

Electrical Principals Theory

Wave analysis using MATLAB.

Richard Woodhams

Gower College Swansea (UWTSD)

<u>Contents</u>	<u>Page</u>
Title page	1
Contents	2
Introduction	3
Task 1	4 - 5
Task 2	6 - 9
Task 3	10 - 11
Task 4	12 - 15
References	16

Introduction

This assignment explores the principles, applications, and computational methods of the Fourier transform (FT) as a critical analytical tool in science and engineering. It is structured into four key tasks, integrating theoretical understanding and practical implementation.

Task 1 focuses on identifying and analysing various waveforms, comparing Fourier transform with Fourier series, and understanding harmonics within power systems and waveform patterns. In task 2, I will create different types of waveforms using MATLAB, concentrating on rectangular and sawtooth waves. I will then analyse the affect that adding more harmonics has to the shape of the waveform. Task 3 looks at the Fourier transform of aperiodic signals, using manual methods, as well as MATLAB, to be able to compare each approach whilst visualising the frequency domain representation of a signal. Task 4 involves analysing an infinite duration signal using fast Fourier transform (FFT) and inverse fast Fourier transform (IFFT). I will recover the signal from the frequency domain the time domain, applying both all and partial frequency components and compare them to the original signal and compare results.

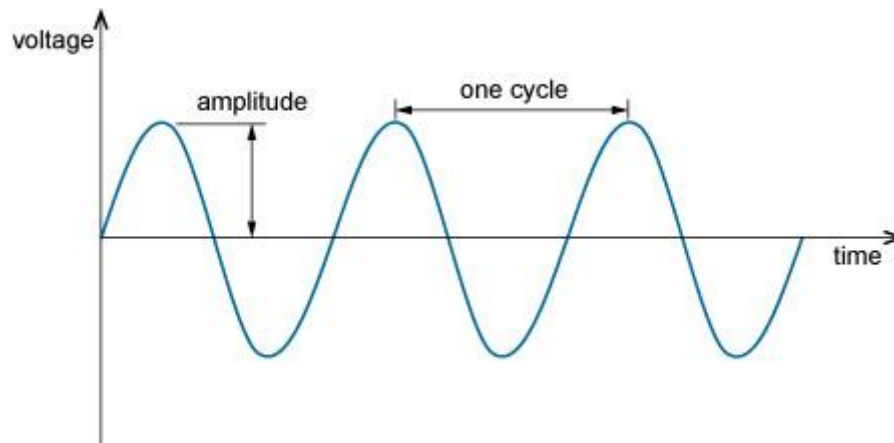
By combining analytical theory, MATLAB coding, and critical evaluation, this assignment will show my understanding of Fourier analysis and its real-world applications.

Task 1

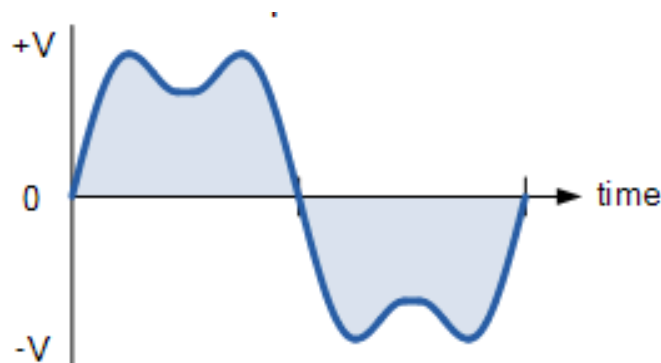
1. Types of Waveforms: Explain how to recognise simple periodic, complex periodic, impulse, and noise waveforms by their shapes.

The four types of waveforms mentioned above can, ultimately, be identified, visually, by two key characteristics – the pattern and the shape of the waveform.

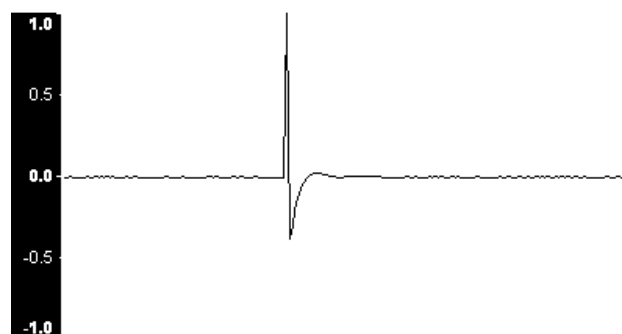
Simple periodic waveform - A basic sine or cosine wave that repeats at regular intervals. It has a single frequency and is predictable.



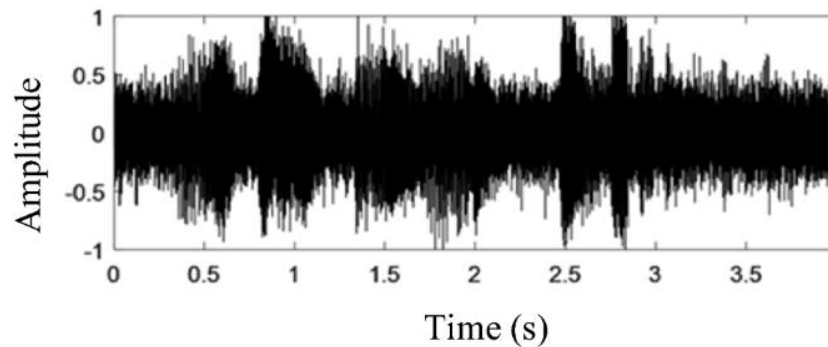
Complex periodic waveform - A waveform formed by adding together multiple sine waves (harmonics), each with a different frequency. Despite being complex, they still repeat periodically.



Impulse waveform - A short-duration, high-amplitude pulse followed by zero or near-zero amplitude. Impulses are not periodic so will occur as single events or repeat at irregular intervals.



Noise waveform - A non-periodic, random signal with no clear structure or repetition. It is made up of a wide range of frequencies. They can be used to show real world sounds like speech and ambient sounds.



