RCNDE Signal Processing Coursework

Written by

Euan Alexander Foster

August 2019

Email: **e.foster@strath.ac.uk**
Tel: **07539 450593**

Contents

1 EXERCISE 1 1
1.1 Trend Removal 1
1.2 Coherent Noise 2
1.3 Signal Correlation 3
1.4 Signal Correlation 4
1.5 Signal Correlation 5
1.6 Signal Correlation 6
1.7 Use of the FFT 8
1.8 Use of the FFT 10
1.9 Use of the FFT 13
1.10 Use of the FFT 14
1.11 Use of the FFT 17
1.12 Digital Filters 18
1.13 Digital Filters 20
1.14 Digital Filters 22
1.15 Digital Filters 22
2 EXERCISE 1 26
2.1 NDE of an Aerospace Composite Component 26
2.2 Acoustic Emission Experiment 29
3 REFERENCE 32

1 EXERCISE 1

1.1 Trend Removal

Trends were removed from three given voltage time traces. Figure 1 shows the three voltage time traces with the trend present, removed and the isolated trend.

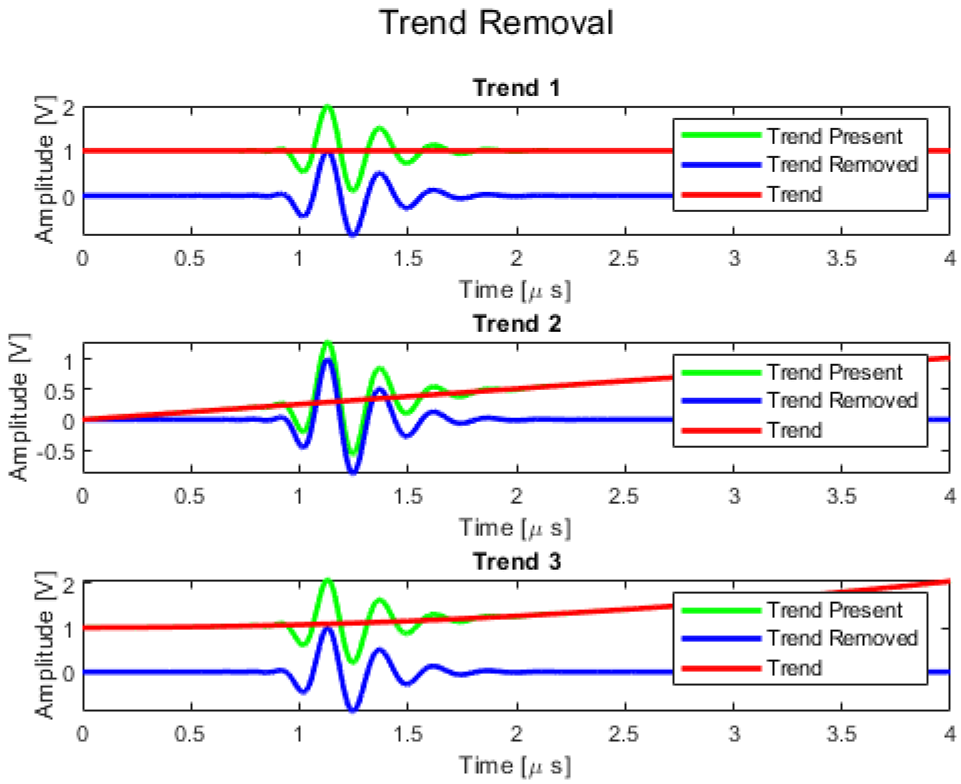

Figure 1 - Trend Removal

The trend was established via the polyfit [1] and polyval [2] MATLAB functions, and subtracted from the original signals. The first trend was noted to be DC offset, whilst the other two trends were noted to be linear and a second order polynomial in nature.

1.2 Coherent Averaging

Coherent averaging was employed on 100 signals that comprised of 1000 samples. It was shown that the noise was reduced by approximately by factor of $\sqrt{1000}$ at 29.98 by taking the standard deviation of the averaged signal.

The following code was used to demonstrate this:

```
%% Exc1_2
close all
clear all

prompt = {'How many signals would you like to coherently average'};
title = 'Signals?';
dims = [1 35];
definput = {'1000'};
m = inputdlg(prompt,title,dims,definput);
m = str2double(m);

prompt = {'How many samples would you like to coherently average'};
title = 'Samples';
dims = [1 35];
definput = {'100'};
n = inputdlg(prompt,title,dims,definput);
n = str2double(n);

signals = zeros(n,5);

for i = 1:m
   signals(:,i) = rand(n,1);
end

st_dev_bef = mean(std(signals));
co_ave = (1/m)*sum(signals,2);
st_dev_aft = std(co_ave);

sprintf('The expected value is %.2f. The actual value is %.2f\n', sqrt(m),
st_dev_bef/st_dev_aft)
```

1.3 Signal Correlation

Two identical arrays comprising of a two cycle in phase cosines were multiplied together in an element wise fashion. Figure 2 shows the two arrays, as well as the element wise product.

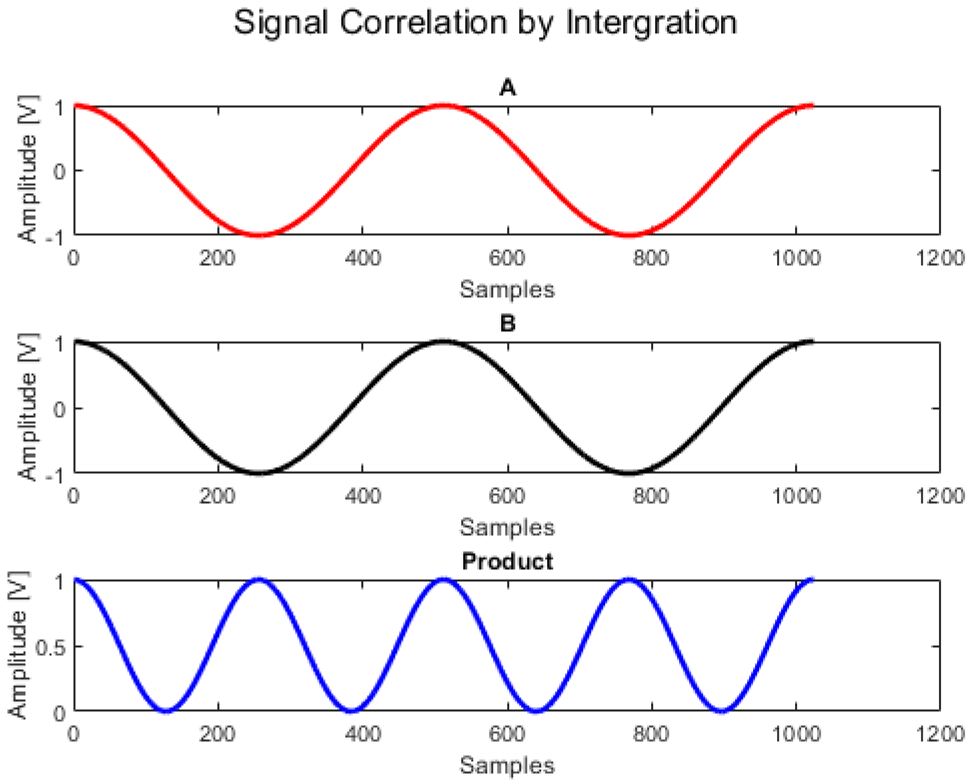

Figure 2 - In Phase 2-Cycle Cosines

The integral of the product was estimated by taking the sum of the product array. For this instance, the area was estimated to be 512. This was in good agreement with the MATLAB trapz function [3], demonstrating that the summation was a good approximation to integrate discretised signals.

1.4 Signal Correlation

Similar analysis was performed on two signals comprising of a 2 cycle cosine a twenty cycle sine wave. This is shown in Figure 3.

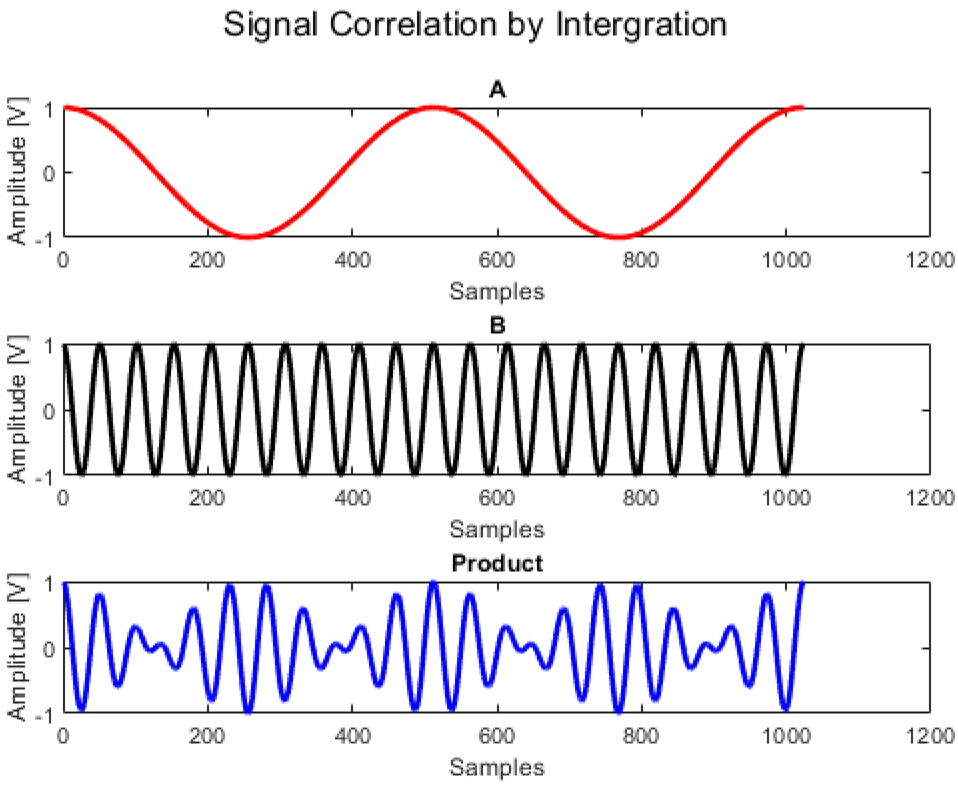

Figure 3 - Two Cycle Cosine & 20 Cycle Sine Wave

The integral was performed in an identical manner, but tended towards a small value at -3.64e-14. This was expected as a sine wave is a 90 degree out of phase cosine wave.

1.5 Signal Correlation

Again, similar analysis was performed on a two cycle cosine and a two cycle sine wave. This is shown in Figure 4.

