***DSP LAB***

***Experiment 1 (PLOTTING)***

x = [0:0.1:10];

y = sin (x);

z = cos (x);

subplot (3,1,1);

plot (x,y);

grid on;

subplot (3,1,2);

plot (x,z);

grid on;

hold on;

subplot (3,1,3);

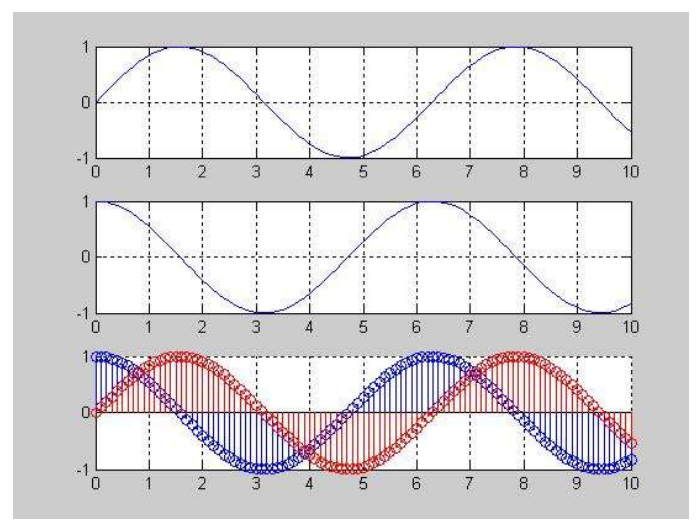
stem (x,z);

grid on;

hold on;

subplot (3,1,3);

stem (x,y, ,'r');



***Experiment 2 (Generating a Signal)***

% Generation of discrete time signals

% 2sin(2πτ-π/2)

T = [-5:0.01:5];

x=2\*sin((2\*pi\*t) - (pi/2));

plot(t,x)

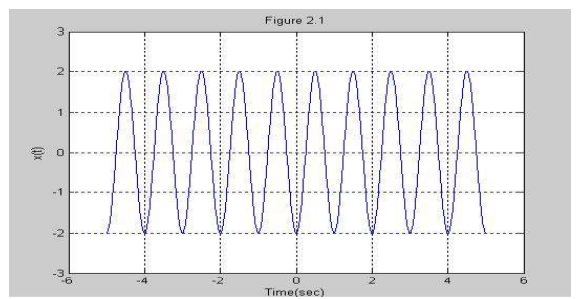
grid on;

axis ([-6 6 -3 3])

ylabel ('x(t)')

xlabel ('Time(sec)')

title ('Figure 2.1')



***Experiment 3 (Generating a Signal)***

% Generation of discrete time signals

n = [-5:5];

x = [0 0 1 1 -1 0 2 -2 3 0 -1];

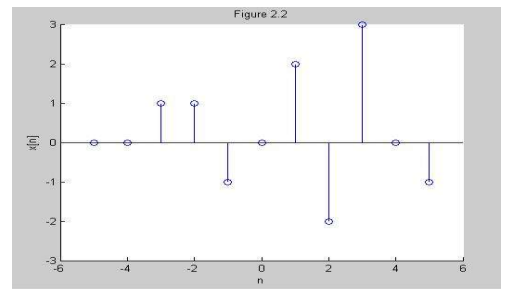
stem (n,x);

axis ([-6 6 -3 3]);

xlabel ('n'); ylabel

('x[n]'); title

('Figure 2.2');



***Experiment 4 (Generating a Signal)***

%Generation of random sequence

n = [0:10];

x = rand (1, length (n));

y = randn (1, length (n));

plot (n,x) ;

grid on;

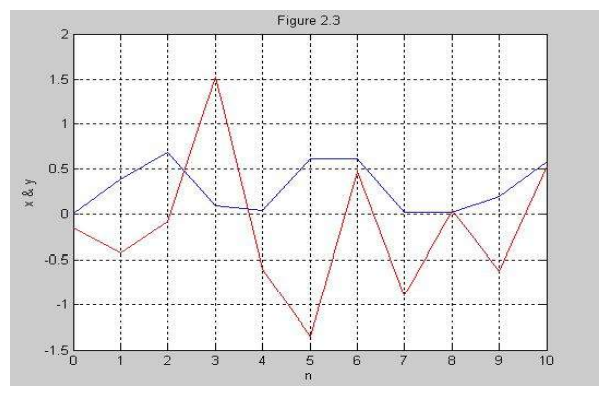
hold on;

plot(n,y,'r');

ylabel ('x & y')

xlabel ('n')

title ('Figure 2.3');



