INDEX

Date: __________

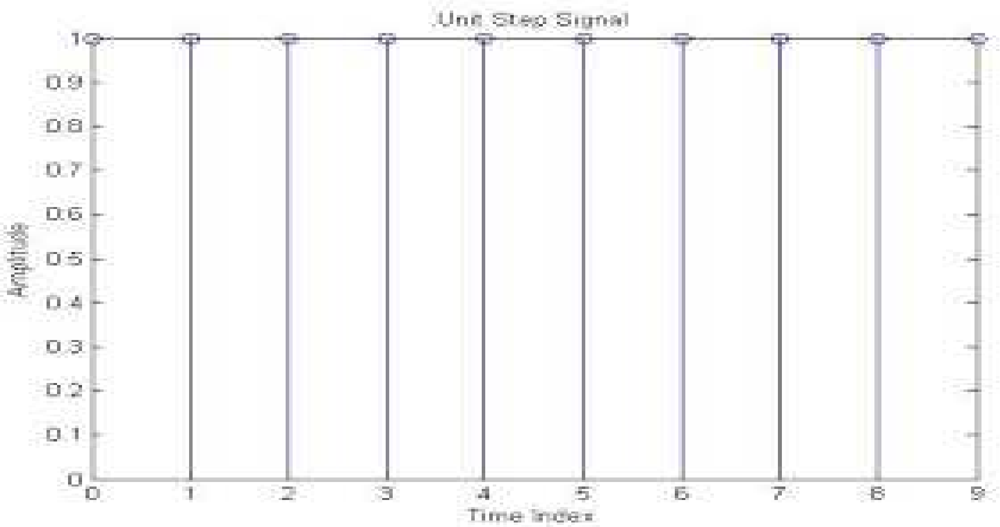
Practical No.: 1

Aim: Write a MATLAB program to generate the following elementary signals:

iv)	Unit step sequence	iv) Sine signal
v)	Unit step signal	v) Cosine signal
vi)	Unit ramp	vi) Exponential signal

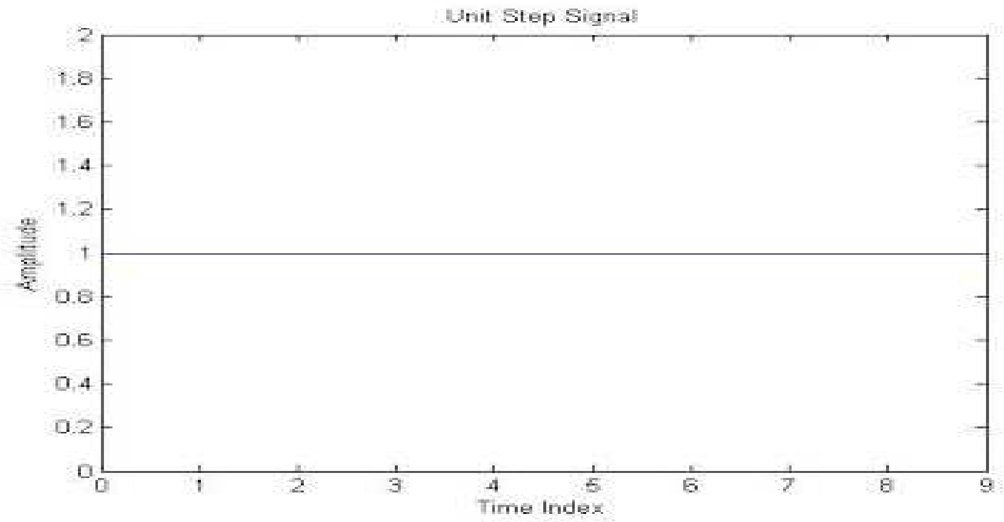
(i) MATLAB program to generate Unit Step sequence.

```
n=input ('Enter the length of the step sequence N=');
                   % Get the length of the require sequence from the user
t=0:n-1;           % defines the time axis
y=ones(1,n);     % defines an 1 x n matrix which is filled with ones
stem(t,y);         %displays the data as lines
ylabel ('Amplitude');                    % name the Y axis
xlabel ('Time Index');                   %Name the x axis
TITLE ('Unit Step Signal');              % Giving the title for the plot
```



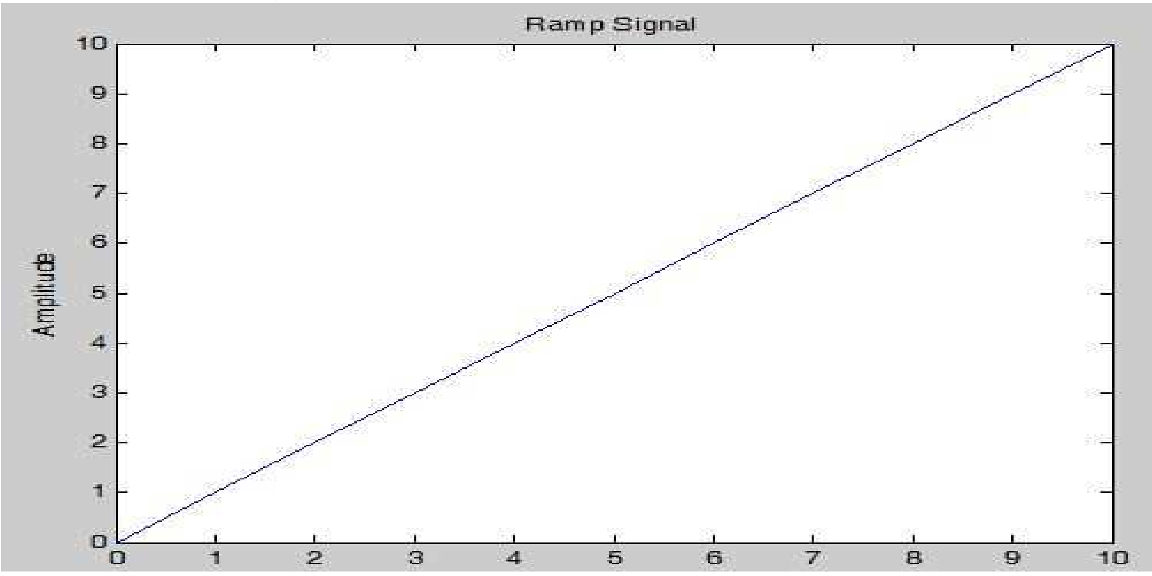
(ii) MATLAB program to generate Unit Step signal.

```
n=input ('Enter the lenght of the step sequence N=');
                    % Get the length of the require sequence from the user
t=0:n-1;           % defines the time axis
y=ones(1,n);       % defines an 1 x n matrix which is filled with one
plot(t,y);         % Plot the graph
ylabel ('Amplitude');              % name the Y axis
xlabel ('Time Index');              %Name the x axis
TITLE ('Unit Step Signal');       % Giving the title for the plot
```



(iii) MATLAB program to generate Unit ramp Signal.

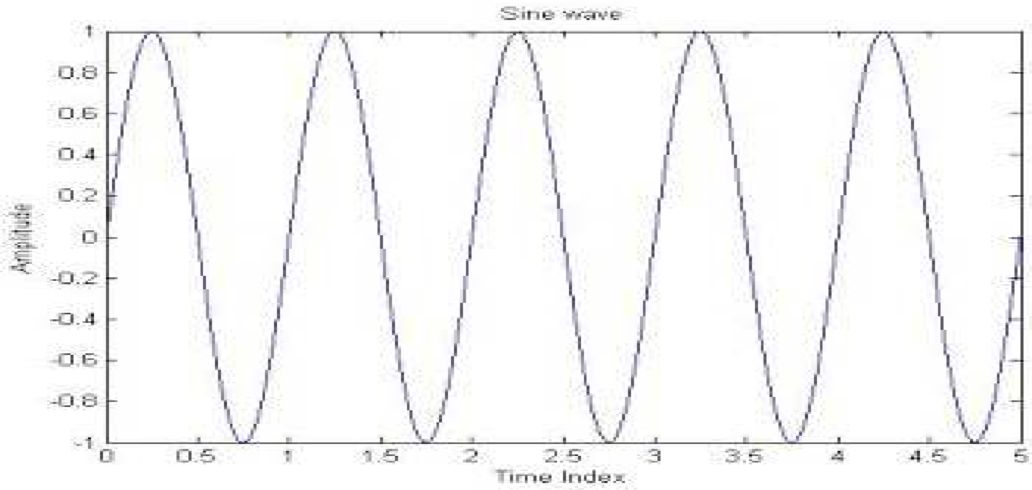
```
n=input ('Enter the length of the sequence N= ');
t=0:n;
y=t;
plot(t,y);
ylabel ('Amplitude');
xlabel ('Time Index');
TITLE ('Ramp Signal');
```



(iv) MATLAB program to generate Sine Wave Signal.

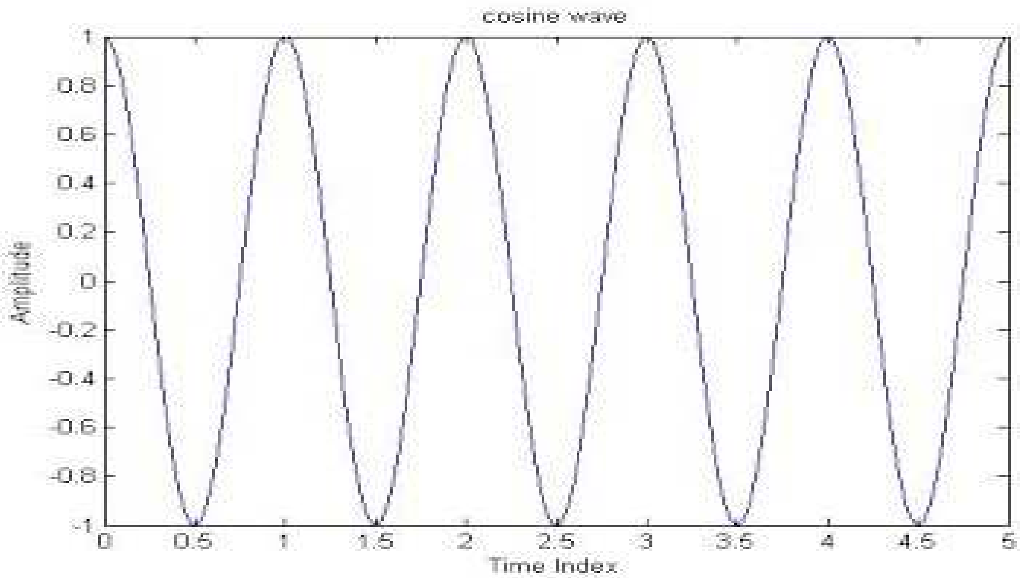
```
f= input('enter the frequency in hertz of the sine wave');
t=0:.0001:5;
y=sin(2*pi*f*t);
plot(t,y);
ylabel ('Amplitude');
xlabel ('Time Index');
```

```
TITLE ('Sine wave');
```



(v) MATLAB program to generate Cosine Wave Signal.

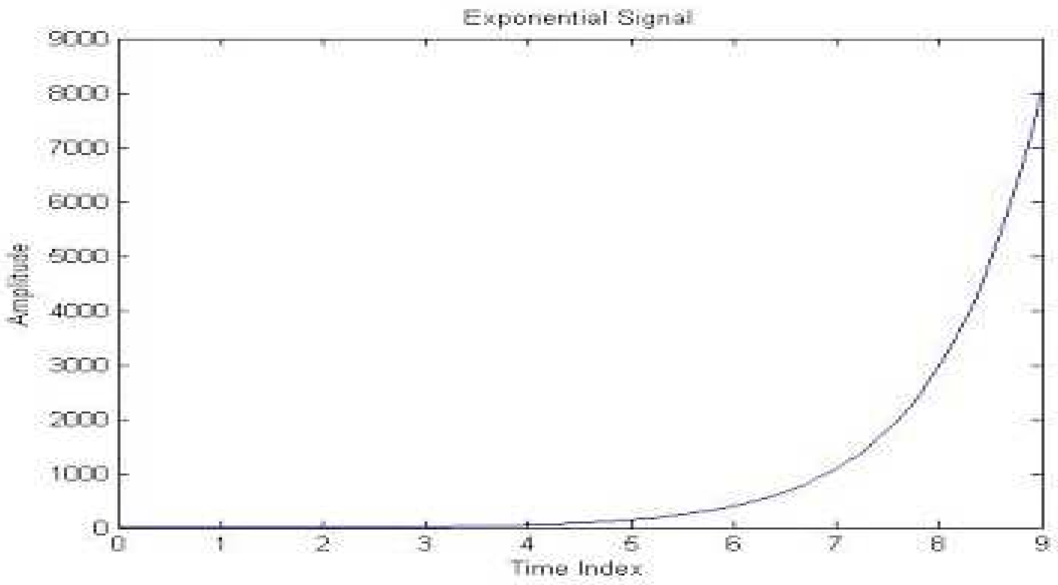
```
f= input('enter the frequency in hertz of the sine wave');
t=0:.0001:5;
y=cos(2*pi*f*t);
plot(t,y);
ylabel ('Amplitude');
xlabel ('Time Index');
TITLE ('cosine wave');
```



(vi) MATLAB program to generate Exponential signal Signal.

```
n=input('Enter the duration of the signal N = ');
a=input ('Enter the scaling factor a = ');
t=0:.1:n-1;
y=exp(a*t);
```

```
plot(t,y);
ylabel ('Amplitude');
xlabel ('Time Index');
TITLE ('Exponential Signal');
```



Date: __________

Practical No.: 2

Aim: Write a MATLAB program to generate the following sequences.

a) 0.5*n*cos(2*∏*n+(∏/4))

b) x1=cos((2*∏*n)/256)