2. Fourier Transform vs. Fourier Series: Describe what the Fourier Transform is and how it differs from the Fourier series.

Fourier transform is used to represent non-periodic signals, like impulse signals, as a continuous sum and to work out the possible frequencies that make it up. Whereas Fourier series is used to represent periodic signals as a sum of its sin and cosine signals. Fourier series can be used to find the exact frequencies that make up a signal.

3. Using Fourier Series in Electrical Analysis: Explain why the Fourier series is useful for analysing electrical waveforms and breaking down periodic signals.

Fourier Series is useful for analysing electrical waveforms as it allows complex periodic waveforms to be deconstructed into their harmonic components. This simplifies analysis by enabling engineers to evaluate circuit behaviour in the frequency domain rather than the time domain. It also aids in the design and understanding of filters, communication systems, and power electronics.

4. Harmonics in Power Systems: Describe how harmonics relate to the Fourier series and why they are important in power systems for managing signal distortion.

In power systems, harmonics can lead to signal distortion, increased heating, and reduced efficiency in electrical devices. The Fourier Series provides a way to quantify these harmonics and helps engineers design systems that counteract their impact.

5. Harmonic Patterns in Waveforms: Explain the mathematical patterns that determine harmonic amplitudes in triangular, sawtooth, and square waveforms.

Different waveforms exhibit characteristic harmonic patterns. A square wave, for instance, contains only odd harmonics whose amplitudes decrease proportionally to $1/n$. A sawtooth waveform contains both even and odd harmonics, with amplitudes also decreasing by $1/n$. A triangular waveform includes only odd harmonics, and their amplitudes fall off more rapidly at a rate of $1/n^2$.

6. Limits of the FFT (Fast Fourier Transform): Discuss the limitations of using the FFT, such as resolution limits, noise sensitivity, and the need for periodic signals.

Although the Fast Fourier Transform (FFT) is a fast and powerful tool for analysing signals, it does have some important limitations. First, it assumes that the signal being analysed repeats itself perfectly within the window of time used for the calculation. If this is not true, the results may not accurately reflect the actual frequencies present. Second, how clearly we can distinguish between different frequencies (known as frequency resolution) depends on how often the signal is sampled and how long the signal is observed. Finally, FFT is sensitive to any noise in the signal, and if the signal is not shaped or "windowed" properly before applying the FFT, it can cause something called spectral leakage, where energy from one frequency spills into others and distorts the results.

Task 2

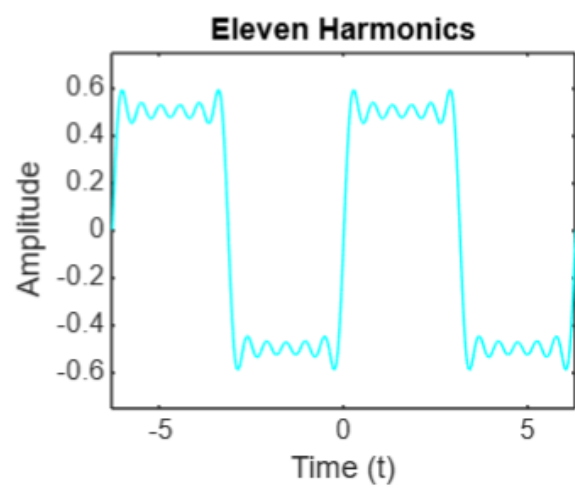
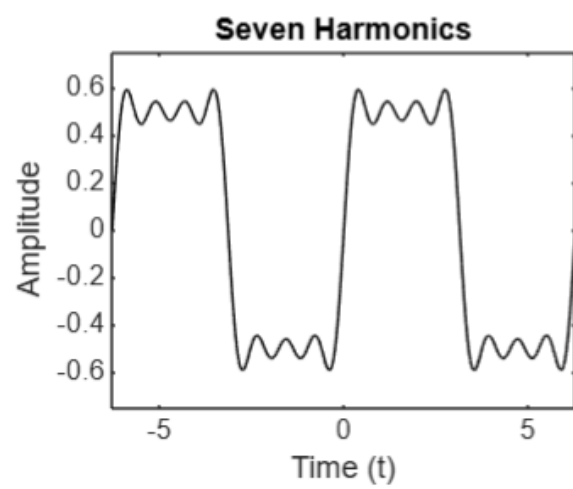
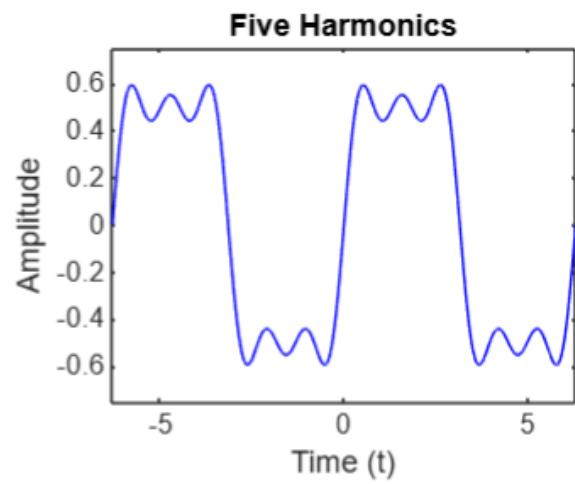
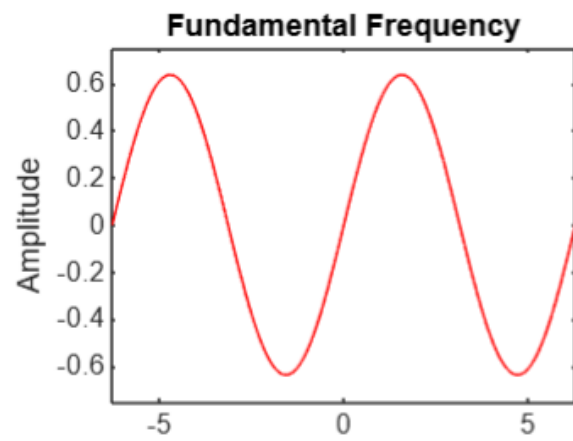
Rectangular waveform generation.

