



Independent University Bangladesh

Department of Electrical and Electronics Engineering

Lab Report 01

Name: Injamamul Haque Sourov

Id: 1820170

Course code: EEE 321L

Couse name: Digital Signal Processing Lab

Lab no: 01

Lab title: Introduction to EEE321L

Date: 11/11/2020

a) Generating sinusoidal signals with phase shifts

Code:

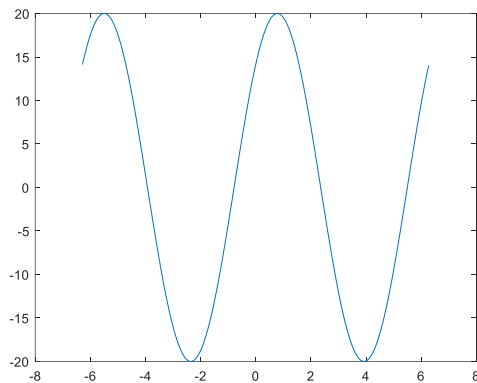
```
% define signals
x = -2*pi:0.01:2*pi;    % specify domain
y = 20*sin(x+pi/4);     % leading waveform
z = 20*sin(x-pi/4);     % lagging waveform

% plot(independent, dependent var)
plot(x,y)               % plot of y wrt x
plot(x,z)               % plot of z wrt x
plot(x,y,x,z)           % plot both in same graph
grid                    % toggles grid
```

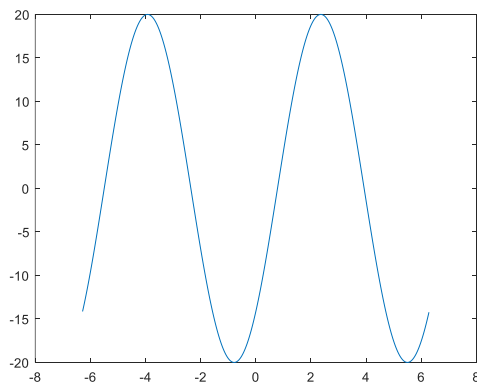
^ Plots were obtained one after another, utilizing the same figure window;

Outputs:

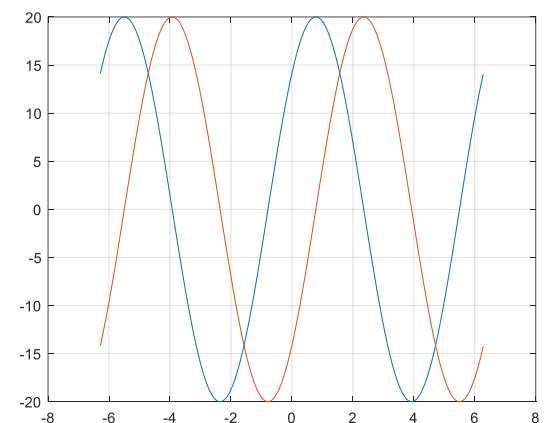
i. Plot of y with respect to x



ii. Plot of z with respect to x



iii. Plot of both curves in same graph

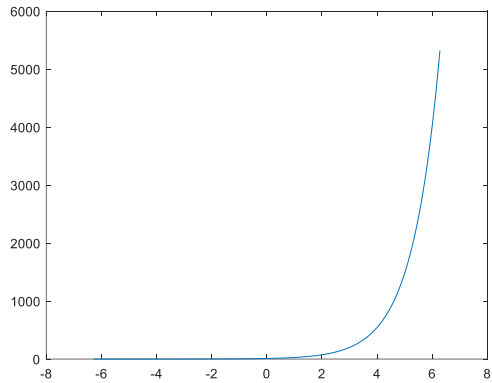


b) Generating exponential signals

Code:

```
% exponential signal
expo = 10*exp(x);          % exp = exponetial function
plot(x, expo)
```

Output:



c) Working with complex variable (i)

Code:

```
% operations with complex numbers
a = 3+4i;
b = 3-8i;
c = a+b
```

Output:

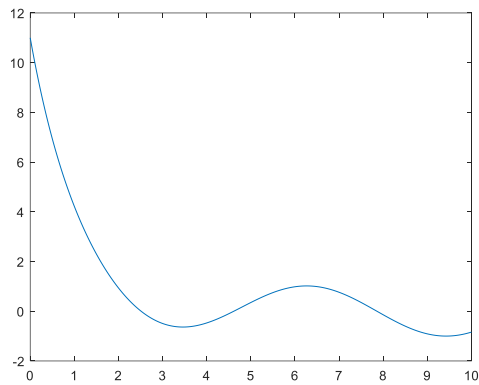
```
c =
    6.0000 - 4.0000i
```

d) Combining multiple signals

Code:

```
% combining cosine and exponetial signal
x = 0:0.01:10;
comb = 10*exp(-x) + cos(x);
plot(x, comb)
```

Output:

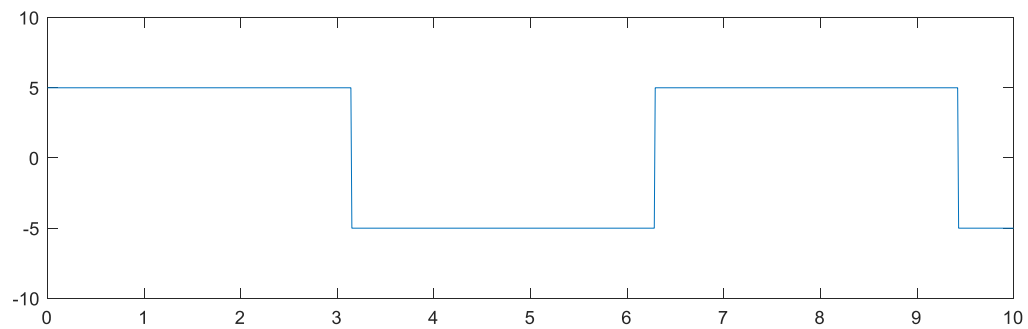


e) Generating square wave

Code:

```
% square wave  
x = 0:0.01:10;  
y = 5*square(x);  
plot(x,y)
```

Output:

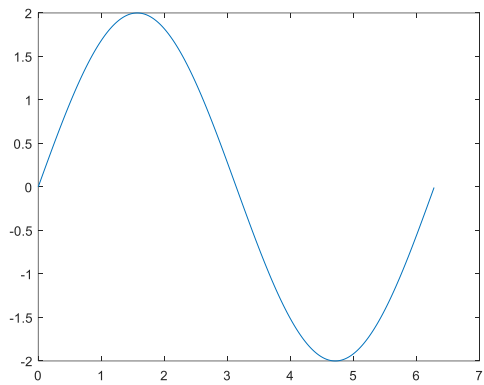


f) Periodic signals

Code:

```
% periodic signal  
t = 0:0.01:2*pi;  
x = 2*sin(t);  
plot(t,x)
```

Output:



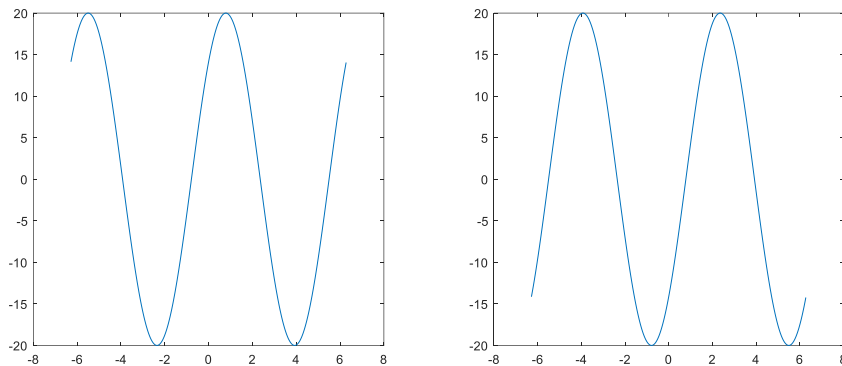
g) Sub-plotting signals

Code:

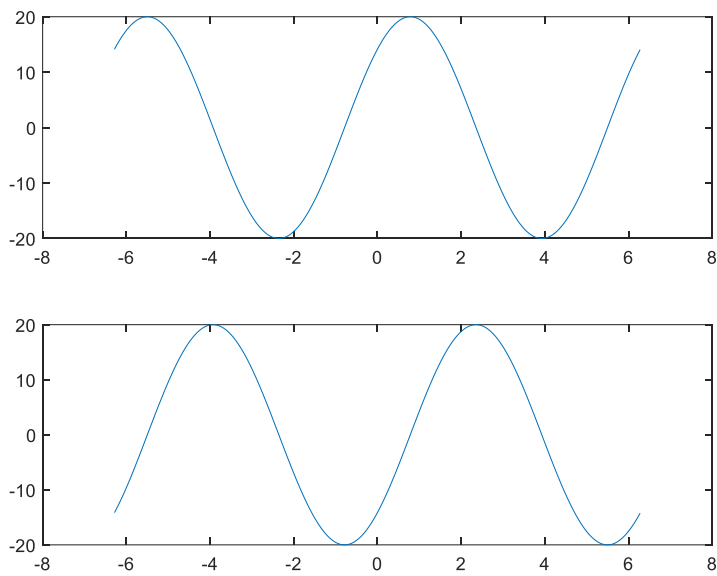
```
% subplots
x = -2*pi:0.01:2*pi;
y = 20*sin(x+pi/4);
z = 20*sin(x-pi/4);
% side by side (1 row, 2 col)
figure(1)           % open figure window
subplot(1,2,1)      % (row, col, handle)
plot(x,y)
subplot(1,2,2)
plot(x,z)
% one above other (2 row, 1 col)
figure(2)           % open figure window
subplot(2,1,1)      % (row, col, handle)
plot(x,y)
subplot(2,1,2)
plot(x,z)
```

Output:

i. Side by side plots (1 row, 2 columns)



ii. One above the other (2 rows, 1 column)



h) Generating random matrices (signals)

Code:

```
% random signal
rand(4)           % random 4x4 matrix with elements between 0 to 1
randn(4)          % negative included
rand(1,5)         % 1 row, 5 col, 0 to 1
```

Output:

- i. Random 4x4 matrix with values between 0 and 1

ans =

0.3171	0.3816	0.4898	0.7547
0.9502	0.7655	0.4456	0.2760
0.0344	0.7952	0.6463	0.6797
0.4387	0.1869	0.7094	0.6551

- ii. Random 4x4 matrix with where element can be negative

ans =

-1.2141	-0.7697	-1.0891	1.5442
-1.1135	0.3714	0.0326	0.0859
-0.0068	-0.2256	0.5525	-1.4916
1.5326	1.1174	1.1006	-0.7423

- iii. Random 5x1 matrix with elements between 0 and 1

ans =

0.2575	0.8407	0.2543	0.8143	0.2435
--------	--------	--------	--------	--------

- i) Defining and calling functions (defined as separate scripts)

- i. Function to add two numbers

Definition: [addition.m]

```
% function to add two numbers
function [z] = addition(x,y)    % function [output] = name(comma_seperated_inputs)
z = x + y;                    % can only have general expressions
end                            % completes loop
```

Call and output:

```
>> % function call
      addition(5,10)

ans =

      15
```

ii. (Assignment) Function to calculate factorial

Definition: [factorial2.m]

```
% function to calculate factorial
function [f] = factorial2(a)
    f = 1; % default value
    while a > 1
        f = f*a;
        a = a-1;
    end
end
```

Calls and outputs:

```
>> % assignment: create a function for factorial
      factorial2(0)

ans =

      1

>> factorial2(1)

ans =

      1

>> factorial2(5)

ans =

     120
```