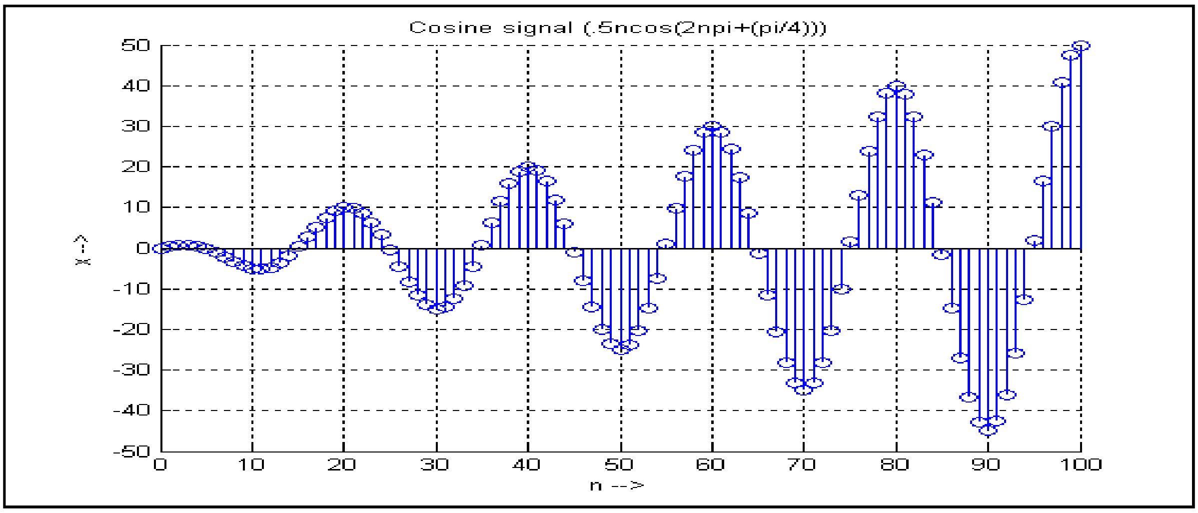
x2=cos((6*∏*n)/256)

x=x1+x2

MATLAB Program.

```
% Matlab program <cosine_signal.m>
% This program is for generating the given cosine signal

clc;                                    %Clear the window
close all;                              %Close all files
clear all;                              %Clear the screen

n=0:1:100;                              %Range of n
y=cos(.05*((2*pi*n)+(pi/4)));
x=.5*n.*y
stem(n,x);
grid on;
title('Cosine signal (.5ncos(2npi+(pi/4)))');
xlabel('n -->');
ylabel('x -->');
```

MATLAB Figure:

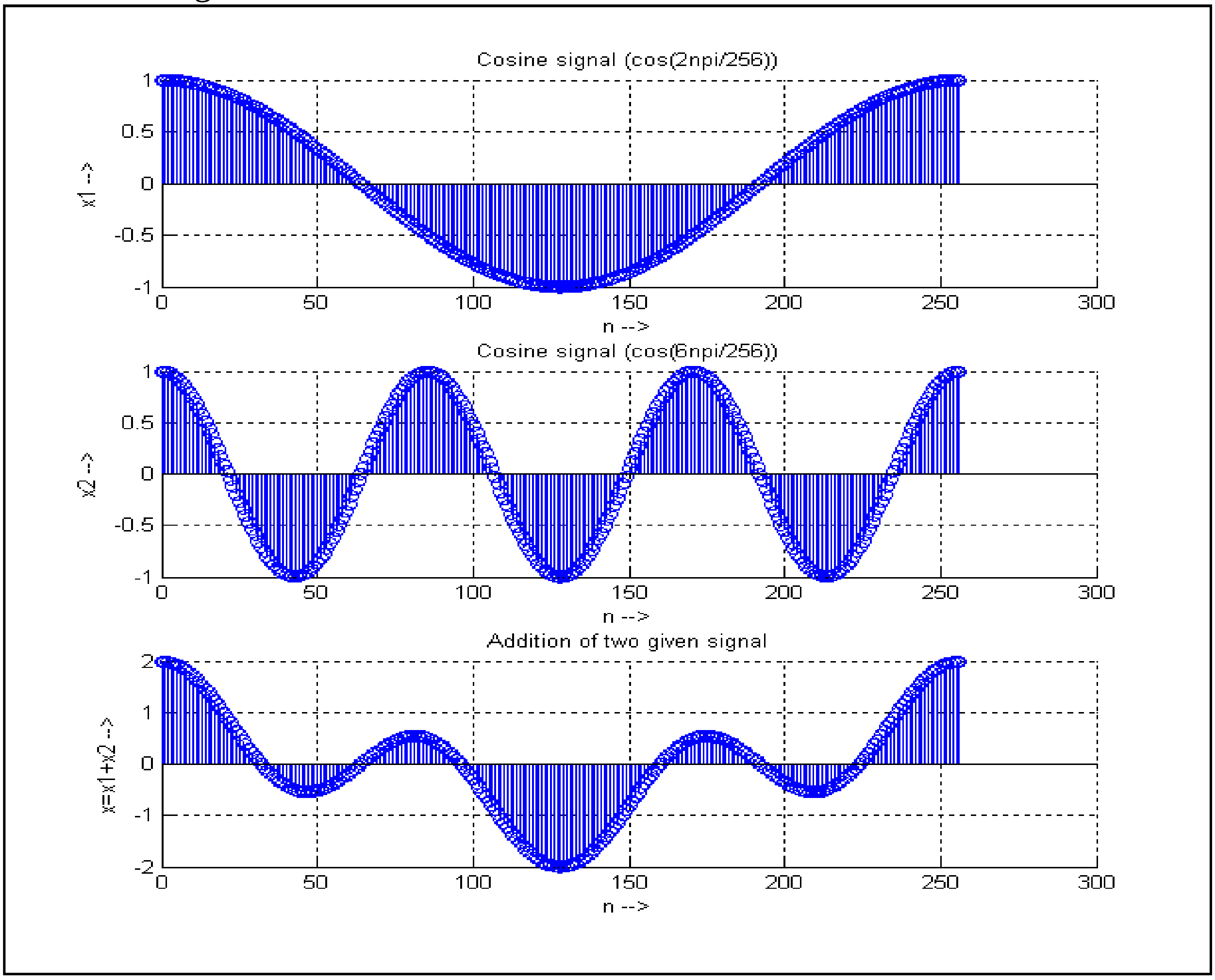


MATLAB Program.

```
% Matlab program <addition_of_signal.m>
% This program is for adding the given two cosine signal

clc;                                    %Clear the window
close all;                              %Close all files
clear all;                              %Clear the screen


n=0:1:256                               %Range of n

x1=cos(2*pi*n/256);
subplot(3,1,1);
stem(n,x1);
grid on;
title ('Cosine signal (cos(2npi/256))');
xlabel('n -->');
ylabel('x1 -->');

x2=cos(6*pi*n/256);
subplot(3,1,2);
stem(n,x2);
grid on;
title ('Cosine signal (cos(6npi/256))');
xlabel('n -->');
ylabel('x2 -->');
```

```
x=x1+x2;
subplot(3,1,3);
stem(n,x);
grid on;
title ('Addition of two given signal');
xlabel('n -->');
ylabel('x = x1+x2 -->');
```

MATLAB Figure:



CONCLUSION:

Date: __________

Practical No.: 3

Aim : Write a MATLAB program for following:

a) To illustrate the effect of up-sampling in frequency domain.

b) To illustration of Interpolation process.

MATLAB Program.

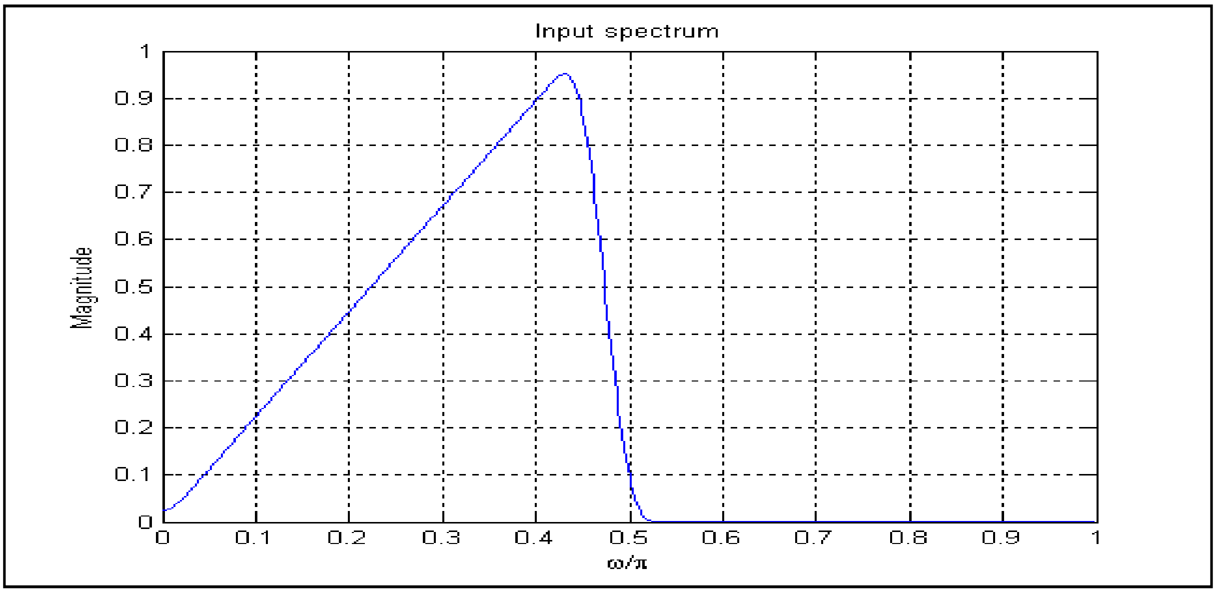
```
% Matlab program <Effect_upsampling_freq.m>
% This program gives effect of up sampling in frequency domain.

clc;                                              %Clear the window
close all;                               `         %Close all files
clear all;                                        %Clear the screen

freq = [0 0.45 0.5 1];
mag = [0 1 0 0];
x = fir2(99, freq, mag);
[Xz, w] = freqz(x, 1, 512);                       % Evaluate &plot input spectrum

figure(1)
plot(w/pi, abs(Xz));
grid on;
xlabel('\omega/\pi');
ylabel('Magnitude');
title('Input spectrum');

L = input('Enter the up-sampling factor: ');      % Generate up-sampled sequence

y = zeros(1, L*length(x));
y([1: L: length(y)]) = x;
[Yz, w] = freqz(y, 1, 512);                       % Evaluate &plot o/p  spectrum

figure(2);
plot(w/pi, abs(Yz));
grid on;
xlabel('\omega/\pi');
ylabel('Magnitude');
```

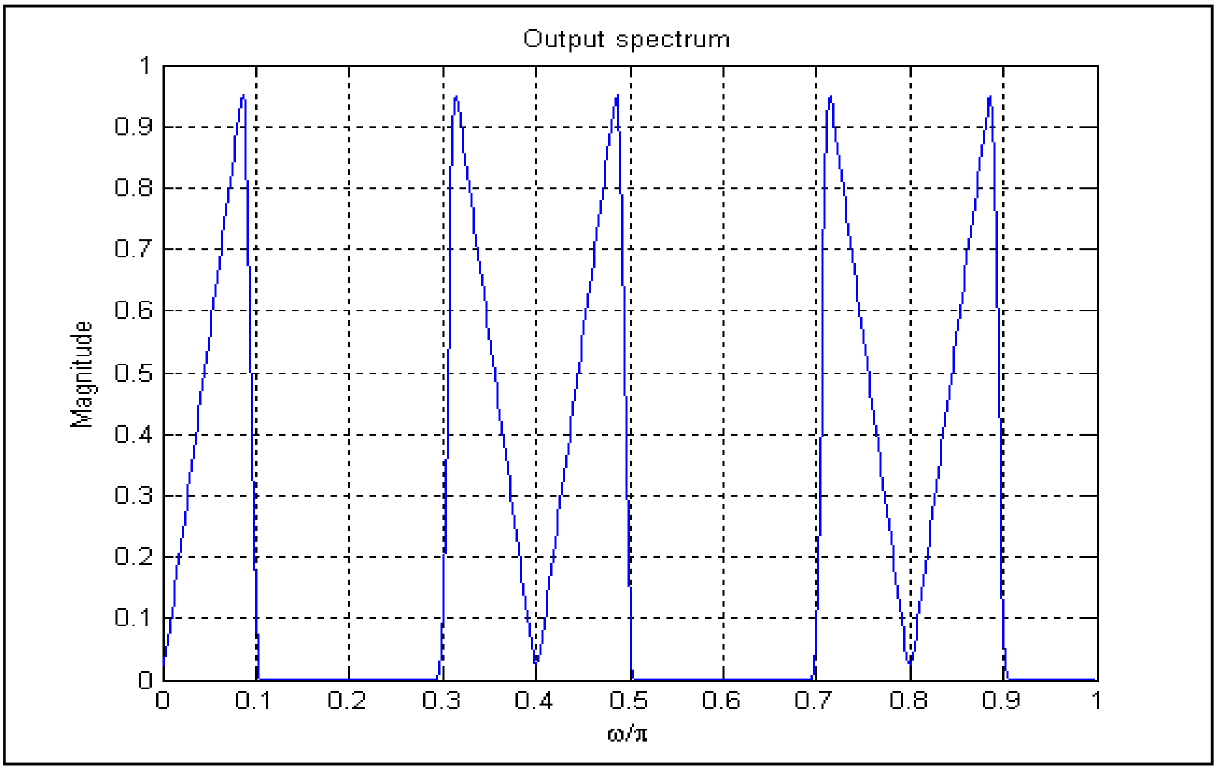
title('Output spectrum');

COMMAND WINDOW

Enter the up-sampling factor: 5

MATLAB Figures.