I have written the code below to generate a rectangular waveform. I began by writing the formula for the fundamental frequency and then coding each odd harmonic formula thereafter. I then wrote code to add the desired harmonics together and plot them on a two-by-two subplot, each with their own title.

```
1 %RW EPT Assignment 1 Task 2
2 %Setting parameters
3 t=-2*pi:0.0001:2*pi;
4 w=1;%2*pi*f=1
5 Y1=(2/pi)*sin(w*t);%first harmonic
6 Y2=(2/(3*pi))*sin(3*w*t);%third harmonic
7 Y3=(2/(5*pi))*sin(5*w*t);%fifth harmonic
8 Y4=(2/(7*pi))*sin(7*w*t);%seventh harmonic
9 Y5=(2/(9*pi))*sin(9*w*t);%ninth harmonic
10 Y6=(2/(11*pi))*sin(11*w*t);%eleventh harmonic
11 y1=Y1;%Fundamental Frequency
12 y2=Y1+Y2+Y3;%Five Harmonics
13 y3=Y1+Y2+Y3+Y4;%Seven Harmonics
14 y4=Y1+Y2+Y3+Y4+Y5+Y6;%Eleven Harmonics
15 subplot(2,2,1);
16 plot(t,y1,'r');
17 ylim([-0.75,0.75]);
18 xlim([-2*pi,2*pi]);
19 title("Fundamental Frequency");
20 ylabel("Amplitude");
21 subplot(2,2,2);
22 plot(t,y2,'b');
23 ylim([-0.75,0.75]);
24 xlim([-2*pi,2*pi]);
25 title("Five Harmonics");
26 ylabel("Amplitude");
27 xlabel("Time (t)");

28 subplot(2,2,3);
29 plot(t,y3,'k');
30 ylim([-0.75,0.75]);
31 xlim([-2*pi,2*pi]);
32 title("Seven Harmonics");
33 ylabel("Amplitude");
34 xlabel("Time (t)");
35 subplot(2,2,4);
36 plot(t,y4,'c');
37 ylim([-0.75,0.75]);
38 xlim([-2*pi,2*pi]);
39 title("Eleven Harmonics");
40 ylabel("Amplitude");
41 xlabel("Time (t)");
42 clear
```

Below are the resulting waveforms.