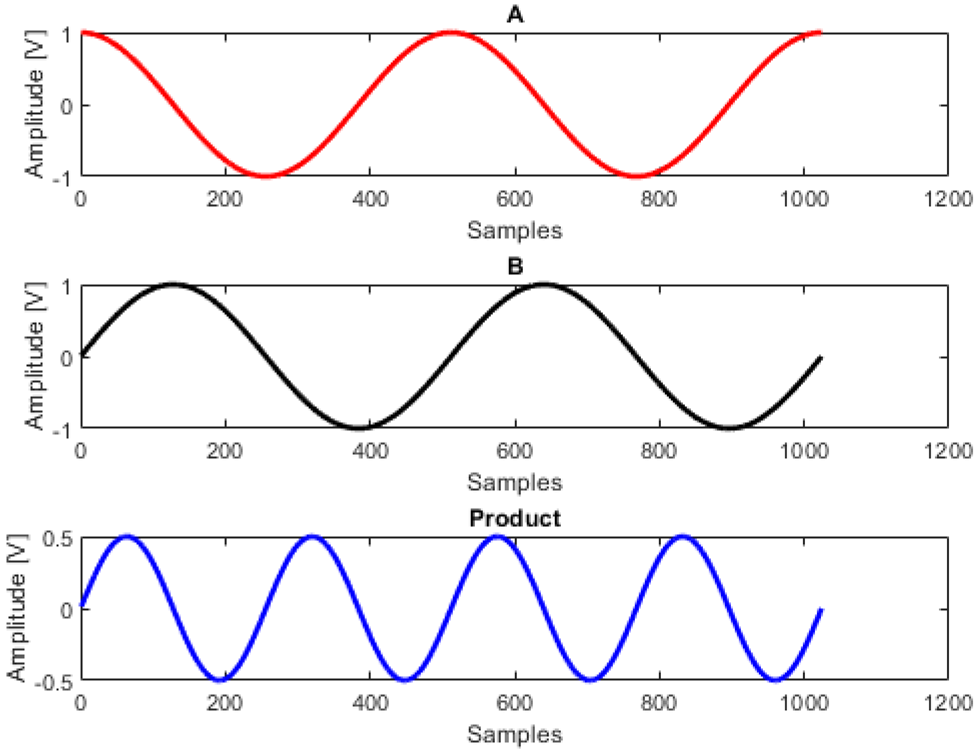

Figure 4 - Two Cycle Sine & Cosine

The was estimated to be 3.952674e-14. This again is expected due to the out of phase nature between a cosine and sine functions previously discussed.

1.6 Signal Correlation

A function was developed to cross correlate two signals based on the following formula:

$$r(k) = \frac{\sum_{n=1}^{N}[x(n) - \bar{x}][y(n+k) - \bar{y}]}{\left(\sum_{n=1}^{N}[x(n) - \bar{x}]^2 \sum_{n=1}^{N}[y(n) - \bar{y}]^2\right)^{\frac{1}{2}}} \quad (1)$$

The removal of the denominator allows for the cross correlation to be completed with no normalisation, and vice versa. As cross correlation returns a signal of length 2N-1 the signals needed to be padded with zeros to make them twice the length of the original signal. The inbuilt MATLAB circshift function [4] was used to shift the signals over each other in a cyclic fashion and take the summation of the product between the two signals at each point.

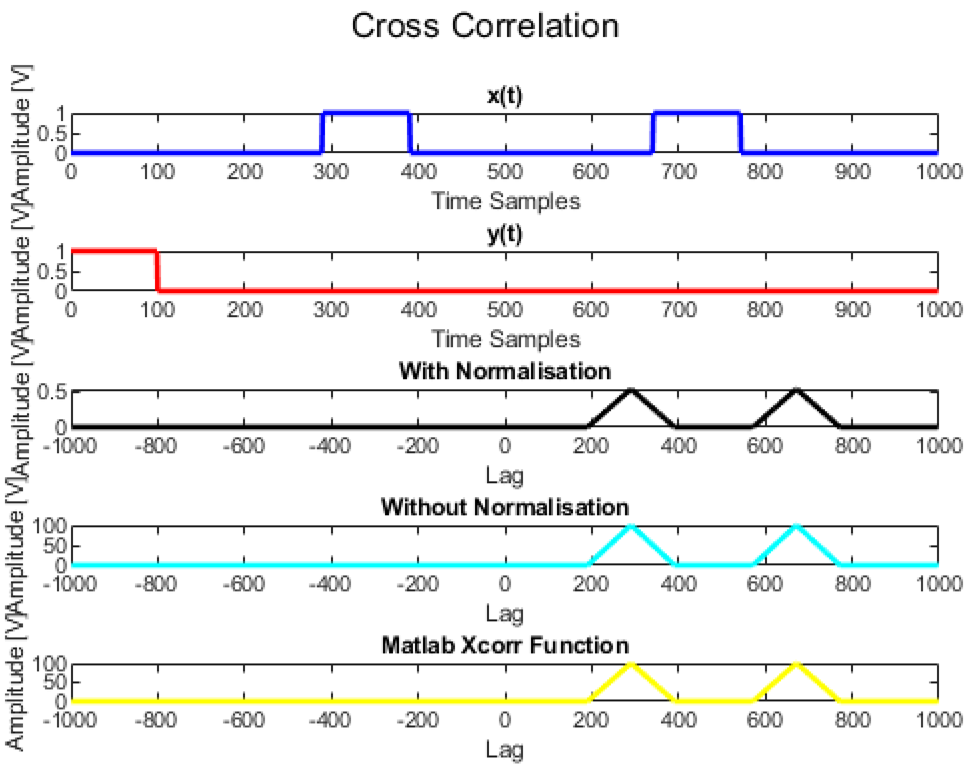

Figure 5 - Cross Correlation

Figure 5 shows two signals, x(t) & y(t), their respective cross correlation and normalised cross correlation based on Eq. 1 as well as the MATLAB cross correlation [5]. As can be seen the developed cross correlation function produced the same result as the MATLAB cross correlation function, giving confidence in the results.

If two of the same signals are cross correlated together, it is known as auto correlation. Figure 6 shows two identical signals being correlated in a similar fashion as previously shown. Auto-correlation can be largely viewed as a test of randomness.

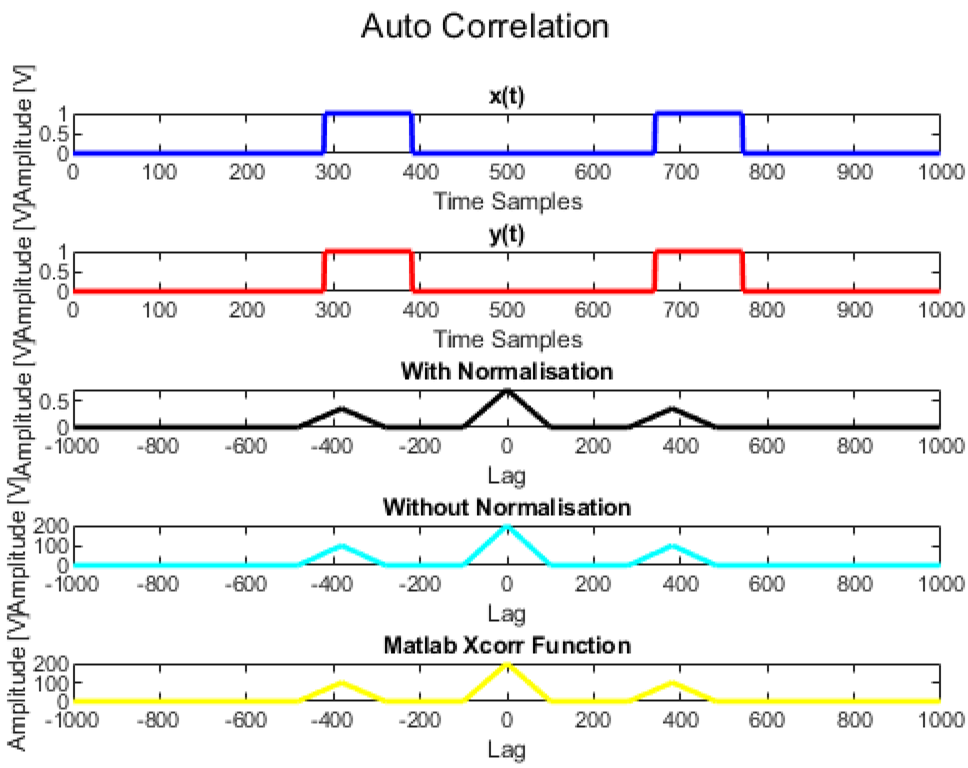

Figure 6 - Auto Correlation

1.7 Use of the FFT

Two 16 point arrays were generated by the following formulas to show the key elements associated the Fast Fourier Transform (FFT) [6]:

$$\mathrm{A}\cos\left(nk\frac{2\pi}{N}\right) \tag{2}$$

$$Bsin\left(nk\frac{2\pi}{n}\right) \tag{3}$$

Initially, an A and B value of 2 was utilised with a K number of 1, and produced the result demonstrated in Figure 7

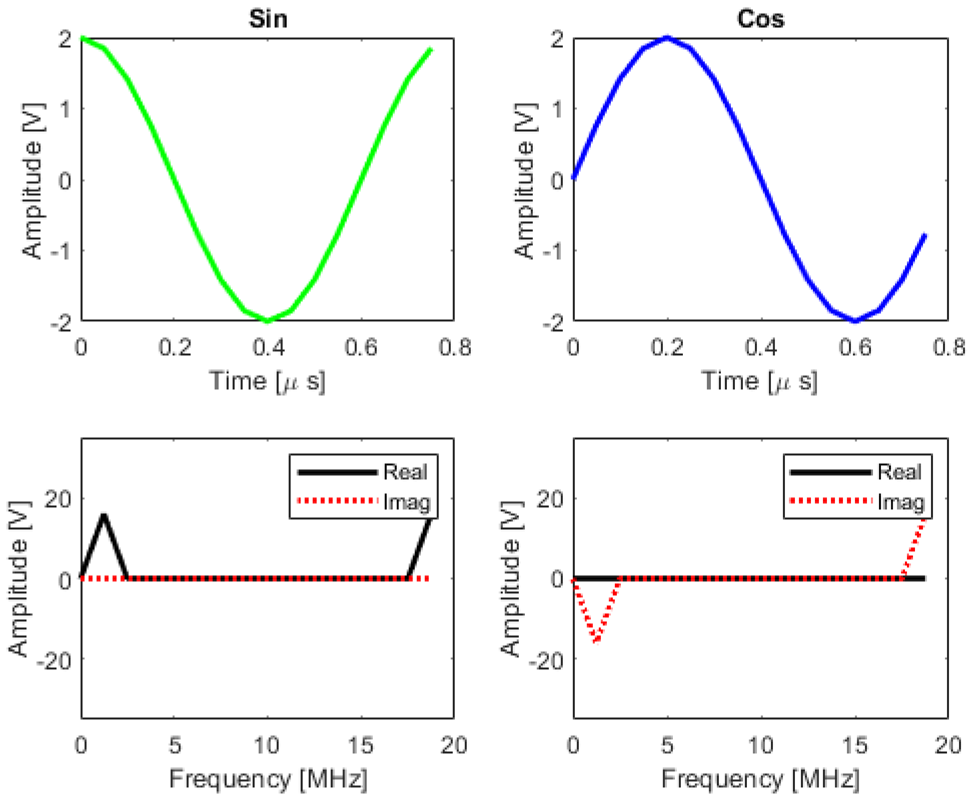

Figure 7 - FFT of Cos and Sine (A=2,B=2,K=1)

The same procedure was repeated over, but with an A and B value of 4 and a K value of 2, and the results are shown in Figure 8.