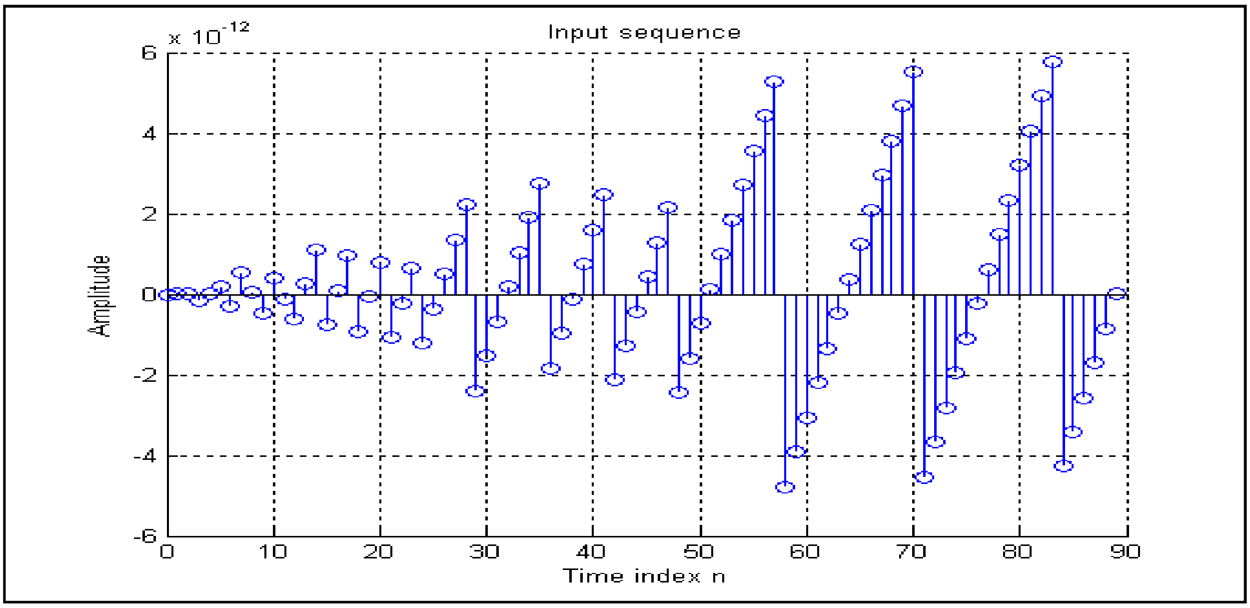
MATLAB Program.

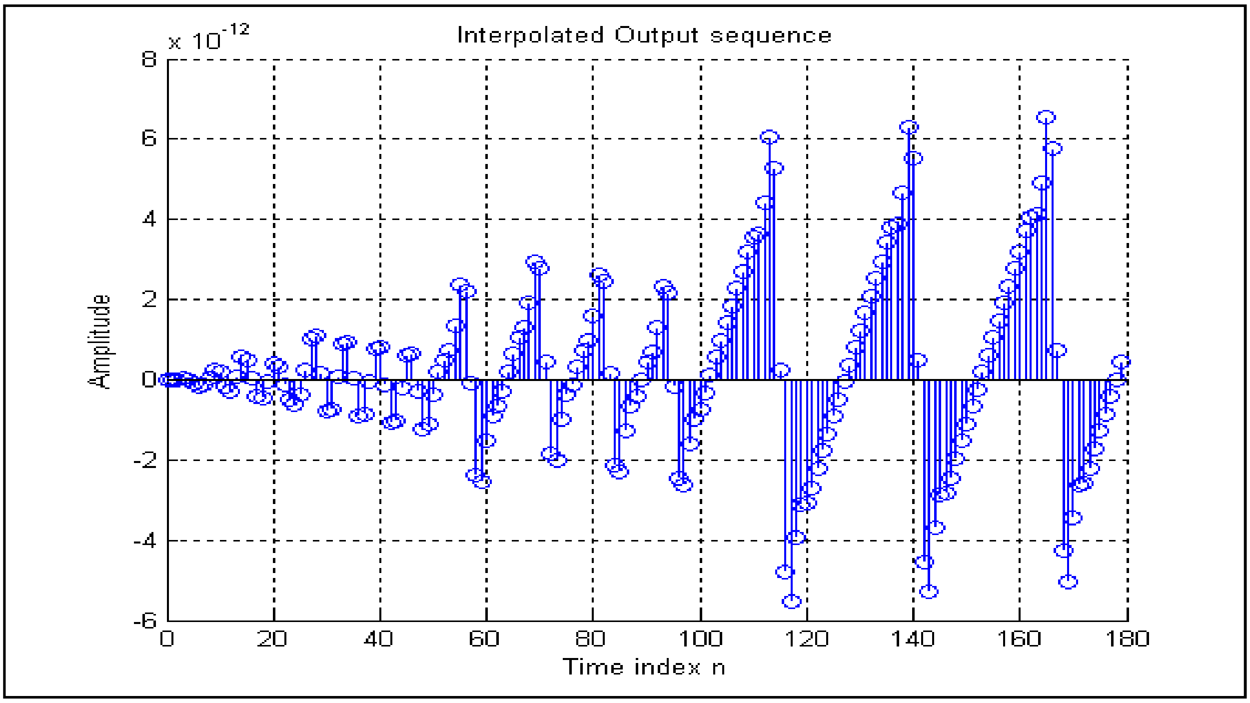
```
% Matlab program <Interpolation.m>
% This program illustrate the interpolation process.

clc;                                          %Clear the window
close all;                 `                  %Close all files
clear all;                                    %Clear the screen


N = input('Length of input signal: ');
L = input('Up-sampling factor: ');
f1 = input('Frequency of first sinusoid: ');
f2 = input('Frequency of second sinusoid: ');

n = 0:N-1;                                    % Generate the input sequence
x = sin(2*pi*f1*n) + sin(2*pi*f2*n);

y = interp(x,L);                              % Generate interpolated o/p

figure(1);                                    % Plot  input & output sequences
stem(n,x(1:N));
title('Input sequence');
xlabel('Time index n');
ylabel('Amplitude');
grid on;

m=0:N*L-1;
figure(2);
stem(m,y(1:N*L));
title(' Interpolated Output sequence');
xlabel('Time index n');
ylabel('Amplitude');
grid on;
```

COMMAND WINDOW

```
Length of input signal: 90
Up-sampling factor: 2
Frequency of first sinusoid: 50
Frequency of second sinusoid: 100
```

MATLAB Figure.


x 10^-12
Input sequence
Amplitude
Time index n
6
4
2
0
-2
-4
-6
0
10
20
30
40
50
60
70
80
90


x 10^-12
Interpolated Output sequence
Amplitude
Time index n
8
6
4
2
0
-2
-4
-6
0
20
40
60
80
100
120
140
160
180

CONCLUSION:

Date: __________

Practical No.: 4

Aim: Write a MATLAB program to illustrate Moving Average Filter for signal smoothing.

MATLAB Program.

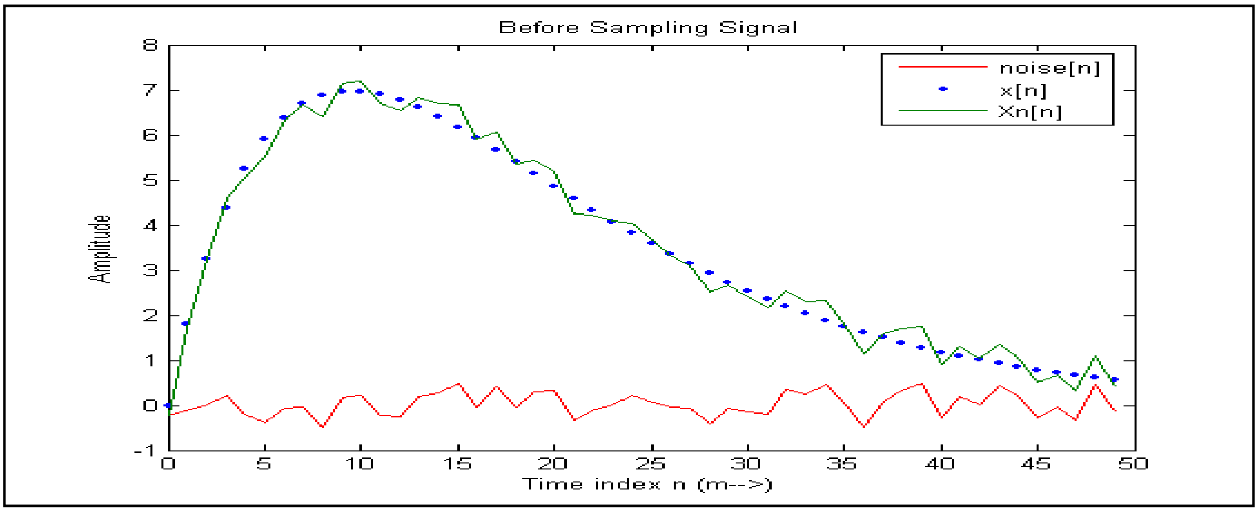
```
% Matlab program <moving_average.m>
% This program is for generating the given cosine signal

clc;                                        %Clear the window
close all;                        `         %Close all files
clear all;                                  %Clear the screen

R = 50;
noise = rand(R,1)-0.5;
m = 0:1:R-1;
x = 2*m.*(0.9.^m);
Xn  = x + noise';

figure(1)
plot(m,noise,'r-',m,x,'.',m,Xn);
title ('Before Sampling Signal');
xlabel('Time index n (m-->)');
ylabel('Amplitude');
legend('noise[n]','x[n]','Xn[n]');

M = input('Number of input samples = ');
b = ones(M,1)/M;
y = filter(b,1,Xn);

M1 = input('Number of input samples = ');
b1 = ones(M1,1)/M1;
y1 = filter(b1,1,Xn);

figure(2)

plot(m,x,'.',m,y,'--',m,y1);
title ('After Sampling Signal');
legend('x[n]','y[n]','y1[n]');
xlabel ('Time index n');
```

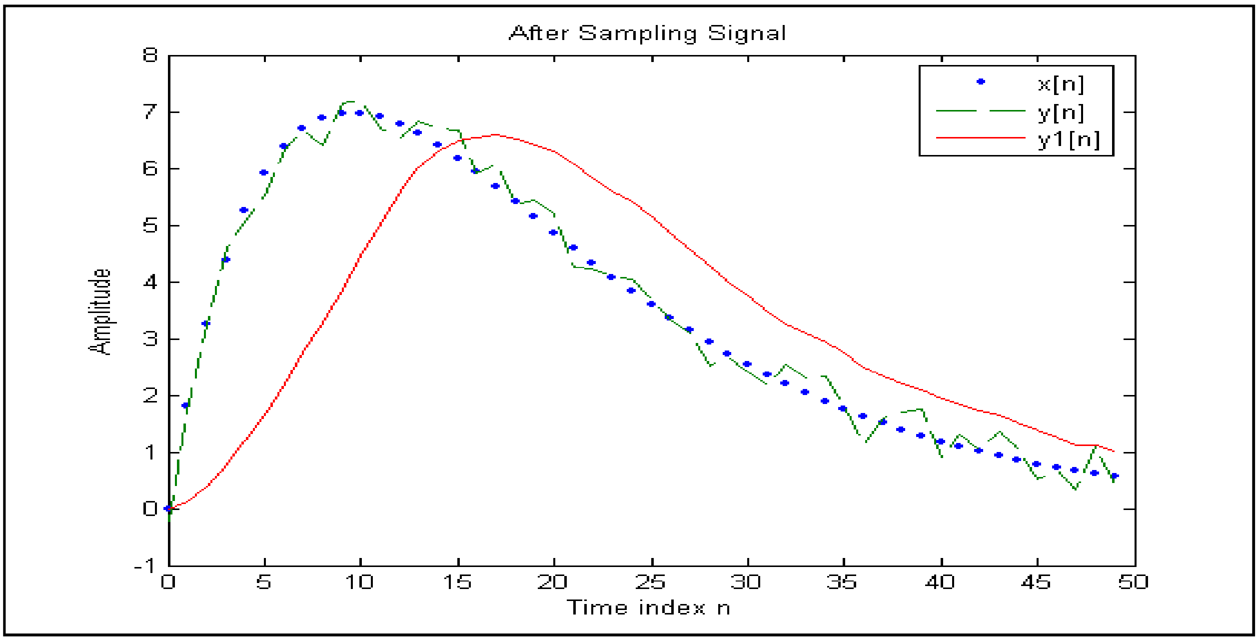
ylabel('Amplitude');

COMMAND WINDOW

Number of input samples = 1
Number of input samples = 12

MATLAB Figure:





CONCLUSION:

Date: __________

Practical No.: 5

Aim: Write a MATLAB program to find the linear convolution of two sequences.

a) Without using MATLAB convolution function.

b) Using MATLAB convolution function.

MATLAB Program.