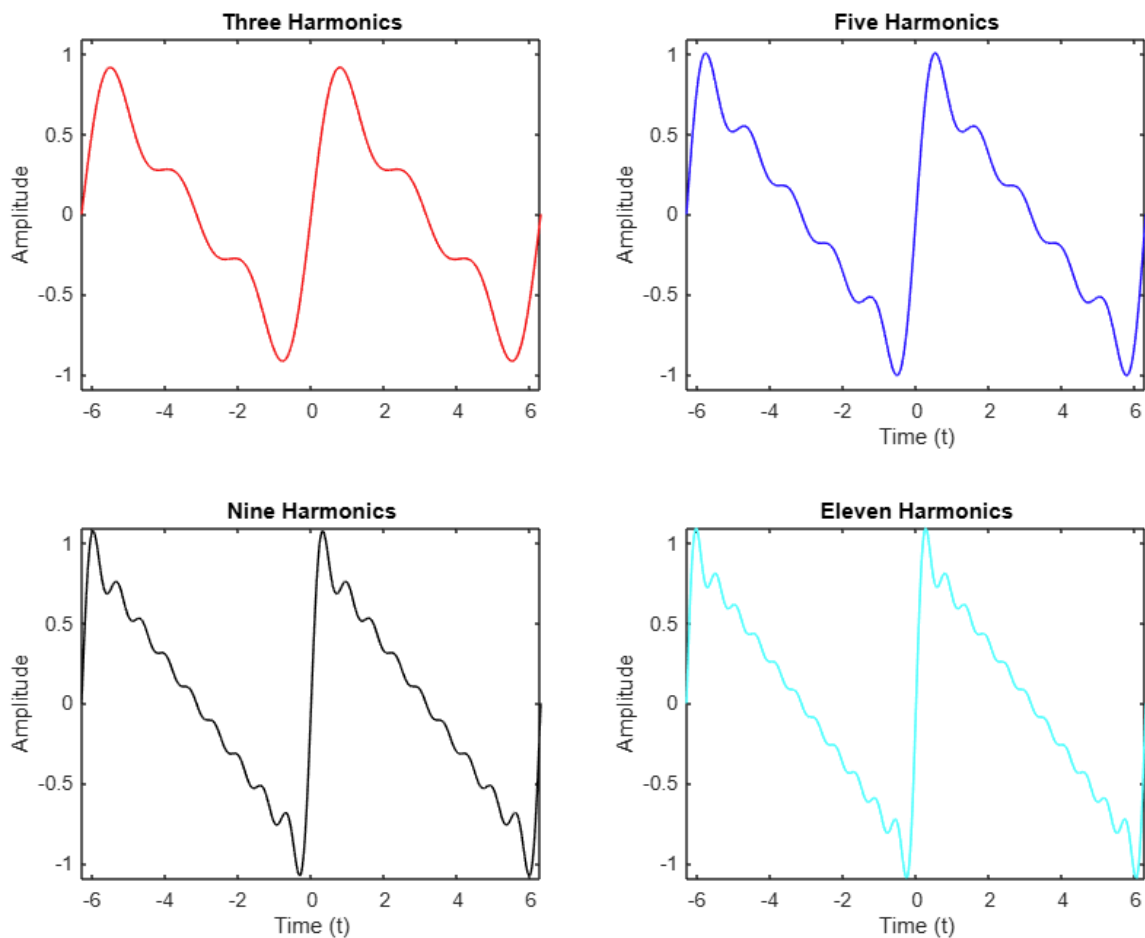


Sawtooth waveform generation

Since sawtooth waveforms contain, both, odd and even harmonics, I have modified my code by adding lines for the missing even harmonics. The full code can be seen below. This also meant that I needed to modify the formula that I used to plot the waveforms. These can be seen in lines sixteen to nineteen.

```
1 %RW EPT Assignment 1 Task 2
2 %Setting parameters
3 t=-2*pi:0.0001:2*pi;
4 w=1;%2*pi*f=1
5 Y1=(2/pi)*sin(w*t);%first harmonic
6 Y2=(2/(2*pi))*sin(2*w*t);%second harmonic
7 Y3=(2/(3*pi))*sin(3*w*t);%third harmonic
8 Y4=(2/(4*pi))*sin(4*w*t);%fourth harmonic
9 Y5=(2/(5*pi))*sin(5*w*t);%fifth harmonic
10 Y6=(2/(6*pi))*sin(6*w*t);%sixth harmonic
11 Y7=(2/(7*pi))*sin(7*w*t);%seventh harmonic
12 Y8=(2/(8*pi))*sin(8*w*t);%eighth harmonic
13 Y9=(2/(9*pi))*sin(9*w*t);%ninth harmonic
14 Y10=(2/(10*pi))*sin(10*w*t);%tenth harmonic
15 Y11=(2/(11*pi))*sin(11*w*t);%eleventh harmonic
16 y1=Y1+Y2+Y3;%Three Harmonics
17 y2=Y1+Y2+Y3+Y4+Y5;%Five Harmonics
18 y3=Y1+Y2+Y3+Y4+Y5+Y6+Y7+Y8+Y9;%Nine Harmonics
19 y4=Y1+Y2+Y3+Y4+Y5+Y6+Y7+Y8+Y9+Y10+Y11;%Eleven Harmonics
20 subplot(2,2,1);
21 plot(t,y1,'r');
22 ylim([-1.1,1.1]);
23 xlim([-2*pi,2*pi]);
24 title("Three Harmonics");
25 ylabel("Amplitude");
26 subplot(2,2,2);
27 plot(t,y2,'b');
28 ylim([-1.1,1.1]);
29 xlim([-2*pi,2*pi]);
30 title("Five Harmonics");
31 ylabel("Amplitude");
32 xlabel("Time (t)");
33 subplot(2,2,3);
34 plot(t,y3,'k');
35 ylim([-1.1,1.1]);
36 xlim([-2*pi,2*pi]);
37 title("Nine Harmonics");
38 ylabel("Amplitude");
39 xlabel("Time (t)");
40 subplot(2,2,4);
41 plot(t,y4,'c');
42 ylim([-1.1,1.1]);
43 xlim([-2*pi,2*pi]);
44 title("Eleven Harmonics");
45 ylabel("Amplitude");
46 xlabel("Time (t)");
47 clear
```

Below are the resulting waveforms.



Analysis and discussion

In the first section, looking at rectangular waveforms, as I add more odd harmonics, the waveform begins to move more and more from a pure sine wave to the desired rectangular wave. The waveform takes on a new shape and has as many peaks and troughs as the number of harmonics that have been added. Every time more odd harmonics are added, the amplitude of those peaks and troughs becomes gradually lesser. Visually, this is shown as a shallower ripple each time. If we were to add a very large number of odd harmonics, the ripple at the top and bottom of the waveform would look almost flat and, therefore, resemble a rectangular waveform.

The second section is much of the same story. As more harmonics are added, the waveform begins to look more and more like a perfect sawtooth waveform. The ripple between each peak and trough flattens until, when there are a very large number of odd and even harmonics added, the ripple is indistinguishable from a flat line without zooming all the way in.

Higher numbers of harmonics are important for shaping rectangular and sawtooth waveforms because they create a more precise waveform.

Task 3

a) Show that the aperiodic signal $x(t)$ has a Fourier Transform:

$$X(f) = 4\text{sinc}(4\pi f)$$

where $x(t)$ is defined as:

$$x(t) = \begin{cases} 1 & -2 \leq t \leq 2 \\ 0 & \text{otherwise} \end{cases}$$

The Fourier Transform is given by:

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft} dt$$

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft} dt$$

$$X(f) = \int_{-2}^{+2} e^{-j2\pi ft} dt$$

$$\text{Since: } \int e^{kt} = \frac{e^{kt}}{k}$$

$$X(f) = \left[\frac{e^{-j2\pi ft}}{-j2\pi f} \right]_{-2}^{+2} = \frac{1}{-j2\pi f} \cdot [e^{-j2\pi ft}]_{-2}^{+2}$$

$$X(f) = \frac{1}{-j2\pi f} \cdot (e^{-j2\pi f(2)} - e^{-j2\pi f(-2)}) = \frac{1}{-j2\pi f} \cdot (e^{-j4\pi f(2)} - e^{j4\pi f})$$

Since; $e^{-j\theta} = \cos \theta - j \sin \theta$ and $e^{j\theta} = \cos \theta + j \sin \theta$, therefore, $e^{-j\theta} - e^{j\theta} = -2j \sin \theta$

[Betterexplained.com, 2019]

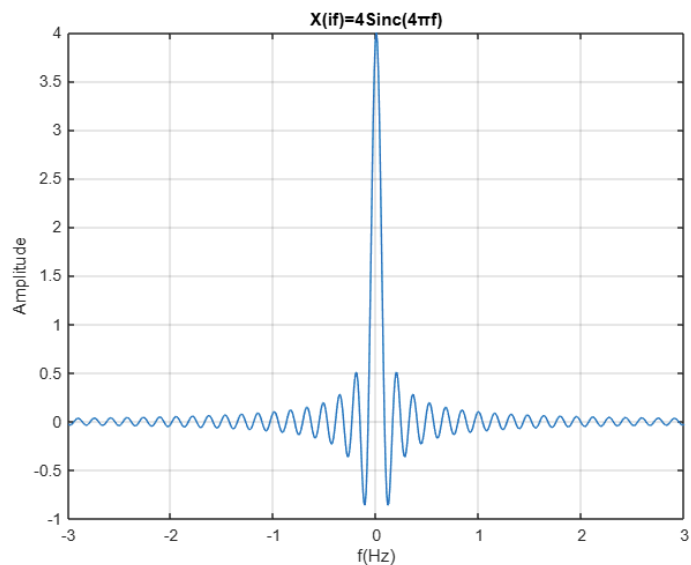
$$X(f) = \frac{1}{-j2\pi f} \cdot (-2j \sin(4\pi f)) = \frac{\sin(4\pi f)}{\pi f} = \frac{4 \sin(4\pi f)}{4\pi f}$$