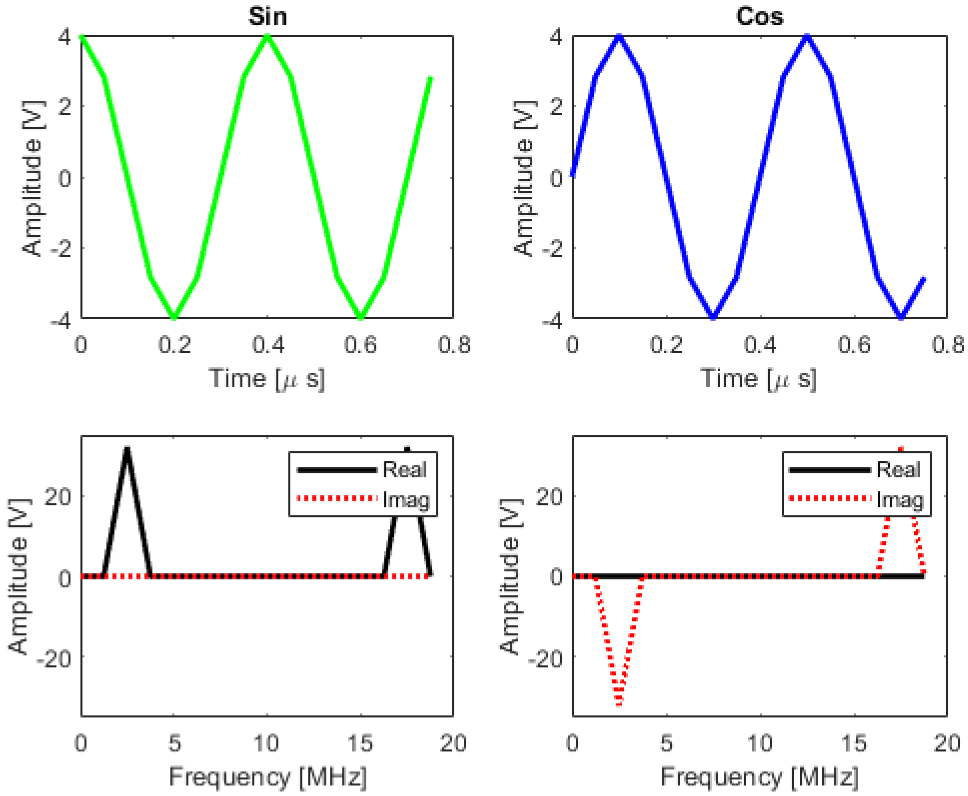

Figure 8 - FFT of Cos and Sine (A=4,B=4,K=2)

From Figure 7 & Figure 8, it can be seen that by doubling the A and B value (i.e. the amplitude of the sine and cosine function) the amplitude response in the frequency domain double as well. Furthermore, it can also be observed that by doubling the wavenumber (k value), the dominant frequency moved from 1.25 MHz to 2.5MHz i.e. 1 frequency bin further up.

As there are only 16 points in this, sample the Nyquist limit can be observed by increasing the k value to being greater than 8.

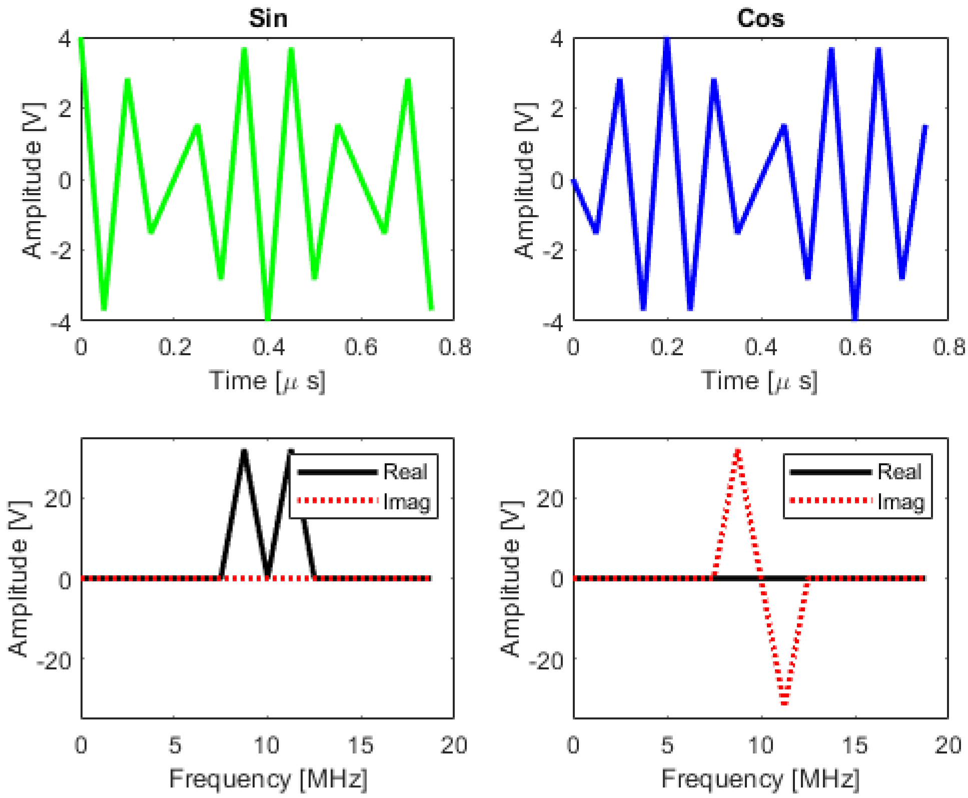

Figure 9 - FFT of Cos and Sine (A=4,B=4,K=9)

Figure 9 shows the same result as the previous analyses, when k is increased to 9. As the FFT is continuous, it can be seen that the energy has wrapped around and all the energy has been allocated to the 7^{th} and 9^{th} frequency bin of 8.75 & 11.25 MHz respectively.

1.8 Use of the FFT

The FFT can be utilised to perform interpolation. This can be achieved by zero padding the frequency response to give the impression that the signal was sampled at a much higher rate.

For the example signal provided, the frequency spectrum was padded to be four times that of its original length to up the sampling frequency from 40MHz to 160MHz. Care must be taken to ensure that symmetry in the Fourier domain remains constant, and that the resulting amplitude is scaled accordingly. This is shown to be observed in the results provided in Figure 10.

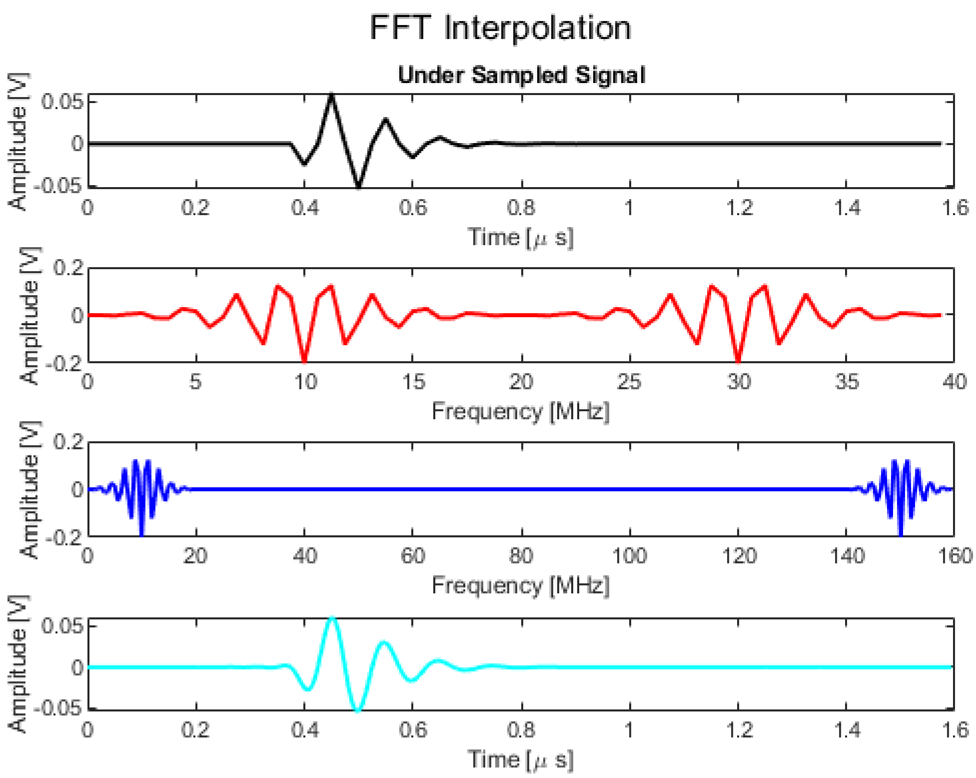

Figure 10 - FFT Interpolation

The original and interpolated signal is compared in Figure 11. If viewed at the start and the end of the signal, the Gibb's phenomenon is observed [7]. This is due to the nature of representing a nonlinear signal with a finite number of Fourier coefficients in the Fourier series.

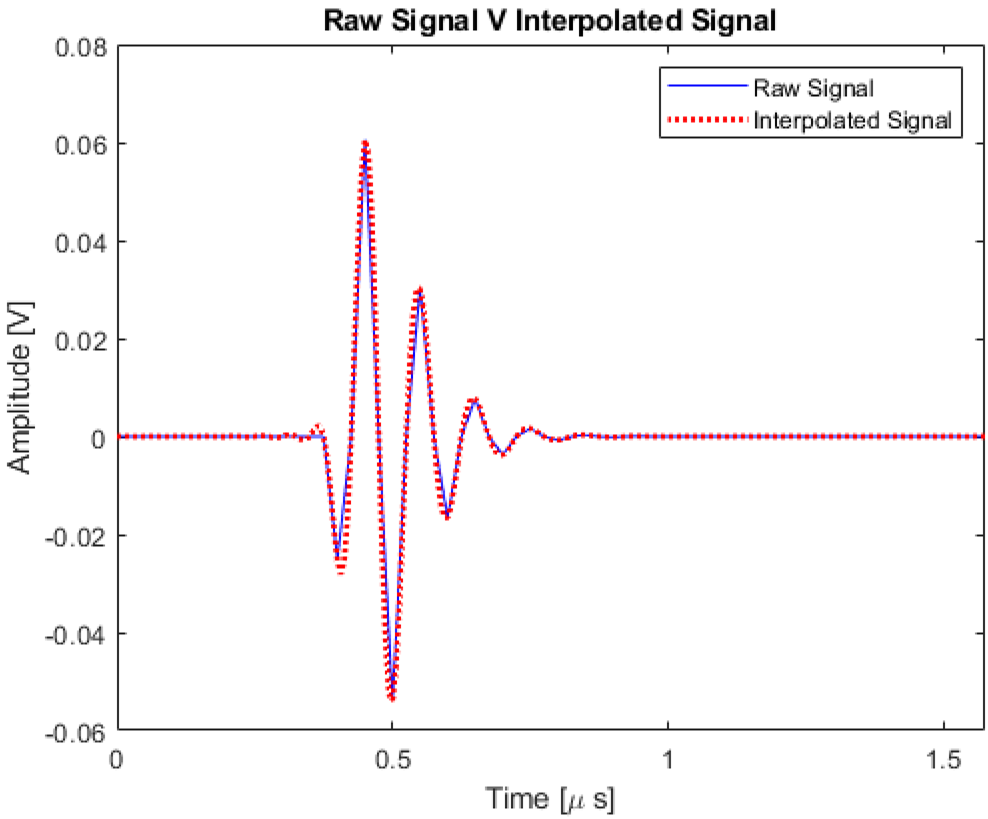

Figure 11 - Gibbs Phenomenon

1.9 Use of the FFT

An ultrasonic time trace was provided that has been contaminated the with a single radio frequency generating unwanted incoherent noise. By taking the FFT of the signal, the frequency of the transducer and noise can be observed.

