Signal Processing – Lab 1 – Basic Signal Processing with MATLAB

Student name: Paul Doherty

Student number: 10387129

Introduction:

The following is a short report on my findings and observations from assignment one which asked me to create specific sine waves and experiment their behaviour using MATLAB.

Exercise 3: FFT Demonstration

The sine wave in exercise 3 requires 128 samples, amplitude of 5.0, frequency of 2kHz and is to be sampled at 8kHz. The following is a graph of “x”, the sine waves amplitude against Time.



“X”, the Fast Fourier Transform (FFT) of the sine wave was then computed using the fft() command. It is graphed below as amplitude vs. frequency.



This graph was achieved by use of the fftshift() command which moves the zero frequency component to the centre of the spectrum. This graph shows separately the imaginary part of the Fourier Transform (red) and the real part of the Fourier Transform (blue). This was done by using the “hold on” command.

Below is a graph of the FFT without using fftshift() command on the imaginary numbers. Notice how the fftshift() command swaps the 1st and 2nd quadrant.



A plot of the magnitude of “X” was also carried out. It is plotted below as amplitude vs. frequency.



This graph shows the peak amplitude at 2000Hz.

On changing the length of the wave to 127 samples, interesting results begin to emerge. Most notably is the graph of “X”.



The graph shows skewed results. This is due to a few reasons most notably spectral leakage. Spectral leakage is a result of frequency sampling. Because our sampling frequency isn’t a power of 2 the algorithm is not compiling efficiently. The most notable and inaccurate result from this sampling frequency (127) is that the FFT of the sine wave is not hitting is not hitting its expected peak. On close inspection of the graph of magnitude of “X” it can be noted that the FFT does not return a result at the frequency 2000Hz which corresponds to max amplitude.



This can also be verified on printing the magnitude of the FFT in the MATLAB command window.

On returning the sampling frequency to 128Hz and changing the frequency to 3kHz expected results are noted such as max amplitude is hit at 3kHz.

More interesting results are yielded when the frequency is set to 4kHz. This is because there are not enough samples to handle the high frequency of the wave to yield accurate results. These results will show an unexpected graph of the sine wave and the sine waves magnitude. . There is also aliasing present which is a result of half the sampling frequency (aka the Nyquist frequency) not being greater than the frequency. This causes distortion in the reconstructed signal. Below are examples of my results at frequency set to 4kHz.





Conclusion:

On these results I can say that in order calculate an accurate FFT the following precautions have to be taken:

* Number of samples should be a power of two for efficient use of algorithm
* The sampling frequency should be more than double the frequency of the wave being sampled (Nyquist)

Exercise 4: Zero Padding

The sine wave in exercise 4 requires 256 samples, frequency of 330.5Hz and is to be sampled at 1024Hz. I have chosen the amplitude of the wave to be 5.0. The FFT of this signal was found and below is a plot of its magnitude represented in discrete plots, plotted with amplitude against frequency.



The time vector was then zero padded to contain 2048 samples. The FFT was then found of the new zero padded vector and its magnitude was then plotted as a continuous line on the same axis as the original time vector. Below is a graph of this FFT plotted as amplitude vs. frequency.



Below is a “zoomed” graph of the region of the main spectral energy. It contains the original time vector whose samples are represented as discrete points and the zero padded vector whose samples are represented as a continuous function.



This graph shows the zero padded vector hitting the true amplitude and the original not. This is because the zero padded FFT contains more samples without actually affecting the signal.

The benefits of zero padding can be seen better when the original vector is not a power of 2 and the zero padding fills out the vector until it is a power of 2.

Conclusion:

Zero padding should be used to get a more accurate approximation of the FFT of a signal especially when original sample is not a power of 2 or a particularly low power of 2.