$$\text{Since: } \frac{\sin(x)}{x} = \text{sinc}(x)$$

$$X(f) = 4\text{sinc}(4\pi f)$$

Below is my MatLab code and plot of $X(f) = 4\text{sinc}(4\pi f)$.

```
1 f=-3:0.01:3;
2 X=4*sinc(4*pi*f);
3 plot(f,X);
4 xlabel("f(Hz)");
5 ylabel("Amplitude");
6 grid on;
7 title("X(f)=4Sinc(4*pi*f)");
```



b)
 $y(t) = e^{-7|t|}$

$$Y(if) = \int_{-\infty}^{\infty} y(t) \cdot e^{-j2\pi ft} dt$$

Since; $y(t) = e^{-7|t|} = \begin{cases} e^{-7t}, & \text{if } t \geq 0 \\ e^{7t}, & \text{if } t < 0 \end{cases}$, the integration will need to be broken down into two parts.

$$Y(if) = \int_{-\infty}^0 e^{7t} \cdot e^{-j2\pi ft} dt + \int_0^{\infty} e^{-7t} \cdot e^{-j2\pi ft} dt$$

For part 1 (p1),

$$p1 = \int_{-\infty}^0 e^{7t} \cdot e^{-j2\pi ft} dt = \int_{-\infty}^0 e^{(7-j2\pi f)t} dt$$

$$p1 = \left[\frac{e^{(7-j2\pi f)t}}{7-j2\pi f} \right]_{-\infty}^0$$

Since the real part of; $7 - j2\pi f > 0$, when $t \rightarrow -\infty$, $e^{(7-j2\pi f)t} \rightarrow 0$.

$$\text{Since; } e^{(7-j2\pi f)0} = e^0 = 1$$

$$p1 = \left[e^{(7-j2\pi f)t} \right]_{-\infty}^0 = \frac{1}{7-j2\pi f} - \frac{0}{7-j2\pi f} = \frac{1}{7-j2\pi f}$$

For part 2 (p2),

$$p2 = \int_0^{\infty} e^{-7t} \cdot e^{-j2\pi ft} dt = \int_0^{\infty} e^{(-7-j2\pi f)t} dt$$

$$p2 = \left[\frac{e^{(-7-j2\pi f)t}}{-7-j2\pi f} \right]_0^{\infty}$$

Since the real part of; $-7 - j2\pi f < 0$, when $t \rightarrow \infty$, $e^{(-7-j2\pi f)t} \rightarrow 0$.

$$\text{Since; } e^{(-7-j2\pi f)0} = e^0 = 1$$

$$p2 = \left[e^{(-7-j2\pi f)t} \right]_{-\infty}^0 = \frac{0}{-7-j2\pi f} - \frac{1}{-7-j2\pi f} = \frac{1}{7+j2\pi f}$$

$$Y(if) = \int_{-\infty}^0 e^{7t} \cdot e^{-j2\pi ft} dt + \int_0^{\infty} e^{-7t} \cdot e^{-j2\pi ft} dt = p1 + p2$$

$$Y(if) = \frac{1}{7-j2\pi f} + \frac{1}{7+j2\pi f} = \frac{(1)(7+j2\pi f)}{(7-j2\pi f)(7+j2\pi f)} + \frac{(1)(7-j2\pi f)}{(7+j2\pi f)(7-j2\pi f)} = \frac{(7+j2\pi f)+(7-j2\pi f)}{(7-j2\pi f)(7+j2\pi f)} = \frac{14}{(7-j2\pi f)(7+j2\pi f)}$$

$$Y(if) = \frac{14}{(7-j2\pi f)(7+j2\pi f)} = \frac{14}{49-(j)^2(2\pi f)^2} = \frac{14}{49-(-1)(2\pi f)^2} = \frac{14}{49+4\pi^2 f^2}$$