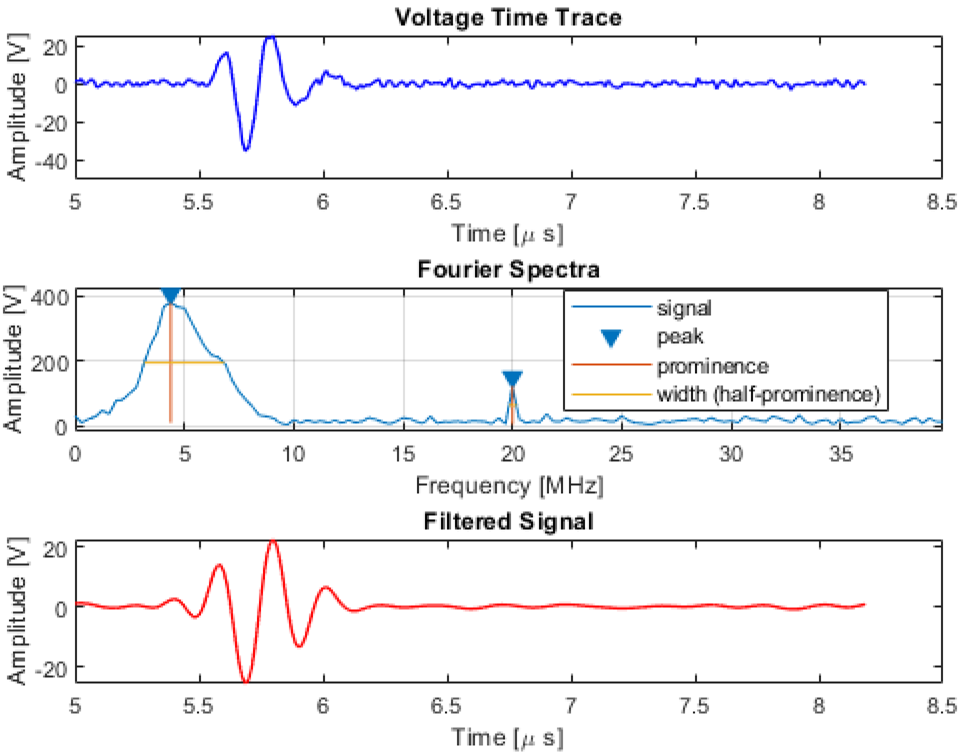

Figure 12 - Filtering Noisy Ultrasonic Data

Figure 12 shows the original noisy ultrasonic data and it's respective FFT. It can clearly see that the ultrasound transducer has a centre frequency of 4.75 MHz, and the noise has been generated with a frequency of 20MHz.

By simple windowing and multiplication in the Fourier domain, a low pass filter can be employed. This is accomplished by identifying all frequency components with a -6dB (equivalent to a full width half maximum) drop above the centre frequency of the transducer and setting them to zero with a hanning window. This multiplication in the Fourier domain is equivalent to convolution in the time domain.

The result is shown in the last graph in Figure 12. As can be seen much of the noise has been removed from the signal.

1.10 Use of the FFT

The same ultrasonic signal as in Section 0 and it's respective frequency spectrum shown in Figure 13. This signal was subject to a single pole filter defined by Eq. 4-6

$$H(k) = \frac{k_0}{k_0 + jk} \tag{4}$$

$$k = \frac{\omega}{\omega_s} N \tag{5}$$

$$k_o = \frac{\omega_0}{\omega_s} N \text{ where } \omega_0 = 0.2\omega_s \tag{6}$$

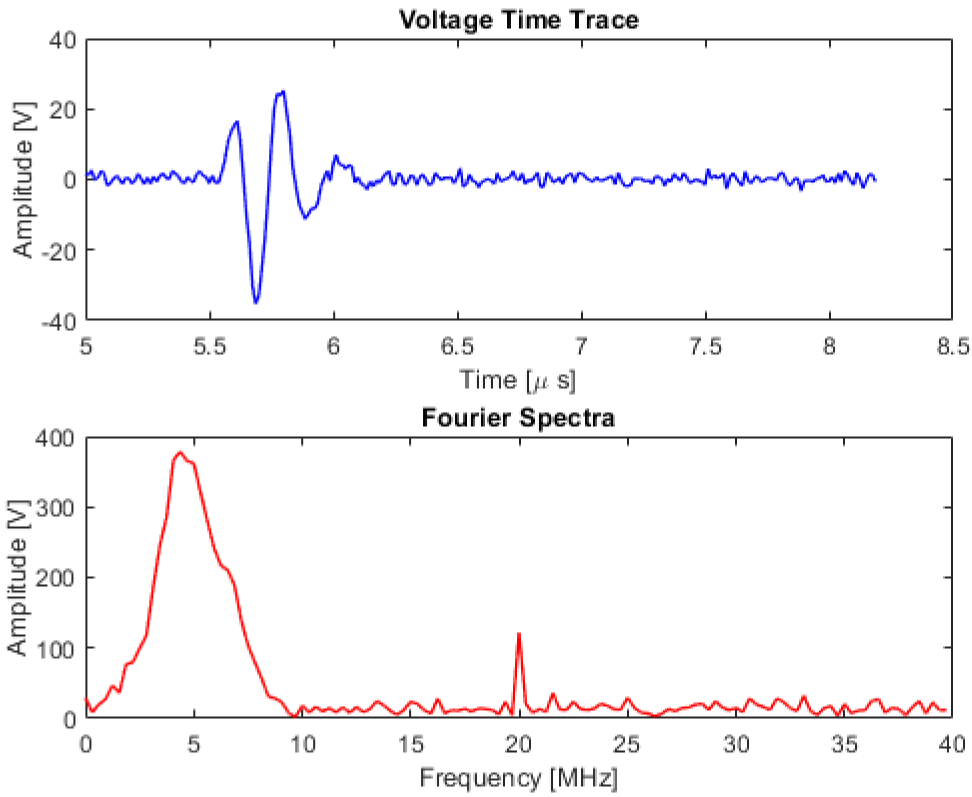

Figure 13 - Ultrasonic Time Trace and Frequency Spectrum

From Eq. 4-6, the filter produced the frequency response given in Figure 14. In designing this filter, the second half must be flipped and the complex conjugate taken to allow for the Fourier properties to remain valid. This is shown in Figure 15.

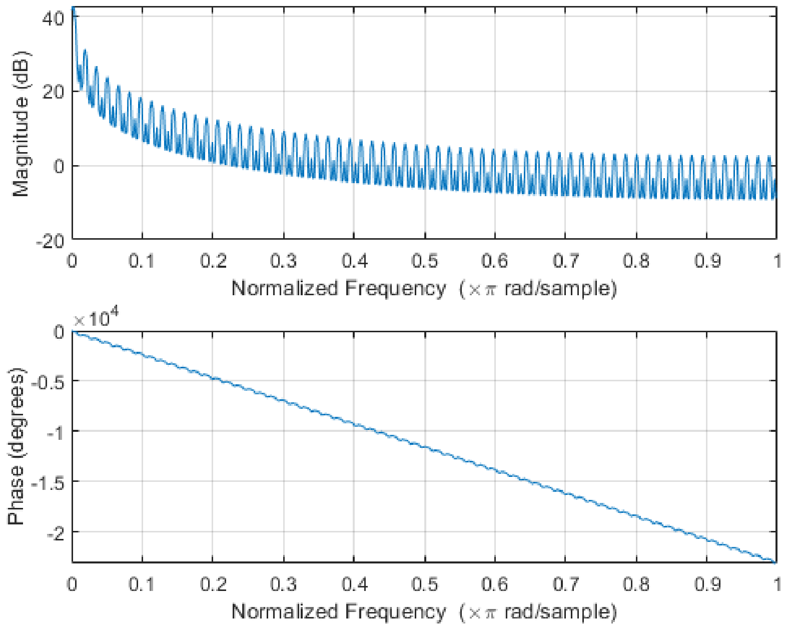

Figure 14 - Frequency response of Single Pole Filter

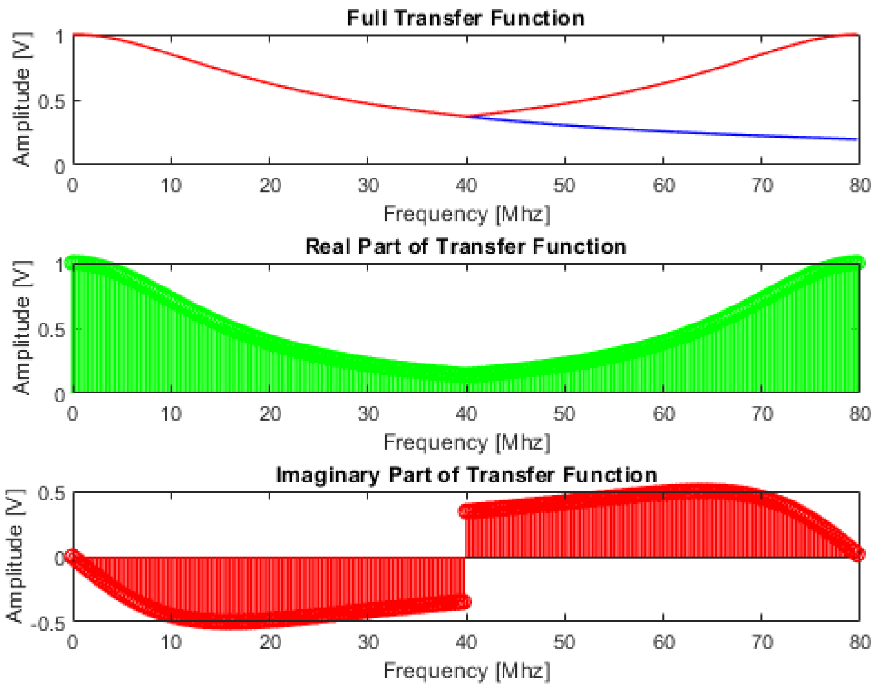

Figure 15 - Real and Imaginary Part of Transfer Function

The filter can be performed on the original signal by performing point wise multiplication with the original signal and taking the Inverse Fast Fourier Transform (IFFT). The real and imaginary elements of the IFFT are shown in Figure 16. The imaginary elements oscillate around zero, and are an artefact associated with performing such analyses. As such they have no physical meaning.

It can be seen that some of the high frequency components have been removed from the time trace, but it is of a very low magnitude suggesting this filter has poor performance when dealing with this type of noise.

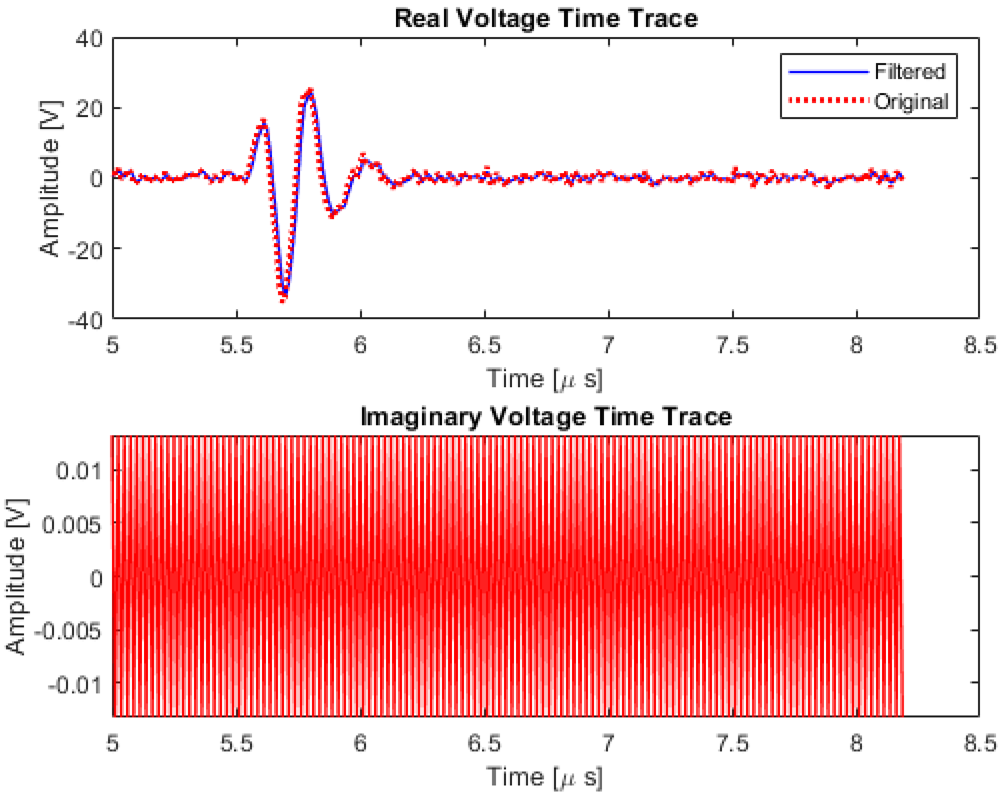

Figure 16 - Real and Imaginary Traces of Filtered Signal

1.11 Use of the FFT

The same analysis was repeated from Section 1.10, but the analysis was performed in the time over the frequency domain. As alluded to earlier, convolution in the time domain is the same as performing multiplication in the frequency domain for a linear time invariant system, such as this [8].

$$h(n) = e^{-\frac{n2\pi f_0}{f_s}} \tag{7}$$

Once the array that represents this filter is established, it is convolved with the original ultrasound signal by the following:

$$r(k) = \sum_{n=1}^{N} x(k-n)\,y(n) \tag{8}$$

Figure 17 shows the filtered signal and the original signal. Filtering was also performed by utilising the inbuilt convolution function in MATLAB [9] to act as a bench mark. As can be seen, good agreement between the results are observed.