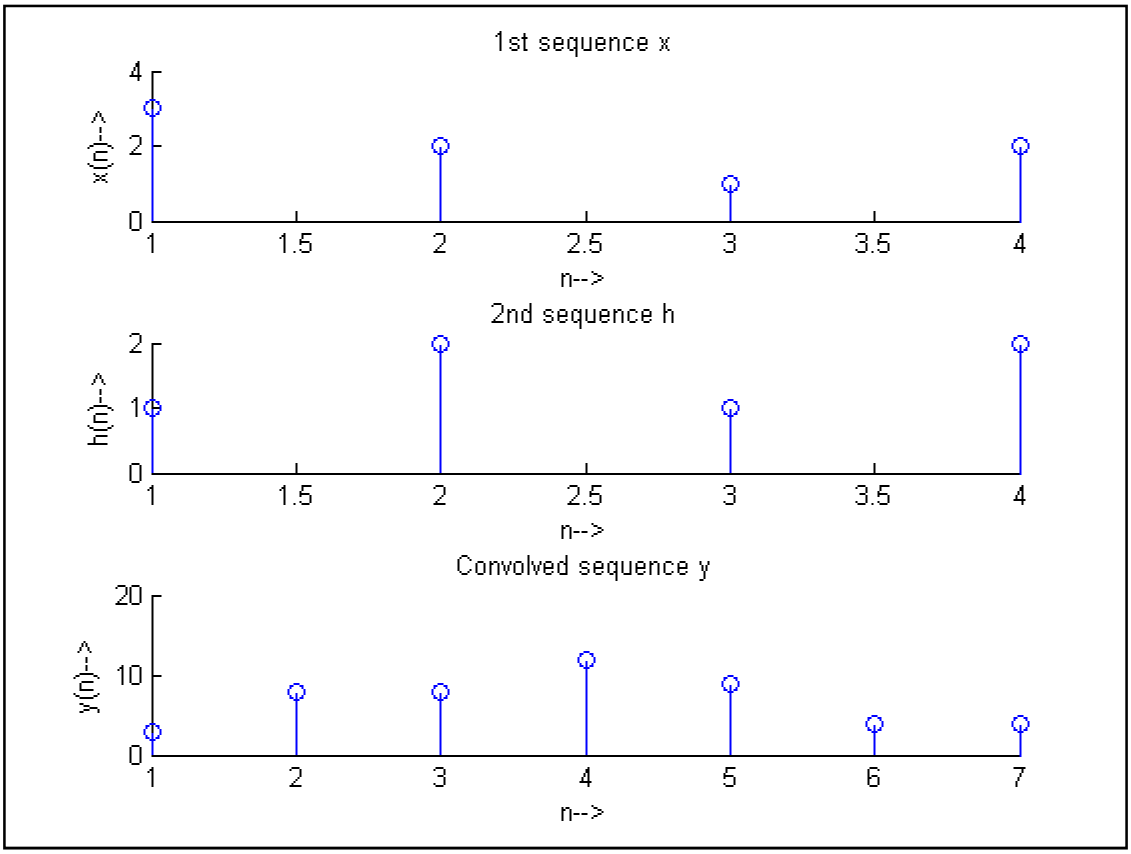
```
% Matlab program <linear_convolution.m>
% This program performs convolution of two positive sequences.

clc;                                          %Clear the window
close all;                         `          %Close all files
clear all;                                    %Clear the screen

x=input('Enter the 1st sequence x:');         %Enter the 1st sequence
subplot(3,1,1);
stem(x);
title('1st sequence x');
xlabel('n-->');
ylabel('x(n)-->');

h=input('Enter the 2nd sequence h:');         %Enter the 2nd  sequence
subplot(3,1,2);
stem(h);
title('2nd sequence h');
xlabel('n-->');
ylabel('h(n)-->');

x1=length(x);                                 %Finds the length of  x
h1=length(h);                                 %Finds the length of  h
n1=length(x)+length(h)-1;                     %Gives the max. Range of n

for n=1:n1                                    %MATLAB  starts any array
                                                 from index 1
   sum(n)=0;
         for k=1:x1                           %Specifies summation limit
```

```
        m=n+1-k;                                    %m=n-k
        if m<=h1                                    %To keep m below h1
            if k<n+1
            sum(n)=sum(n)+(x(k)*h(m));              %Convolution
            end
        else
        end
      end
  y(n)=sum(n);
end


disp ('The convolution of x(n)& h(n)is y(n)= x(n)*h(n)');
y                                                   %Convolved output

subplot(3,1,3);
stem(y);
title('Convolved sequence y');
xlabel('n-->');
ylabel('y(n)-->');
```

COMMAND WINDOW

```
Enter the sequence x:[3 2 1 2]
Enter the sequence h:[1 2 1 2]
The convolution of x(n)& h(n)is y(n)= x(n)*h(n)

y =    3    8    8   12    9    4    4
```

MATLAB Figures:



MATLAB Program.

```
% Matlab program <linear_convolution.m>
% This program performs convolution of two positive sequences by MATLAB function.

clc;                                        %Clear the window
close all;                                  %Close all files
clear all;                                  %Clear the screen

x=input('Enter the 1st sequence x:');       %Enter the 1st sequence

subplot(3,1,1);
stem(x);
title('1st sequence x');
xlabel('n-->');
ylabel('x(n)-->');

h=input('Enter the 2nd sequence h:');       %Enter the 2nd sequence
subplot(3,1,2);
stem(h);
title('2nd sequence h');
```

```
xlabel('n-->');
ylabel('h(n)-->');

y=conv(x,h);                                              %convolution function
disp ('The convolution of x(n)& h(n) is  y(n) =  x(n)*h(n) ')
y

subplot(3,1,3);
stem(y);
title('Convolved sequence y');
xlabel('n-->');
ylabel('y(n)-->');
```

COMMAND WINDOW

```
Enter the sequence x:[3 2 1 2]
Enter the sequence h:[1 2 1 2]
The convolution of x(n)& h(n) is  y(n) =  x(n)*h(n)

y =     3     8     8    12     9     4     4
```

CONCLUSION:

Date: __________

Practical No.:6

Aim : Write a MATLAB program for following:

a) To obtain partial fraction expansion of rational Z-transform.

b) To obtain Z-transform from partial fraction expansion.

c) To obtain power series expansion of Z-transform.

d) Stability test for Z-transform.

MATLAB Program.

```
% Matlab program <PFE_of_Ztrans.m>
% This program gi.ves partial fraction expansion of  Z-transform.

clc;                                                    %Clear the window
close all;                        `                     %Close all files
clear all;                                              %Clear the screen

num = input('Enter the numerator coefficients: ');      % num=(1+(1/z))
den = input('Enter the denominator coefficients: ');    %den=(1-(5/6)(1/z)+(1/6)(1/z^2))

[r,p,k] = residuez(num,den);
disp('Residues');
disp(r')

disp('Poles');
disp(p')

disp('Constants');
disp(k)
```

COMMAND WINDOW

```
Enter the numerator coefficients: [1 1]

Enter the denominator coefficients: [1 -5/6 1/6]
```

Residues
    9.0000   -8.0000

Poles
    0.5000   0.3333

Constants []

MATLAB Program.

```
% Matlab program <Ztrans_from_PFE.m>
% This program gives Z-transformer from partial fraction expansion.

clc;                                        %Clear the window
close all;                  `               %Close all files
clear all;                                  %Clear the screen

r = input('Enter the residues: ');
p = input('Enter the poles: ');
k = input('Enter the constants: ');

[num, den] = residuez(r,p,k);
disp('Numerator polynomial coefficients');
disp(num)

disp('Denominator polynomial coefficients');
disp(den)
```

COMMAND WINDOW

```
Enter the residues: [9 -8]

Enter the poles: [.5 .3333]

Enter the constants: []

Numerator polynomial coefficients
        1.0000    1.0003

Denominator polynomial coefficients
        1.0000   -0.8333    0.1666
```

MATLAB Program.

```
% Matlab program <PSE_of_ztrans.m>
% This program gives power series expansion of Z-transform.

clc;                                                    %Clear the window
close all;                                              %Close all files
clear all;                                              %Clear the screen

N = input('Enter the length of output vector: ');
num = input('Enter the numerator coefficients: ');      % 1/(1-2(1/z))
den = input('Enter the denominator coefficients: ');

x = [1 zeros(1, N-1)];                                  % Compute the desired number
                                                        of inverse transform coefficients
y = filter(num, den, x);
disp('Coefficients of the power series expansion');
disp(y)
```

COMMAND WINDOW

```
Enter the length of output vector: 4

Enter the numerator coefficients: [1]

Enter the denominator coefficients: [1 -2]

Coefficients of the power series expansion
    1    2    4    8
```

MATLAB Program.

```
% Matlab program <Stability_test.m>
% This program is used for stabilities test of  Z-transform.

clc;                                                    %Clear the window
close all;                                              %Close all files
clear all;                                              %Clear the screen
```

```
num = input('Enter the numerator vector: ');
den = input('Enter the denominator vector: ');
N = max(length(num),length(den));

x = 1;
y0 = 0;
S = 0;
zi = zeros(1,N-1);

for n = 1:1000
    [y,zf] = filter(num,den,x,zi);
    if abs(y) < 0.000001,
            break,
    end
    x = 0;
    S = S + abs(y);
    y0 = y;zi = zf;
end

if n < 1000
    disp('Stable Transfer Function');
else
    disp('Unstable Transfer Function');
end
```

COMMAND WINDOW

```
Enter the numerator vector: [1 1]

Enter the denominator vector: [1 -5/6 1/6]

Stable Transfer Function
```

CONCLUSION:

Date: ___________

Practical No.: 7

Aim : Write a MATLAB program to:

a) Obtain N-point DFT of a sequence.

b) Obtain N-point IDFT of a sequence.

c) Compute linear convolution using DFT.

MATLAB Program.

```
% Matlab program <DFT.m>
% This program performs  DFT of the entered sequence.

clc;                                              %Clear the window
close all;                          `             %Close all files
clear all;                                        %Clear the screen


N=input('Enter the N-point to be calculated for DFT: ');
x=input('Enter the sequence: ');

for k=2:1:N+1
sum(k-1)=0;
   for n=1:1:N
       y(n)=x(n)*exp(-(j*2*pi*(n-1)*(k-2)/N));
       sum(k-1)=sum(k-1)+y(n);
   end
end

z=sum;

fprintf('The %d point DFT of the given input sequence is ', N)
z
```

COMMAND WINDOW

Enter the N-point to be calculated for DFT: 4
Enter the sequence: [0 1 2 3]
The 4 point DFT of the given input sequence is

z =

6.0000 -2.0000 + 2.0000i -2.0000 - 0.0000i -2.0000 - 2.0000i

MATLAB Program.

```
% Matlab program <IDFT.m>
% This program performs  IDFT of the entered sequence.

clc;                                    %Clear the window
close all;                    `         %Close all files
clear all;                              %Clear the screen

N=input('Enter the N-point to be calculated for DFT: ');
x=input('Enter the sequence: ');

for k=2:1:N+1
sum(k-1)=0;
    for n=1:1:N
        y(n)=x(n)*exp((j*2*pi*(n-1)*(k-2)/N));
        sum(k-1)=sum(k-1)+y(n);
    end
end

out=(1/N)*sum;

z=out;

fprintf ('The %d point DFT of the given input sequence is ', N)
z
```

COMMAND WINDOW

```
Enter the N-point to be calculated for DFT: 4
Enter the sequence: [2 1+i 0 1-i]
The 4 point DFT of the given input sequence is

z =

   1.0000           -0.0000 + 0.0000i   0.0000 + 0.0000i   1.0000 - 0.0000i
```

MATLAB Program.

```
% Matlab program <linear_convolution_DFT.m>
% This program performs  linear convolution of the entered sequence using DFT.

clc;                                              %Clear the window
close all;                              `         %Close all files
clear all;                                        %Clear the screen

x=input('Enter the sequence x: ');
h=input('Enter the sequence h: ');
xa=[x zeros(1,length(h)-1)];
ha=[h zeros(1,length(x)-1)];

N=length(xa);
for k1=2:1:(N+1)
      sum1(k1-1)=0;
      for n1=1:1:N
            y1(n1)=xa(n1)*exp(-(j*2*pi*(n1-1)*(k1-2)/N));
            sum1(k1-1)=sum1(k1-1)+y1(n1);
     end
end

Xk=sum1;

for k2=2:1:(N+1)
     sum2(k2-1)=0;
     for n2=1:1:N
            y2(n2)=ha(n2)*exp(-(j*2*pi*(n2-1)*(k2-2)/N));
            sum2(k2-1)=sum2(k2-1)+y2(n2);
     end
end
```

```
Hk=sum2;

Yk= Xk.*Hk;

M=length(Yk);




for k=2:1:(M+1)
    sum(k-1)=0;
    for n=1:1:M
        y(n)=Yk(n)*exp((j*2*pi*(n-1)*(k-2)/M));
        sum(k-1)=sum(k-1)+y(n);
    end
end

Y=(1/M)*sum;

disp('The Linear Convolution by DFT of given sequences is:' )
Y
```

COMMAND WINDOW

```
Enter the sequence x: [1 2 2 1]
Enter the sequence h: [1 2 3]

The Linear Convolution by DFT of given sequences is:

Y = 1.0000 - 0.0000i    4.0000 - 0.0000i    9.0000 - 0.0000i    11.0000 - 0.0000i
    8.0000 + 0.0000i    3.0000 - 0.0000i
```

CONCLUSION:

Date: __________

Practical No.: 8

Aim: Write a MATLAB program to design following filters.

a) To design Butterworth Low Pass Filter

b) To design Butterworth High Pass Filter

c) To design Butterworth Band Pass Filter

d) To design Butterworth Band Reject Filter

MATLAB Program.