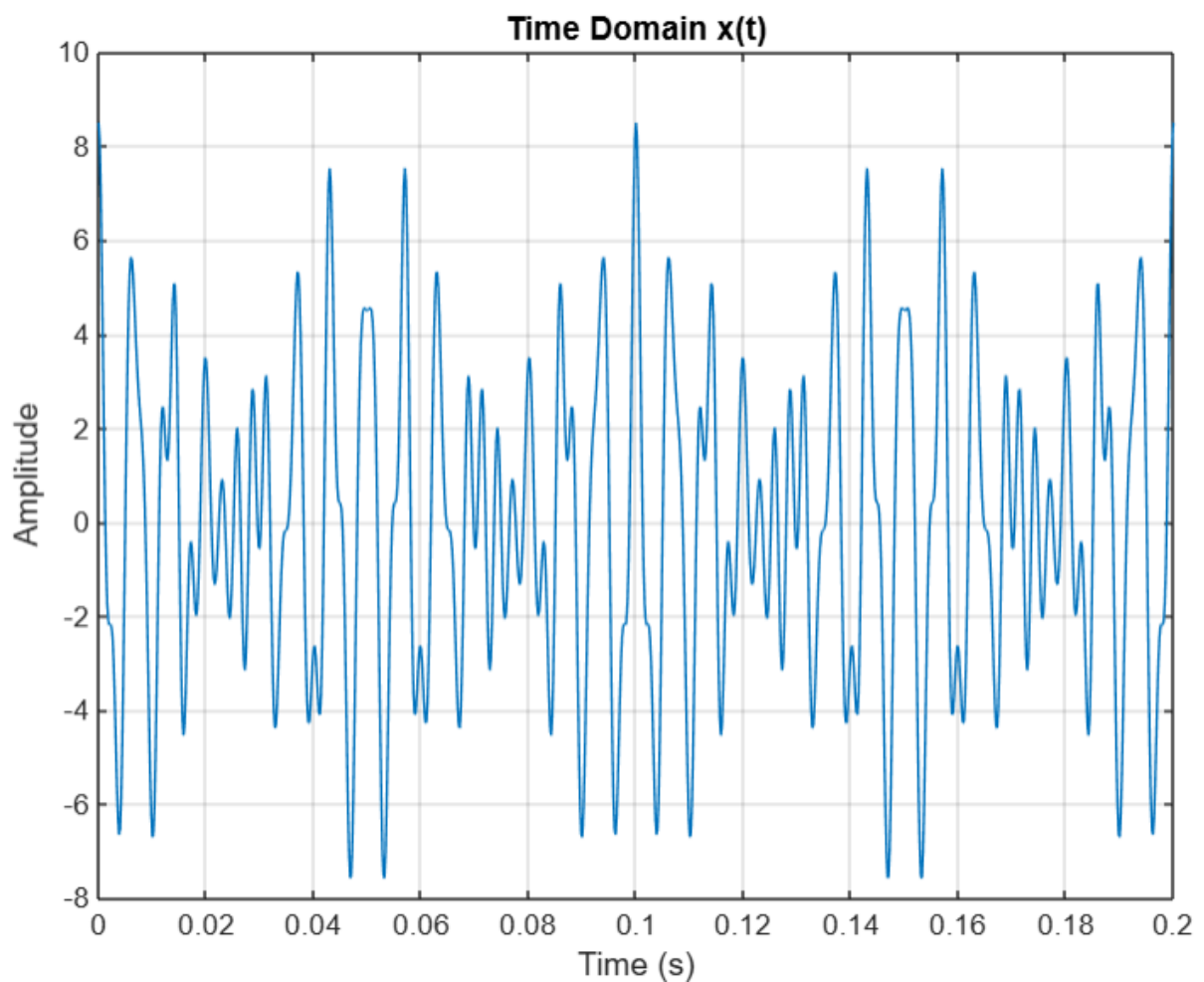
$$Y(if) = \frac{14}{49+4\pi^2 f^2}$$

Task 4

a) $x(t) = 3\cos(2\pi 140t) + 3\cos(2\pi 160t) + 2\cos(2\pi 350t) + 0.5\cos(2\pi 420t)$

The code below plots the signal above with 1000 samples within a 0.2 second timespan. This equates to a sample every 0.0002s.

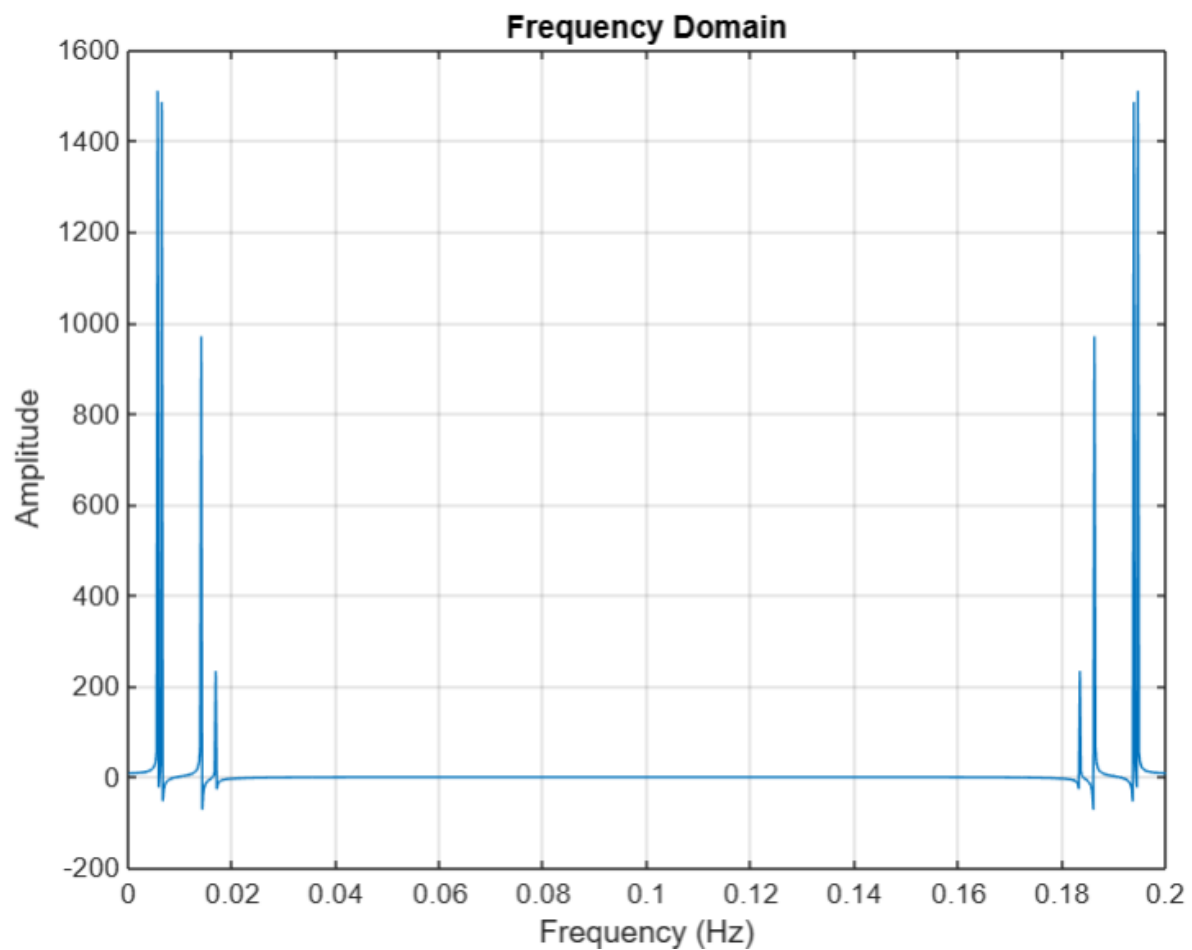
```
1 Ts = 0.0002; % Sampling period (s)
2 N = 1000; % Length of the signal
3 t = 0:Ts:N*Ts; % Time increments
4 x = 3*cos(2*pi*140*t) + 3*cos(2*pi*160*t) + 2*cos(2*pi*350*t) + 0.5*cos(2*pi*420*t);
5 plot(t, x);
6 title('Time Domain x(t)');
7 xlabel('Time (s)');
8 ylabel('Amplitude');
9 grid on;
```



b) Fast fourier transform of above signal.