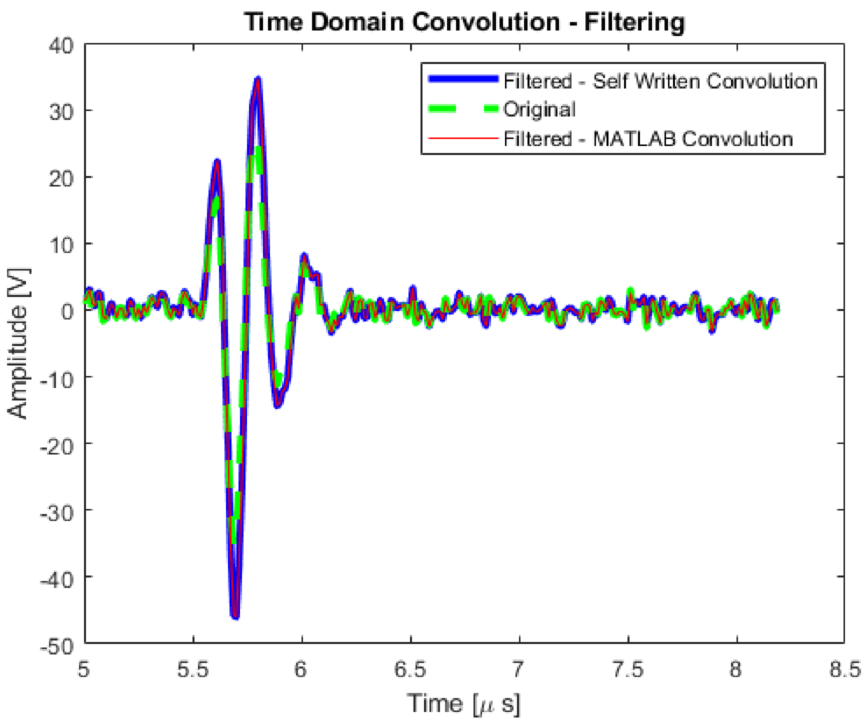

Figure 17 - Filtering Via Convolution

Figure 18 shows the original and filtered Fourier spectrum. This filter provides a boost to the frequencies of interest, but has low attenuation of the noisy frequency at 20 Mhz.

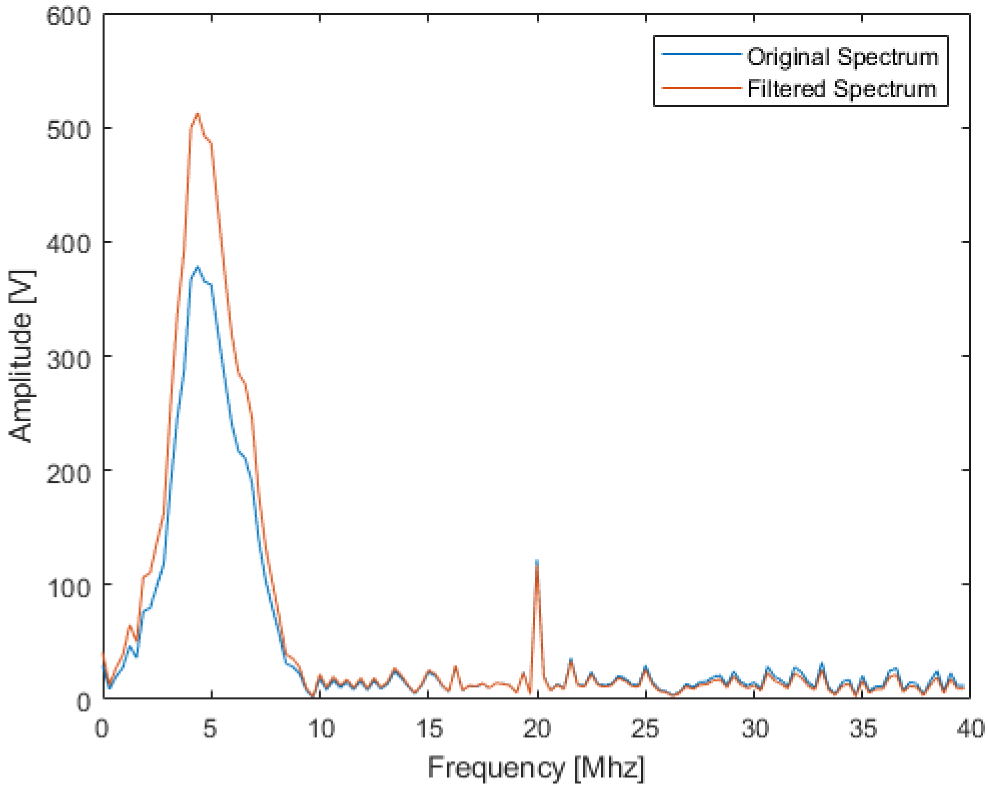

Figure 18 - Original and Filtered Fourier Spectrum

1.12 Digital Filters

A digital filter in the form of Eq. 9 was implemented on the same noisy ultrasonic time trace as in Section 0:

$$H(z) = \frac{z+1}{z\left[1+\frac{2CR}{T}\right] - \left[\frac{2CR}{T} - 1\right]} = \frac{z+1}{Az - B} \tag{9}$$

Where,

$$A = 1 + \frac{2CR}{T}, \qquad B = \frac{2CR}{T} - 1, \quad \frac{CR}{T} = \frac{2\pi\omega_0}{\omega_s}, \qquad H(z) = \frac{Y(z)}{x(z)} \tag{10}$$

From the formula specified in Eq. 10, it is possible to arrive at Eq. 11. By inspection and carrying out the reverse z-transform, the discrete time implementation of this filter is arrived at in Eq. 12

$$Y(z) = \frac{1}{A}[[1 + z^{-1}]x(z) + Bz^{-1}Y(z)] \tag{11}$$

$$y(n) = \frac{1}{A}[x(n) + x(n-1) + y(n-1)B] \tag{12}$$

This filter was then employed on the given ultrasonic signal, and is shown in Figure 19.

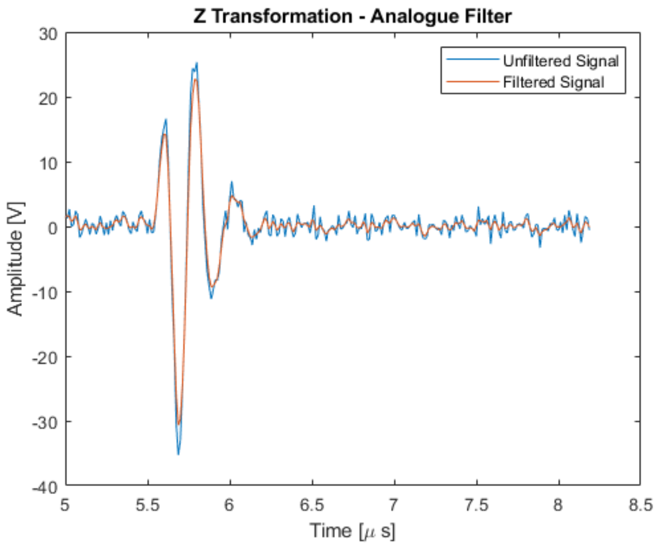

Figure 19 - Digital Filter with Analogue Counterpart

Like the previous analysis, the filtered and unfiltered Fourier spectra are given in Figure 20. This filter slightly attenuates the frequency of interest around 5MHz, and does a much better job of attenuating the interference at 20MHz than the filter previously presented in Section 1.9.

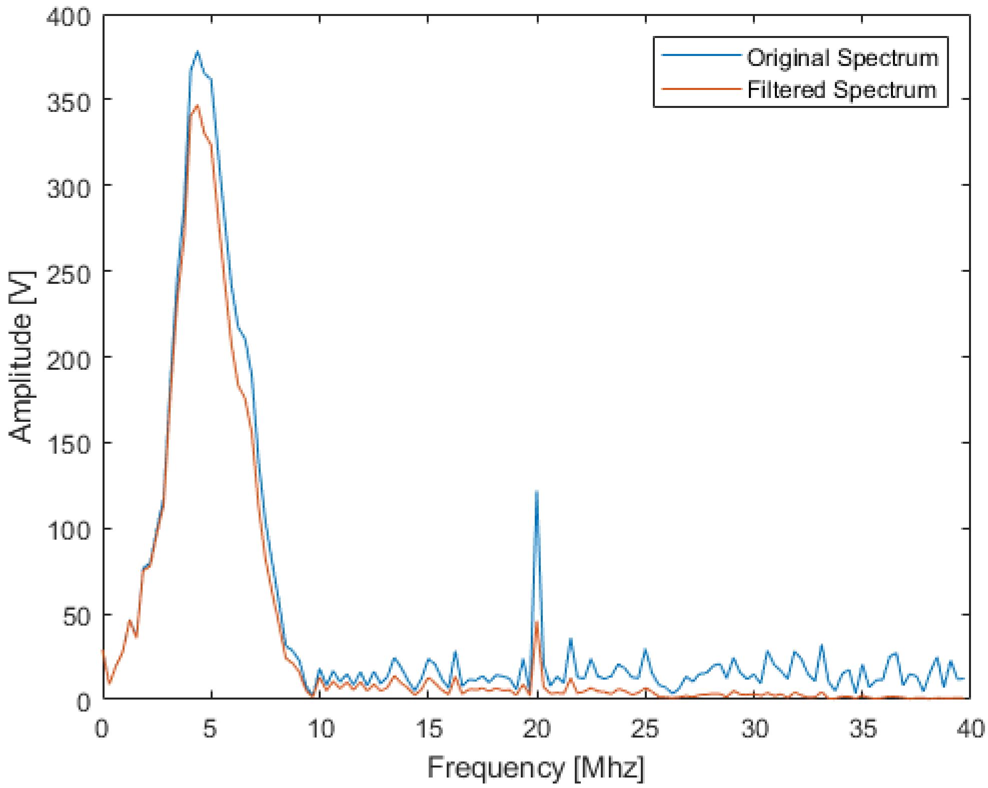

Figure 20 - Original and Filtered Fourier Spectra

1.13 Digital Filters

A moving average filter in form given in Eq. 13 was implemented on the same ultrasonic time trace as in Section 0. The filtered and unfiltered signal and Fourier Spectra are presented in Figure 21 & Figure 22 respectively. As can be seen the filtered signal greatly amplifies the frequency of interest and slightly attenuates the noise.

