Pandit Deendayal Petroleum Univerisity

Digital Signal Processing (19EE406P)

Ankit Bandyopadhyay
17BEE005

Lab Manual

Course Coordinator: Dr. Alok Jain

Table of Contents

Sr no.	Aim of the experiment	**Page no.**
1	To generate discrete sequence using software tool	1
2	To study and generate different operation of signal	8
3	To study and generate the signal in the form of CTS (Continuous time signal) & DTS (Discrete Time signal)	15
4	To find out the convolution of sequences of two given sequences and also by using properties.	32
5	To find out Cross co-relation of the given sequence.	39
6	To study the sampling theorem and to find out Nyquist rate and interval of the given signals.	41
7	To find out the convolution, Z-Transform and inverse Z-Transform of the given signals and also to find out poles, zeros and gain from Z-Transform	44
8	Design of FIR filter using window function	51
9	Design the low and high pass filter with the transfer function and cut-off frequency and also draw its characteristics bode plot and root locus	57
10	Designing of IIR Butterworth low and high pass Filter	61

EXPERIMENT NO. 01

Aim: To generate discrete sequence using software tool.

1. sin (x)
2. cos (x)
3. sin (-x)
4. cos (-x)
5. x (n), x(n/2), x(2n)
6. x (n-2)
7. x (n+2)
8. x (-n)
9. 3D plot of cos (x) and sin (x)

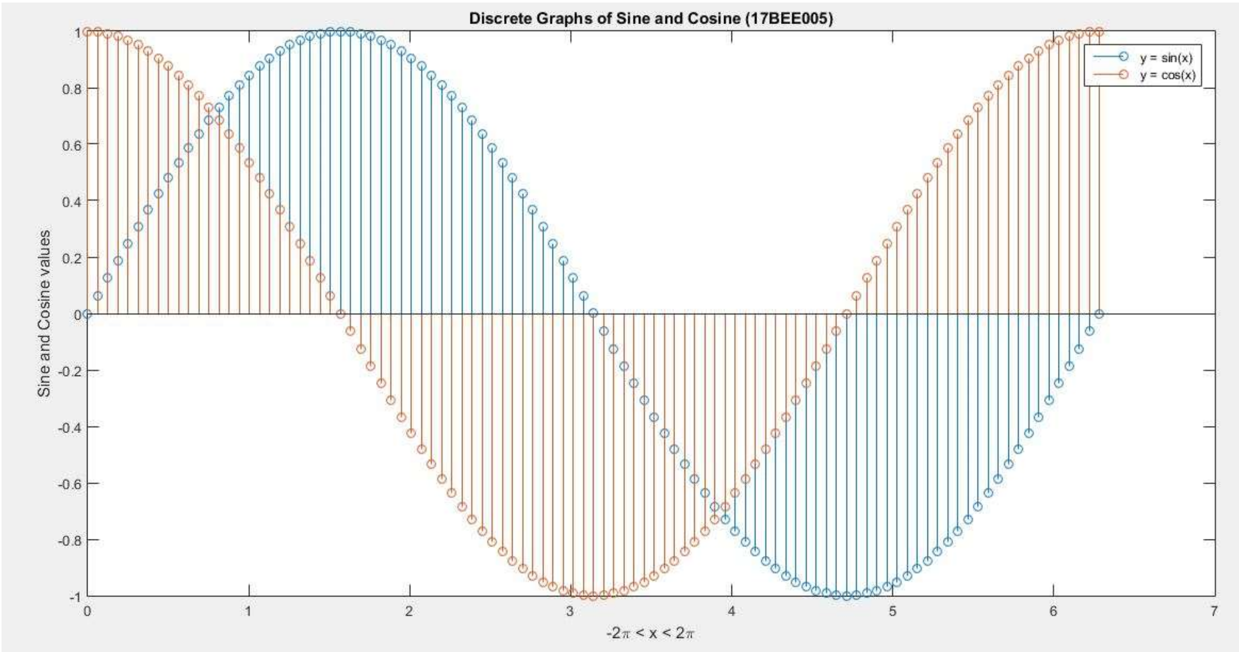
- **sin(x) and cos(x) :**

Code:

% Generation of sin(x) and cos(x)

```
x=[0:pi/50:2*pi]
y=sin(x)
z=cos(x)
figure, stem(x,y)
hold on
stem (x,z)
title('Discrete Graphs of Sine and Cosine (17BEE005)')
xlabel('-2\pi < x < 2\pi')
ylabel('Sine and Cosine values')
legend('y = sin(x)','y = cos(x)')
```

Figure:



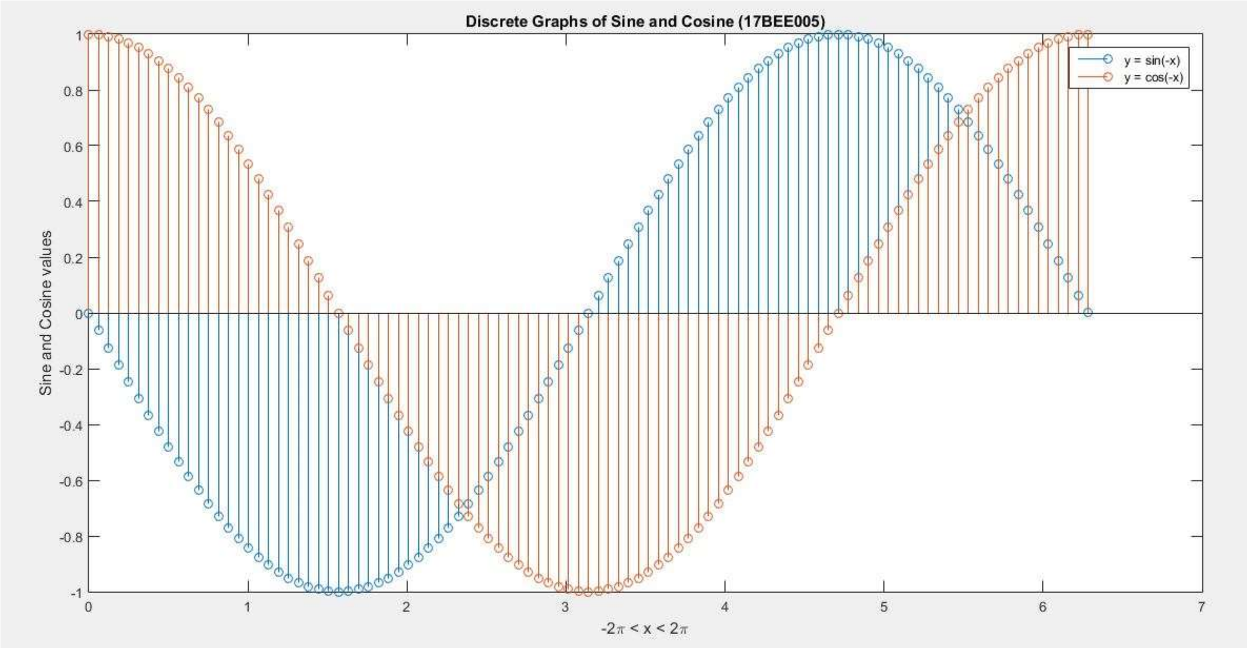
- **sin(-x) and cos(-x):**

  Code:

  % Generation of sin(-x) and cos(-x)

```
x=[0:pi/50:2*pi]
y=sin(-x)
z=cos(-x)
figure, stem(x,y)
hold on
stem (x,z)
title('Discrete Graphs of Sine and Cosine (17BEE005)')
xlabel('-2\pi < x < 2\pi')
ylabel('Sine and Cosine values')
legend('y = sin(-x)','y = cos(-x)')
```

Figure:



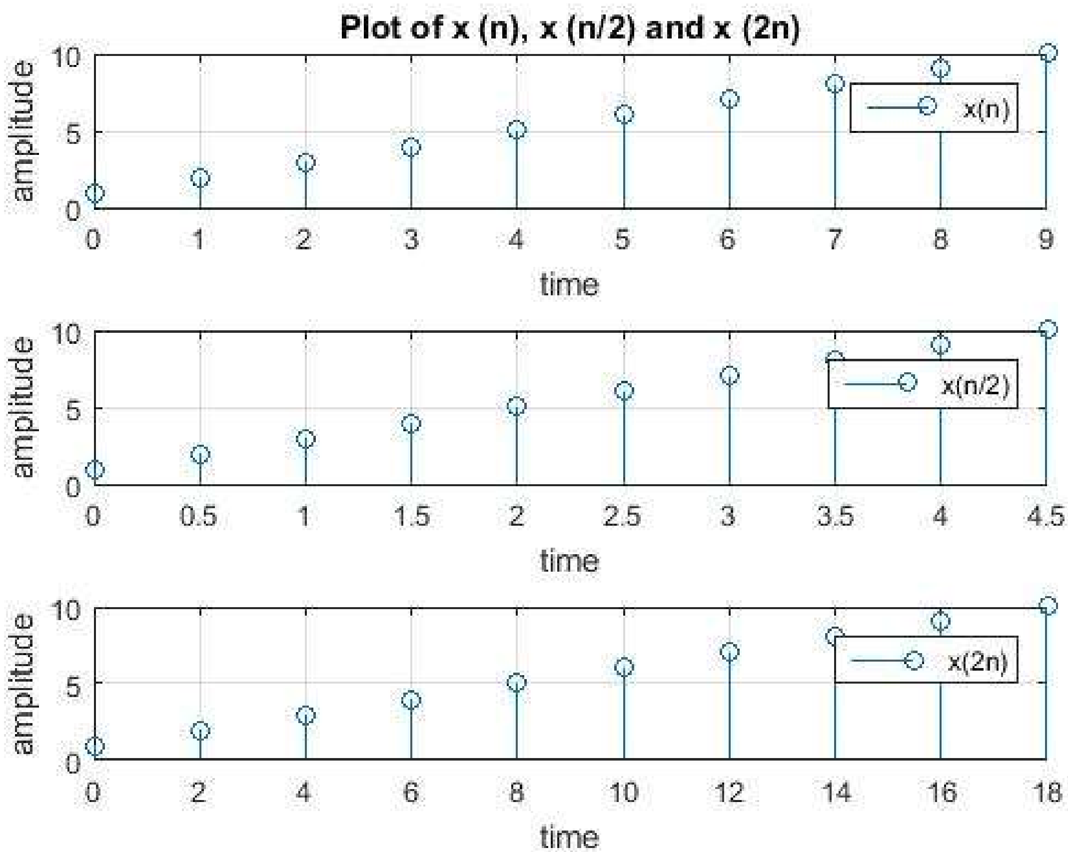
- **x(n), x(n/2), x(2n):**

  Code:

```
%% Generation of x(n), x(n/2), x(2n)

clc;
clear all;
close all;
n=0:9;
x=1:10;
subplot(3,1,1)
stem(n,x)
legend('x(n)')
grid on
title('Plot of x (n), x (n/2) and x (2n)')
xlabel('time')
ylabel('amplitude')
subplot(3,1,2)
stem(n/2,x)
legend('x(n/2)')
grid on
xlabel('time')
ylabel('amplitude')
subplot(3,1,3)
```

```
stem(2*n,x)
legend('x(2n)')
grid on
xlabel('time')
ylabel('amplitude')
```

Figure:

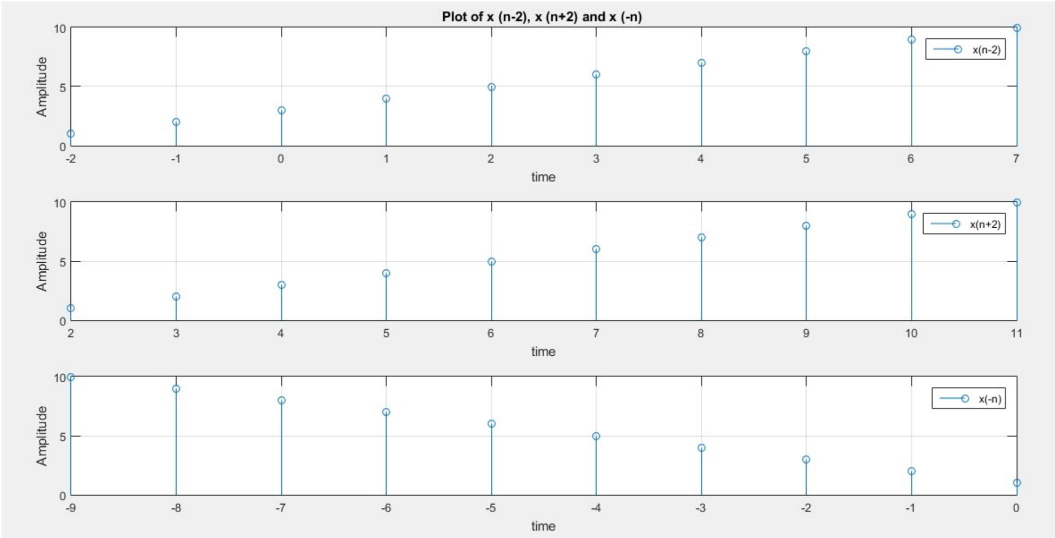


- **x(n-2),x(n+2),x(-n):**

  Code:

```
%% Generation of x(n-2), x(n+2), x(-n)

clc;
clear all;
close all;
n=0:9;
x=1:10;
subplot(3,1,1)
stem(n-2,x)
legend('x(n-2)')
grid on
title('Plot of x (n-2), x (n+2) and x (-n) ')
xlabel('time')
ylabel('Amplitude')
subplot(3,1,2)
stem(n+2,x)
legend('x(n+2)')
grid on
xlabel('time')
ylabel('Amplitude')
subplot(3,1,3)
stem(-n,x)
legend('x(-n)')
grid on
xlabel('time')
ylabel('Amplitude')
```

Figure:

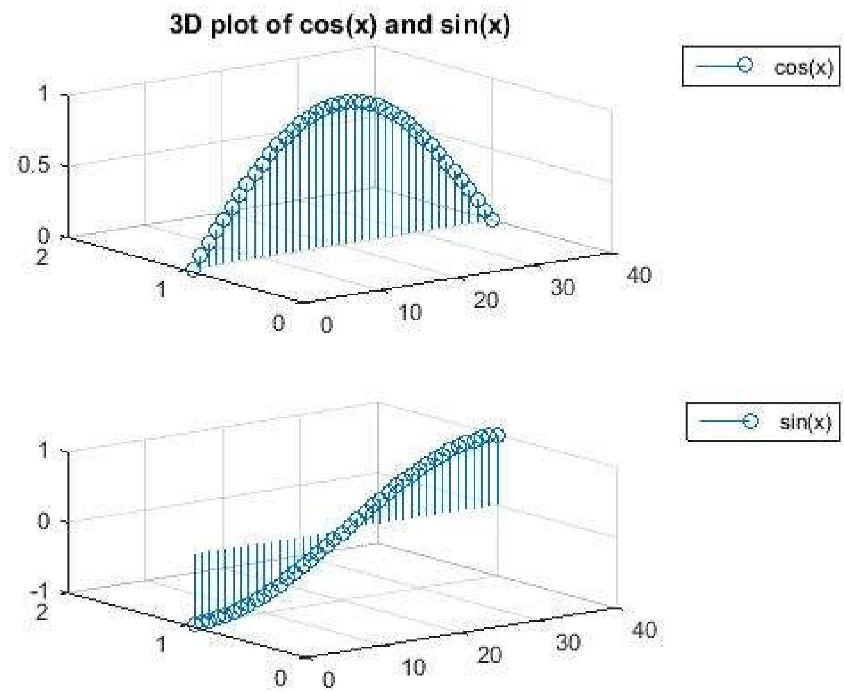


- **3D plots of sin(x) and cos(x):**

Code:

```
%Generation of 3D plots of sin(x) and cos(x)

clc;
clear all;
close all;
x=linspace(-pi/2,pi/2,40);
y=cos(x);
subplot(2,1,1)
stem3(y)
legend('cos(x)')
title('3D plot of cos(x) and sin(x)')
z=sin(x);
subplot(2,1,2)
stem3(z)
legend('sin(x)')
```

Figure:


3D plot of cos(x) and sin(x)
cos(x)
1
0.5
0
2
1
0
0
10
20
30
40
sin(x)
1
0
-1
2
1
0
0
10
20
30
40

EXPERIMENT NO. 02

Aim: To study and generate different types of signal.