The code below plots the fast fourier transform of the signal above with 1000 samples within a 0.2 second timespan. This equates to a sample every 0.0002s.

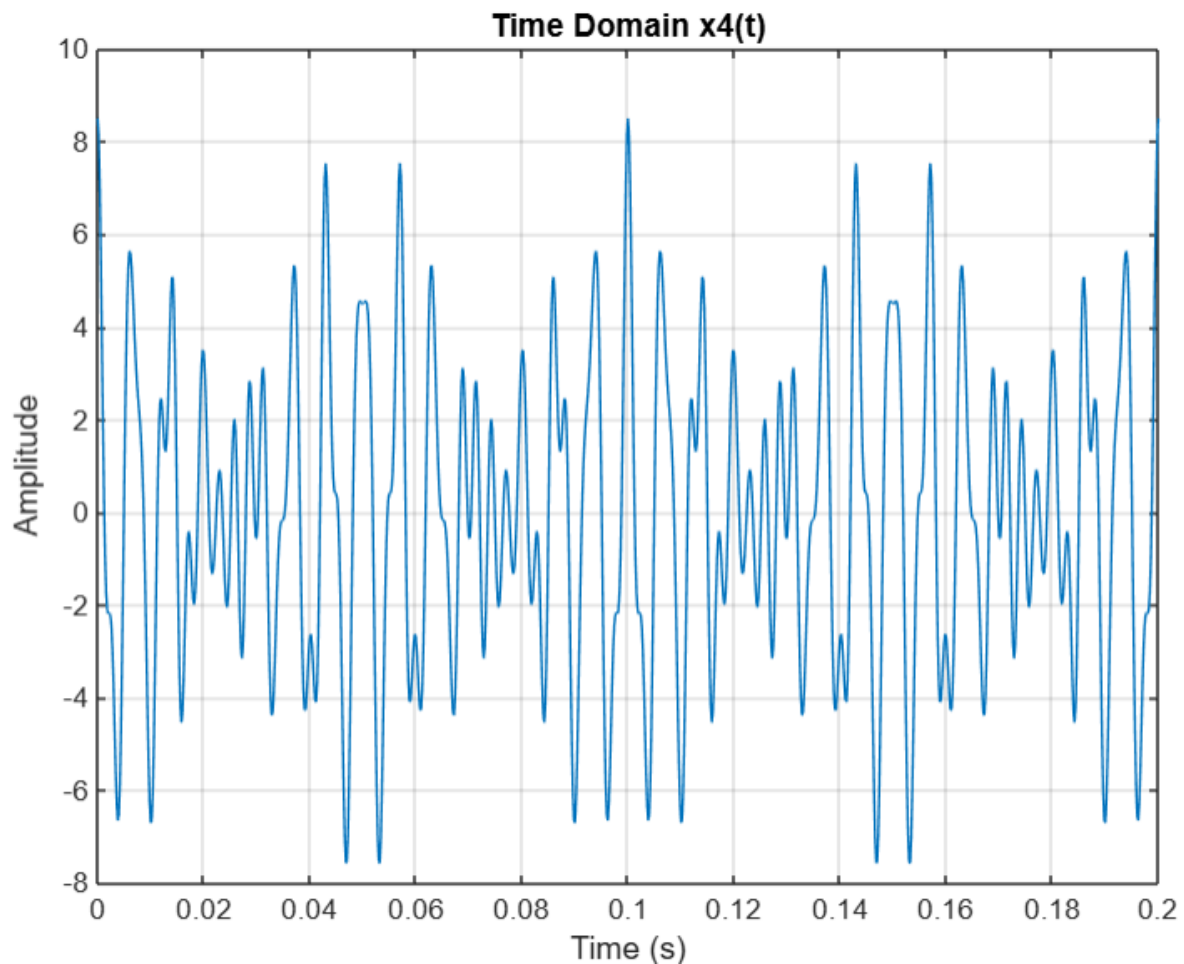
```
1  clc;
2  clear;
3  Ts = 0.0002;           % Sampling period (s)
4  N = 1000;              % Length of the signal
5  t = 0:Ts:N*Ts;         % Time increments
6  x = 3*cos(2*pi*140*t) + 3*cos(2*pi*160*t) + 2*cos(2*pi*350*t) + 0.5*cos(2*pi*420*t);
7  x4=fft(x); %x4 is the fourier transform of x
8  plot(t, x4);
9  title('Frequency Domain');
10 xlabel('Frequency (Hz)');
11 ylabel('Amplitude');
12 grid on;
```



c) Signal recovery using IFFT – full frequency components.

