Notes on ECEN220 Signals and Systems Lab 1

Part 1

Plotting a continuous time (CT) sinewave

In part 1, we want to plot a continuous time (CT) signal $x(t) = \sin(2\pi f_0 t)$ with $f_0 = 4$ Hz.

Matlab plots a CT sine waveform by connecting up "dots", that is, the "sample points" of the original waveform, so Matlab sees $\sin(2\pi f_0 t)$ as sin(a) where a is a vector of our defined sample points.

$2\pi f_0$ is defined as rad/second, and if $f_0 = 1$ Hz, we expect to see one cycle of sine wave. Given that $f_0 = 4$ Hz, this means there will be 4 cycles of sine wave. Notice that once defined, $2\pi f_0$ becomes a constant. So, in order to set the number of points, we can define a vector of t, so that when multiplied with $2\pi f_0$, we will have a vector of $2\pi f_0 t$ that we can supply as the input argument to $\sin(2\pi f_0 t)$.

We must be careful to define enough points. If the points are not enough, the resulting sine wave will not look smooth, or worse still, we will get funny looking graph that does not look like a sine wave at all. For example, if you only set 10 points between 0 and 1 second (which is the time range that we want to see), there will not be enough points to get a proper sinewave. (Try it!). The more points that you have, the better looking sine wave that you will get, BUT it will take a longer time for Matlab to produce the plot for you (Try it!). So, you need to decide on a balance.

We can set the vector t as t = 0:dt:1 where dt is the time resolution which is the time difference between successive samples. For example, a dt of 0.5 means that the time difference between two samples is 0.5 second, and this will only give you 3 sample points from 0 to 1 second. (Why 3?) Imagine having only 3 sample points to plot 4 cycles of sine wave – this is obviously not going to work (Try and see the waveform!).

Plotting a discrete-time (DT) sinewave

After getting the CT sinewave, we want to obtain the DT version of it. This is done through a process known as "sampling". In sampling, we need to define our sampling frequency, for example, $f_s = bf_0$. Corresponding to the sampling frequency is the sampling period, $T_s = 1/f_s = 1/bf_0$, meaning to say that for every T_s, we will get one sample point. Remember that in DT, the discrete number of sample points is defined by an integer number of n, so this means there will be nT_s sample points in total. Putting into equation, we have

$x[n] = \sin(2\pi f_0 \mathbf{nTs})$ *instead of t.*

Substituting in $T_s = 1/f_s = 1/bf_0$, we have $x[n] = \sin(2\pi f_0 \mathbf{n/bf_0}) = \sin(2\pi \mathbf{n/b})$ and if we know the sampling frequency, we will know b so can put the value of b in.

To plot a DT sinewave, you need to define a vector of n; you also cannot use plot but have to use **stem**.

On Periodicity of DT signal

Now we know that $x[n] = \sin(2\pi \mathbf{n/b})$. We also know that a DT signal is periodic if $2\pi \mathbf{n/b} = 2\pi \mathbf{m}$. That is to say that the argument of $2\pi \mathbf{n/b}$ is equal to a multiple $\mathbf{m}$ of 2π. We can simplify the condition as saying that **n** = **mb** (note that we know the value **b**). So, by choosing an appropriate **m**, we can determine the fundamental period **N** of x[n] where **N** = **mb.**

Part 2

Test on Linearity

A system is said to be linear, if and only if, the output of the system to a linear combination of individual inputs, **is equal** to the linear combination of the outputs of the system to individual inputs.

For example, let's say we have 2 individual signals, m(t) and n(t). If we want to test a system A with this response A = $e^{(signal)}$, to see if it is a linear system, we can do the following:

Firstly, we find the output of the system to a linear combination of individual inputs, for example, A1 = $e^{(2*m(t)+3*n(t))}$.

Secondly, we find the linear combination of the outputs of the system to individual inputs, for example, A2 = $2*e^{m(t)} + 3*e^{n(t)}$.

Lastly, we compare A1and A2. The system A is linear if and only if A1 = A2.

Part 3

DT Convolution

a) We want to plot y[n] = x[n] convolve with h[n]. In Matlab, this is done by using y=conv(x,h). So first, you have to define your vectors of x and h according to the requirement. Then proceed to get the convolution, and plot the result.

b) To verify by hands, you can choose to use either the graphical method, or you can use purely analytical (ie mathematical) method, to arrive at the final answer. For how to use the analytical method, **read example 2.4** of textbook on page 85 for the hints => that example shares similarity to this question. Once you have obtain the results calculated by hands, plot both sets of results on the same graph to see if they agree with each other (They should).