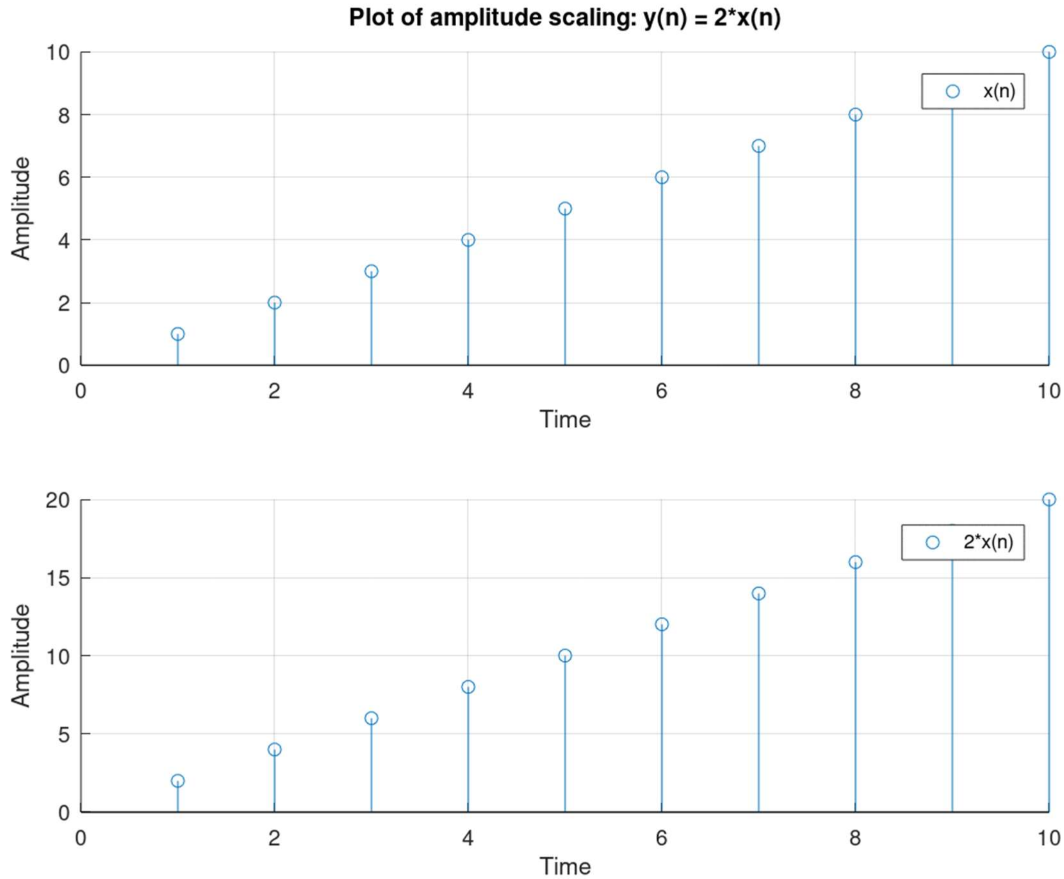
1. Amplitude scaling: y(n) = 2*x(n)
2. Time scaling: y(n) = x(2*n)
3. Time Shifting: y(n) = x(n+2)
   y(n) = x(n-2)
4. Time reversal: y(n) = x(-n)

- **Amplitude scaling: y(n) = 2*x(n):**

Code:

```
%Generation of signal using amplitude scaling
y(n) = 2*x(n)
clc;
clear all;
close all;
n=1:10;
x=1:10;
subplot(2,1,1)
stem(n,x)
legend('x(n)');
title('Plot of amplitude scaling: y(n) = 2*x(n)');
xlabel('Time')
ylabel('Amplitude')
subplot(2,1,2)
stem(n,2*x)
legend('2*x(n)')
xlabel('Time')
ylabel('Amplitude')
```

Figure:



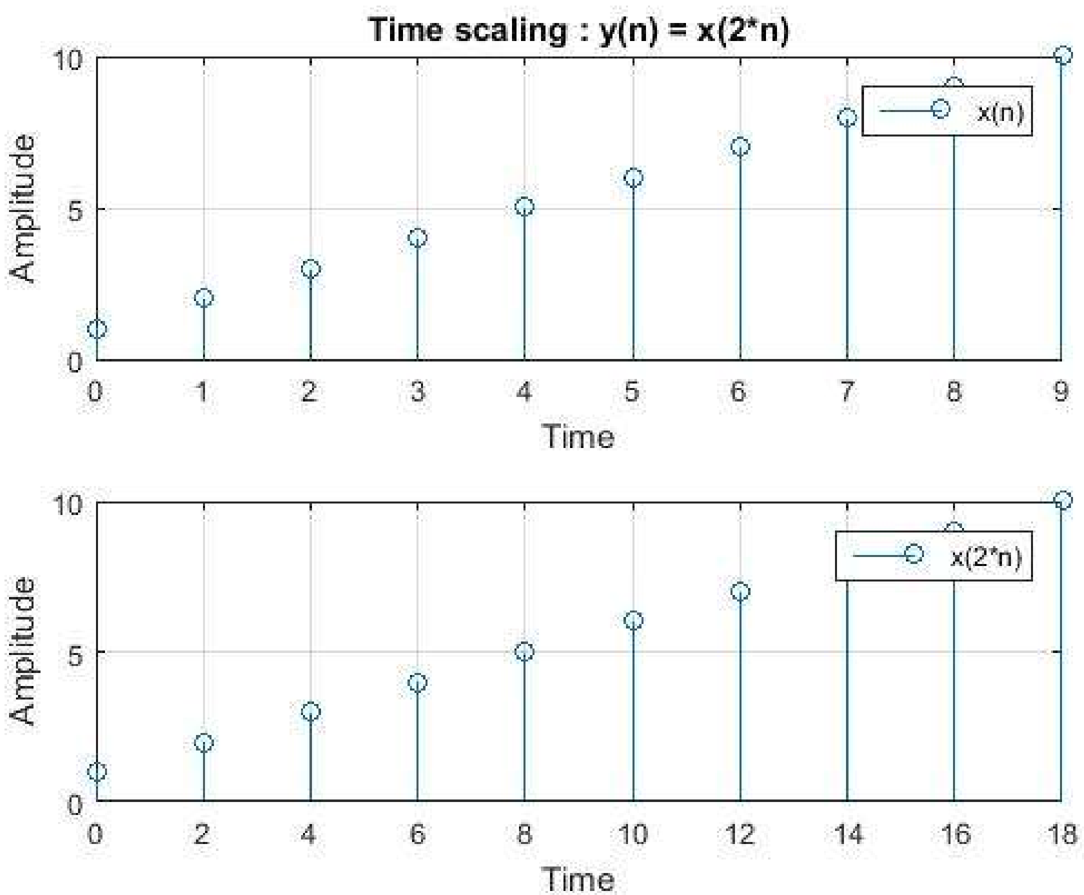
- **Time scaling: y(n) = x(2*n):**

Code:

```
%Generation of signal using Time scaling:
y(n) = x(2*n)

clc;
clear all;
close all;
n=1:10;
x=1:10;
```

```
subplot(2,1,1)
stem(n,x)
legend('x(n)')
title('Plot of amplitude scaling: y(n) = 2*x(n)')
xlabel('Time')
ylabel('Amplitude')

subplot(2,1,2)
stem(2*n,x)
legend('x(2*n)')
xlabel('Time')
ylabel('Amplitude')
```

Figure:

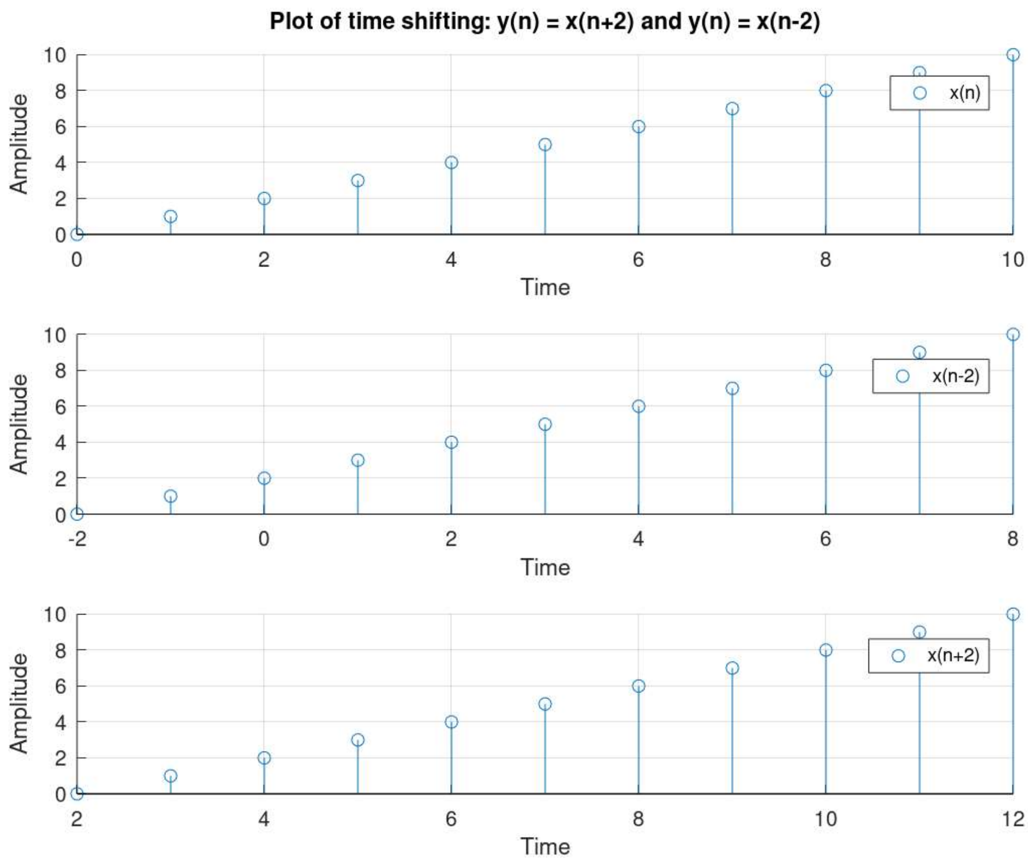


- **Time shifting: y(n) = x(n-2)**
  y(n) = x(n+2)

  Code:

```
%Generation of signal using amplitude scaling

clc;
clear all;
close all;
n=0:10;
x=0:10;

subplot(3,1,1)
stem(n,x)
legend('x(n)')
title('Plot of amplitude scaling: y(n) = x(n+2) and
y(n) = x(n-2)')
xlabel('Time')
ylabel('Amplitude')

subplot(3,1,2)
stem(n-2,x)
legend('x(n-2)')
xlabel('Time')
ylabel('Amplitude')

subplot(3,1,3)
stem(n+2,x)
legend('x(n+2)')
xlabel('Time')
ylabel('Amplitude')
```

Figure:

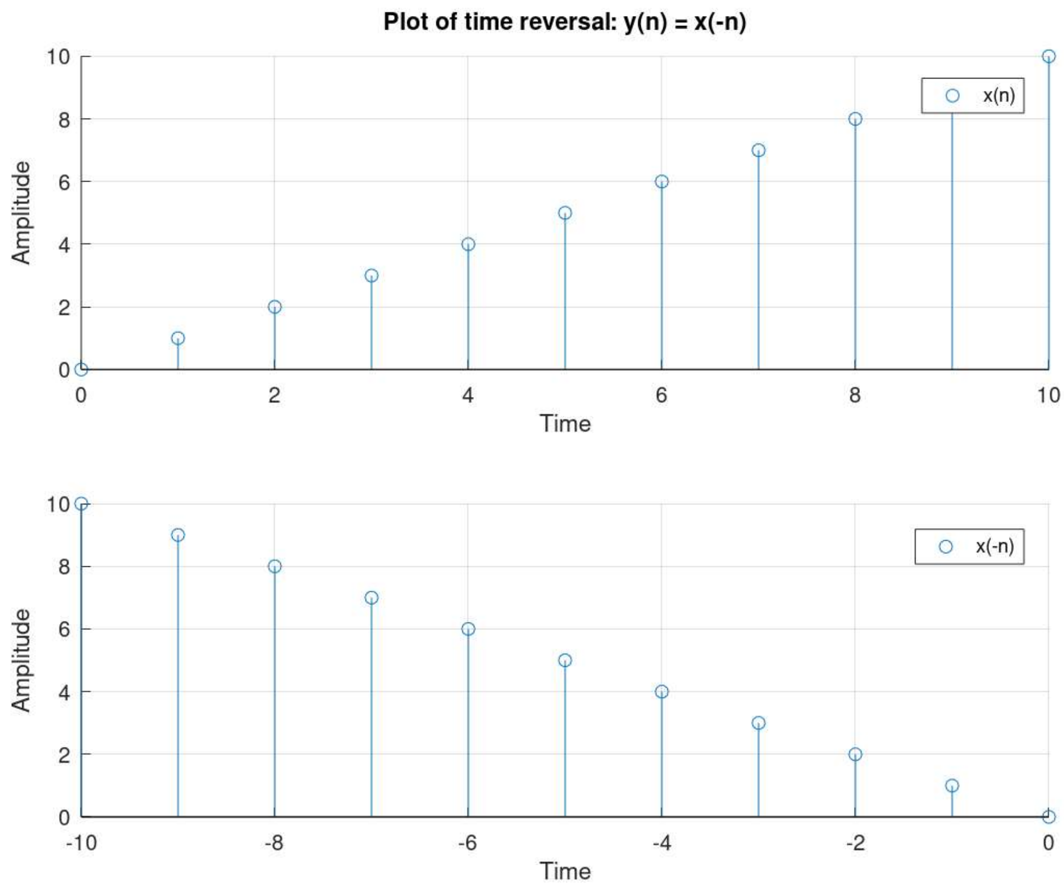

Plot of time shifting: y(n) = x(n+2) and y(n) = x(n-2)
Amplitude
Time
x(n)
x(n-2)
x(n+2)

- **Time reversal: y(n) = x(-n):**

  Code:

```
%Generation of signal using time reversal:
y(n) = x(-n)

clc;
clear all;
close all;
n=0:10;
x=0:10;

subplot(2,1,1)
stem(n,x)
legend('x(n)')
title('Plot of time reversal: y(n) = x(-n)')
xlabel('Time')
ylabel('Amplitude')

subplot(2,1,2)
stem(-n,x)
legend('x(-n)')
xlabel('Time')
ylabel('Amplitude')
```

Figure:


Plot of time reversal: y(n) = x(-n)
x(n)
Amplitude
Time
x(-n)
Amplitude
Time

EXPERIMENT NO. 03

Aim: To generate various types of signals.

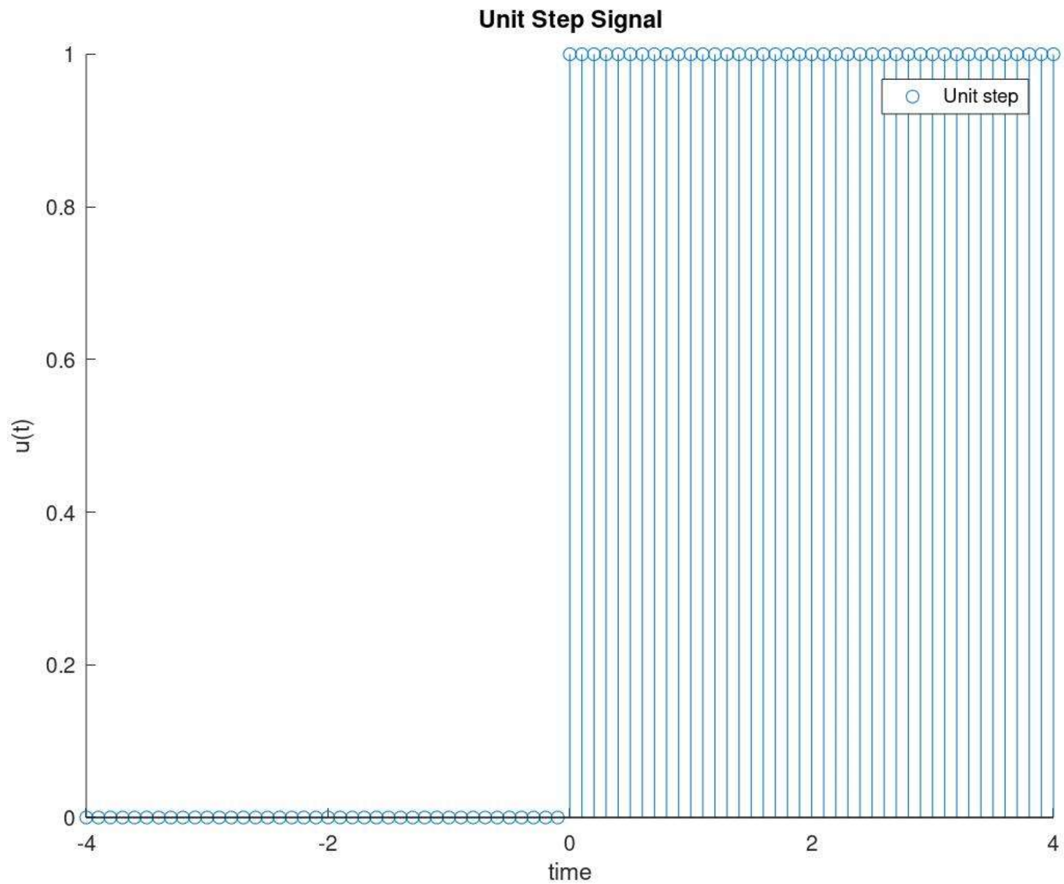
1. Unit step signal
2. Impulse signal
3. Ramp signal (both continuous and discrete)
4. Square wave signal (both continuous and discrete)
5. Exponential signal (both continuous and discrete)
6. Triangular signal (both continuous and discrete)
7. Seesaw signal
8. Sigmoid signal