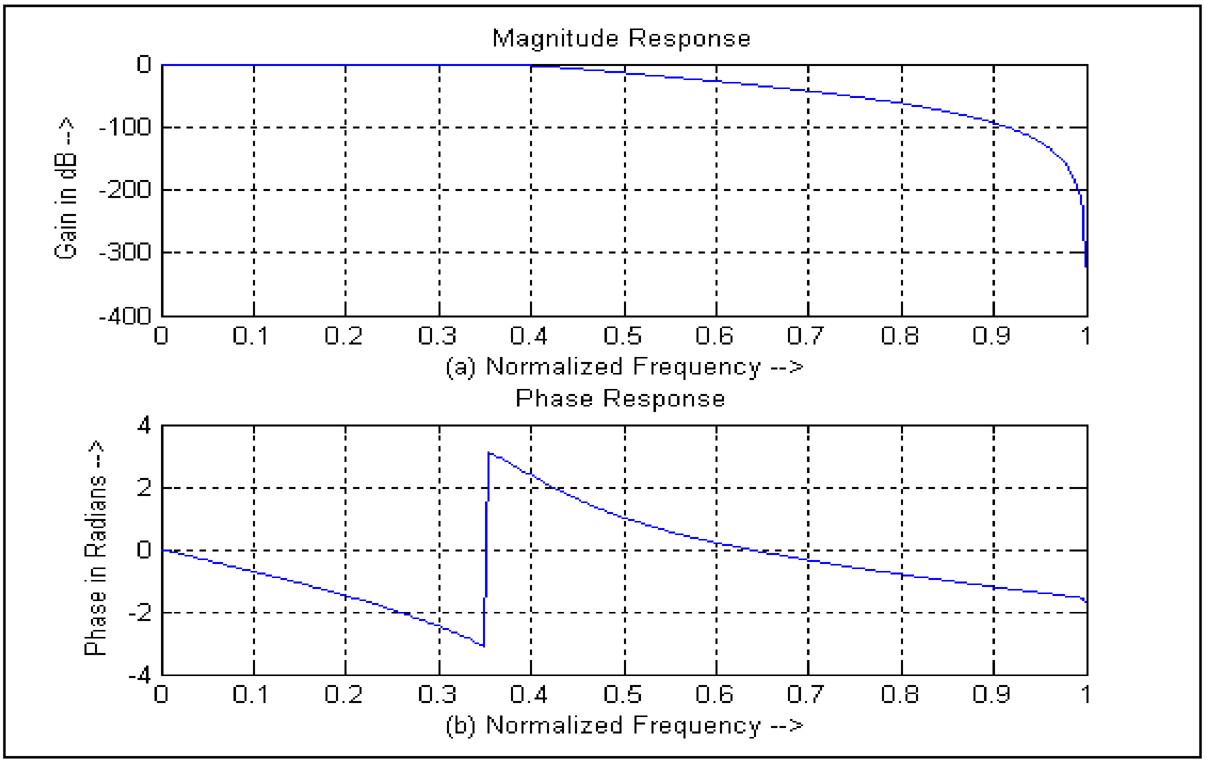
```
% Matlab program <butter_lpf.m>
% This program gives Butterworth low pass filter.

clc;                                              %Clear the window
close all;                          `             %Close all files
clear all;                                        %Clear the screen

Ap=input('Enter the passband attenuation in dB: ');
As=input('Enter the stopband attenuation in dB: ');
f1=input('Enter the Passband digital frequency(between 0 to 1): ');
f2=input('Enter the Stopband digital frequency(between 0 to 1): ');
fs=1;
E=((10^(.1*Ap))-1);
L=((10^(.1*As))-1);

N=ceil((0.5)*(log10(L/E))/log10(f2/f1))           % to find order
fc=f1/(E^(1/(2*N)))

[b,a]=butter(N,fc)
w=0:.01:pi;
[H,NF]=freqz(b,a,w);

M=20*log10(abs(H));                                %digital
subplot(2,1,1)
plot((NF/pi),M)
grid on;
title('Magnitude Response');
```

```
xlabel(' (a) Normalized Frequency -->');
ylabel(' Gain in dB -->');

an=angle(H);
subplot(2,1,2)
plot((NF/pi),an)
grid on;
title('Phase Response');
xlabel(' (b) Normalized Frequency -->');
ylabel(' Phase in Radians -->');
```

COMMAND WINDOW

```
Enter the passband attenuation in dB: 1
Enter the stopband attenuation in dB: 15
Enter the first digital normalised freq.(in between 0 to 1): .35
Enter the second digital normalised freq.(in between 0 to 1): .6

N =5
fc =0.4006
b = 0.0221    0.1104    0.2208    0.2208    0.1104    0.0221
a = 1.0000   -0.9790    0.9695   -0.3832    0.1104   -0.0112
```

MATLAB figure:



MATLAB Program.

```
% Matlab program <butter_hpf.m>
% This program gives Butterworth high pass filter.

clc;                                                    %Clear the window
close all;                          `                   %Close all files
clear all;                                              %Clear the screen


Ap=input('Enter the passband attenuation in dB: ');
As=input('Enter the stopband attenuation in dB: ');
f1=input('Enter the Passband digital frequency(between 0 to 1): ');
f2=input('Enter the Stopband digital frequency(between 0 to 1): ');
fs=1;

E=((10^(.1*Ap))-1);
L=((10^(.1*As))-1);

N=ceil((0.5)*(log10(L/E))/log10(f2/f1))                 % to find order
fc=f1/(E^(1/(2*N)))

[b,a]=butter(N,fc,'high')
w=0:.01:pi;
[H,NF]=freqz(b,a,w);

M=20*log10(abs(H));                                     %digital
subplot(2,1,1)
plot((NF/pi),M)
grid on;
title('Magnitude Response');
xlabel(' (a) Normalized Frequency -->');
ylabel(' Gain in dB -->');

an=angle(H);
subplot(2,1,2)
plot((NF/pi),an)
grid on;
title('Phase Response');
xlabel(' (a) Normalized Frequency -->');
ylabel(' Phase in Radians -->');
```

COMMAND WINDOW

Enter the passband attenuation in dB: 1
Enter the stopband attenuation in dB: 15
Enter the first digital normalised freq.(in between 0 to 1): .35
Enter the second digital normalised freq.(in between 0 to 1): .6

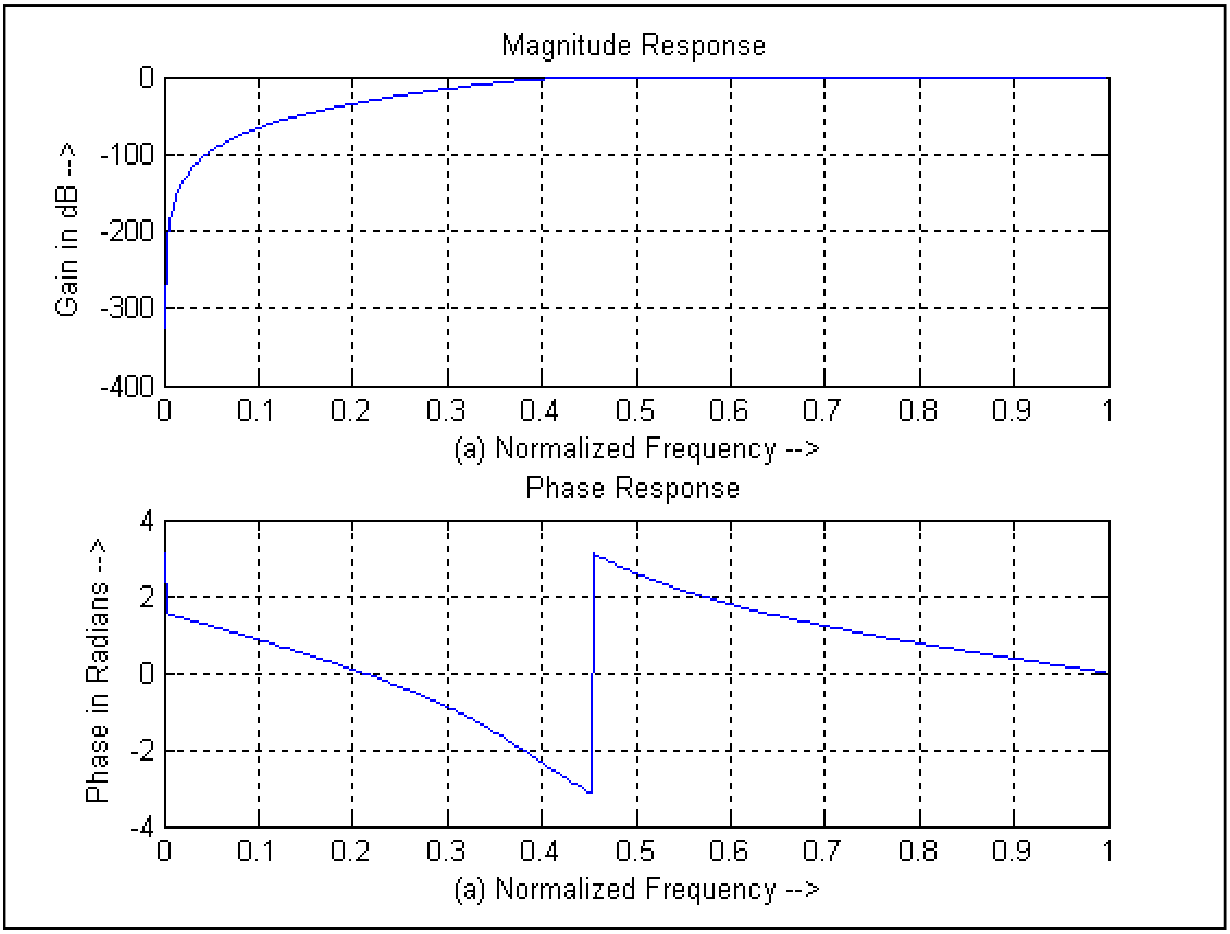
N = 5

fc = 0.4006

b = 0.1079 -0.5396 1.0792 -1.0792 0.5396 -0.1079

a = 1.0000 -0.9790 0.9695 -0.3832 0.1104 -0.0112

MATLAB figure:



MATLAB Program.

```
% Matlab program <butter_bpf.m>
% This program gives Butterworth band pass filter.

clc;                                                      %Clear the window
close all;                          `                     %Close all files
clear all;                                                %Clear the screen

Ap=input('Enter the passband attenuation in dB: ');
As=input('Enter the stopband attenuation in dB: ');
f1=input('Enter the first digital normalised freq.(in between 0 to 1): ');
f2=input('Enter the second digital normalised freq.(in between 0 to 1): ');
fs=1;

E=((10^(.1*Ap))-1);
L=((10^(.1*As))-1);

N=ceil((0.5)*(log10(L/E))/log10(f2/f1))                  % to find order
fc=[f1 f2]

[b,a]=butter(N,fc,'bandpass')
w=0:.01:pi;
[H,NF]=freqz(b,a,w);

M=20*log10(abs(H));                                       %digital
subplot(2,1,1)
plot((NF/pi),M)
grid on;
title('Magnitude Response');
xlabel(' (a) Normalized Frequency -->');
ylabel(' Gain in dB -->');

an=angle(H);
subplot(2,1,2)
plot((NF/pi),an)
grid on;
title('Phase Response');
xlabel(' (b) Normalized Frequency -->');
ylabel(' Phase in Radians -->');
```

COMMAND WINDOW

Enter the passband attenuation in dB: 1
Enter the stopband attenuation in dB: 15
Enter the first digital normalised freq.(in between 0 to 1): .35
Enter the second digital normalised freq.(in between 0 to 1): .6

N = 5

fc = 0.3500   0.6000

b = Columns 1 through 11

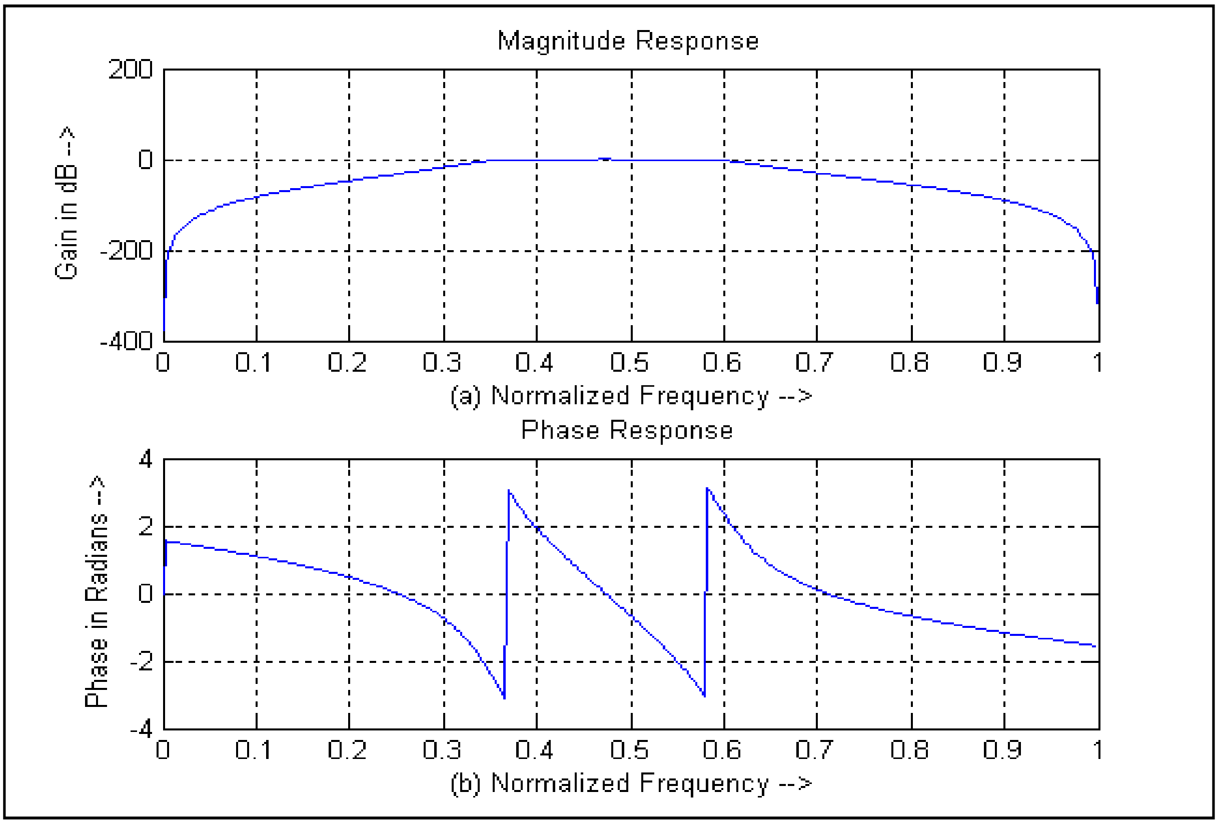
0.0033   0   -0.0164   0   0.0328   0   -0.0328   0   0.0164 0   -0.0033

a = Columns 1 through 11

1.0000   -0.6348   2.6382   -1.3394   3.0780   -1.1743   1.8627   -0.4816   0.5776

-0.0769   0.0723

MATLAB Figure



MATLAB Program.

