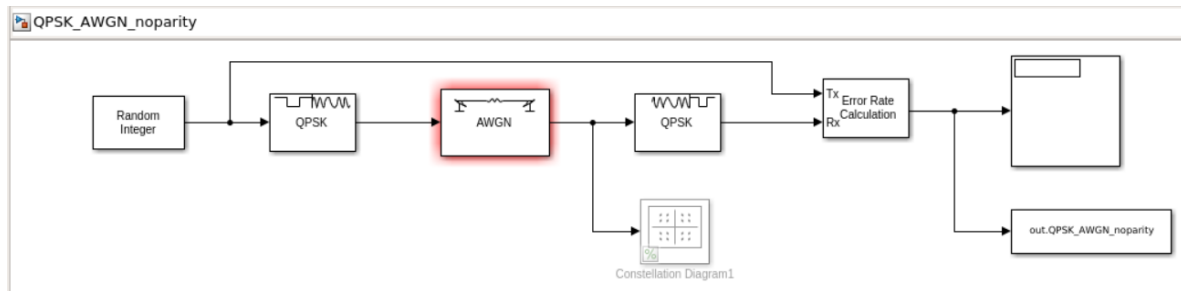
BPSK MODEL WITH NO HAMMING CODE



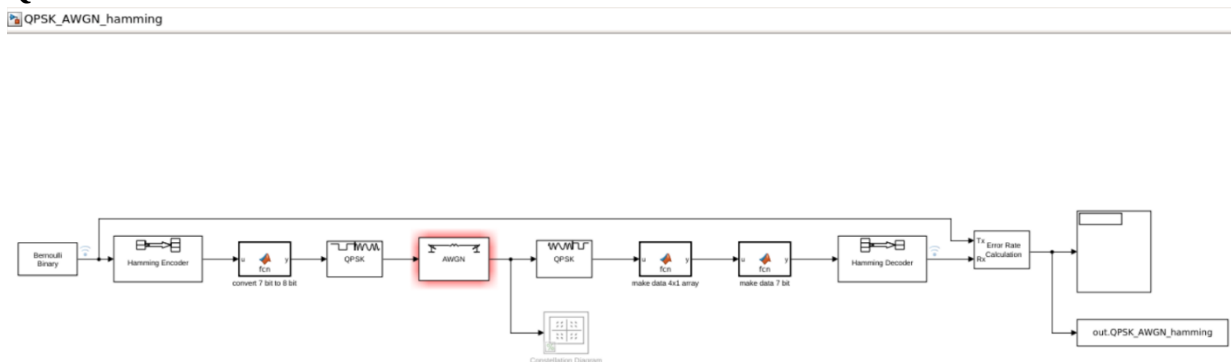
BPSK MODEL WITH HAMMING CODE



QPSK MODEL WITH NO HAMMING CODE



QPSK MODEL WITH HAMMING CODE



.M FILE

```
1. %{
2. Version: 1.0
3. author: https://github.com/matt-t251
4. CC: BY-SA
5. Language: matlab
6.
7. Function:
8.     investigate the proformance of BPSK and QPSK in he pressence of
      AWGN.
9.     investigate the effects of modifying the hamming code.
10.
11. Associated files:
12.     QPSKparity.slx (redundant)
13.     BPSK_AWGN_noparity.slx
14.     BPSK_AWGN_parity.slx
15.     QPSK_AWGN_noparity.slx
16.     QPSK_AWGN_parity.slx (redundant)
17.     QPSK_AWGN_hamming.slx
18.
19. Output:
20.     Final is table: Nb/no - BPQK(noparity) - BPQK(parity) -
      QPQK(noparity)
21.     - QPQK(parity)
22. %}
23.
24. % setup
25.
26. clear
27.
28. % Bitrate = 10000000;
29. % Bit_period = 1/Bitrate;
30. % SymbolPeriod = 2 * Bit_period;
31. % delayVector = 1.0e-6*[0 .2 .4 .8];
32. % pathVector = [0 -9 -12 -15];
33. EbNorange = 1:15;
34. EbNo = 5;
35. t = 0.01;
36. codeword_length = 7;
37. messege_length =4;
38.
39. % Main
40.
41.
42.
43. for i = EbNorange
44.     EbNo = i;
45.     final(i,1) = i;
46. end
47.
48. for i = EbNorange
49.     EbNo = i;
50.     sim('BPSK_AWGN_noparity', t);
```

```

51.     final(i,2) = ans.BPSK_AWGN_noparity(end,1);
52.end
53.
54.for i = EbNorange
55.     EbNo = i;
56.     sim('QPSK_AWGN_noparity', t);
57.     final(i,3) = ans.QPSK_AWGN_noparity(end,1);
58.end
59.
60.codeword_length = 7;
61.messege_length = 4;
62.for i = EbNorange
63.     EbNo = i;
64.     sim('BPSK_AWGN_parity', t);
65.     final(i,4) = ans.BPSK_AWGN_parity(end,1);
66.end
67.
68.codeword_length = 15;
69.messege_length = 11;
70.
71.for i = EbNorange
72.     EbNo = i;
73.     sim('BPSK_AWGN_parity', t);
74.     final(i,5) = ans.BPSK_AWGN_parity(end,1);
75.end
76.
77.codeword_length = 31;
78.messege_length = 26;
79.for i = EbNorange
80.     EbNo = i;
81.     sim('BPSK_AWGN_parity', t);
82.     final(i,6) = ans.BPSK_AWGN_parity(end,1);
83.end
84.
85.for i = EbNorange
86.     EbNo = i;
87.     sim('QPSK_AWGN_hamming', t);
88.     final(i,2) = ans.QPSK_AWGN_hamming(end,1);
89.end
90.
91. %{
92.codeword_length = 7;
93.messege_length = 4;
94.for i = EbNorange
95.     EbNo = i;
96.     sim('QPSK_AWGN_parity', t);
97.     final(i,7) = ans.QPSK_AWGN_parity(end,1)
98.end
99.
100.     codeword_length = 15;
101.     messege_length = 11;
102.     for i = EbNorange
103.         EbNo = i;

```

```
104.         sim('QPSK_AWGN_parity', t);
105.         final(i,8) = ans.QPSK_AWGN_parity(end,1)
106.     end
107.
108.     codeword_length = 31;
109.     messege_length = 26;
110.     for i = EbNorange
111.         EbNo = i;
112.         sim('QPSK_AWGN_parity', t);
113.         final(i,9) = ans.QPSK_AWGN_parity(end,1)
114.     end
115.     %}
116.
117.     filename = 'testdata.xlsx';
118.     writematrix(final,filename,'Sheet',1,'Range','A1')
```