- **Unit step signal:**

  Code:

```
%% Generation of unit step signal

clear all;
clc;

j=0;

for i=-4:0.1:4
j=j+1;
t(j)=i;

if j<42
  u(j)=0;

else u(j)=1;

endif

endfor

plot(t,u);
title('Unit Step Signal');
xlabel('time');
ylabel('u(t)');
legend('Unit step')
```

Figure:

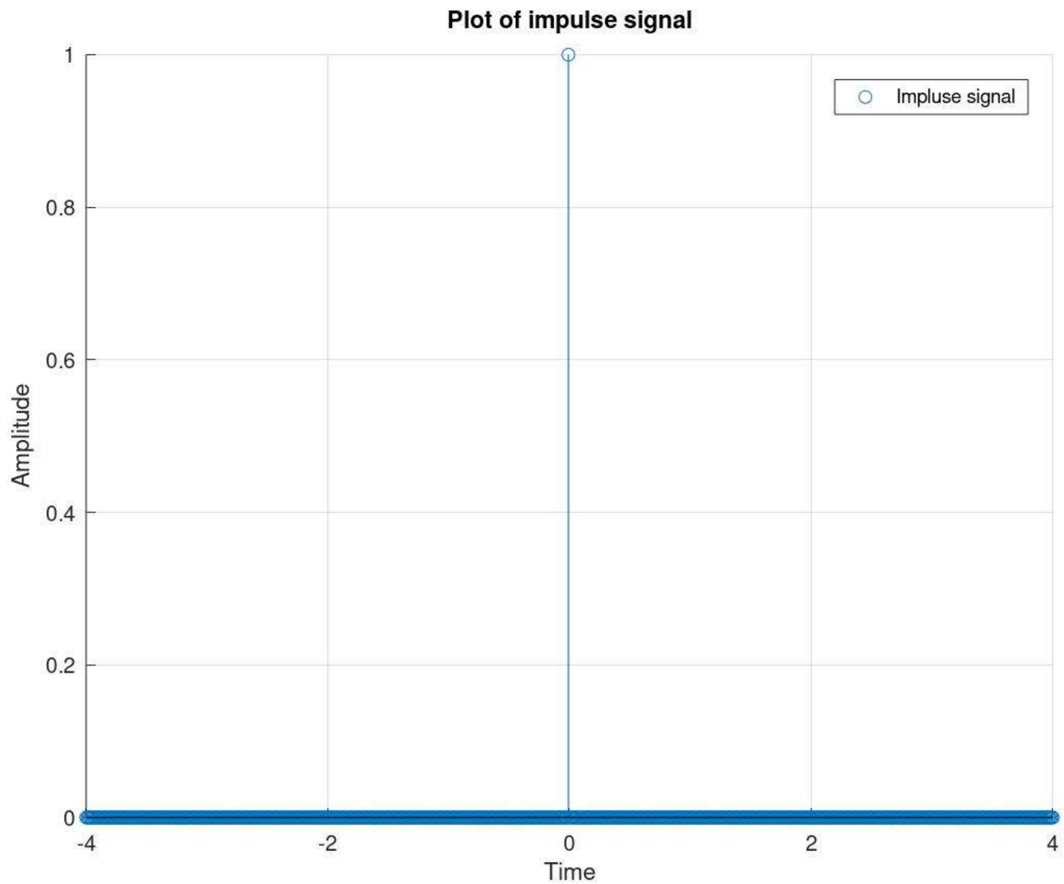


- **Impulse signal:**

  Code:

```
% Generation of impulse signal

clc;
clear all;
close all;

j=0;

for i=-4:0.01:4
  j=j+1;
  t(j)=i;

  if j==400
    del(j)=1;

  else
    del(j)=0;

  endif

endfor

stem(t,del)
legend('Impluse signal')
title('Plot of impulse signal')
xlabel('Time')
ylabel('Amplitude')
```

Figure:



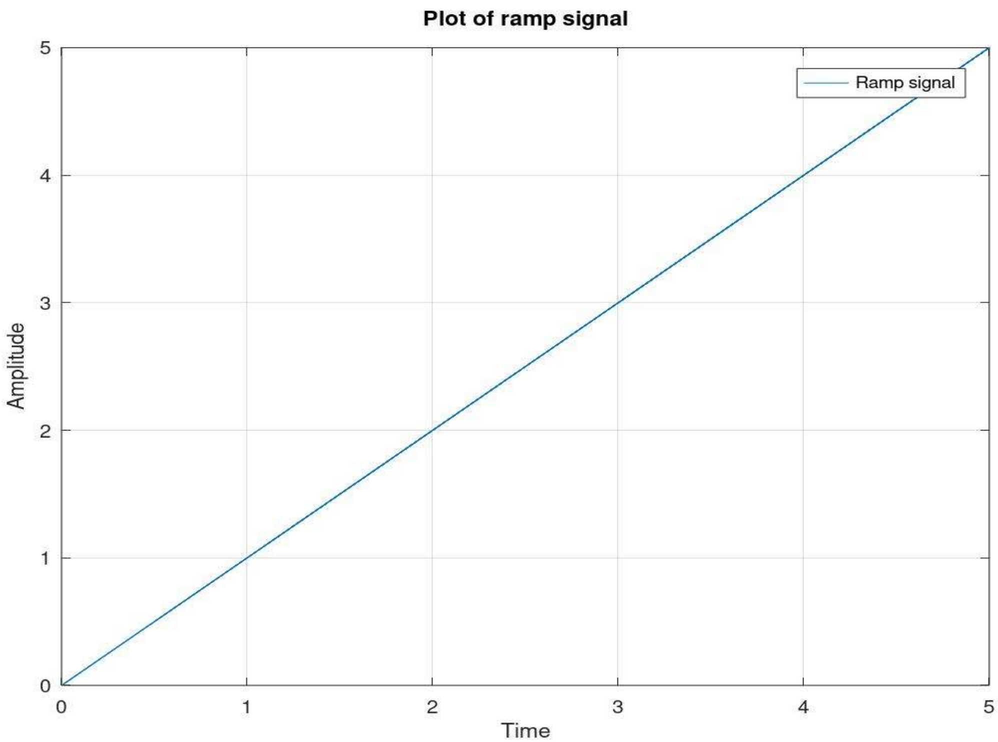
- **Ramp signal (Continuous):**

Code:

```
%Generation of Ramp signal

j=0;
for i=0:0.001:5
  j=j+1;
  t(j)=i;
  r=t;
endfor
plot(t,r);
```

```
legend('Ramp signal')
title('Plot of ramp signal')
xlabel('Time')
ylabel('Amplitude')
```

Figure:



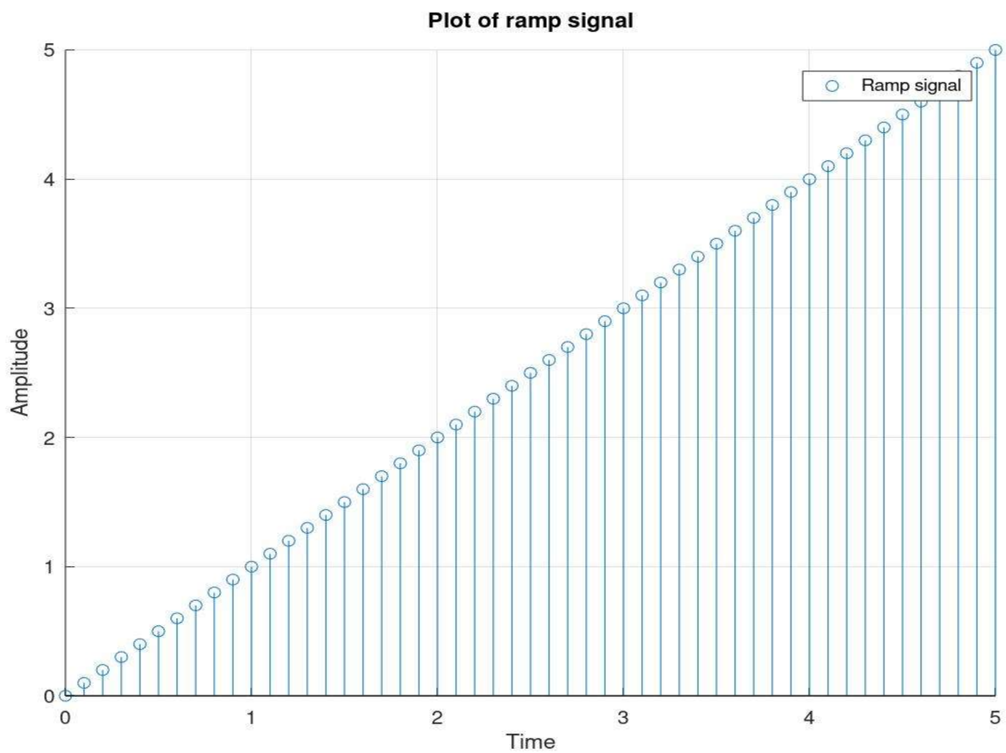
- **Ramp Signal (Discrete):**

Code:

```
%Generation of Ramp signal

Clc;
Clear all;
Close all;
i=0:0.1:5
r=i;
stem(t,r);
legend('Ramp signal')
title('Plot of ramp signal')
```

```
xlabel('Time')
ylabel('Amplitude')
```

Figure:



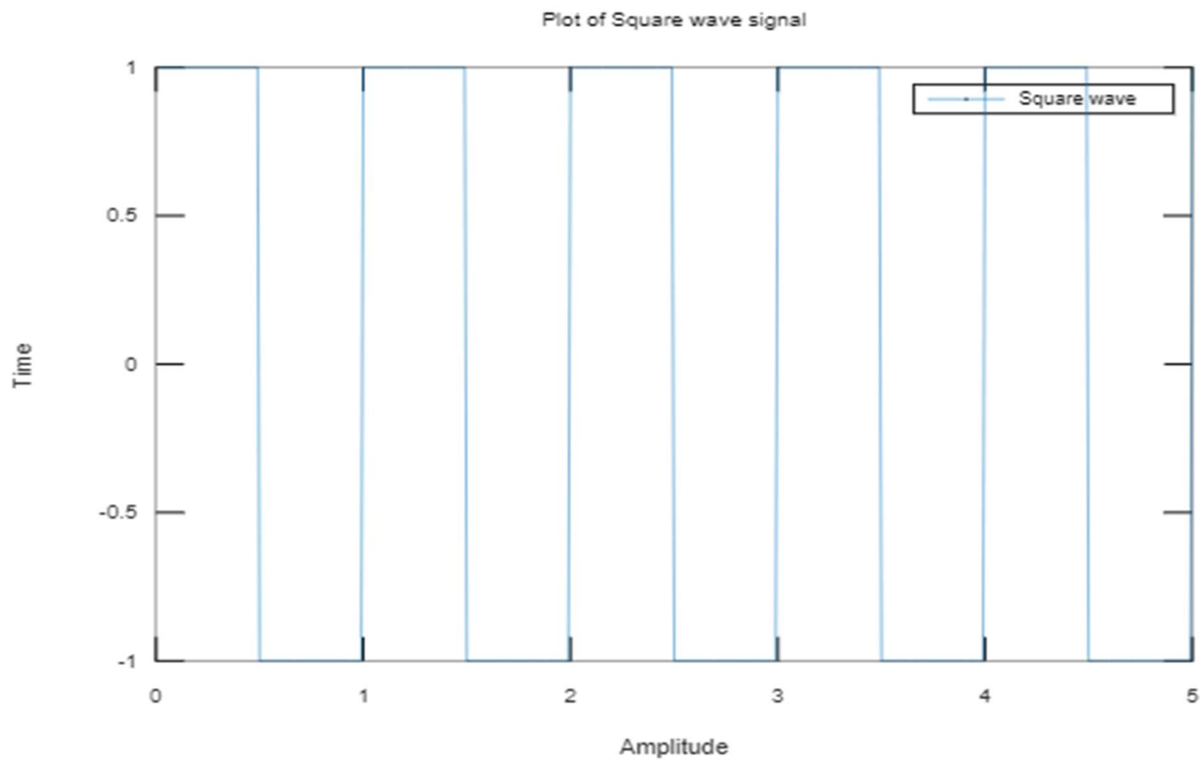
- **Square wave signal (Continuous):**

Code:

```
%Generation of Squarewave signal

clc;
clear all;
close all;

t = (0:0.01:5);
swave = square(2*pi*t);
plot(t,swave);
title('Plot of Square wave signal')
legend('Square wave');
```

```
ylabel('Time');
xlabel('Amplitude');
```

Figure:

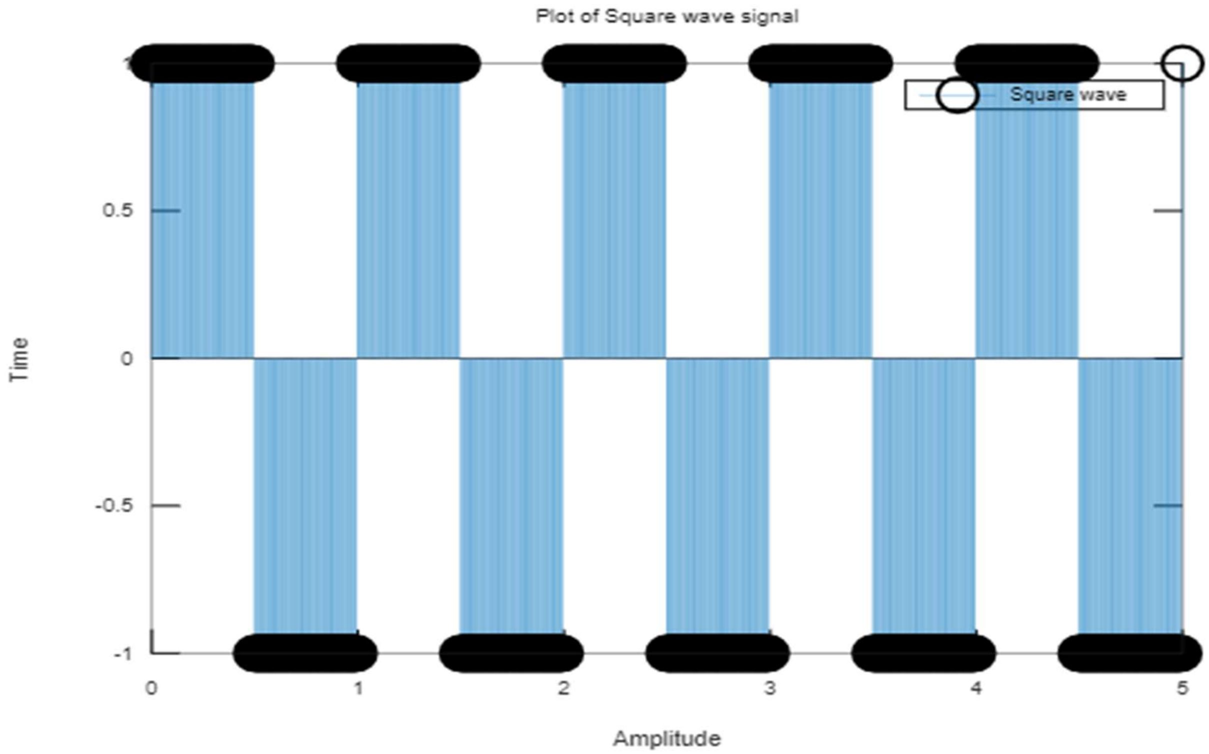


- **Square wave signal (Discrete):**

Code:

```
%Generation of Squarewave signal

clc;
clear all;
close all;
t = (0:0.01:5);
swave = square(2*pi*t);
stem(t,swave);
title('Plot of Square wave signal')
legend('Square wave');
xlabel('Amplitude');
ylabel('Time');
```

Figure:

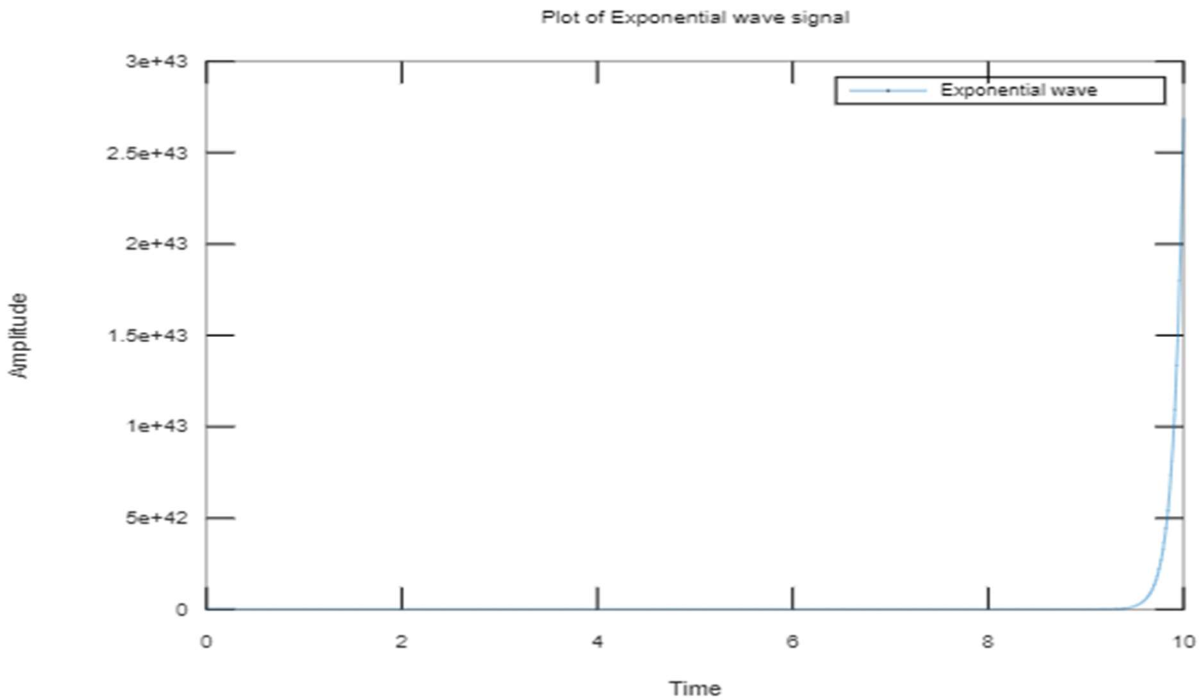


- **Exponential wave signal (Continuous):**

Code:

```
% Generation of exponential wave signal

clc;
clear all;
close all;

t = (0:0.01:10);
a = 10;
x = exp(a*t);
plot(t,x);
title('Plot of Exponential wave signal');
legend('Exponential wave');
xlabel('Time');
ylabel('Amplitude');
```

Figure:

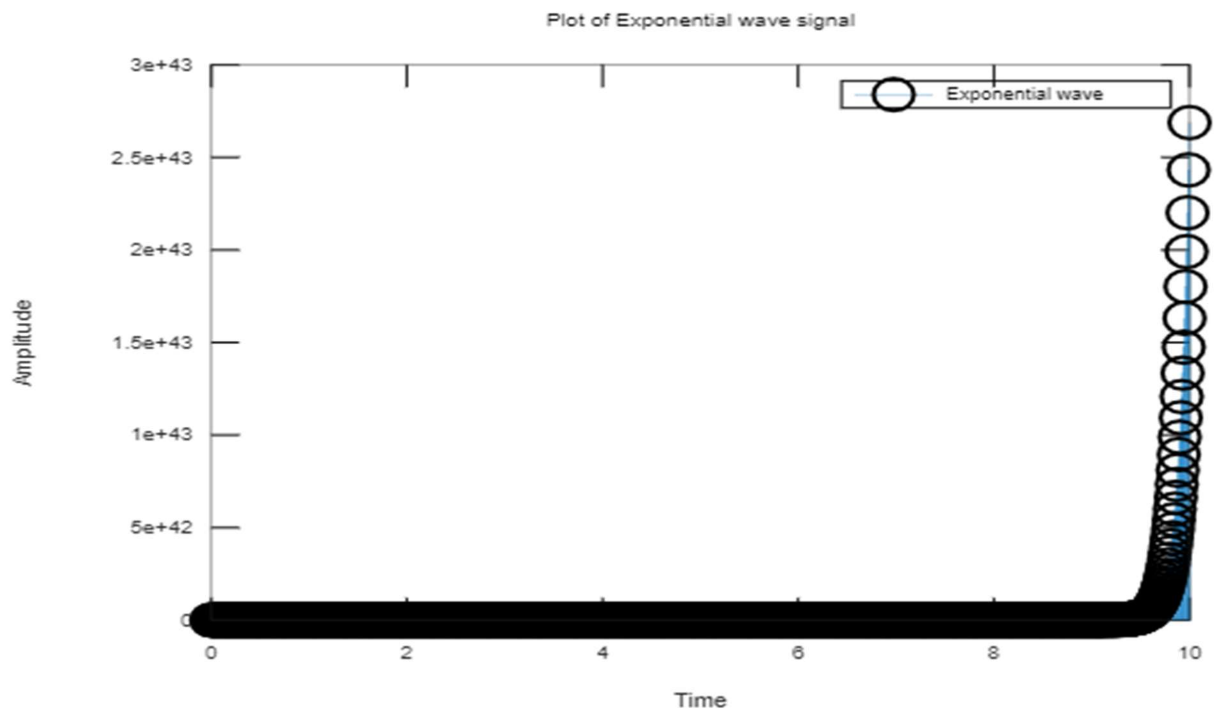


- **Exponential wave signal (Discrete):**

Code:

```
% Generation of exponential wave signal

clc;
clear all;
close all;

t = (0:0.01:10);
a = 10;
x = exp(a*t);
stem(t,x);
title('Plot of Exponential wave signal');
legend('Exponential wave');
xlabel('Time');
ylabel('Amplitude');
```

Figure:

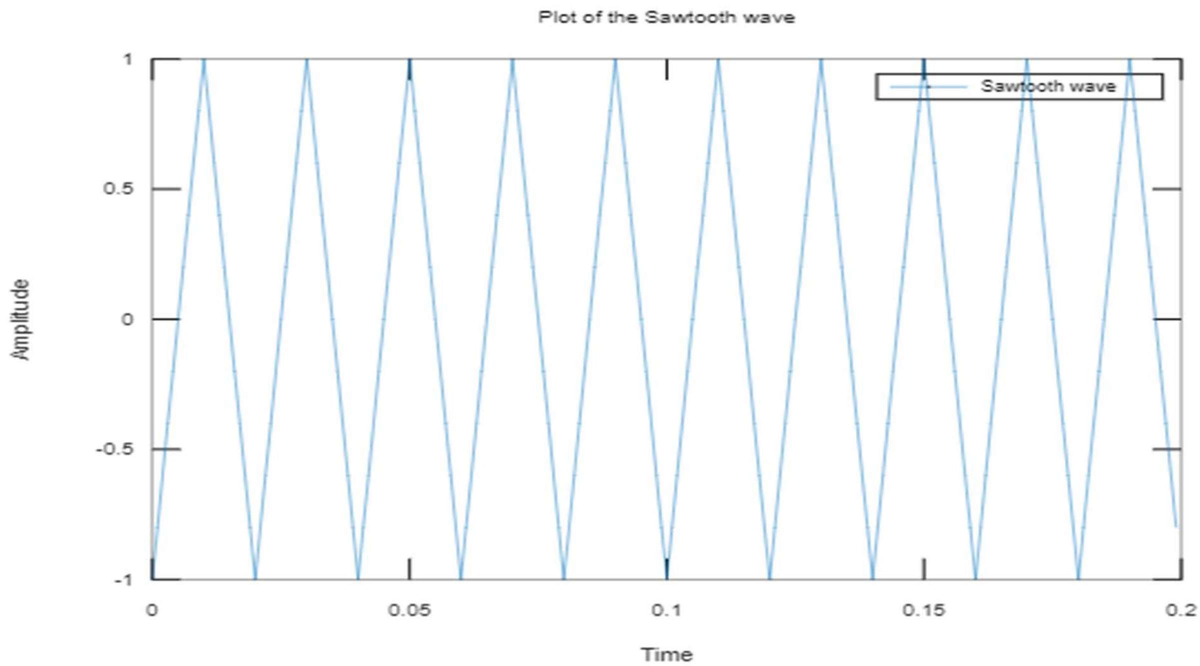


- **Sawtooth wave signal (Continuous):**

Code:

```
% Generation of Sawtooth wave

clc;
clear all;
close all;

T = 10*(1/50);
fs = 1000;
t = 0:1/fs:T-1/fs;
x = sawtooth(2*pi*50*t,1/2);
plot(t,x);
title('Plot of the Sawtooth wave');
legend('Sawtooth wave');
xlabel('Time');
ylabel('Amplitude');
```

Figure:

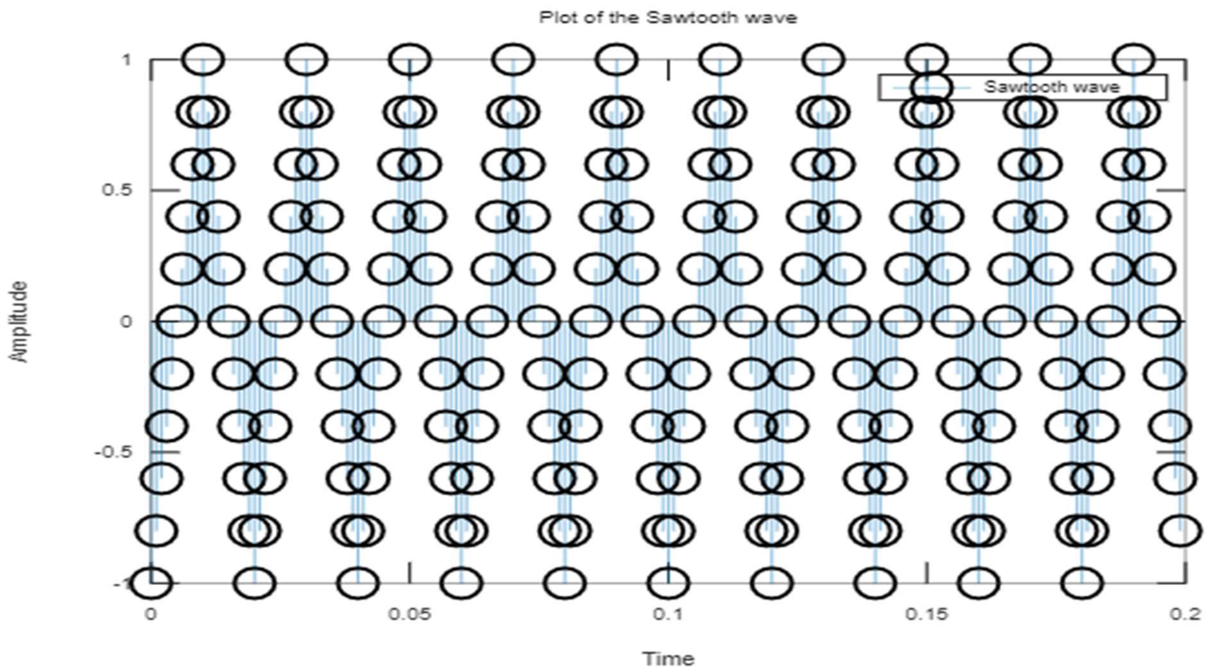


- **Sawtooth wave signal (Discrete):**

Code:

```
% Generation of Sawtooth wave

clc;
clear all;
close all;

T = 10*(1/50);
fs = 1000;
t = 0:1/fs:T-1/fs;
x = sawtooth(2*pi*50*t,1/2);
stem(t,x);
title('Plot of the Sawtooth wave');
legend('Sawtooth wave');
xlabel('Time');
ylabel('Amplitude');
```

Figure:

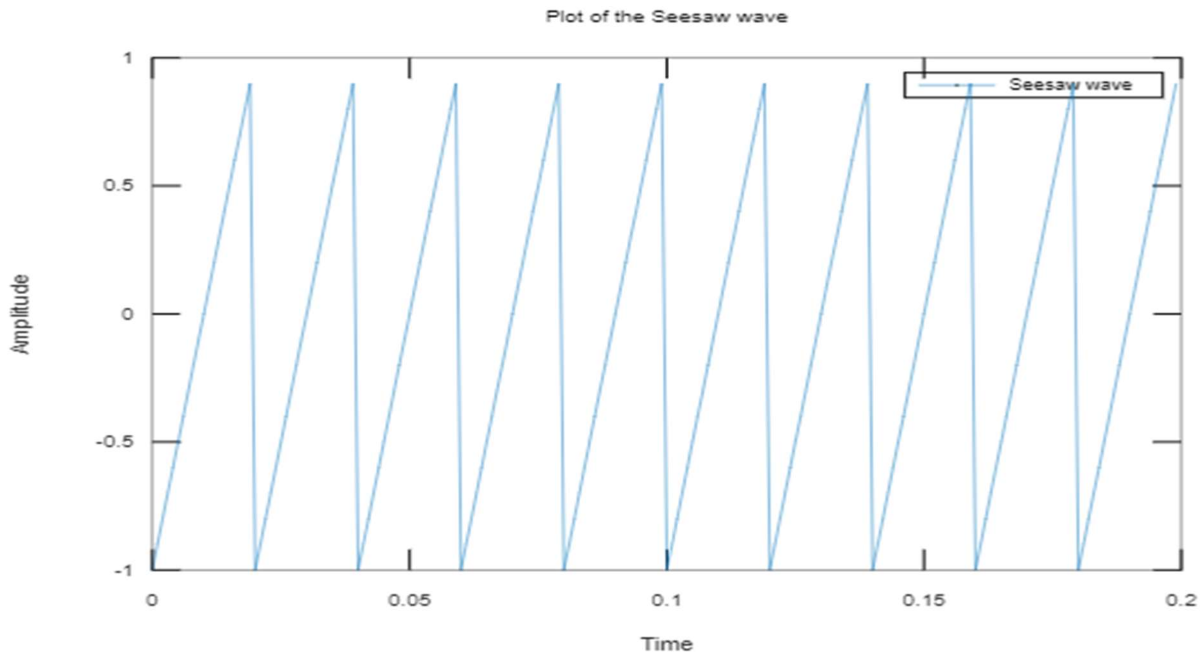


- **Seesaw wave signal (Continuous):**

Code:

```
% Generation of Seesaw wave

clc;
clear all;
close all;

T = 10*(1/50);
fs = 1000;
t = 0:1/fs:T-1/fs;
x = sawtooth(2*pi*50*t);
plot(t,x);
title('Plot of the Seesaw wave');
legend('Seesaw wave');
xlabel('Time');
ylabel('Amplitude');
```

Figure:

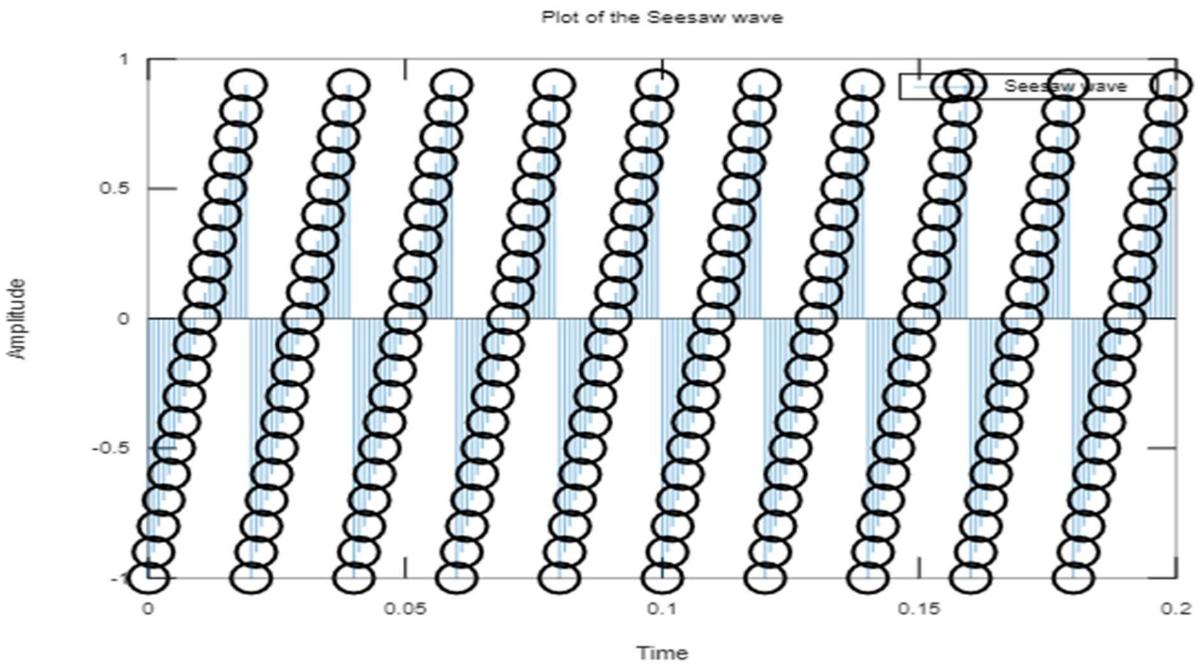


- **Seesaw wave signal (Discrete):**

Code:

```
% Generation of Seesaw wave

clc;
clear all;
close all;

T = 10*(1/50);
fs = 1000;
t = 0:1/fs:T-1/fs;
x = sawtooth(2*pi*50*t);
stem(t,x);
title('Plot of the Seesaw wave');
legend('Seesaw wave');
xlabel('Time');
ylabel('Amplitude');
```