$$y(n) = y(n-1) + x(n) - x(n-N) \tag{13}$$

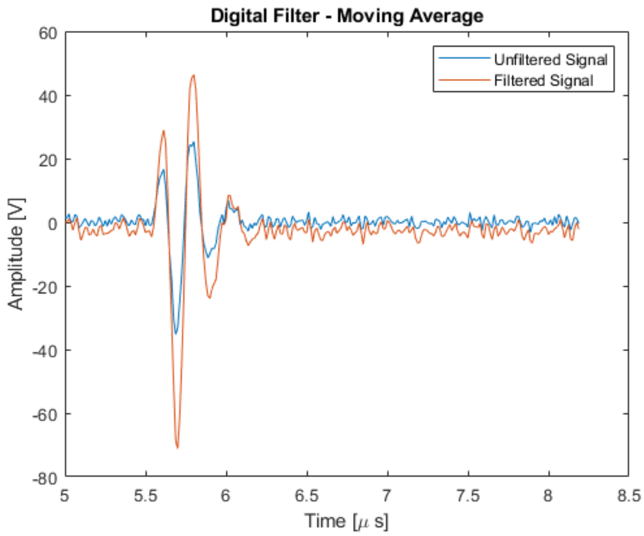

Figure 21 - Filtered and Unfiltered Signal

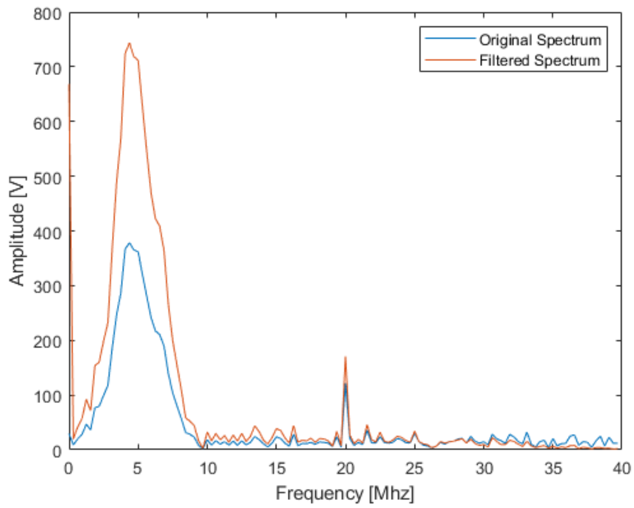

Figure 22 - Filtered and Unfiltered Fourier Spectrum

1.14 Digital Filters

A band and high pass filter as defined in Eq. 14 & 15, was implemented on the same ultrasonic signal given in Section 0, and presented in Figure 23.

$$y(n) = -y(n-2) + x(n) - x(n-4) \tag{14}$$

$$y(n) = -y(n-1) - x(n) - x(n-4) \tag{15}$$

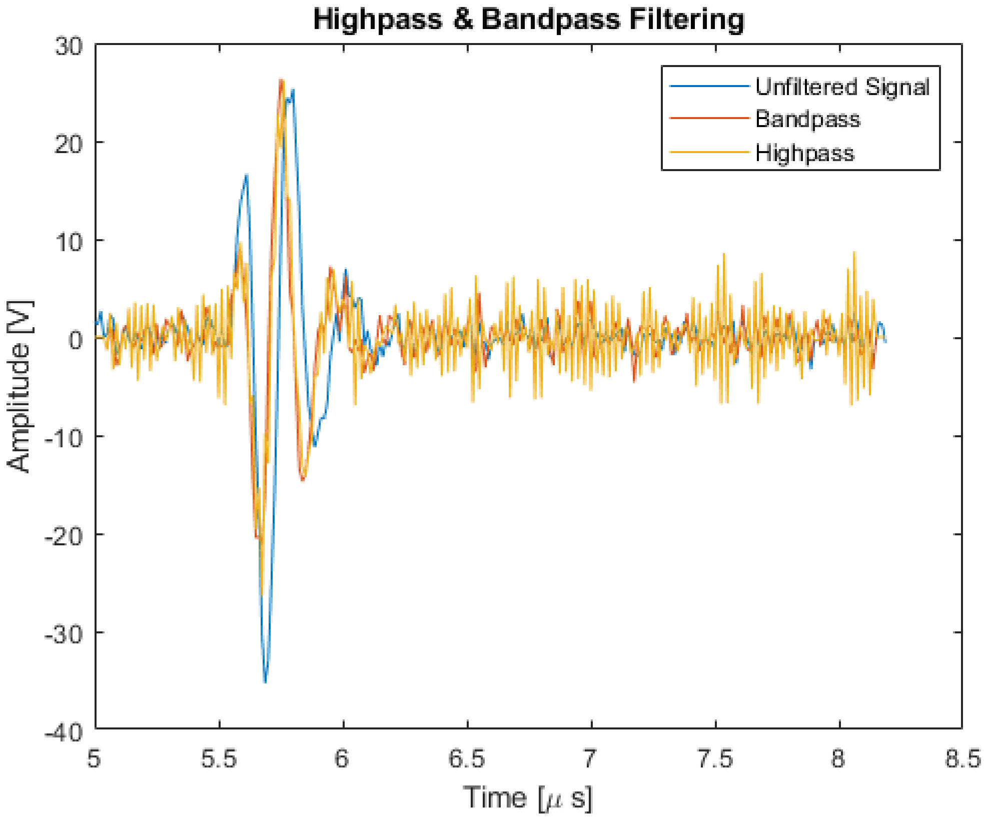

Figure 23 - Band pass, High Pass and Original Signals

The respective Fourier Spectra are plotted in Figure 24. As can be seen the high pass filter attenuates the lower frequencies and amplifies the higher frequencies. It can also be seen that the band pass filter attenuates the higher frequency very well but also attenuates the frequency of interest at around 5MHz, suggesting the band needs to be altered, for optimum performance.

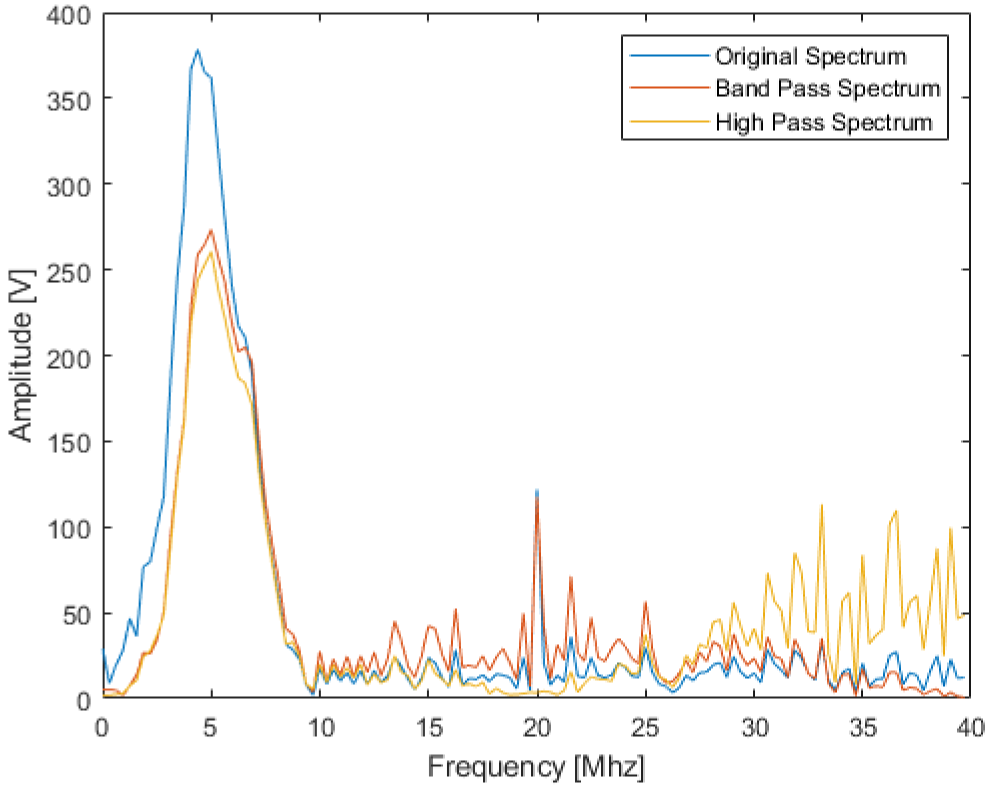

Figure 24 - Original, Band Pass, High Pass Spectrum

1.15 Digital Filters

A notch filter in the form of Eq. 16 was implemented on the same ultrasonic signal as given in Section 0.

$$y(n) = b_1 y(n-1) + b_2 y(n-2) + x(n) + a_1 x(n-1) + x(n-2) \quad (16)$$

Where,

$$p = 0.9, \quad \theta = , \quad a_1 = -2\cos\theta, \quad b_1 = 2pcos\theta, \quad b_2 = -p^2$$

The filtered and original signal and their respective Fourier Spectra are plotted in Figure 25 & Figure 26. As can be seen the single interfering frequency is well attenuated, but some noise is still present. This is thought to be due to the windowed nature of the interference generating its own Fourier Spectra. It is thought that if a more wideband filter were to be used, better results would be obtained.