***Experiment 5 (Generating a discrete periodic signal Signal)***

n = [0:4];

x = [1 1 2 -1 0];

subplot (2,1,1);

stem (n,x);

grid on;

axis ([0 14 -1 2]);

xlabel ('n');

ylabel ('x(n)');

title ('Figure 2.4(a)');

xtilde = [x,x,x];

length\_xtilde = length (xtilde);

n\_new = [0:length\_xtilde-1];

subplot (2,1,2);

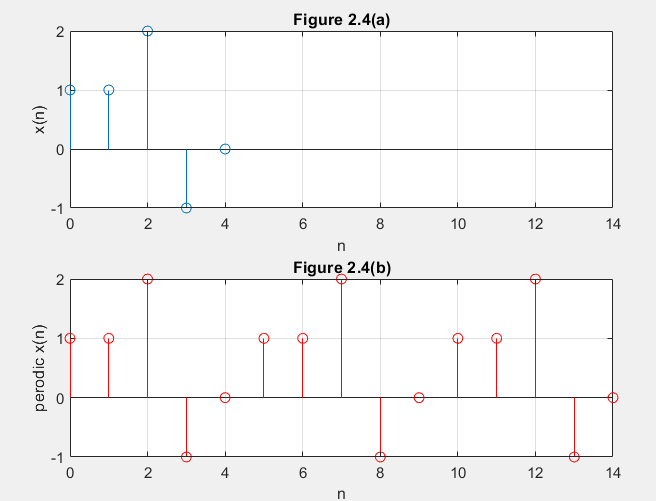
stem (n\_new, xtilde,'r');

grid on;

xlabel ('n');

ylabel ('perodic x(n)');

title ('Figure 2.4(b)');



***Experiment 6 (Generating Square wave) using loop***

clear;

clc;

n = input ('Insert the value of odd n:');

t = 0:.001:1;

sum=0;

for f=1:2:n

w=sin(2\*pi\*f\*t);

sum=sum+w;

end

subplot(1,1,1)

plot(t,sum)

grid on;

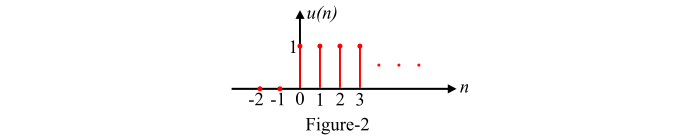
***Experiment 7 (Generating Unit Step Discrete Time Signal)***

**Experiment Name:** Generating and Plotting Unit Step Discrete Time Signal.

**Discrete Time Unit Step Signal:**

It is denoted by u[n]. Mathematically, the discrete-time unit step signal or sequence u[n] is defined as follows –

The graphical representation of the discrete-time unit step signal u[n] is shown in the following figure:



%Generating and Plotting Unit Step Discrete Time Signal.

clc; %clears the command window

clear all; %clears the current variables which are being used

close all; %close programs that are running behind in MATLAB

N=input('Enter the range: ');

n=-N:1:N;

y= [zeros(1,N),1,ones(1,N)];

stem(n,y);

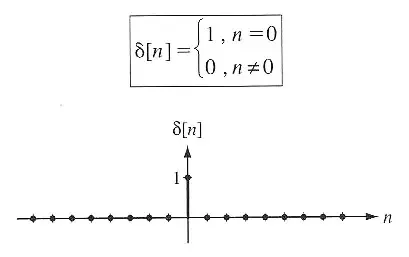
axis([-(N+1) N+1 -0.5 1.5]); % [-x x -y y]

xlabel('Time');

ylabel('Amplitude of Y');

title('Generating Unit Step Function');

***Experiment 8 :*** Unit impulse signal is mathematically defined as,



Code of implementation of impulse signal in Matlab:

%Code of implementation of impulse signal in Matlab:

clc;

clear all;

m1=input('enter the value of x-axis in negative side:');

m2=input('enter the value of x-axis in positive side:');

n=m1:m2;

x=(n==0);%it works as if statement like n=-5:5( 0 0 0 0 0 1 0 0 0 0 0 0)