Figure:

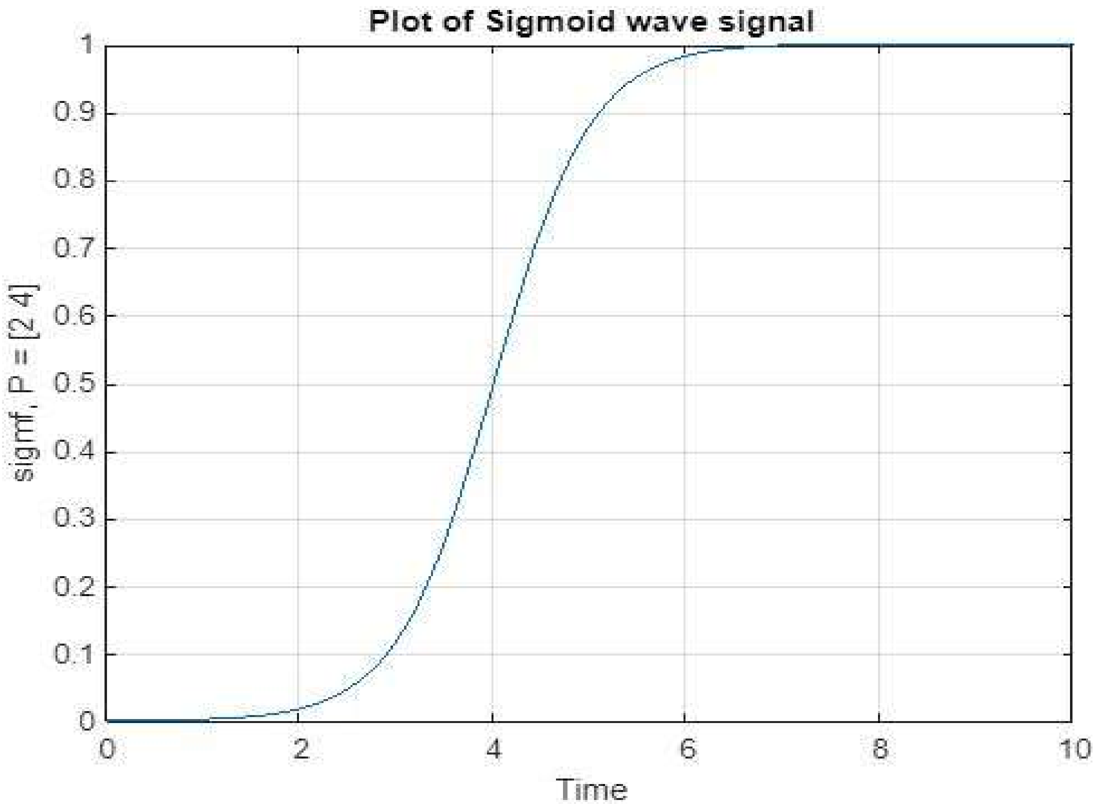


- **Sigmoid wave signal (Continuous):**

Code:

```
% Generation of sigmoid function

clc;
clear all;
close all;

x = 0:0.1:10;
y = sigmf(x,[2 4]);
plot(x,y)
title('Plot of Sigmoid wave signal');
ylabel('sigmf, P = [2 4]');
xlabel('Time');
grid on
```

Figure:

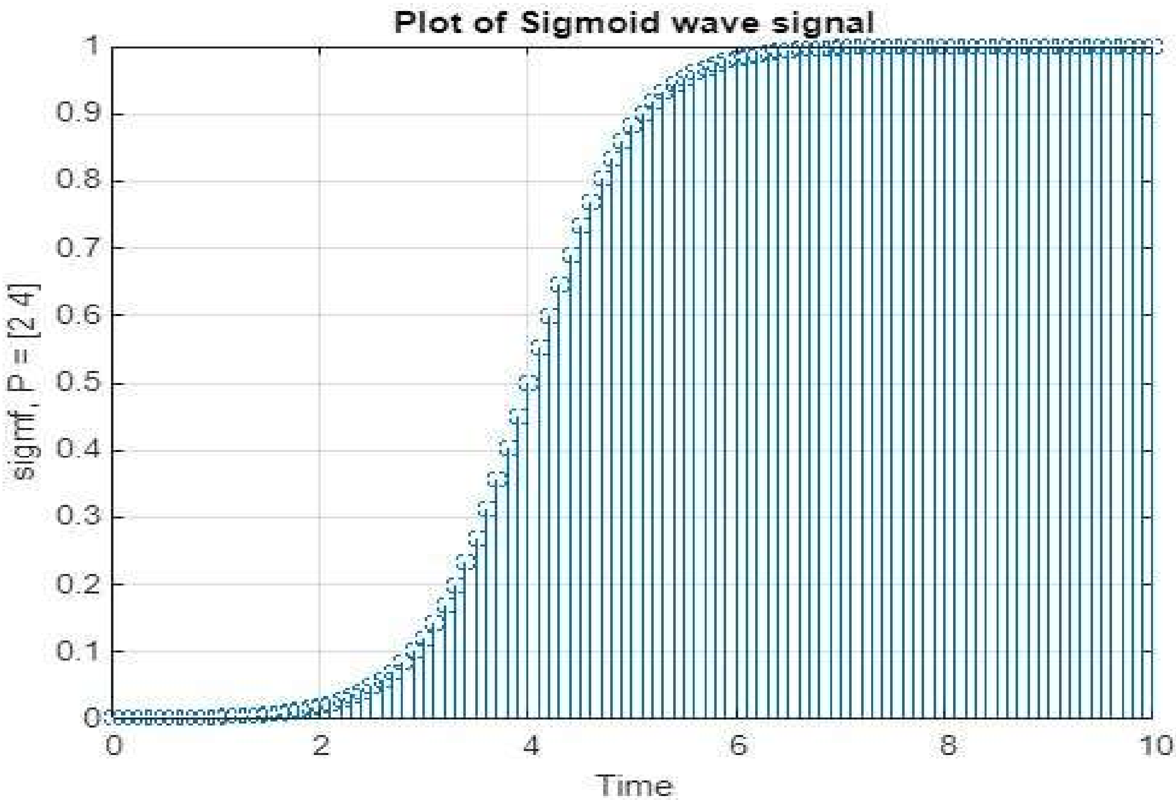


- **Sigmoid wave signal (Discrete):**

Code:

```
% Generation of sigmoid function

clc;
clear all;
close all;

x = 0:0.1:10;
y = sigmf(x,[2 4]);
stem(x,y)
title('Plot of Sigmoid wave signal');
ylabel('sigmf, P = [2 4]');
xlabel('Time');
grid on
```

Figure:


Plot of Sigmoid wave signal
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
sigmf, P = [2 4]
0
2
4
6
8
10
Time

EXPERIMENT NO. 04

Aim: To study and generate the convolution of two signals using different properties.

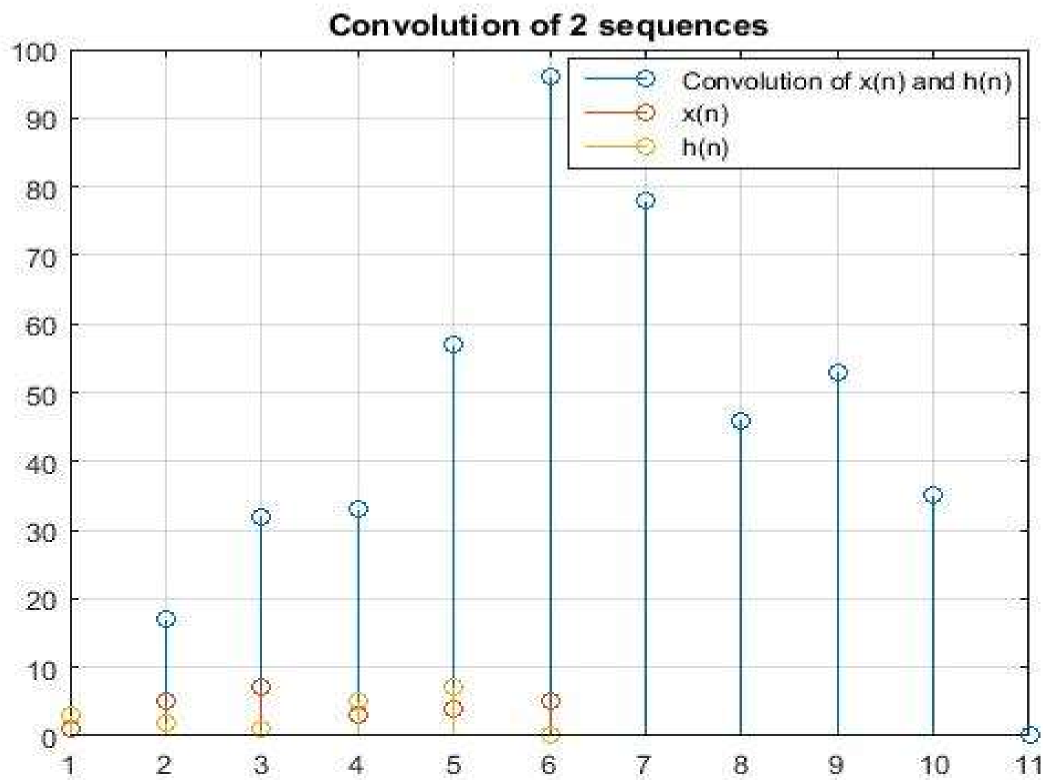
Que: 1

x(n) = {1,5,7,3,4,5} & h(n) = {3,2,1,5,7,0}

Code:

```
clc;
clear all;
n = [0.5];
x = [1 5 7 3 4 5];
h = [3 2 1 5 7 0];
m = conv(x,h);
stem(m)
title('Convolution of 2 sequences')
hold on
stem (x)
grid on
hold on
stem (h)
grid on
legend('Convolution of x(n) and h(n)','x(n)','h(n)')
```

Figure:



Convolution of 2 sequences
Convolution of x(n) and h(n)
x(n)
h(n)
100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
11

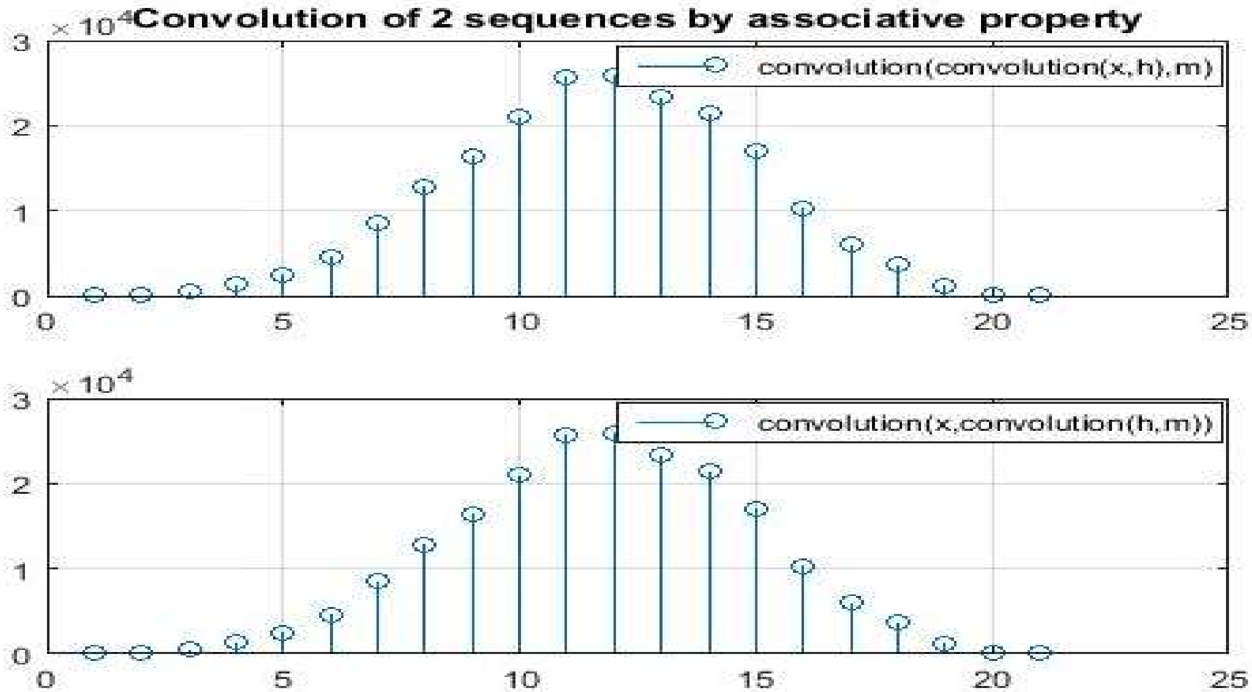
Que: 2.

Using Associative Property

Code:

```
clc;
clear all;
n = [0.5];
x = [1 5 7 3 4 5];
h = [3 2 1 5 7 0];
m = conv(x,h);
z = conv(conv(x,h),m);
p = conv(x,conv(h,m));
subplot(2,1,1)
stem(z)
grid on
title('Convolution of 2 sequences by associative
property');
legend('convolution(convolution(x,h),m)');
subplot(2,1,2)
stem(p)
grid on
legend('convolution(x,convolution(h,m))')
```

Figure:



Que: 3.

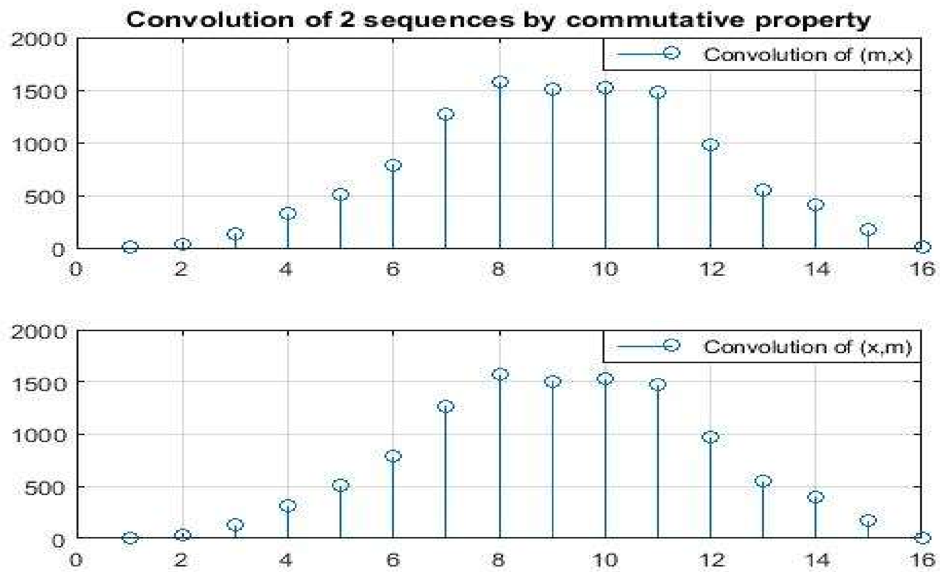
Using Commutative Property

Code:

```
clc;
clear all;
n = [0.5];
x = [1 5 7 3 4 5];
h = [3 2 1 5 7 0];
m = conv(x,h);
z = conv(m,x);
p = conv(x,m);
subplot(2,1,1)
stem(z)
title('Convolution of 2 sequences by commutative
property');
legend('Convolution of (m,x)')
grid on
```

```
subplot(2,1,2)
stem(p)
legend('Convolution of (x,m)')
grid on
```

Figure:



Que: 4.