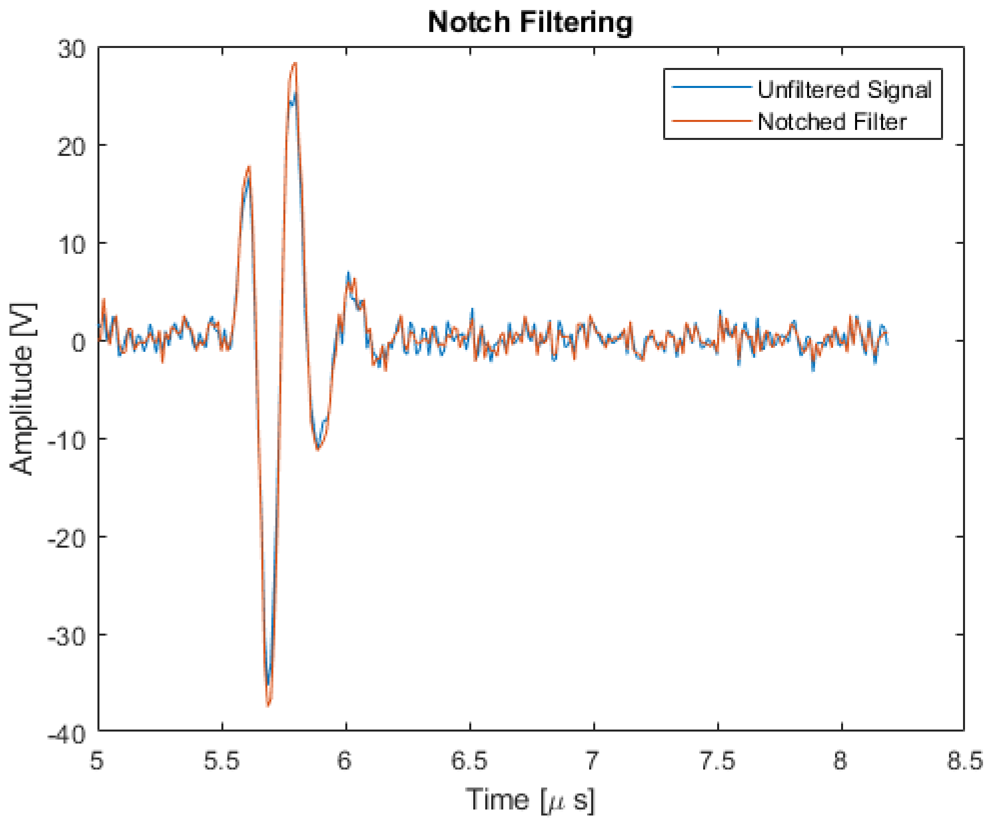

Figure 25 - Filtered and Original Signal

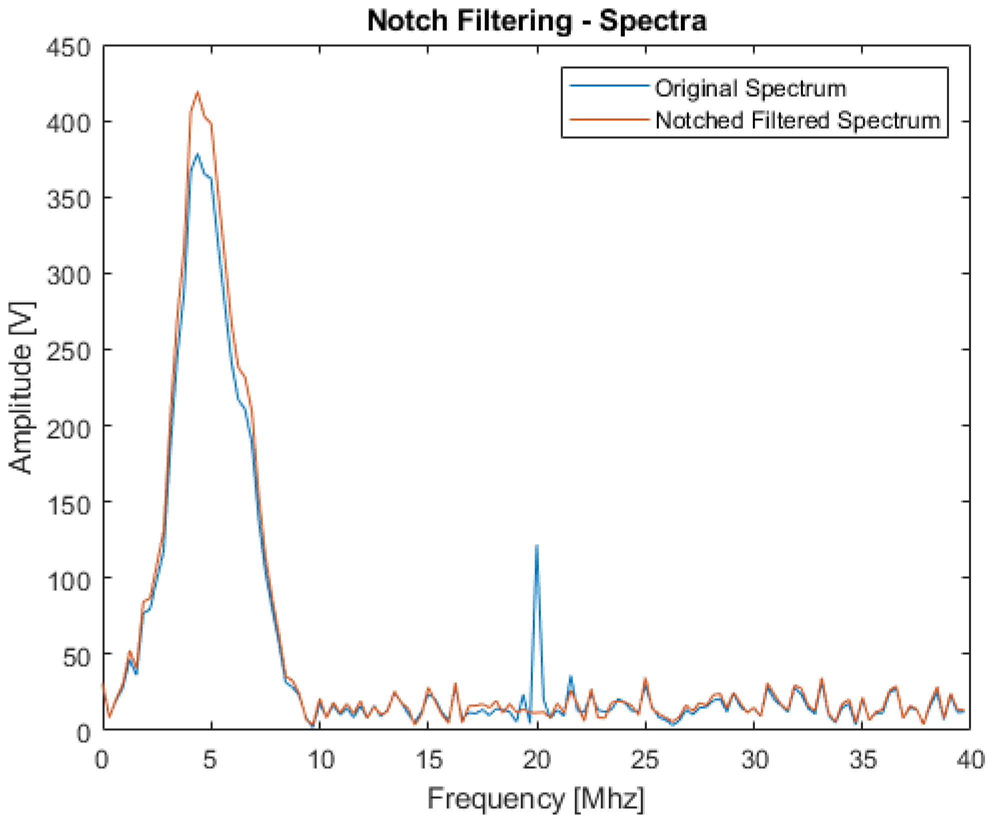

Figure 26 - Original & Notched Fourier Spectra

2 EXERCISE 1

2.1 NDE of an Aerospace Composite Component

An ultrasonic A-scan time domain signal sampled at 40 MHz was provided of a 4 mm thick aerospace component with the intention for the attenuation coefficient to be derived. This can then be correlated to the components structural health.

Figure 27 shows the A scan signal with the front wall and back wall isolated at approximately 1.75 µs and 4.5 µs respectively. This was isolated use a tapered cosine window using the inbuilt MATLAB tukey function [10]. Within this function, two parameters can be specified the window length and the taper value. The window length was determined by manual analysis of the data, and the window taper was set to 0.8.

With the front and back wall echoes isolated in time, the back wall echo was multiplied by 1/0.47, as the reflection coefficient was given to be 0.53, so hence the transmission coefficient can be determined to be 0.47. This has the effect of normalising the back wall echo as if there had been zero attenuation due to the difference in paths taken in the two parts of the signal.

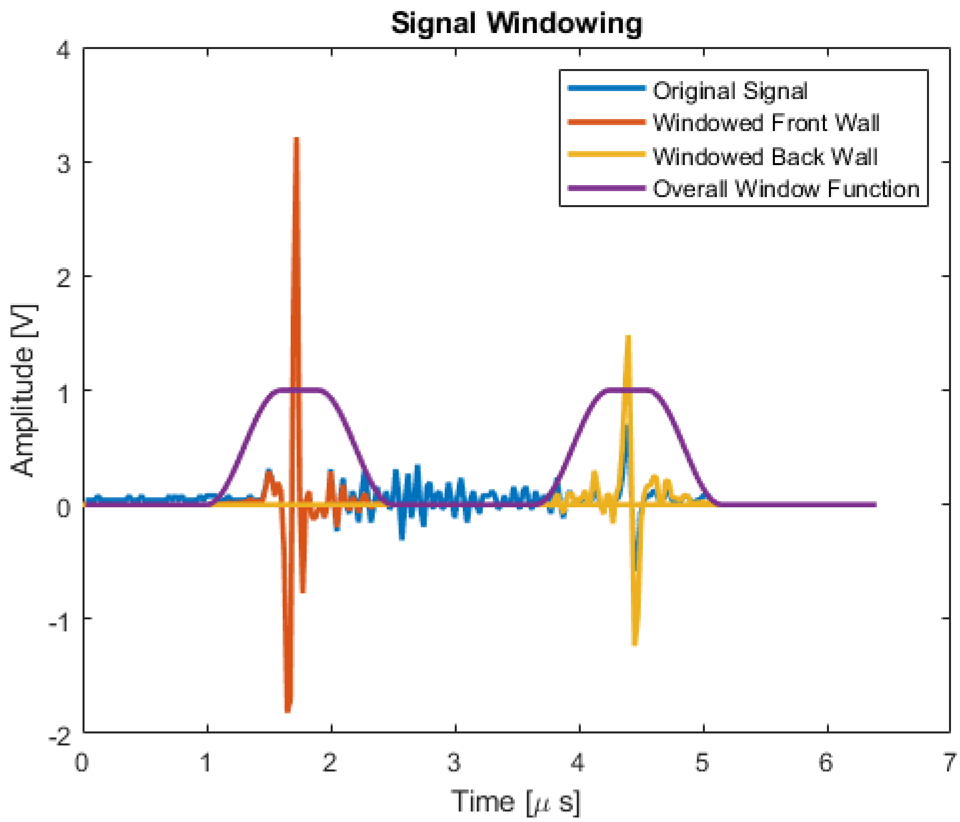

Figure 27 - Ultrasonic Time Trace of a Composite Component

The Fourier spectra of the back and front wall signals are given in Figure 28. As can be seen they vary drastically over the frequency range.

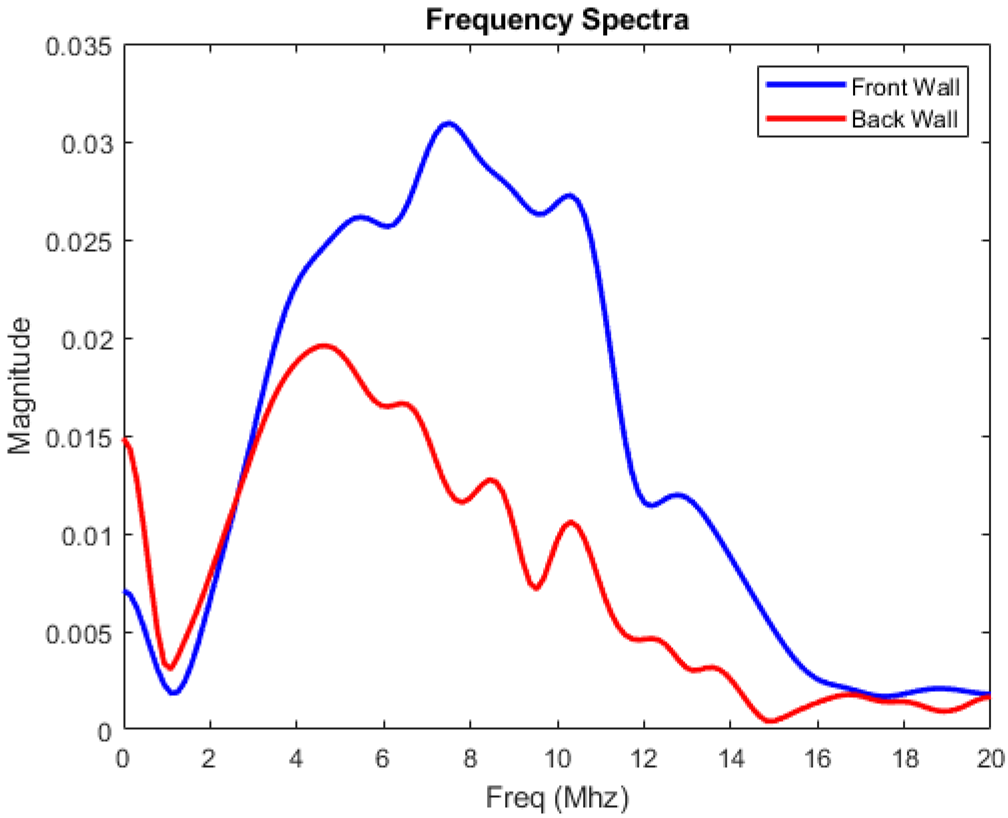

Figure 28 - Fourier Spectrum of Front and Back Wall

From, the FFT of both the front wall and back, the attenuation coefficient as function of frequency can be calculated by Eq. 17, where d is the component thickness, A is the modulus of the back wall generated from the FFT of the back wall, and A_0 is the modulus of the front wall generated from the FFT of the front wall.

$$\alpha(f) = -\frac{1}{2d}\ln\left(\frac{A(f)}{A_0(f)}\right) \quad (17)$$

The attenuation coefficient as a function of frequency is given in Figure 29. As can be seen the attenuation coefficient generally increases with frequency, which is broadly in line with what is expected.

