Exercise 1 — EE6733 – Reading Course

Target Deadline: Wednesday May 15th, 2019

TIME DOMAIN REPRESENTATION OF SINUSOIDS

ALL SCRIPTS SHOULD APPLY THE CODING STANDARDS WE DISCUSS IN CLASS.
READABILITY, EFFICIENCY, MODULARLIZATION AND GENERALIZATION ARE IMPORTANT CONSIDERATIONS BEYOND FUNCTIONALITY

Sinusoidal signals are at the foundation of signal theory. They can be fully represented by their Amplitude (A), frequency (f_0) and phase (φ) and when they are expressed as a function of time, they are generally written as:

$$g(t) = A \cdot \cos(2\pi f_0 t + \varphi) \tag{1}$$

but can be also written as:

$$g(t) = A \cdot \cos(2\pi f_0 (t - t_0)) \tag{2}$$

Equations 1 and 2 are equivalent. Equating the argument in (1) with that in (2) simply yields the relationship between the shift in a sinusoid in terms of radians and the shift in terms of time.

Sinusoidal signals can also be written in terms of their in-phase and quadrature components:

$$g(t) = B \cdot sin(2\pi f_0 t) + C \cdot cos(2\pi f_0 t) \tag{3}$$

Equations 1, 2 and 3 are equivalent as long as B and C are chosen such that $A = \sqrt{B^2 + C^2}$ and $\varphi = tan^{-1}(C/B)$. This can easily be demonstrated mathematically with the following identity:

$$\sin(x + y) = \sin(x)\cos(y) + \cos(x)\sin(y) \tag{I1}$$

Useful functions for this assignment: `cos()`, `length()`, `size()`, `isempty()`, `eval()`, `plot()`, `subplot()`, `xlabel()`, `ylabel()`, `title()`, `text()`, `sqrt()`

Question 1: Create a matlab function called `makesin()` which generates a sinusoid (use a cosine). The function should set default parameters as delineated in table 1:

Amplitude (A):	1	mV
Frequency (fo):	60	Hz (cycles/second)
Phase (phi):	0	degrees
Duration (T):	1	Seconds
Sampling Rate (fs):	1000	Hz (samples/second)

Set up your function so that default values are used unless the user specifies a value for one by passing an argument. To do this, use `makesin(varargin)`. Make sure the user knows which order to pass the variables by setting a help line as follows:

```
%inputs:  A, fo, phi (in degrees), T, fs; use [] to skip a parameter
```

To test your function, write a script which uses the function to plot a 1 sec sinusoid signal against time. Use a sampling rate fs = 100 Hz, an Amplitude A = 5mV, a frequency f0 = 2Hz, and a phase, φ = 0 degrees. Make sure you plot the signal against time (you will need to create the time scale using `t=0:dt:T-dt` (where `dt = 1/fs`)).

Question 2: Write a script which uses your makesin() function to plot two sinusoid signals. The first signal should have the same parameters as the test parameters listed in question 1. The second signal should have the same parameters except for the phase. Shift this signal by 90 degrees. Plot both signals on the same graph. Plot the first signal with a blue solid line (`'b-'`). Plot the second signal with a red dashed line (`'r:'`).

Use the plot to estimate the shift in time (you can do this manually by inspection). Does this estimate coincide with what you would expect theoretically (calculate it mathematically)? Delineate the relationship between shift in degrees and time on the plot using text as follows:

```
Shift:  90 degrees, 0.125 seconds
```

Question 3: Create a matlab function called makesinQ() which generates a sinusoid (use a cosine) given in-phase and quadrature components. The function should set default parameters as delineated in table 1:

In-Phase Component (Aip):	$\frac{1}{2}$	mV
Quadrature Component (Aqu)	$\frac{\sqrt{3}}{2}$	mV
Frequency (fo):	60	Hz (cycles/second)
Duration (T):	1	Seconds
Sampling Rate (fs):	1000	Hz (samples/second)

Set up your function so that default values are used unless the user specifies a value for one by passing an argument. To do this, use `makesinQ(varargin)`. Make sure the user knows which order to pass the variables by setting a help line as follows:

```
%inputs:  Aip, Aqu, fo, T, fs; use [] to skip a parameter
```

To test your function, write a script which uses the function to plot a 1 sec sinusoid signal against time (use a blue solid line (`'b-'`)). Use a sampling rate fs = 100 Hz, and keep all other parameter values as defaults.

Then, superimposed on the same plot use the makesin() function you created in Question 1 to plot the same sinusoid based on A-f_0-φ parameters. To do this, you will have to calculate the input parameter values based on the relationship between them and the in-phase and quadrature components. Make sure you plot the signals against time (you will need to create the time scale using `t=0:dt:T-dt` (where `dt = 1/fs`)).