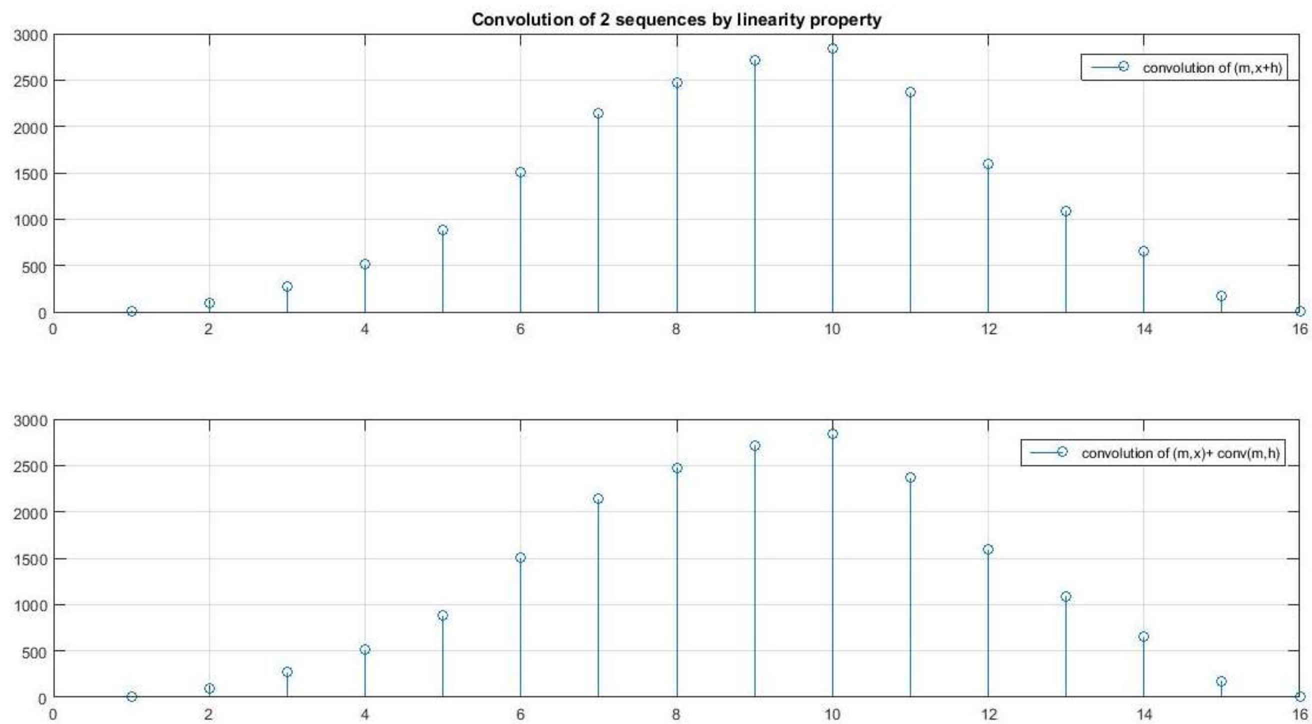
Using Linearity Property

Code:

```
clc;
clear all;
n = [0.5];
x = [1 5 7 3 4 5];
h = [3 2 1 5 7 0];
m = conv(x,h);
z = conv(m,x+h);
p = conv(m,x)+conv(m,h);
subplot(2,1,1)
stem(z)
title('Convolution of 2 sequences by linearity
property');
grid on
```

```
legend('convolution of (m,x+h)')
subplot(2,1,2)
stem(p)
grid on
legend('convolution of (m,x)+ conv(m,h) ')
```

Figure:



Que: 5.

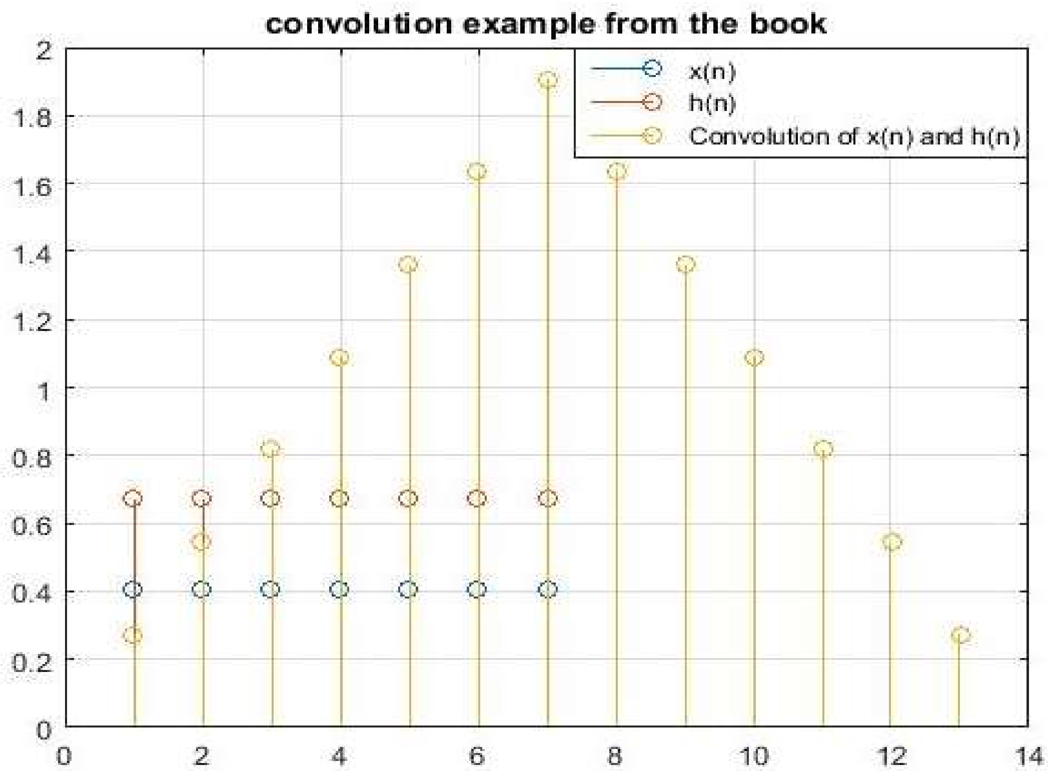
Convolution example from the book

Code:

```
clc;
clear all;
t = -pi:pi;
x = sin(2*pi*4*t);
h = cos(2*pi*8*t);
m = conv(x,h);
stem(x)
hold on
stem(h)
```

```
hold on
stem (m)
grid on
title('convolution example from the book')
legend('x(n)','h(n)','Convolution of x(n) and h(n)')
```

Figure:



EXPERIMENT NO. 05

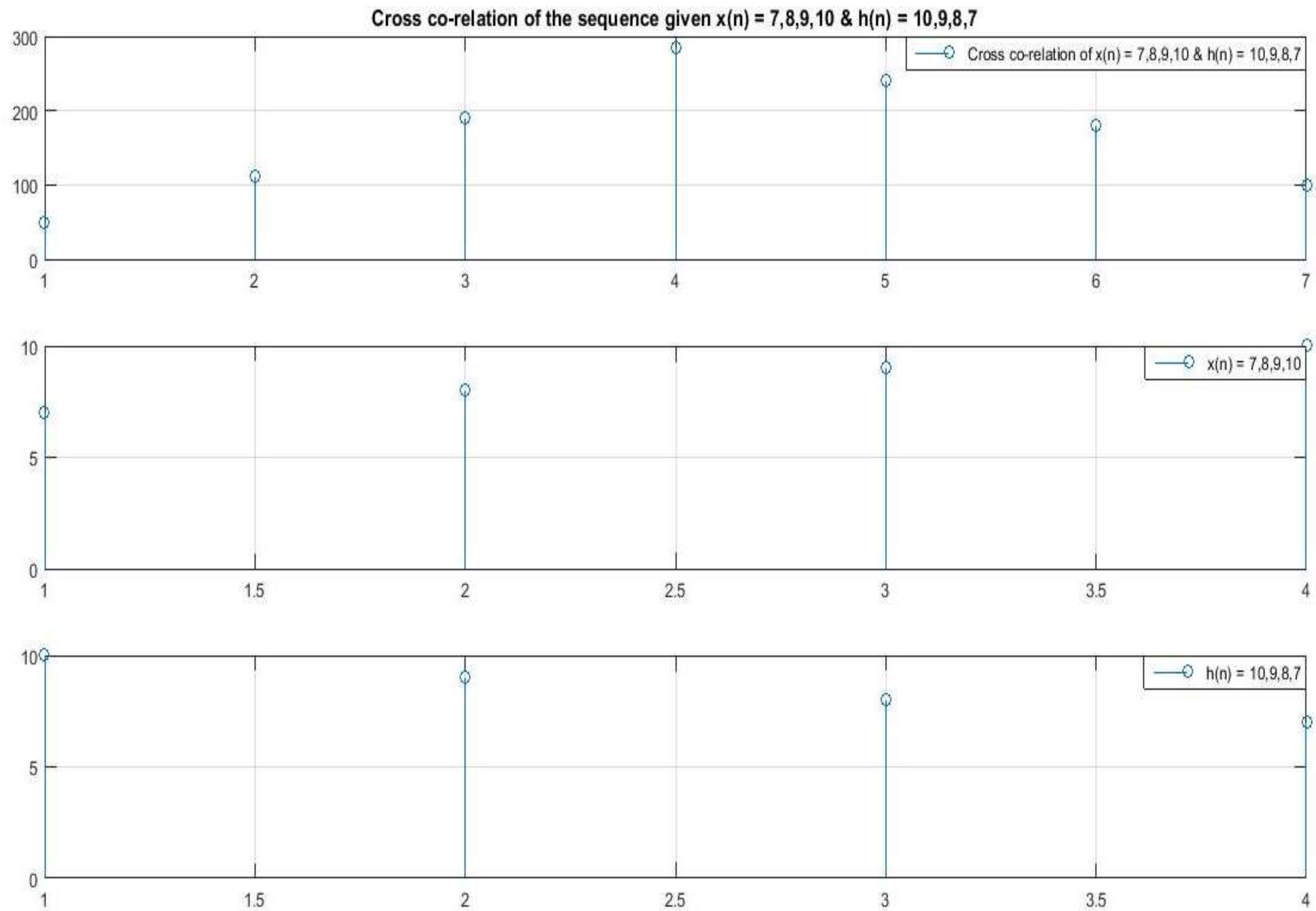
Aim: To find out Cross co-relation of the given sequence.

Que: x(n) = {7,8,9,10} & h(n) = {10,9,8,7}

Code:

```
clc;
clear all;
x = [7 8 9 10];
h = [10 9 8 7];
m = xcorr(x,h);
subplot(3,1,1)
stem(m)
title('Cross co-relation of the sequence given x(n) =
{7,8,9,10} & h(n) = {10,9,8,7}')
grid on
legend('Cross co-relation of x(n) = {7,8,9,10} & h(n) =
{10,9,8,7}')
subplot(3,1,2)
stem(x)
grid on
legend('x(n) = {7,8,9,10}')
subplot(3,1,3)
stem(h)
grid on
legend('h(n) = {10,9,8,7}')
```

Figure:


Cross co-relation of the sequence given x(n) = 7,8,9,10 & h(n) = 10,9,8,7
Cross co-relation of x(n) = 7,8,9,10 & h(n) = 10,9,8,7
x(n) = 7,8,9,10
h(n) = 10,9,8,7

EXPERIMENT NO. 06

Aim: To study the sampling theorem and to find out Nyquist rate and interval of the given signals.

Que: 1. X(t) = 6cos(60*pi*t) + 12sin(150*pi*t) - cos(30*pi*t)

Code:

```
clc;
clear all;
t = -pi:pi;
w1 = input('Enter the value of w1 - ');
w2 = input('Enter the value of w2 - ');
w3 = input('Enter the value of w3 - ');
X = 6*cos(w1*t) + 12*sin(w2*t) - 1*cos(w3*t);
f1 = w1/(2*pi);
f2 = w2/(2*pi);
f3 = w3/(2*pi);
F = [f1 f2 f3];
fm = max(F)
fs = 2*fm
ts = 1/fs
```

Output:

Enter the value of w1 – 60*pi
Enter the value of w2 – 150*pi
Enter the value of w3 – 30*pi

fm =

75

fs =

150

ts =

0.0067

Que: 2. x(t) = (1/4*pi)*(cos(5000*pi*t) + cos(3000*pi*t))

Code:

```
clc;
clear all;
t = -pi:pi;
w1 = input('Enter the value of w1 - ');
w2 = input('Enter the value of w2 - ');
x = ((1/4*pi) *  (cos(w1*t) + cos(w2*t)));
f1 = w1/(2*pi);
f2 = w2/(2*pi);
F = [f1 f2];
fm = max(F)
fs = 2*fm
ts = 1/fs
```

Output:

Enter the value of w1 - 5000*pi
Enter the value of w2 - 3000*pi

fm =

2500

fs =

5000

ts =

2.0000e-04

Que: 3. x(t) = 8*cos(200*pi*t)

Code:

```
clc;
clear all;
t = -pi:pi;
x = 8*cos(200*pi*t);
f1 = 200*pi/(2*pi);
F = [f1];
fm = max(F)
fs = 2*fm
ts = 1/fs
```

Output:

fm =

100

fs =

200

ts =

0.005

EXPERIMENT NO. 07

Aim: To find out the convolution, Z-Transform and inverse Z-Transform of the given signals and also to find out poles, zeros and gain from Z-Transform.

CONVOLUTION

1. $X1(Z) = 3 + 4(Z^{-1}) + 5(Z^{-2})$,
   $X2(Z) = 5 + 6(Z^{-1}) + 7(Z^{-2}) + 8(Z^{-3})$

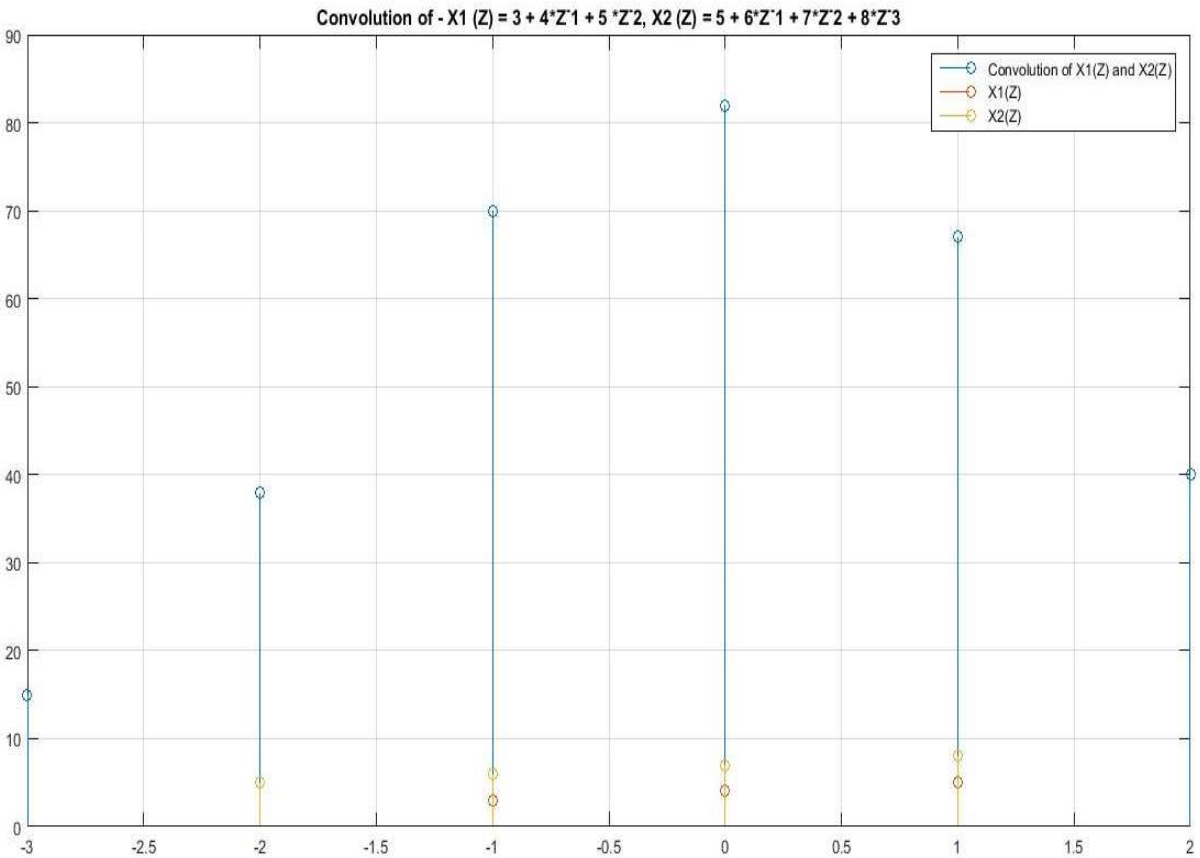
Function File:

```
function [ h,nh ] = conv_m( x,nx,y,ny )
nyb = nx(1) + ny(1);
nye = nx(end) + ny(end);
nh = (nyb:nye);
h = conv(x,y);
end
```