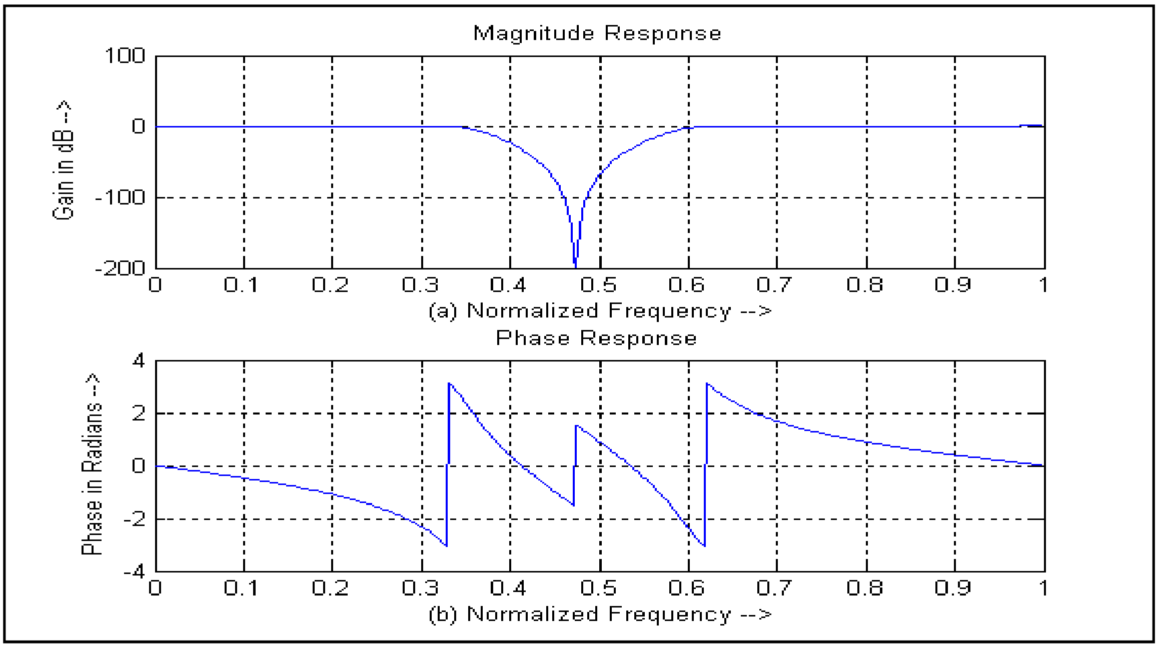
```
% Matlab program <butter_bsf.m>
% This program gives Butterworth band stop filter.

clc;                                              %Clear the window
close all;                              `         %Close all files
clear all;                                        %Clear the screen

Ap=input('Enter the passband attenuation in dB: ');
As=input('Enter the stopband attenuation in dB: ');
f1=input('Enter the first digital normalised freq.(in between 0 to 1): ');
f2=input('Enter the second digital normalised freq.(in between 0 to 1): ');
fs=1;

E=((10^(.1*Ap))-1);
L=((10^(.1*As))-1);

N=ceil((0.5)*(log10(L/E))/log10(f2/f1))          % to find order
fc=[f1 f2]

[b,a]=butter(N,fc,'stop')
w=0:.01:pi;
[H1,NF1]=freqz(b,a,w);

M=20*log10(abs(H));                              %digital
subplot(2,1,1)
plot((NF/pi),M)
grid on;
title('Magnitude Response');
xlabel(' (a) Normalized Frequency -->');
ylabel(' Gain in dB -->');

an=angle(H);
subplot(2,1,2)
plot((NF/pi),an)
grid on;
title('Phase Response');
xlabel(' (b) Normalized Frequency -->');
ylabel(' Phase in Radians -->');
```

COMMAND WINDOW

Enter the passband attenuation in dB: 1
Enter the stopband attenuation in dB: 15
Enter the first digital normalised freq.(in between 0 to 1): .35
Enter the second digital normalised freq.(in between 0 to 1): .6

N = 5
fc = 0.3500   0.6000
b = Columns 1 through 11
0.2689   -0.2284   1.4223   -0.9267   2.9232   -1.3967   2.9232   -0.9267   1.4223
-0.2284   0.2689
a = Columns 1 through 11
1.0000   -0.6348   2.6382   -1.3394   3.0780   -1.1743   1.8627   -0.4816   0.5776
-0.0769   0.0723

MATLAB Figure:



CONCLUSION:

Date: __________

Practical No.: 9

Aim: Write a MATLAB program to design following filters.

a) To design Chebyshev-I Low Pass Filter

b) To design Chebyshev-I High Pass Filter

c) To design Chebyshev-I Band Pass Filter

d) To design Chebyshev-I Band Reject Filter

MATLAB Program.

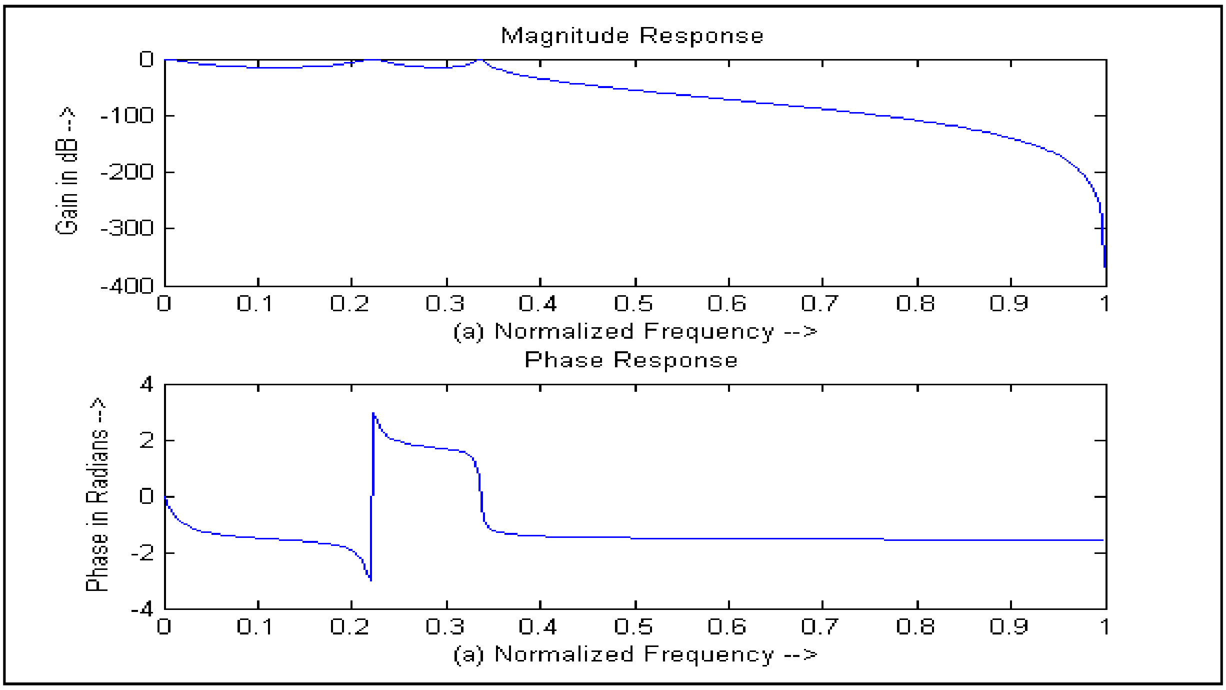
```
% Matlab program <chebyshev1_lpf.m>
% This program gives chebyshev1 low pass filter.

clc;                                                    %Clear the window
close all;                              `               %Close all files
clear all;                                              %Clear the screen

Ap=input('Enter the passband attenuation in dB: ');
As=input('Enter the stopband attenuation in dB: ');
f1=input('Enter the Passband digital frequency(between 0 to 1): ');
f2=input('Enter the Stopband digital frequency(between 0 to 1): ');
fs=1;

E=((10^(.1*Ap))-1);
L=((10^(.1*As))-1);

N=ceil((acosh(sqrt(L/E)))/acosh(f2/f1))
fc=f1

[b,a]=cheby1(N,Ap,fc);
w=0:.01:pi;
[H,NF]=freqz(b,a,w);

M=20*log10(abs(H));

subplot(2,1,1)
plot((NF/pi),M)
```

```
title('Magnitude Response');
xlabel(' (a) Normalized Frequency -->');
ylabel(' Gain in dB -->');

an=angle(H);

subplot(2,1,2)
plot((NF/pi),an)
title('Phase Response');
xlabel(' (a) Normalized Frequency -->');
ylabel(' Phase in Radians -->');
```

COMMAND WINDOW

```
Enter the passband attenuation in dB: 15
Enter the stopband attenuation in dB: 50
Enter the first digital normalised freq.(in between 0 to 1): .35
Enter the second digital normalised freq.(in between 0 to 1): .6

N = 5
fc = 0.3500
b =0.0006    0.0030    0.0061    0.0061    0.0030    0.0006
a =1.0000   -3.4238    5.7337   -5.6050    3.1947   -0.8803
```

MATLAB figure:



MATLAB Program.

```
% Matlab program <chebyshev1_hpf.m>
% This program gives Chebyshev1 high pass filter.

clc;                                    %Clear the window
close all;                  `           %Close all files
clear all;                              %Clear the screen


Ap=input('Enter the passband attenuation in dB: ');
As=input('Enter the stopband attenuation in dB: ');
f1=input('Enter the Passband digital frequency(between 0 to 1): ');
f2=input('Enter the Stopband digital frequency(between 0 to 1): ');
fs=1;

E=((10^(.1*Ap))-1);
L=((10^(.1*As))-1);

N=ceil((acosh(sqrt(L/E)))/acosh(f2/f1))
fc=f1

[b,a]=cheby1(N,Ap,fc,'High')
w=0:.01:pi;
[H,NF]=freqz(b,a,w);

M=20*log10(abs(H));

subplot(2,1,1)
plot((NF/pi),M)
title('Magnitude Response');
xlabel(' (a) Normalized Frequency -->');
ylabel(' Gain in dB -->');

an=angle(H);

subplot(2,1,2)
plot((NF/pi),an)
title('Phase Response');
xlabel(' (b) Normalized Frequency -->');
ylabel(' Phase in Radians -->');
```

COMMAND WINDOW

Enter the passband attenuation in dB: 15
Enter the stopband attenuation in dB: 50
Enter the first digital normalised freq.(in between 0 to 1): .35
Enter the second digital normalised freq.(in between 0 to 1): .6

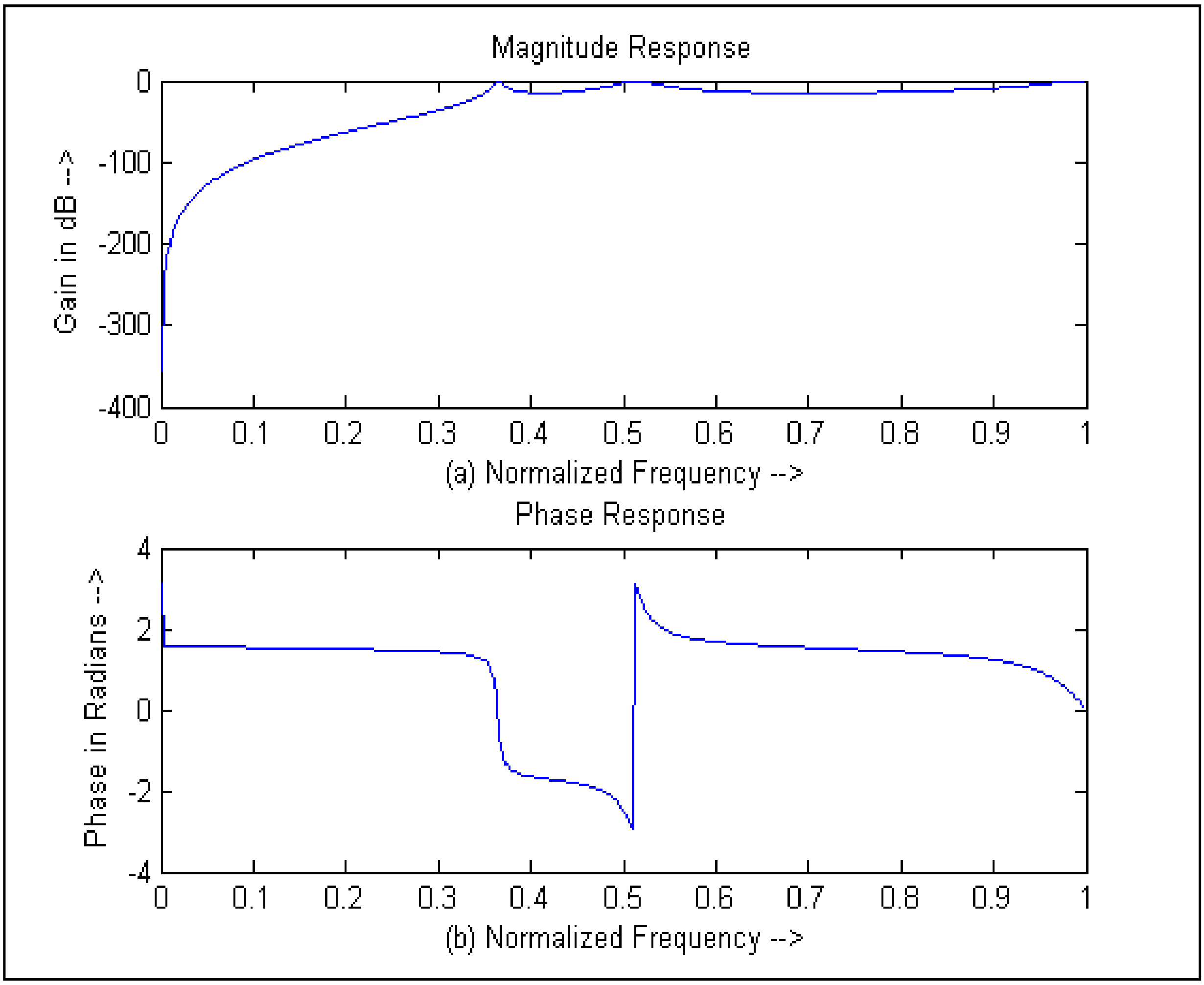
N= 5

fc =0.3500

b = 0.0177 -0.0887 0.1774 -0.1774 0.0887 -0.0177

a =1.0000 0.1460 1.1620 0.9530 0.2932 0.7886

MATLAB Figure



MATLAB Program.

```
% Matlab program <chebyshev1_bpf.m>
% This program gives Chebyshev1 band pass filter.

clc;                                        %Clear the window
close all;                        `         %Close all files
clear all;                                  %Clear the screen

Ap=input('Enter the passband attenuation in dB: ');
As=input('Enter the stopband attenuation in dB: ');
f1=input('Enter the first digital normalised freq.(in between 0 to 1): ');
f2=input('Enter the second digital normalised freq.(in between 0 to 1): ');
fs=1;

E=((10^(.1*Ap))-1);
L=((10^(.1*As))-1);

N=ceil((acosh(sqrt(L/E)))/acosh(f2/f1))
fc=[f1 f2]

[b,a]=cheby1(N,Ap,fc,'bandpass')
w=0:.01:pi;
[H,NF]=freqz(b,a,w);

M=20*log10(abs(H));

subplot(2,1,1)
plot((NF/pi),M)
title('Magnitude Response');
xlabel(' (a) Normalized Frequency -->');
ylabel(' Gain in dB -->');

an=angle(H);

subplot(2,1,2)
plot((NF/pi),an)
title('Phase Response');
xlabel(' (b) Normalized Frequency -->');
ylabel(' Phase in Radians -->');
```