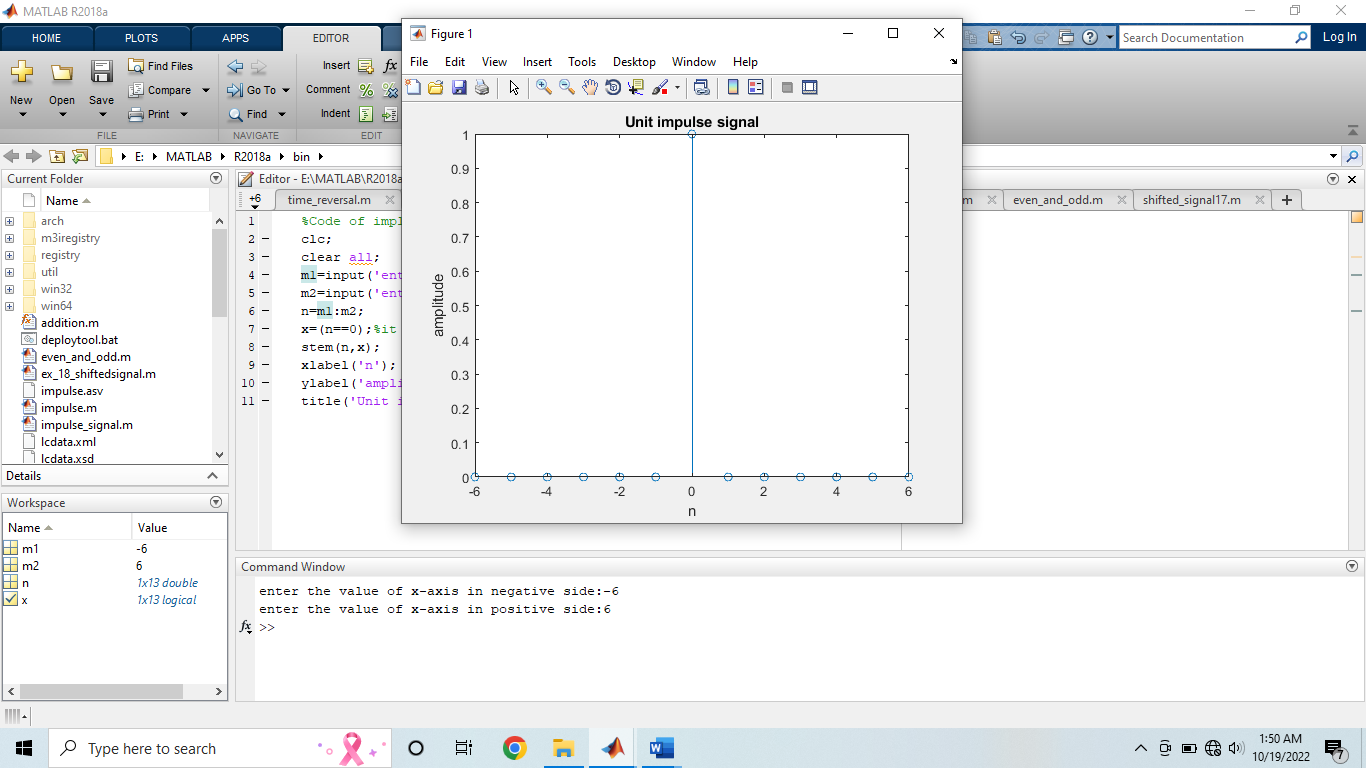
stem(n,x);

xlabel('n');

ylabel('amplitude');

title('Unit impulse signal');

**Output:**

****

The unit impulse can be implemented in different way:

clc;

clear all;

close all;

m1=input('enter the value of x-axis in negative side:');

m2=input('enter the value of x-axis in positive side:');

n=-m1:m2;

d=[zeros(1,m1) 1 zeros(1,m2)];