Main File:

```
clc;

clear all;
nx = -1:1;
ny = -2:1;
x = [3,4,5];
y = [5,6,7,8];
[h,nh] = conv_m(x,nx,y,ny);
stem(nh,h)
hold on
stem(nx,x)
hold on
stem(ny,y)
title('Convolution of - X1 (Z) = 3 + 4*Z^-1 + 5 *Z^-2,
X2 (Z) = 5 + 6*Z^-1 + 7*Z^-2 + 8*Z^-3')
legend('Convolution of X1(Z) and X2(Z)', 'X1(Z)',
'X2(Z)')
grid on
```

Figure:



Convolution of - X1 (Z) = 3 + 4*Z^-1 + 5 *Z^-2, X2 (Z) = 5 + 6*Z^-1 + 7*Z^-2 + 8*Z^-3
Convolution of X1(Z) and X2(Z)
X1(Z)
X2(Z)
90
80
70
60
50
40
30
20
10
0
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

2. $X1(Z) = 2(Z) + 3 + 4(Z^{-1})$,
   $X2(Z) = 3(Z^{2}) + 4(Z) + 5 + 6(Z^{-1})$

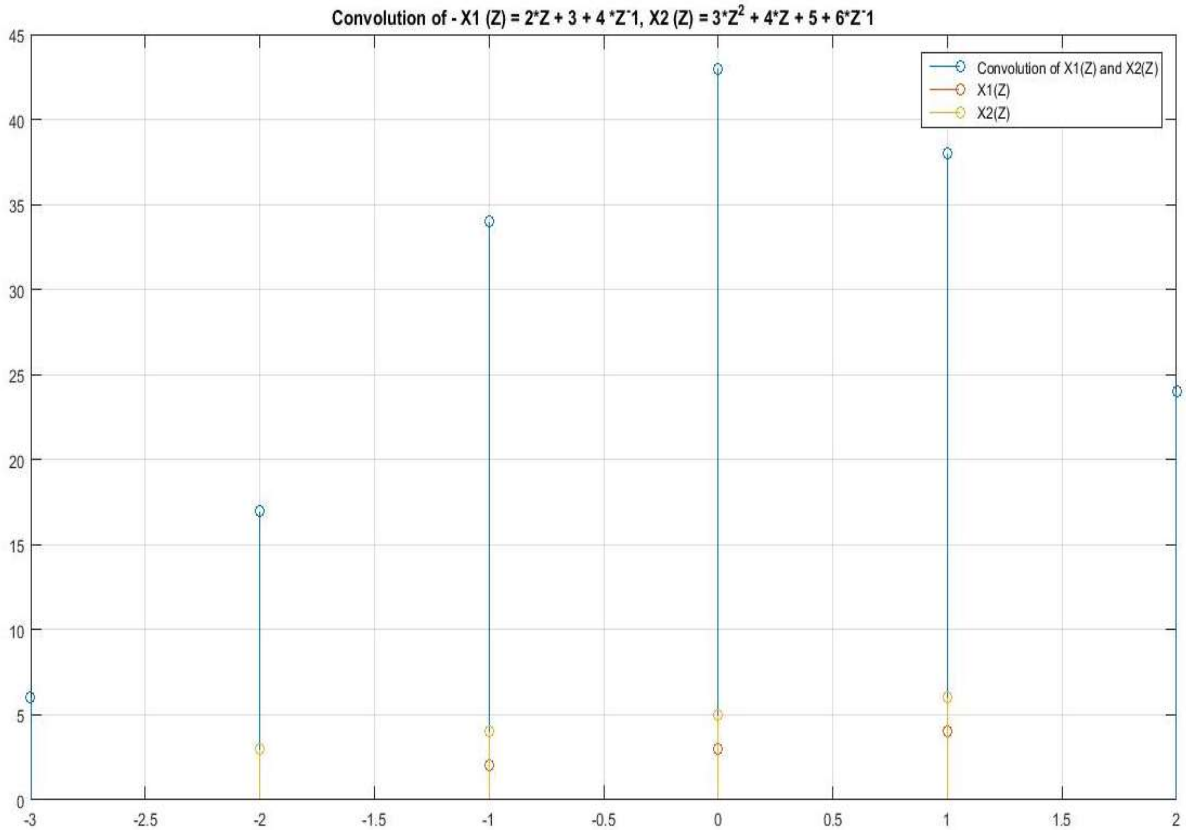
Function File:

```
function [ h,nh ] = conv_m( x,nx,y,ny )
nyb = nx(1) + ny(1);
nye = nx(end) + ny(end);
nh = (nyb:nye);
h = conv(x,y);
end
```

Main File:

```
clc;
clear all;
nx = -1:1;
ny = -2:1;
x = [3,4,5];
y = [5,6,7,8];
[h,nh] = conv_m(x,nx,y,ny);
stem(nh,h)
hold on
stem(nx,x)
hold on
stem(ny,y)
title('Convolution of - X1 (Z) = 2*Z + 3 + 4 *Z^-1, X2 (Z) = 3*Z^2 + 4*Z + 5 + 6*Z^-1')
legend('Convolution of X1(Z) and X2(Z)', 'X1(Z)', 'X2(Z)')
grid on
```

Figure:



Convolution of - X1 (Z) = 2*Z + 3 + 4 *Z-1, X2 (Z) = 3*Z2 + 4*Z + 5 + 6*Z-1
Convolution of X1(Z) and X2(Z)
X1(Z)
X2(Z)
45
40
35
30
25
20
15
10
5
0
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Z-TRANSFORM

1. **3^n**

Code:

```
clc;
clear all;
syms n;
a = 3^n;
b = ztrans(a)
```

Output:

b =

z/(z – 3)

2. **Cos(wn)**

Code:

```
clc;
clear all;
syms n;
w = 2*pi*50;
a = cos(w*n);
b = ztrans(a)
```

Output:

b =

(z*(z - 1))/(z^2 - 2*z + 1)

INVERSE Z-TRANSFORM

Code:

```
clc;
clear all;
a = [1,0.7*sqrt(3)]; % denominator
b = [1,-0.9*sqrt(3),0.2]; % numerator
[r,p,c] = residuez(a,b)
zplane(a,b)
```

Output:

r =

2.0601

-1.0601

p =

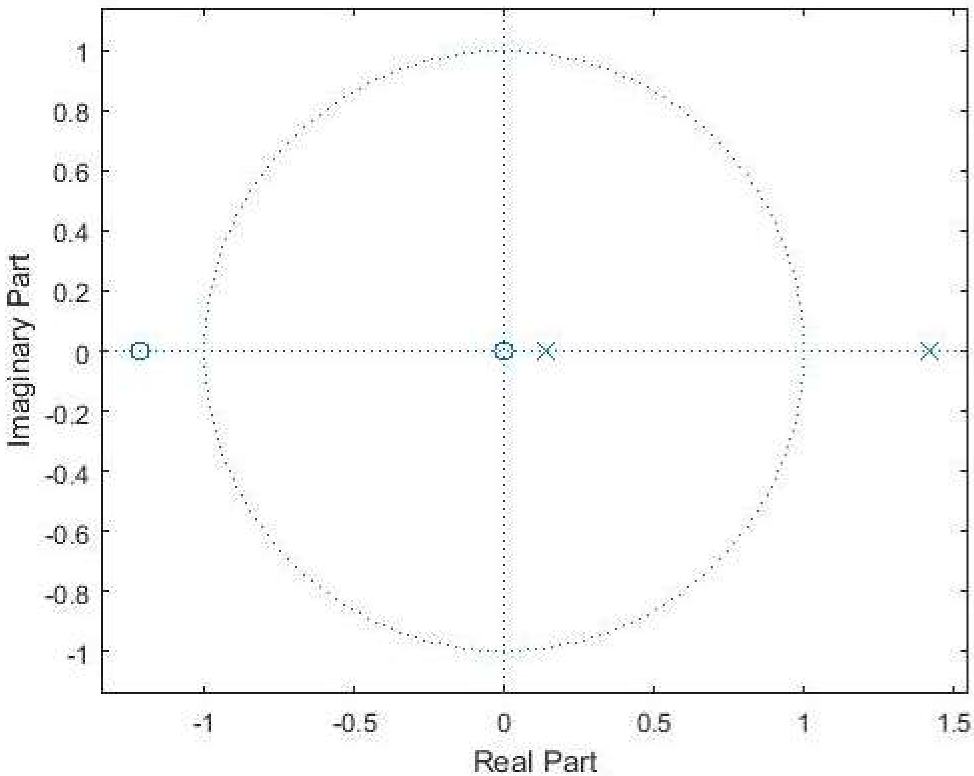
1.4178

0.1411

c =

[]

Figure:



EXPERIMENT NO. 08

Aim: To design FIR filter using window function.

Que: `hd(w) = exp(j*w*(M-1/2))    -wc < w < wc`

Rectangular Window:

Code:

```
clc;
clear all;
Wc = 1;
M = 7;
alpha = (M - 1)./2;
n = 0:1:M-1;
hd = (sin(Wc*(n - alpha)))./(1 - pi*(n - alpha));
error = 0.1;
Wn = boxcar(M); %% rectangular window
Hn = Wn*hd
```

Output:

Hn =

-0.0135 -0.1248 -0.2032 0 -0.3929 -0.1721 -0.0168

-0.0135 -0.1248 -0.2032 0 -0.3929 -0.1721 -0.0168

-0.0135 -0.1248 -0.2032 0 -0.3929 -0.1721 -0.0168

-0.0135 -0.1248 -0.2032 0 -0.3929 -0.1721 -0.0168

-0.0135 -0.1248 -0.2032 0 -0.3929 -0.1721 -0.0168

-0.0135 -0.1248 -0.2032 0 -0.3929 -0.1721 -0.0168

-0.0135 -0.1248 -0.2032 0 -0.3929 -0.1721 -0.0168

1. Hamming Window:

Code:

```
clc;
clear all;
Wc = 1;
M = 7;
alpha = (M - 1)./2;
n = 0:1:M-1;
hd = (sin(Wc*(n - alpha)))./(1 - pi*(n - alpha));
error = 0.1;
Wh = hamming(M); %% Hamming window
Hd = Wh*hd
```

Output:

Hd =

-0.0011	-0.0100	-0.0163	0	-0.0314	-0.0138	-0.0013
-0.0042	-0.0387	-0.0630	0	-0.1218	-0.0534	-0.0052
-0.0104	-0.0961	-0.1564	0	-0.3025	-0.1325	-0.0129
-0.0135	-0.1248	-0.2032	0	-0.3929	-0.1721	-0.0168
-0.0104	-0.0961	-0.1564	0	-0.3025	-0.1325	-0.0129
-0.0042	-0.0387	-0.0630	0	-0.1218	-0.0534	-0.0052
-0.0011	-0.0100	-0.0163	0	-0.0314	-0.0138	-0.0013

2. Hanning Window:

Code:

```
clc;
clear all;
Wc = 1;
M = 7;
alpha = (M - 1)./2;
n = 0:1:M-1;
hd = (sin(Wc*(n - alpha)))./(1 - pi*(n - alpha));
error = 0.1;
Whn = hanning(M); %% Hanning window
Hd = Whn*hd
```

Output:

Hd =

-0.0020 -0.0183 -0.0298 0 -0.0575 -0.0252 -0.0025

-0.0068 -0.0624 -0.1016 0 -0.1965 -0.0861 -0.0084

-0.0116 -0.1066 -0.1734 0 -0.3354 -0.1469 -0.0143

-0.0135 -0.1248 -0.2032 0 -0.3929 -0.1721 -0.0168

-0.0116 -0.1066 -0.1734 0 -0.3354 -0.1469 -0.0143

-0.0068 -0.0624 -0.1016 0 -0.1965 -0.0861 -0.0084

-0.0020 -0.0183 -0.0298 0 -0.0575 -0.0252 -0.0025

3. Blackmann Window:

Code:

```
clc;
clear all;
Wc = 1;
M = 7;
alpha = (M - 1)./2;
n = 0:1:M-1;
hd = (sin(Wc*(n - alpha)))./(1 - pi*(n - alpha));
error = 0.1;
Wb = blackman(M); %% Blackmann window
Hd = Wb*hd
```

Output:

Hd =

```
       0         0         0    0    0         0       0
 -0.0018 -0.0162 -0.0264 0 -0.0511 -0.0224 -0.0022
 -0.0085 -0.0787 -0.1280 0 -0.2475 -0.1084 -0.0106
 -0.0135 -0.1248 -0.2032 0 -0.3929 -0.1721 -0.0168
 -0.0085 -0.0787 -0.1280 0 -0.2475 -0.1084 -0.0106
 -0.0018 -0.0162 -0.0264 0 -0.0511 -0.0224 -0.0022
       0         0         0    0    0         0       0
```

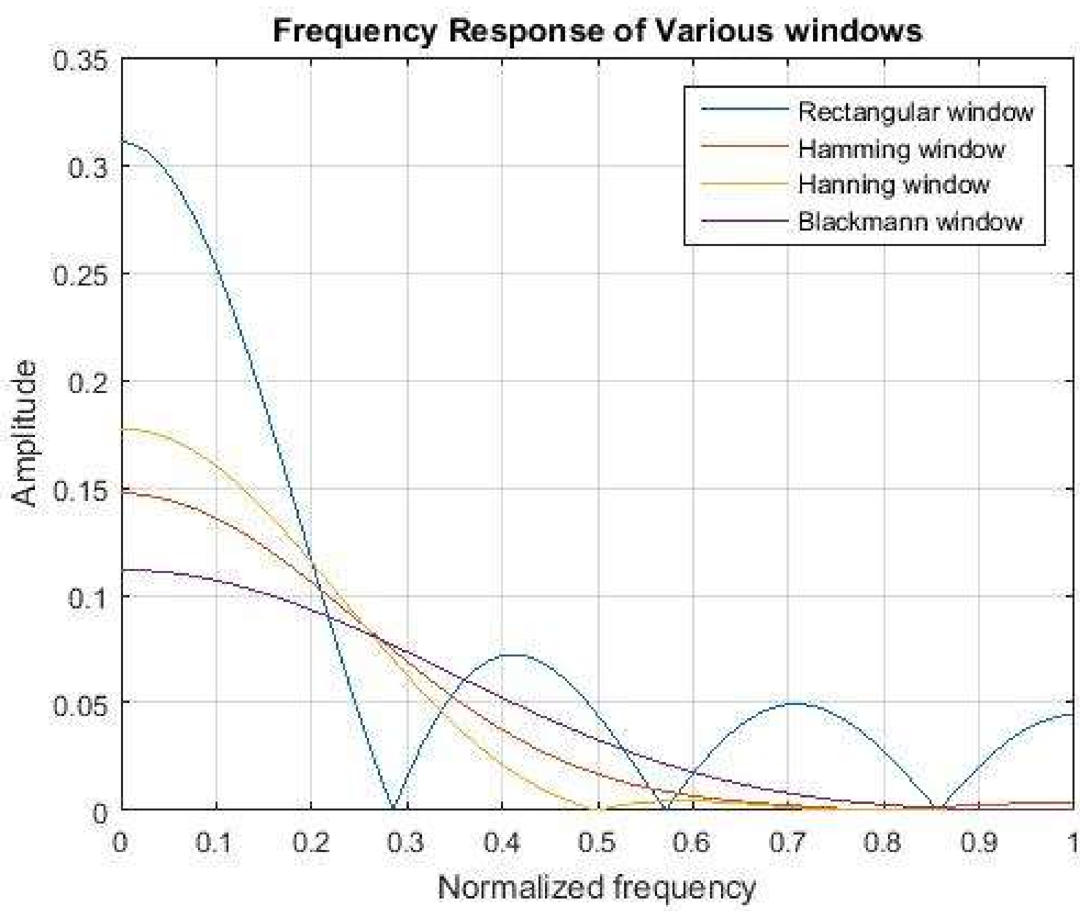
4. Frequency Response of various windows:

Code:

```
clc;
clear all;
wc = 0.5*pi;
M = 7;
a = (M-1)/2;
eps = 0.001;
n = 0:1:M-1;
hd = (sin(wc*(n-a+eps)))/(1-pi*(n-a+eps));

%% rectangular window %%
wh = boxcar(M);
hn = wh'.*hd;
w = 0:0.01:pi;
h = freqz(hn,1,w);

%% hamming window %%
whn = hamming(M);
hn1 = whn'.*hd;
h1 = freqz(hn1,1,w);

%% hanning window %%
wn = hanning(M);
hn2 = wn'.*hd;
h2 = freqz(hn2,1,w);

%% blackmann window %%
wb = blackman(M);
hn3 = wb'.*hd;
h3 = freqz(hn3,1,w);


%% plot all the various window %%
P1 = plot((w/pi),abs(h));
hold on
```

```
P2 = plot((w/pi),abs(h1));
hold on
P3 = plot((w/pi),abs(h2));
hold on
P4 = plot((w/pi),abs(h3));

legend([P1,P2,P3,P4],'Rectangular window','Hamming
window','Hanning window','Blackmann window')
grid on
xlabel('Normalized frequency');
ylabel('Amplitude');
title('Frequency Response of Various window')
```

Figure:



EXPERIMENT NO. 09

Aim: Design the low and high pass filter with the transfer function and cut-off frequency and also draw its characteristics bode plot and root locus.