COMMAND WINDOW

Enter the passband attenuation in dB: 15
Enter the stopband attenuation in dB: 50
Enter the first digital normalised freq.(in between 0 to 1): .35
Enter the second digital normalised freq.(in between 0 to 1): .6

N = 5

fc = 0.3500 0.6000

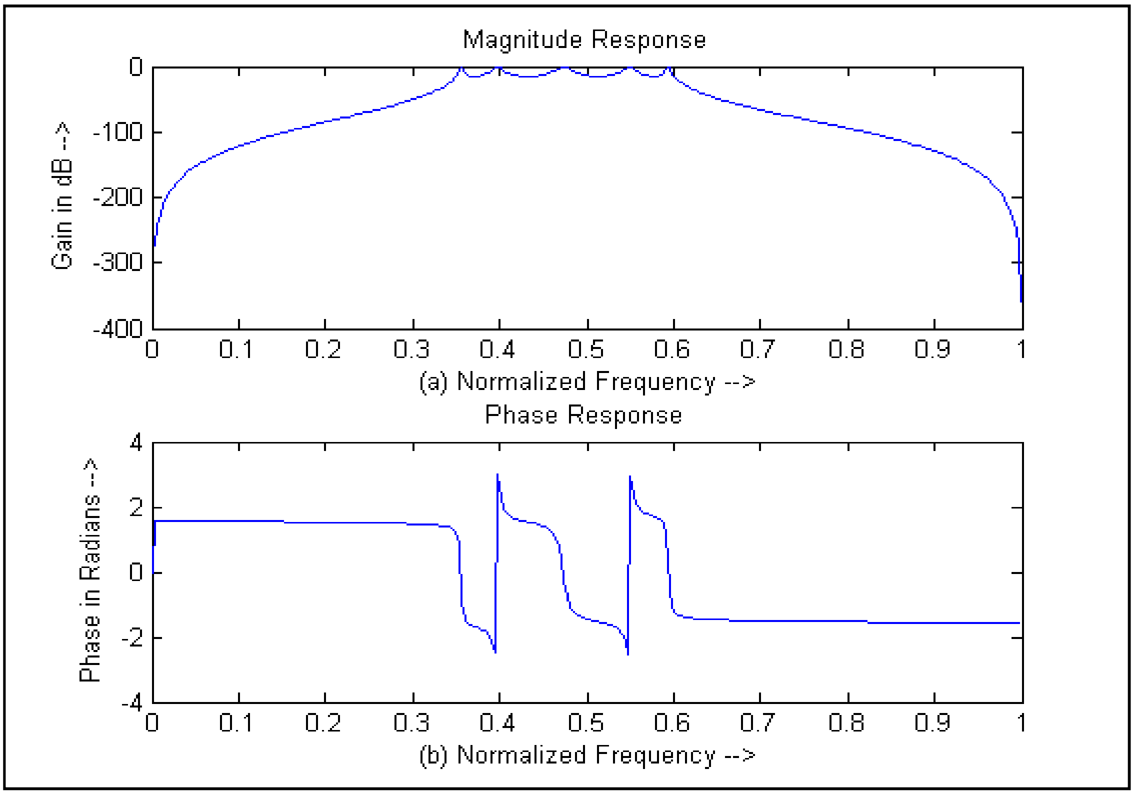
b =Columns 1 through 11

0.0001 0 -0.0005 0 0.0011 0 -0.0011 0 0.0005 0 -0.0001

a =Columns 1 through 11

1.0000 -0.7776 4.4034 -2.7343 8.2207 -3.8892 8.0794 -2.6385 4.1771
-0.7225 0.9128

MATLAB Figure



MATLAB Program.

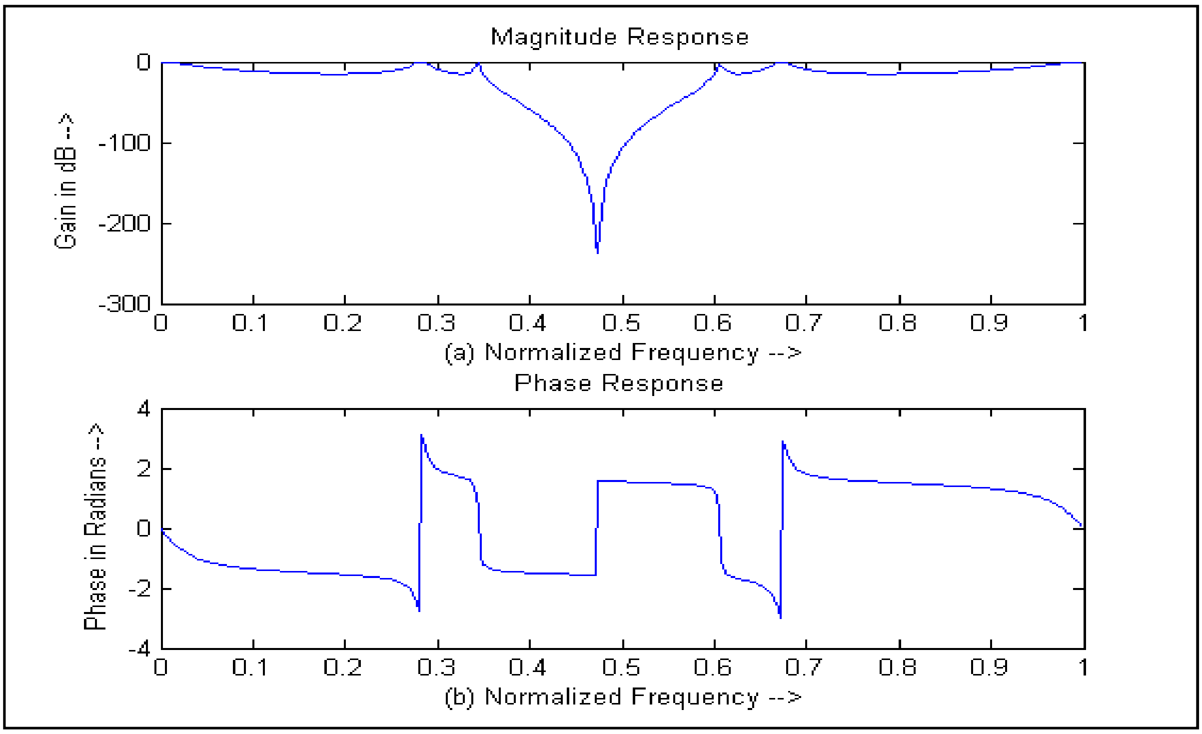
```
% Matlab program <chebyshev1_bsf.m>
% This program gives Chebyshev1 band stop filter.

clc;                                   %Clear the window
close all;                  `          %Close all files
clear all;                             %Clear the screen

Ap=input('Enter the passband attenuation in dB: ');
As=input('Enter the stopband attenuation in dB: ');
f1=input('Enter the first digital normalised freq.(in between 0 to 1): ');
f2=input('Enter the second digital normalised freq.(in between 0 to 1): ');
fs=1;

E=((10^(.1*Ap))-1);
L=((10^(.1*As))-1);

N=ceil((acosh(sqrt(L/E)))/acosh(f2/f1))
fc=[f1 f2]

[b,a]=cheby1(N,Ap,fc,'stop')
w=0:.01:pi;
[H,NF]=freqz(b,a,w);

M=20*log10(abs(H));

subplot(2,1,1)
plot((NF/pi),M)
title('Magnitude Response');
xlabel(' (a) Normalized Frequency -->');
ylabel(' Gain in dB -->');

an=angle(H);

subplot(2,1,2)
plot((NF/pi),an)
title('Phase Response');
xlabel(' (b) Normalized Frequency -->');
ylabel(' Phase in Radians -->');
```

COMMAND WINDOW

Enter the passband attenuation in dB: 15
Enter the stopband attenuation in dB: 50
Enter the first digital normalised freq.(in between 0 to 1): .35
Enter the second digital normalised freq.(in between 0 to 1): .6

```
N = 5
fc =0.3500    0.6000
b = Columns 1 through  11
    0.0426   -0.0361    0.2251   -0.1467    0.4627   -0.2211    0.4627   -0.1467    0.2251
   -0.0361    0.0426

a = Columns 1 through  11
    1.0000   -0.5229    1.2705   -0.5901    1.1767   -0.1640   -0.4844    0.3136   -0.7496
    0.3767   -0.7526
```

MATLAB Figure:



CONCLUSION:

Date: __________

Practical No.: 10

Aim: Write a MATLAB program to design following filters.

a) To design Chebyshev-II Low Pass Filter

b) To design Chebyshev-II High Pass Filter

c) To design Chebyshev-II Band Pass Filter

d) To design Chebyshev-II Band Reject Filter

MATLAB Program.

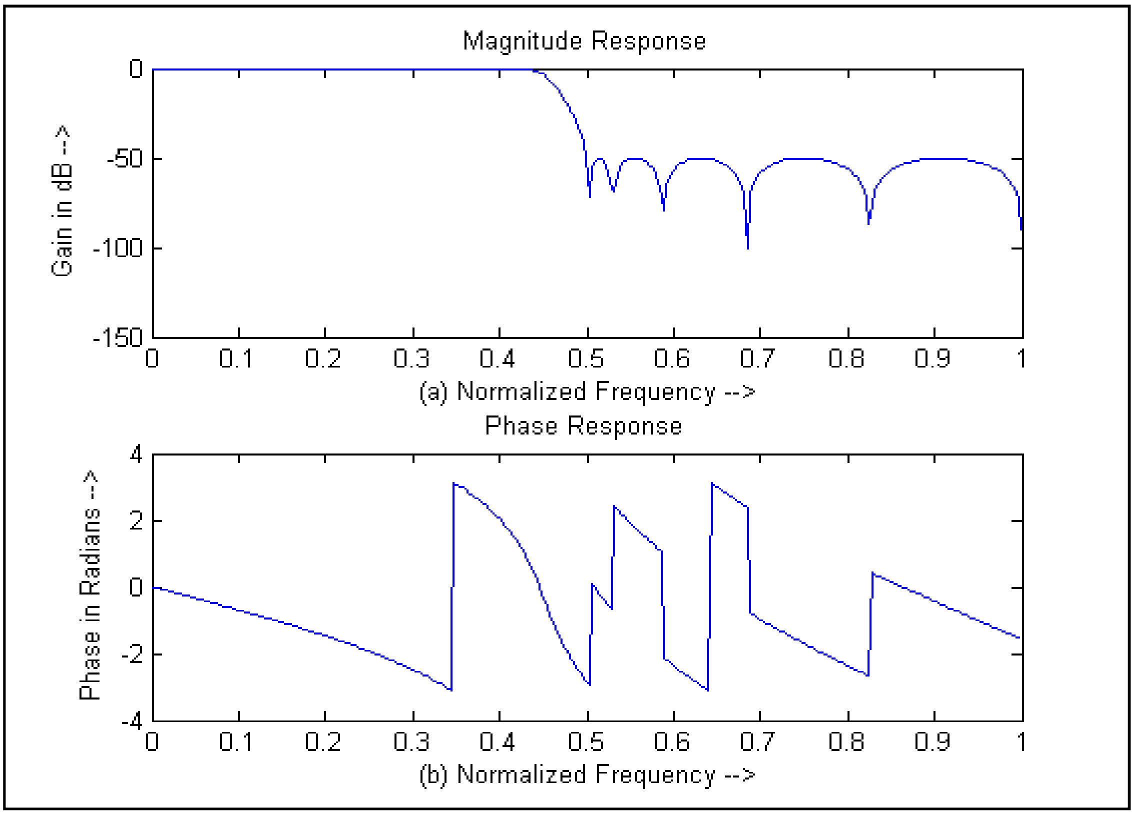
```
% Matlab program <chebyshev2_lpf.m>
% This program gives chebyshev2 low pass filter.

clc;                                          %Clear the window
close all;                      `             %Close all files
clear all;                                    %Clear the screen

Ap=input('Enter the passband attenuation in dB: ');
As=input('Enter the stopband attenuation in dB: ');
f1=input('Enter the Passband digital frequency(between 0 to 1): ');
f2=input('Enter the Stopband digital frequency(between 0 to 1): ');
fs=1;


L=((10^(.1*As))-1);

fc=f1
N=ceil((acosh(sqrt(L)))/acosh(f2/fc))

[b,a]=cheby2(N,As,fc)
w=0:.01:pi;
[H,NF]=freqz(b,a,w);

M=20*log10(abs(H));

subplot(2,1,1)
```

```
plot((NF/pi),M)
title('Magnitude Response');
xlabel(' (a) Normalized Frequency -->');
ylabel(' Gain in dB -->');

an=angle(H);

subplot(2,1,2)
plot((NF/pi),an)
title('Phase Response');
xlabel(' (b) Normalized Frequency -->');
ylabel(' Phase in Radians -->');
```

COMMAND WINDOW

```
Enter the passband attenuation in dB: 10
Enter the stopband attenuation in dB: 50
Enter the first digital normalised freq.(in between 0 to 1): .5
Enter the second digital normalised freq.(in between 0 to 1): .6

fc = 0.5000

N = 11

b =  Columns 1 through  12

    0.0353    0.1607    0.4465    0.8844    1.3582    1.6707    1.6707    1.3582    0.8844
    0.4465    0.1607    0.0353

a =  Columns 1 through  12

    1.0000    0.4636    2.1485    1.2185    1.8473    1.0466    0.7948    0.3697    0.1613
    0.0492    0.0108    0.0012
```

MATLAB figure:



MATLAB Program.