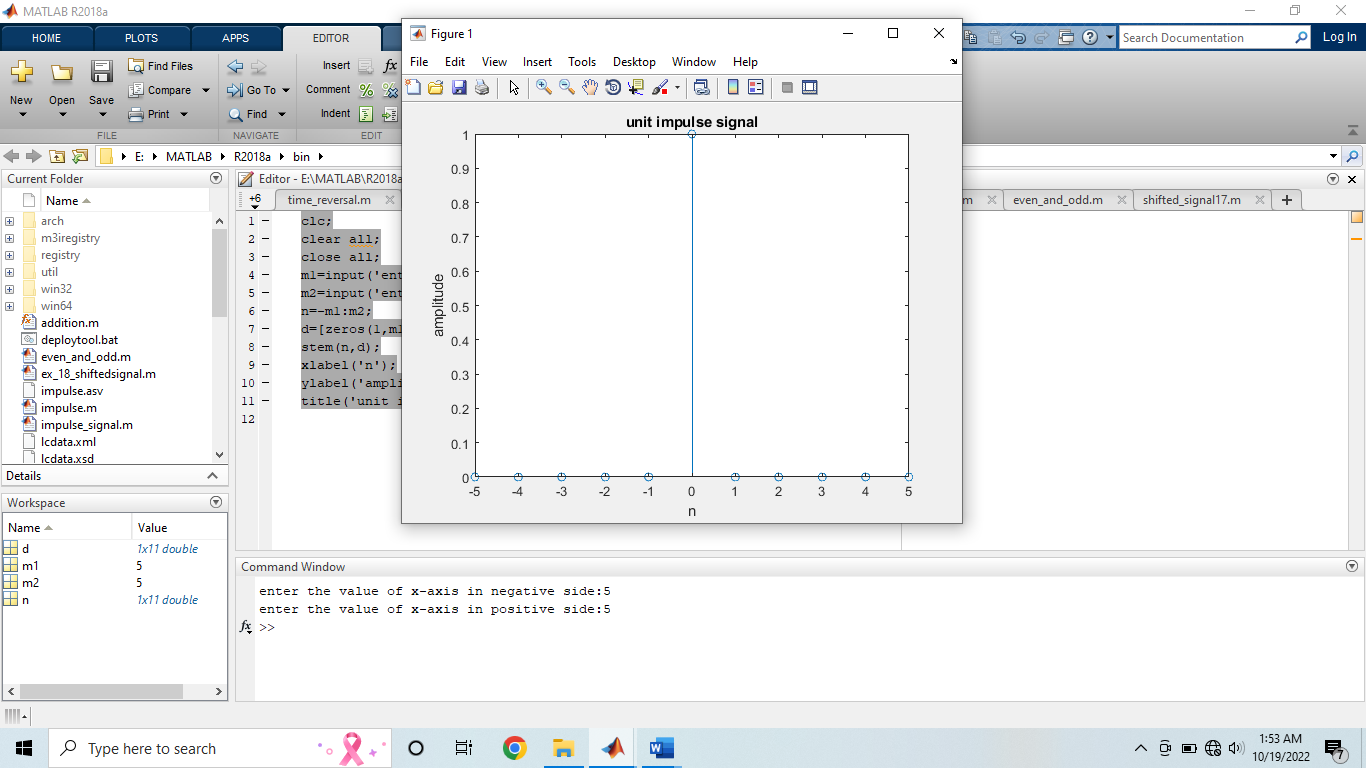
stem(n,d);

xlabel('n');

ylabel('amplitude');

title('unit impulse signal');

**Output:**

****

Experiment 9

Generating and plotting ramp discrete time signal.

The **discrete time unit ramp signal** is that function which starts from n = 0 and increases linearly. It is denoted by r(n). It is signal whose amplitude varies linearly with time n. **mathematically**; the discrete time unit ramp sequence is defined as –

**Experiment 10 (**Time reversal using a discrete sinusoidal function [use of fliplr( ) and values of x-axis(angle) in radian)

%Time reversal using a function (sinusoidal function angle in radian)

close all

clc

t1=0:0.2:2\*pi; %values of x-axis in radian

x1=sin(t1); %values of y-axis

x2=fliplr(x1); %fliplr() -> this function gives the flipped result;

%lr means left right ...flipud() ud means up down

t2= -fliplr(t1); % time values must be flipped and negated

subplot(2,1,1)

stem(t1,x1,'LineWidth',2)

xlim([-10 10])

title('\bf\fontsize{25}Original Signal')

xlabel('\bf\fontsize{20}Samples')

ylabel('\bf\fontsize{20}Amplitude')

grid on;

ax = gca;

ax.XAxis.FontSize = 15;

ax.XAxis.FontWeight = 'bold';

ax.YAxis.FontSize = 15;

ax.YAxis.FontWeight = 'bold';

subplot(212)

stem(t2,x2,'LineWidth',2)

xlim([-10 10])

title('\bf\fontsize{25}Time Reversed Signal')

xlabel('\bf\fontsize{20}Samples')

ylabel('\bf\fontsize{20}Amplitude')

grid on;

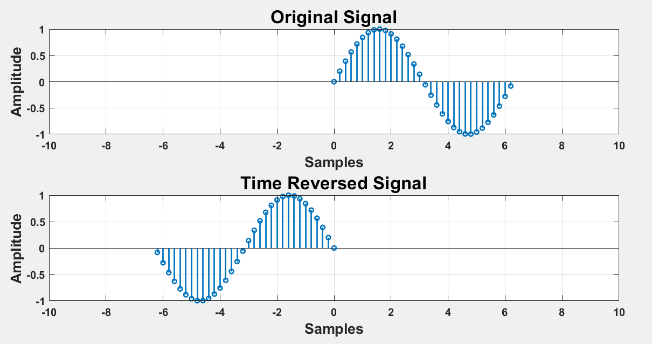
ax = gca;

ax.XAxis.FontSize = 15;

ax.XAxis.FontWeight = 'bold';

ax.YAxis.FontSize = 15;

ax.YAxis.FontWeight = 'bold';



**Experiment 11** Time reversal using a discrete sinusoidal function [use of fliplr()and values of x-axis(angle) in degree]

(Take the concept of experiment 9)