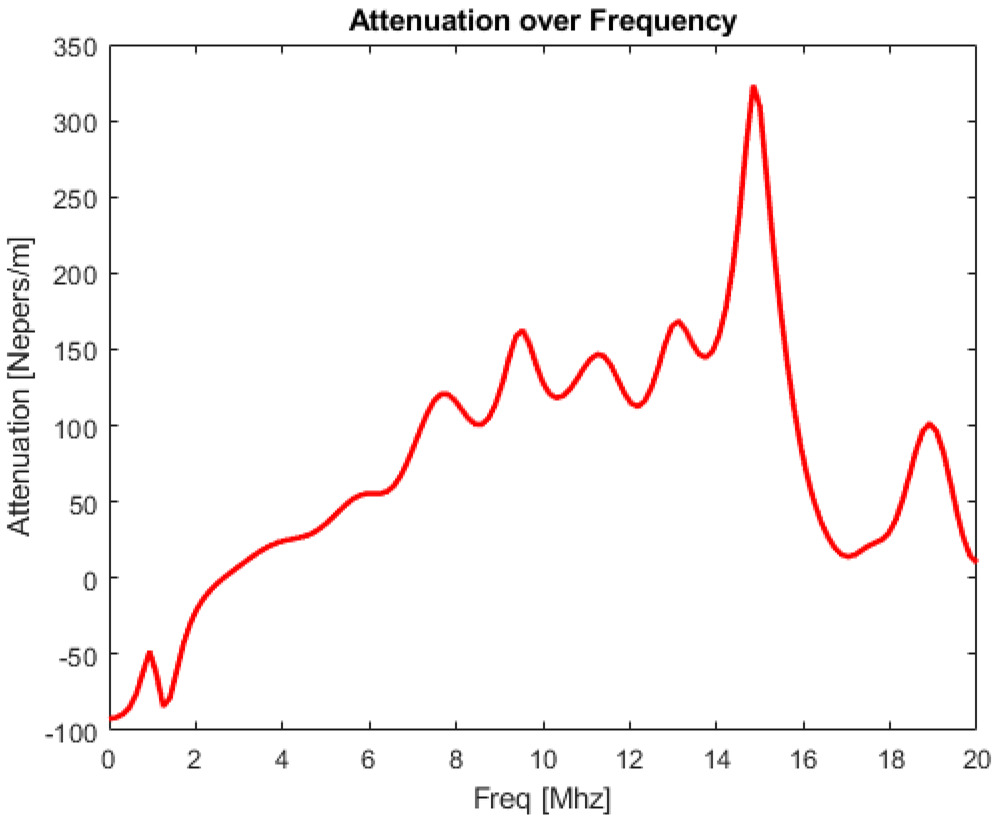

Figure 29 - Attenuation Coefficient As a Function of Frequency

2.2 Acoustic Emission Experiment

An acoustic emission experiment had been simulated and produced two acoustic signals: one that has propagated through plant components (X2) and the other that has not (X1). These two signals are plotted in Figure 30 & Figure 31 respectively.

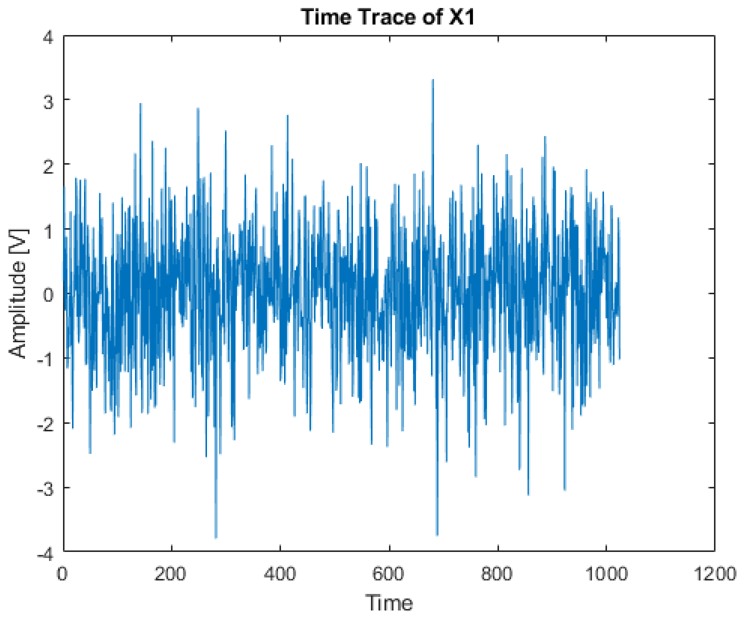

Figure 30 - Time Trace of X1

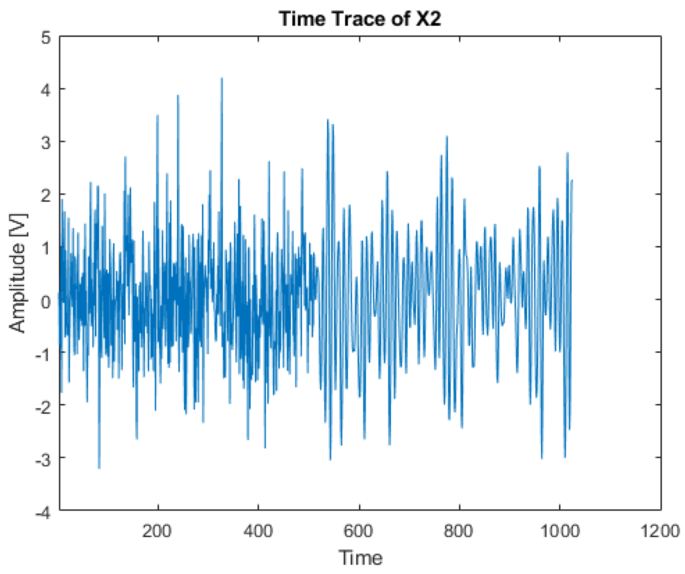

Figure 31 - Time Trace of X2

To prove that the X1 signal was mostly comprised of random noise, the standard deviation, and the auto correlation of the time trace were calculated. The standard deviation was determined to be 0.99, and the auto correlation is given in Figure 32. As can be seen from the figure and the larger standard deviation value, there is no periodicity indicating a high degree of randomness.

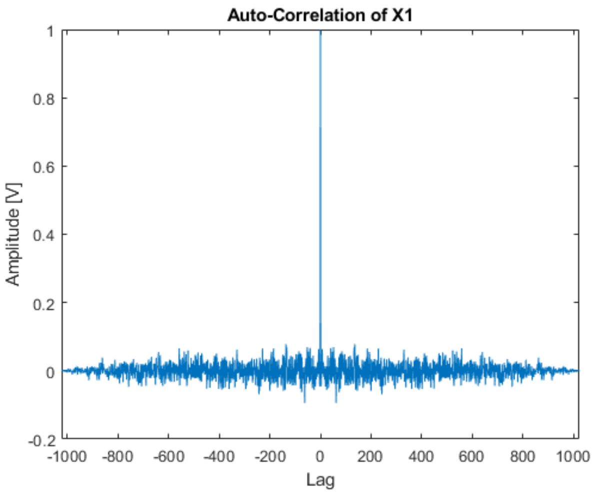

Figure 32 - Auto Correlation of X1

The cross correlation of X1 and X2 was taken next to determine the impulse response and is shown in Figure 33. This in turn allows for the time delay to be calculated where the signals are most similar. The maximum amplitude is observed at 515 samples, and this corresponds to 12.9 ms with a sampling frequency of 40 MHz.

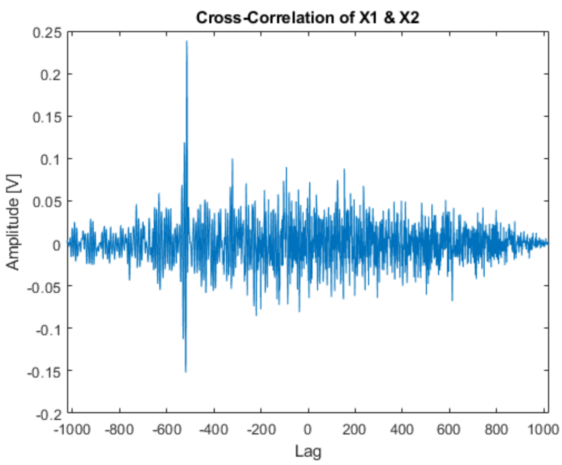

Figure 33 - Cross Correlation between X1 & X2

The Fourier spectra of X1 and X2 are shown in Figure 34 & Figure 35. The spectrum of X1 can be seen be nearly the same amplitude across the frequency range. This is expected due to it's random nature. On the other hand, the Fourier spectrum of X2 clearly shows the introduction of a signal at approximately 4 KHz.

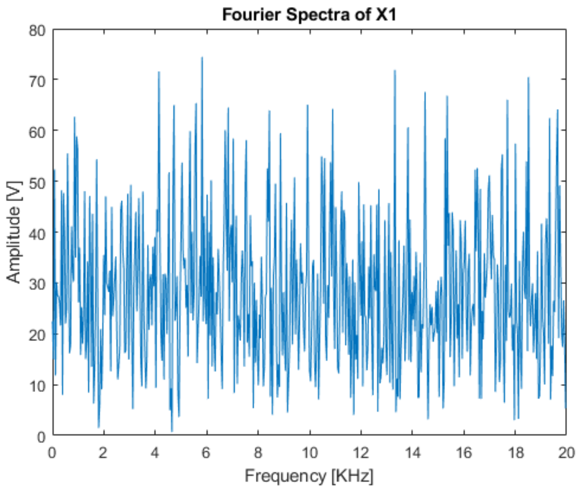

Figure 34 - Fourier Spectrum of X1

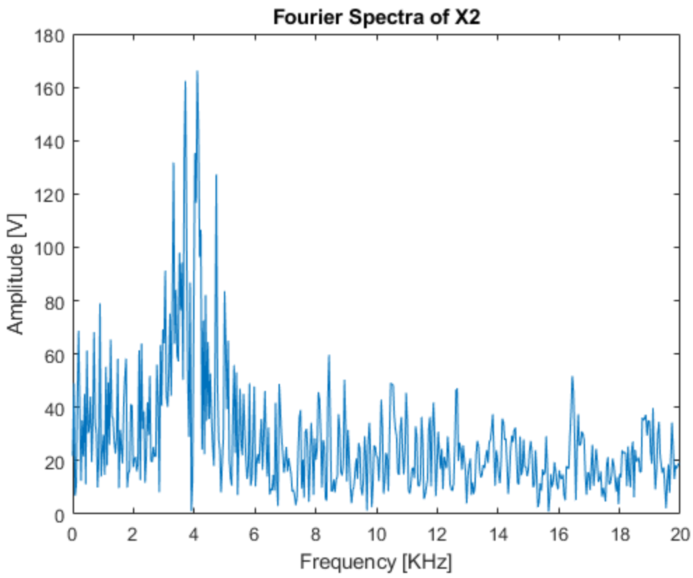

Figure 35 - Fourier Spectrum of X2

It is thought that by monitoring defects with respect to their time lag, and Fourier spectrum a suitable NDT technique and inspection schedule could be established.

3 REFERENCES

[1] "Polynomial curve fitting - MATLAB polyfit - MathWorks United Kingdom." [Online]. Available: https://uk.mathworks.com/help/matlab/ref/polyfit.html.
[2] "Polynomial evaluation - MATLAB polyval - MathWorks United Kingdom." [Online]. Available: https://uk.mathworks.com/help/matlab/ref/polyval.html.
[3] "Trapezoidal numerical integration - MATLAB trapz - MathWorks United Kingdom." [Online]. Available: https://uk.mathworks.com/help/matlab/ref/trapz.html.
[4] "Shift array circularly - MATLAB circshift - MathWorks United Kingdom." [Online]. Available: https://uk.mathworks.com/help/matlab/ref/circshift.html.
[5] "Cross-correlation - MATLAB xcorr - MathWorks United Kingdom." [Online]. Available: https://uk.mathworks.com/help/matlab/ref/xcorr.html.
[6] "Fast Fourier transform - MATLAB fft - MathWorks United Kingdom." [Online]. Available: https://uk.mathworks.com/help/matlab/ref/fft.html.
[7] E. W. Weisstein, "Gibbs Phenomenon." [Online]. Available: http://mathworld.wolfram.com/GibbsPhenomenon.html.
[8] "Linear time-invariant system," *Wikipedia*. 05-Jul-2019.
[9] "Convolution and polynomial multiplication - MATLAB conv - MathWorks United Kingdom." [Online]. Available: https://uk.mathworks.com/help/matlab/ref/conv.html.
[10] "Tukey (tapered cosine) window - MATLAB tukeywin - MathWorks United Kingdom." [Online]. Available: https://uk.mathworks.com/help/signal/ref/tukeywin.html.