# Assignment #1(MATLAB) CE242 : Signals & Systems
# Dept. of Computer Engineering
# Sharif University of Technology
# Fall 2005

Distributed : 8/2 Due: 8/15

* *

---

1. Consider the following Matlab code, which generates a discrete-time sinusoidal signal x[n] and plots it:

```
%------------------------------------------------------------------
%
% Simple Matlab code to generate a discrete-time sinusoid and plot it.
%
n = 0:40; % array n holds the values of the time variable.
omega = 0.1*2*pi; % frequency of the sinusoid.
phase_offset = 0; % phase offset.
A = 1.5; % amplitude.
arg = omega*n - phase_offset; % note that "arg" is an array the same size as n.
xn = A * cos(arg); % values of the signal x[n].
stem(n,xn);
axis([0 40 -2 2]);
grid;
title('Discrete Time Sinusoid');
xlabel('n');
ylabel('x[n]');
```

(a) Type in this code and run it. You can type the commands in line-by-line or, you can use an editor to type the commands into a file. If you type them in line-by-line, then they are executed as soon as you type them. If you put them in a file, then it should have a name that ends with ".m"; in that case, the commands will be executed when you type the name of the file at the Matlab prompt (for example, if you name the file `P1a.m` and put it in the current directory, then the commands will be run when you type `P1a` at the Matlab prompt).

(b) What is the length of the signal x[n]? What is its period?

(c) What is the purpose of the `axis` command?

(d) What is the purpose of the `grid` command?

(e) Replace the `stem` command in the above code with the `plot` command and run the code again. How does this change the plot?

(f) Modify the code so that it generates a sinusoid with length 50, frequency $0.8 * 2\pi$ radians per sample, amplitude 2.5, and a phase offset of $\pi/2$ radians (use the `stem` command for this part). Run the modified code to generate another discrete-time sinusoid.

2- Write a MATLAB program to generate and display a delayed unit step sequence u[n] with a delay of 11 samples.

3. Let

$x[n] = \delta[n] + 2\delta[n-1] - \delta[n-3] + 3\delta[n-4] - \delta[n-5]$

and

$h[n] = 2\,\delta[n] + 2\,\delta[n-2]$:

(a) Write Matlab code to compute and plot the following convolutions: $y_1[n] = x[n] * h[n]$ and $y_2[n] = x[n-2] * h[n]$.

- ➔Plot each convolution result for $0 =< n =< 9$.
- ➔Use the `subplot` command to plot both $y_1[n]$ and $y_2[n]$ in a single figure.
- ➔For each result, use the `stem` command to do the plotting.
- ➔Label each plot with the time index $n$ and the name of the result signal.

**Hint:** The convolution operation $y[n] = x[n] * h[n]$ can be implemented in Matlab using the command `conv`, provided that the two signals to be convolved are both of finite length. For example, the Matlab code below shows how to compute and plot the output of an FIR system with impulse response
$h[n] = 3\delta[n] + 2\,\delta[n-1] + \delta[n-2] - 2\,\delta[n-3] + \delta[n-4] - 4\,\delta[n-6] + 3\,\delta[n-8]$:

Ex:

```
h = [3 2 1 -2 1 0 -4 0 3]; % impulse response
x = [1 -2 3 -4 3 2 1]; % input signal
y = conv(h,x);
n = 0:14;
subplot(2,1,1);
stem(n,y);
```

(b) Based on the plots from part (a), would you guess that the system is *time invariant* or *time varying*?