1. Low Pass Filter

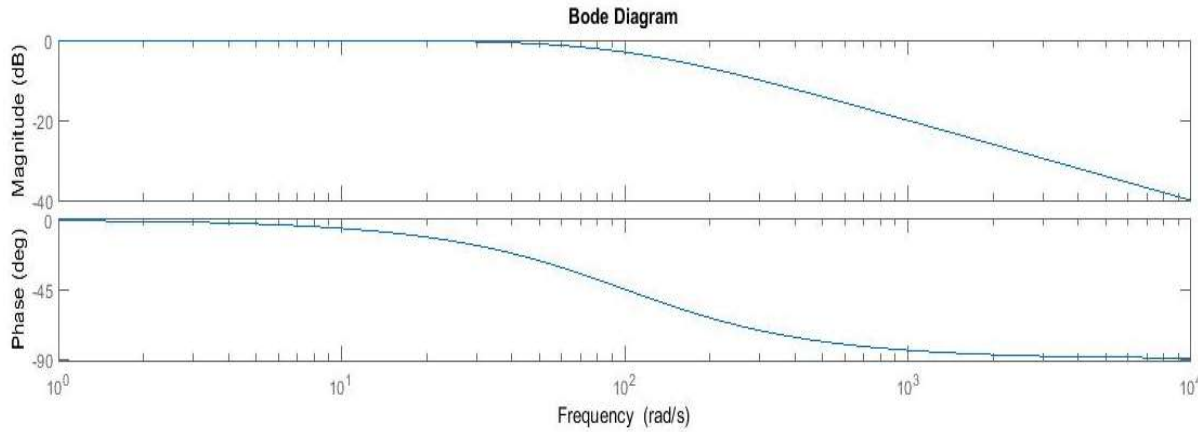
Code:

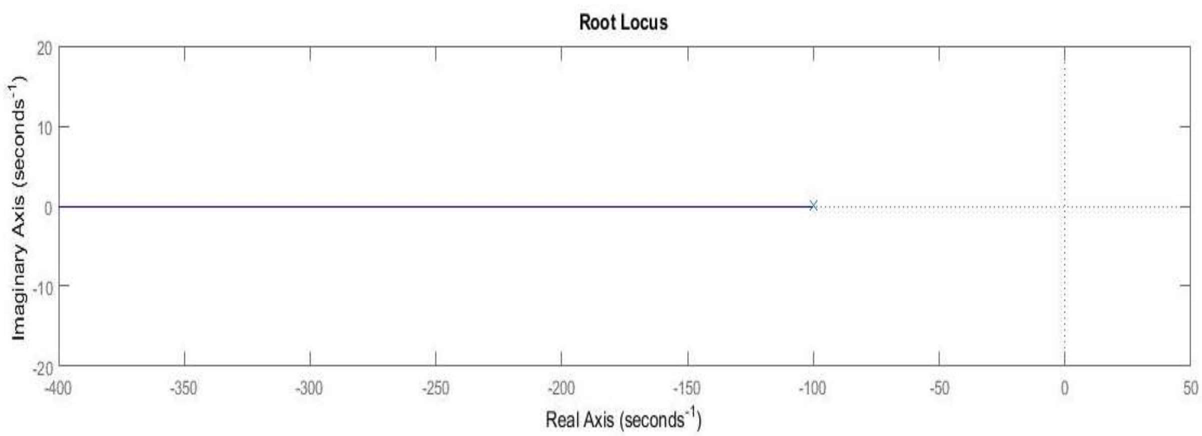
```
clc;
clear all;
close all;
R = 100;
w = input('Enter cut-off frequency (Hz) - ');
C = 1/(R*w);
N = [1]; %% numerator
D = [(R*C), 1]; %% denominator
sys = tf(N,D);
[GM, PM, c, d] = margin(sys);
subplot(2,1,1)
bode(sys);
subplot(2,1,2)
rlocus(sys);
```

Output:

Enter cut-off frequency (Hz) – 100

Figure:


Bode Diagram
Magnitude (dB)
0
-20
-40
Phase (deg)
0
-45
-90
10^0
10^1
10^2
10^3
10^4
Frequency (rad/s)


Root Locus
Imaginary Axis (seconds^{-1})
20
10
0
-10
-20
-400
-350
-300
-250
-200
-150
-100
-50
0
50
Real Axis (seconds^{-1})

2. High Pass Filter

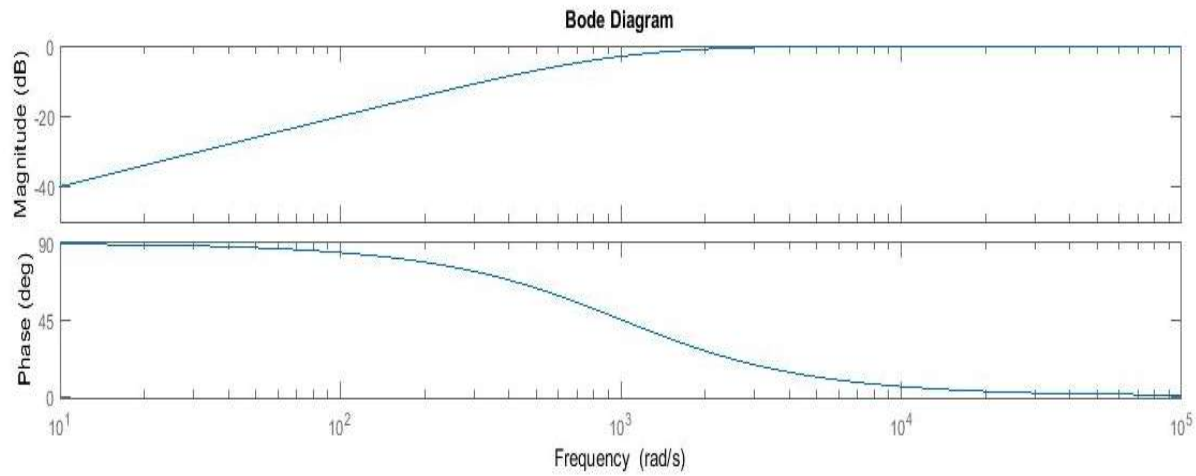
Code:

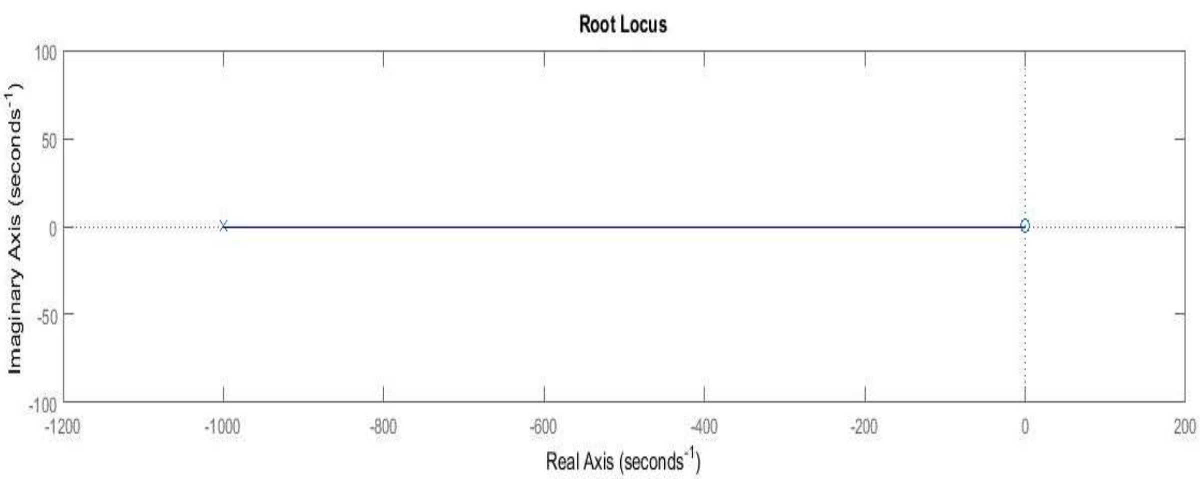
```
clc;
clear all;
close all;
R = 100;
w = input('Enter cut-off frequency (Hz) - ');
C = 1/(R*w);
N = [1, 0]; %% numerator
D = [1, (1/(R*C))]; %% denominator
sys = tf(N,D);
[GM, PM, c, d] = margin(sys);
subplot(2,1,1)
bode(sys);
subplot(2,1,2)
rlocus(sys);
```

Output:

Enter cut-off frequency (Hz) – 100

Figure:


Bode Diagram
Magnitude (dB)
0
-20
-40
Phase (deg)
90
45
0
10^1
10^2
10^3
10^4
10^5
Frequency (rad/s)


Root Locus
Imaginary Axis (seconds^{-1})
100
50
0
-50
-100
-1200
-1000
-800
-600
-400
-200
0
200
Real Axis (seconds^{-1})

EXPERIMENT NO. 10

Aim: To design the IIR Butterworth low pass and high pass filter.

Que. Pass band ripple = 0.5
Stop band ripple = 40
Pass band frequency = 2000
Stop band frequency = 3000
Sample frequency = 10,000

1. Low Pass Filter

Code:

```
clc;
close all;
clear all;
rp = input('Enter the value of pass band ripple - ');
rs = input('Enter the value of stop band ripple - ');
fp = input('Enter the value of pass band frequency -
');
fs = input('Enter the value of stop band frequency -
');
f  = input('Enter the vaue of sampling frequency - ');
w1 = 2*fp/f;
w2 = 2*fs/f;
[n,wn] = buttord(w1,w2,rp,rs);
[b,a] = butter(n,wn);
w = 0:0.1:pi;
[h,p] = freqz(b,a,w);
g = 20*log10(abs(h));
A = angle(h);

%% Plotting the Amplitude Plot
subplot(2,1,1);
plot(p/pi,g);
ylabel('Amplitude');
xlabel('Normalized Frequency');
grid on
title('Amplitude Plot');
```

```
%% Plottong the Phase Plot
subplot(2,1,2);
plot(p/pi,A);
ylabel('Amplitude');
xlabel('Normalized Frequency');
grid on
title('Phase Plot');
```

Output:

Enter the value of pass band ripple - 0.5

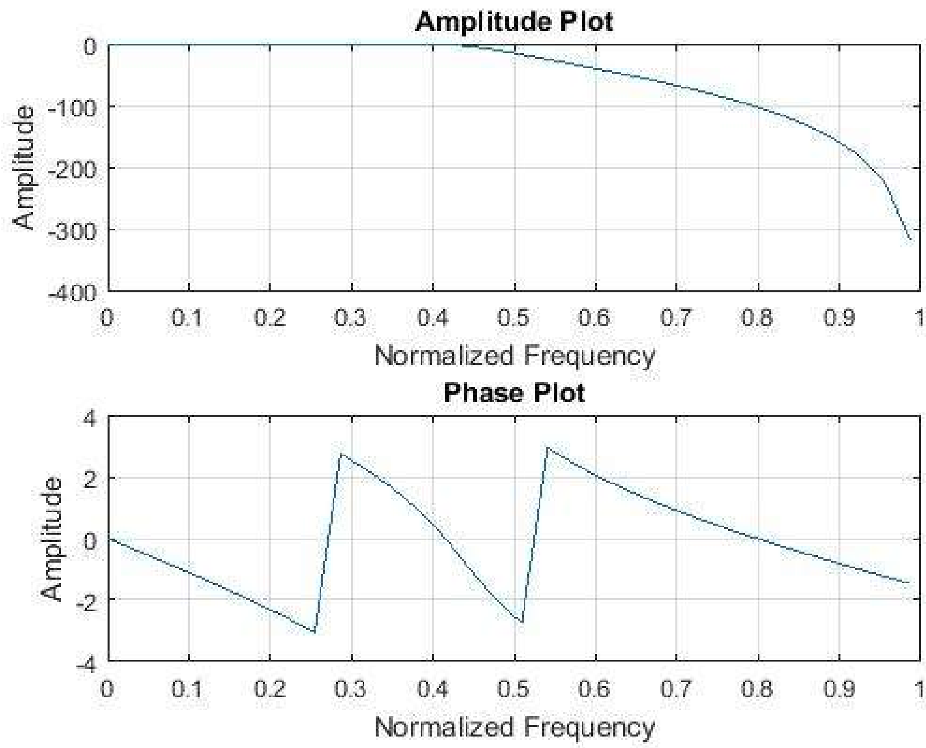
Enter the value of stop band ripple - 40

Enter the value of pass band frequency - 2000

Enter the value of stop band frequency - 3000

Enter the vaue of sampling frequency – 10000

Figure:



2. High Pass Filter

Code:

```
clc;
close all;
clear all;
rp = input('Enter the value of pass band ripple - ');
rs = input('Enter the value of stop band ripple - ');
fp = input('Enter the value of pass band frequency -
');
fs = input('Enter the value of stop band frequency -
');
f  = input('Enter the vaue of sampling frequency - ');
w1 = 2*fp/f;
w2 = 2*fs/f;
[n,wn] = buttord(w1,w2,rp,rs);
[b,a] = butter(n,wn,'high');
w = 0:0.1:pi;
[h,p] = freqz(b,a,w);
```

```
g = 20*log10(abs(h));
A = angle(h);

%% plotting the Amplitude Plot
subplot(2,1,1);
plot(p/pi,g);
ylabel('Amplitude');
xlabel('Normalized Frequency');
grid on
title('Amplitude Plot');

%% Plottong the Phase Plot
subplot(2,1,2);
plot(p/pi,A);
ylabel('Amplitude');
xlabel('Normalized Frequency');
grid on
title('Phase Plot');
```

Output:

Enter the value of pass band ripple - 0.5

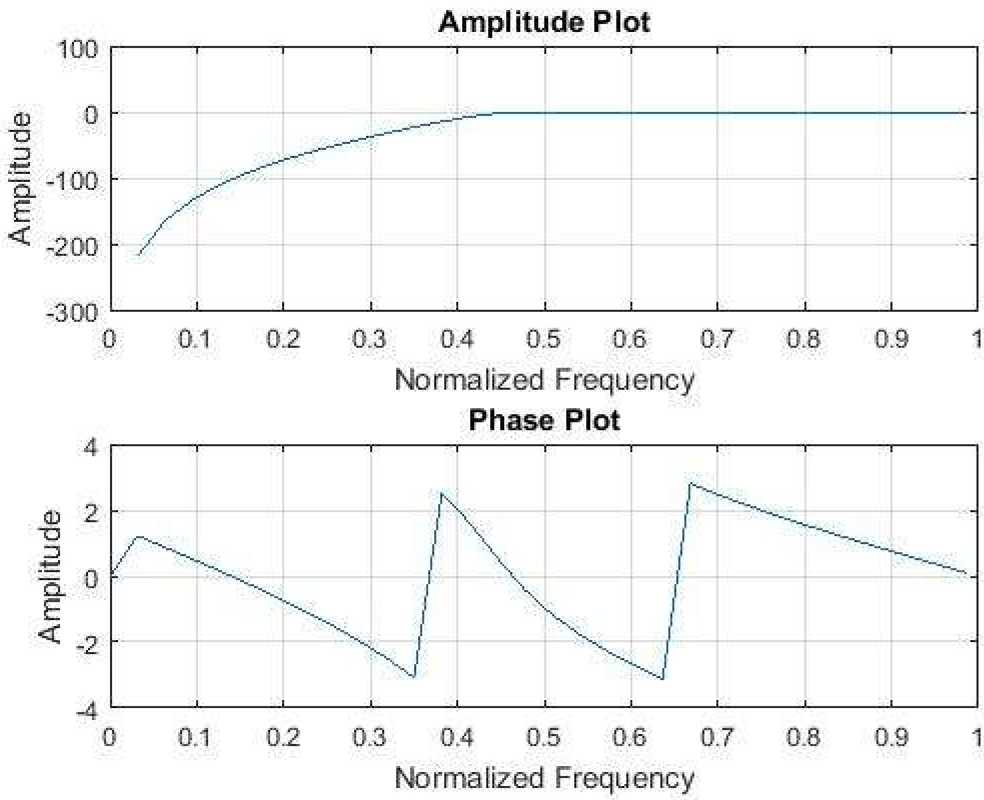
Enter the value of stop band ripple - 40

Enter the value of pass band frequency - 2000

Enter the value of stop band frequency - 3000

Enter the vaue of sampling frequency – 10000

Figure:


Amplitude Plot
100
0
-100
-200
-300
Amplitude
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Normalized Frequency
Phase Plot
4
2
0
-2
-4
Amplitude
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Normalized Frequency