The code below uses Matlab to plot the inverse fast fourier transform of the fast fourier transform signal above with 1000 samples within a 0.2 second timespan. This equates to a sample every 0.0002s. this should then look the same as the original signal from part a.

```
1  clc;
2  clear;
3  Ts = 0.0002;           % Sampling period (s)
4  N = 1000;              % Length of the signal
5  t = 0:Ts:N*Ts;         % Time increments
6  x = 3*cos(2*pi*140*t) + 3*cos(2*pi*160*t) + 2*cos(2*pi*350*t) + 0.5*cos(2*pi*420*t);
7  x4=fft(x); %x4 is the fourier transform of x
8  ix=ifft(x4);
9  plot(t, ix);
10 title('Time Domain x4(t)');
11 xlabel('Time (s)');
12 ylabel('Amplitude');
13 grid on;
```

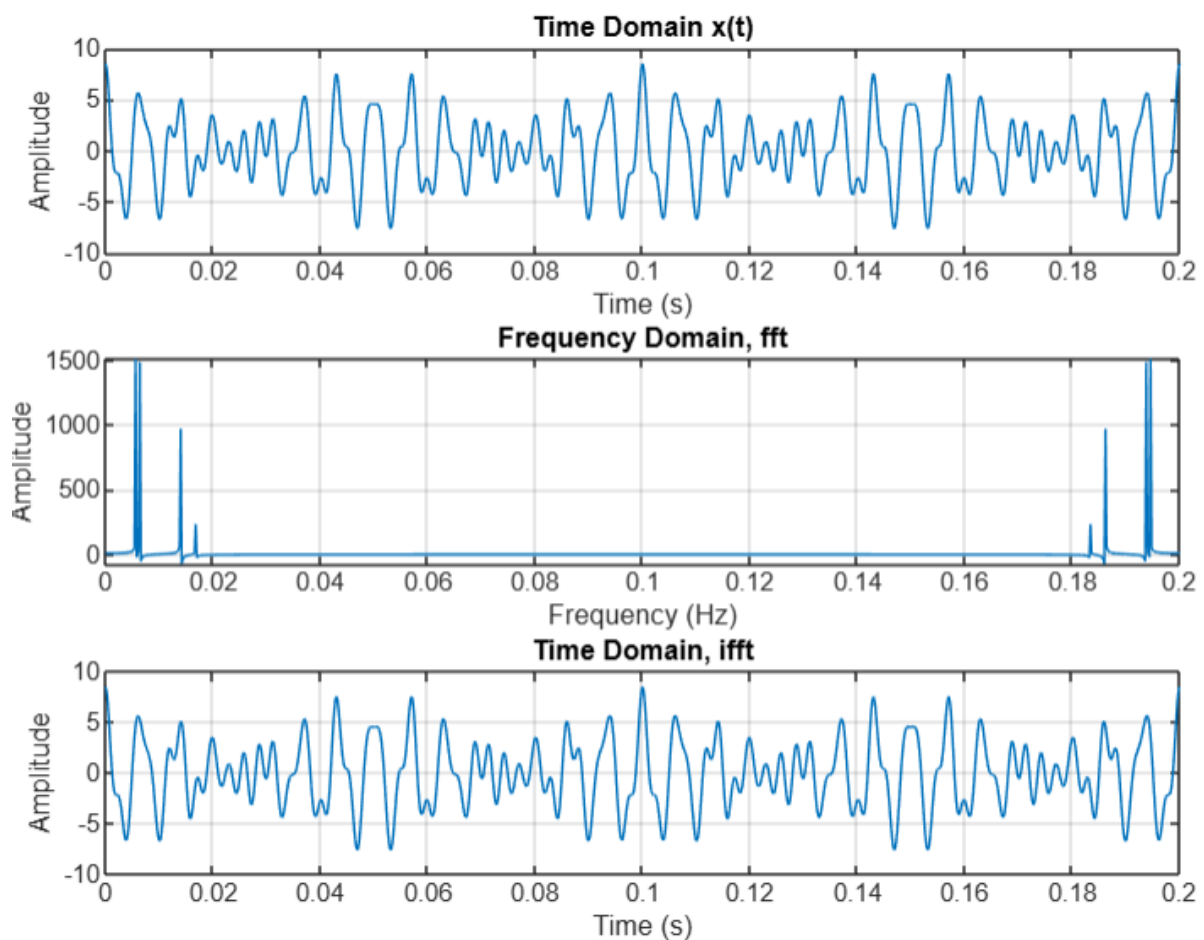


When I combine the codes above and create subplots, it is easier to see that the ifft plot is the same as the original meaning that the signal has been successfully recovered to the time domain with all frequency components.

```

1  clc;
2  clear;
3  Ts = 0.0002;           % Sampling period (s)
4  N = 1000;              % Length of the signal
5  t = 0:Ts:N*Ts;         % Time increments
6  x = 3*cos(2*pi*140*t) + 3*cos(2*pi*160*t) + 2*cos(2*pi*350*t) + 0.5*cos(2*pi*420*t);
7  x4=fft(x); %x4 is the fourier tranform of x
8  ix=ifft(x4);
9  subplot(3,1,1);
10 plot(t, x);
11 title('Time Domain x(t)');
12 xlabel('Time (s)');
13 ylabel('Amplitude');
14 grid on;
15 subplot(3,1,2);
16 plot(t, x4);
17 title('Frequency Domain, fft');
18 xlabel('Frequency (Hz)');
19 ylabel('Amplitude');
20 grid on;
21 subplot(3,1,3);
22 plot(t, ix);
23 title('Time Domain, ifft');
24 xlabel('Time (s)');
25 ylabel('Amplitude');
26 grid on;

```



References

Betterexplained.com, 2019 – Intuitive Understanding of Euler’s Formula. [online]

Available at: betterexplained.com/eulers-formula/.

Accessed: 18 April 2025

Lboro.ac.uk, 2008 – HELM Workbook 24 – Fourier Transforms. [online]

Available at: lboro.ac.uk/FourierTransforms.pdf

Accessed 29 April 2025

Mathcentre.ac.uk, 2009 – The complex conjugate. [online]

Available at: mathcentre.ac.uk/sigma-complex6-2009-1.pdf

Accessed: 18 April 2025

Poole I, N.D – Electronic Waveforms & Signals: sine, square, triangle, ramp, etc. [online]

Available at: electronics-notes.com/electronic-electrical-waveforms

Accessed: 30 April 2025

Smith, S.W, 1999 – The scientist and engineer’s guide to digital signal processing. San Diego, Calif. California Technical Pub.