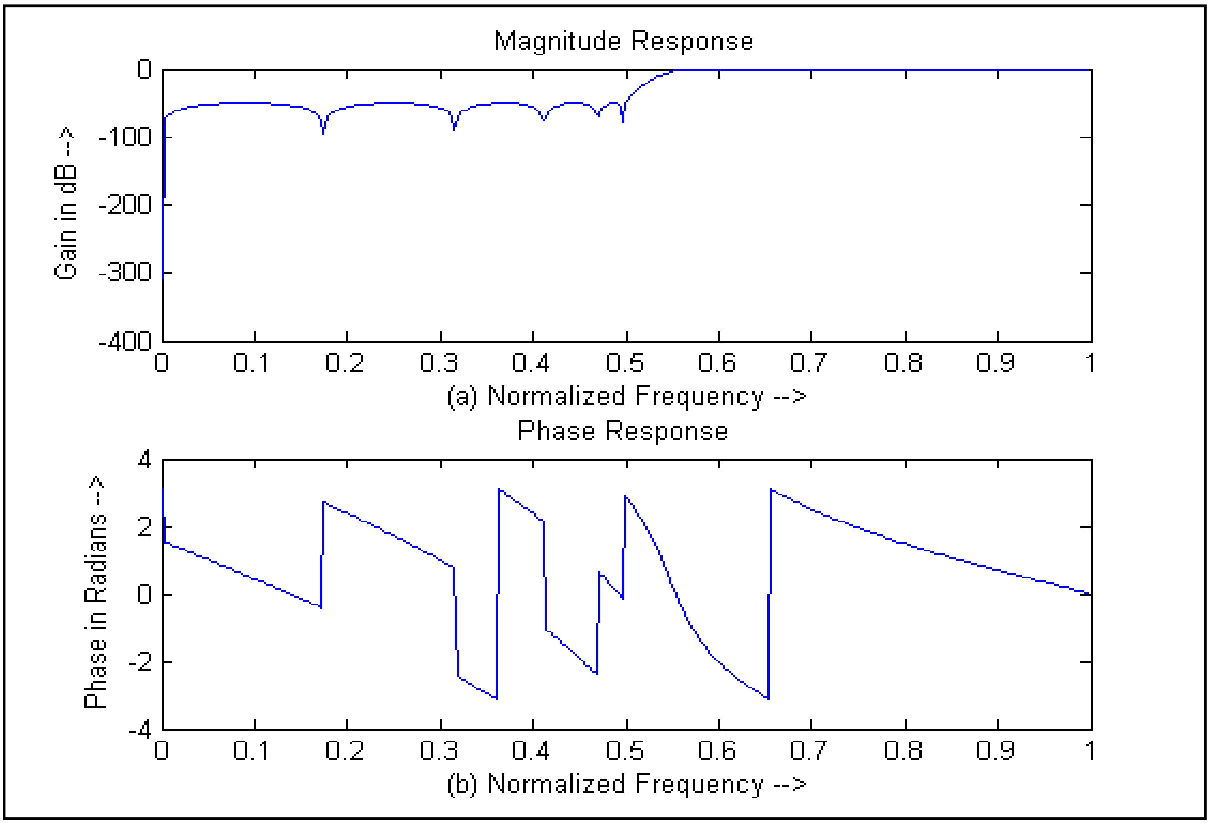
```
% Matlab program <chebyshev2_hpf.m>
% This program gives Chebyshev2 high pass filter.

clc;                          %Clear the window
close all;              `     %Close all files
clear all;                    %Clear the screen


Ap=input('Enter the passband attenuation in dB: ');
As=input('Enter the stopband attenuation in dB: ');
f1=input('Enter the Passband digital frequency(between 0 to 1): ');
f2=input('Enter the Stopband digital frequency(between 0 to 1): ');
fs=1;

L=((10^(.1*As))-1);

fc=f1
N=ceil((acosh(sqrt(L)))/acosh(f2/fc))

[b,a]=cheby2(N,As,fc,'High')
```

```
w=0:.01:pi;
[H,NF]=freqz(b,a,w);

M=20*log10(abs(H));

subplot(2,1,1)
plot((NF/pi),M)
title('Magnitude Response');
xlabel(' (a) Normalized Frequency -->');
ylabel(' Gain in dB -->');

an=angle(H);

subplot(2,1,2)
plot((NF/pi),an)
title('Phase Response');
xlabel(' (b) Normalized Frequency -->');
ylabel(' Phase in Radians -->');
```

COMMAND WINDOW

```
Enter the passband attenuation in dB: 10
Enter the stopband attenuation in dB: 50
Enter the first digital normalised freq.(in between 0 to 1): .5
Enter the second digital normalised freq.(in between 0 to 1): .6

fc =  0.5000

N =  11

b =  Columns 1 through  12

     0.0353   -0.1607    0.4465   -0.8844    1.3582   -1.6707    1.6707   -1.3582    0.8844
    -0.4465    0.1607   -0.0353

a =  Columns 1 through  12

    1.0000   -0.4636    2.1485   -1.2185    1.8473   -1.0466    0.7948   -0.3697    0.1613
   -0.0492    0.0108   -0.0012
```

MATLAB Figure:



MATLAB Program.

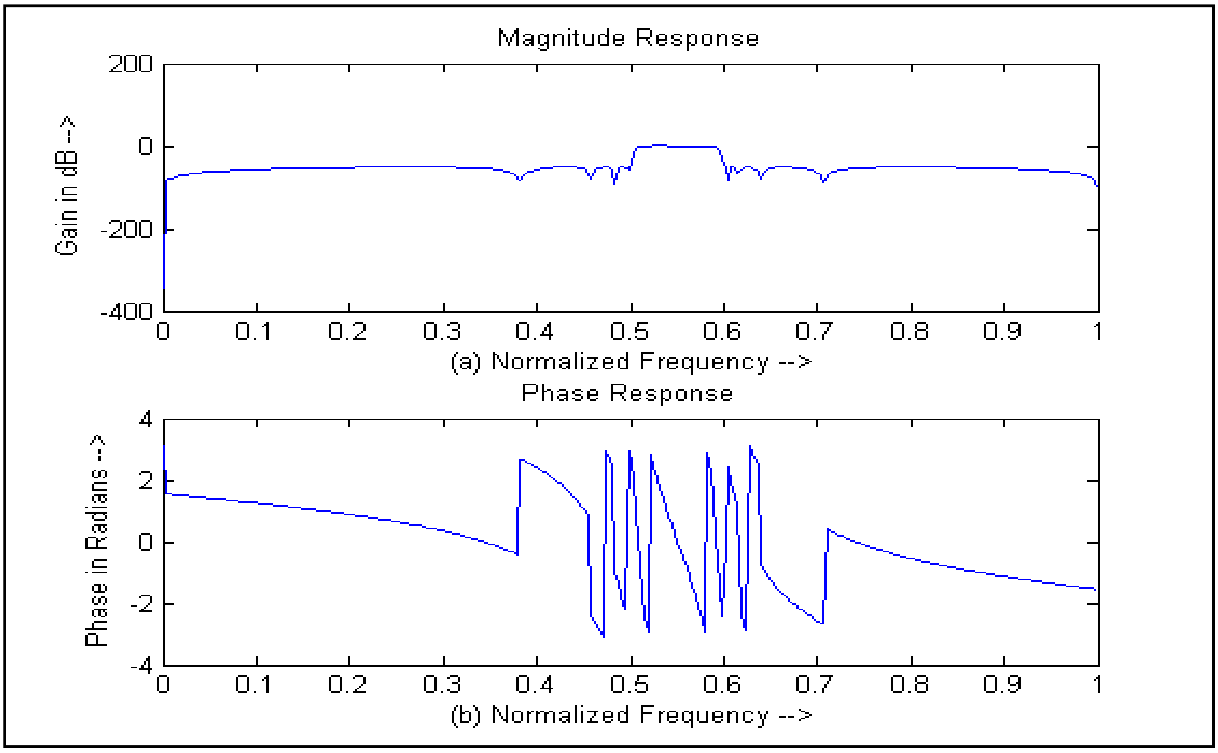
```
% Matlab program <chebyshev2_bpf.m>
% This program gives Chebyshev2 band pass filter.

clc;                                    %Clear the window
close all;                    `         %Close all files
clear all;                              %Clear the screen

Ap=input('Enter the passband attenuation in dB: ');
As=input('Enter the stopband attenuation in dB: ');
f1=input('Enter the first digital normalised freq.(in between 0 to 1): ');
f2=input('Enter the second digital normalised freq.(in between 0 to 1): ');
fs=1;

L=((10^(.1*As))-1);

N=ceil((acosh(sqrt(L)))/acosh(f2/f1))
fc=[f1 f2]

[b,a]=cheby2(N,As,fc,'Bandpass')
w=0:.01:pi;
```

```
[H,NF]=freqz(b,a,w);

M=20*log10(abs(H));

subplot(2,1,1)
plot((NF/pi),M)
title('Magnitude Response');
xlabel(' (a) Normalized Frequency -->');
ylabel(' Gain in dB -->');

an=angle(H);

subplot(2,1,2)
plot((NF/pi),an)
title('Phase Response');
xlabel(' (b) Normalized Frequency -->');
ylabel(' Phase in Radians -->');
```

COMMAND WINDOW

```
Enter the passband attenuation in dB: 10
Enter the stopband attenuation in dB: 50
Enter the first digital normalised freq.(in between 0 to 1): .5
Enter the second digital normalised freq.(in between 0 to 1): .6

N = 11

fc =  0.5000    0.6000

b =  Columns 1 through  23

    0.0028    0.0082    0.0315    0.0629    0.1365    0.2041    0.3097    0.3517    0.3824
    0.3016    0.1805   -0.0000   -0.1805   -0.3016   -0.3824   -0.3517   -0.3097   -0.2041
   -0.1365   -0.0629   -0.0315   -0.0082   -0.0028

a =  Columns 1 through 23

    1.0000    3.1297   13.2170   28.7983   69.3560  115.5898  200.5175  267.4663  362.6265
   395.9562  435.1143  392.2682  354.9256  263.4227  196.7771  118.3305   72.3419
   33.9780   16.6198    5.6174    2.1010    0.4044    0.1053
```

MATLAB Figure:



MATLAB Program.

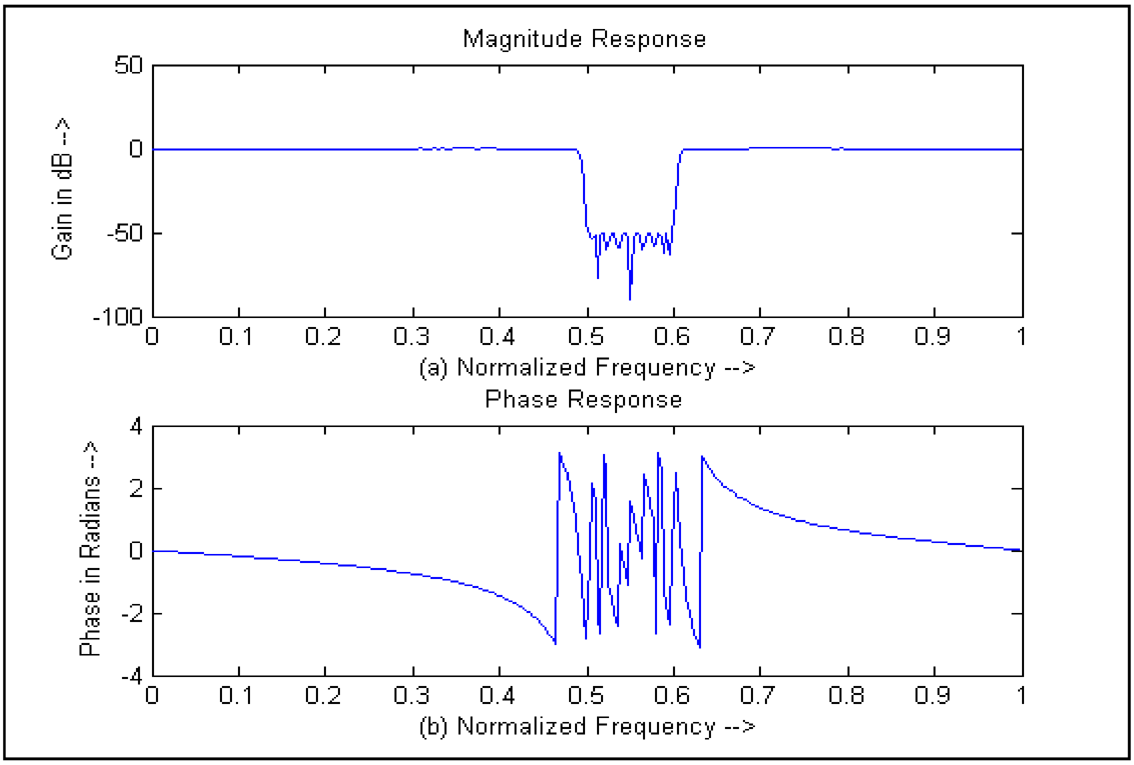
```
% Matlab program <chebyshev2_bsf.m>
% This program gives Chebyshev2 band stop filter.

clc;                                              %Clear the window
close all;                                        %Close all files
clear all;                                        %Clear the screen

Ap=input('Enter the passband attenuation in dB: ');
As=input('Enter the stopband attenuation in dB: ');
f1=input('Enter the first digital normalised freq.(in between 0 to 1): ');
f2=input('Enter the second digital normalised freq.(in between 0 to 1): ');
fs=1;

L=((10^(.1*As))-1);

N=ceil((acosh(sqrt(L)))/acosh(f2/f1))
fc=[f1 f2]

[b,a]=cheby2(N,As,fc,'stop')
w=0:.01:pi;
[H,NF]=freqz(b,a,w);
```

```
M=20*log10(abs(H));

subplot(2,1,1)
plot((NF/pi),M)
title('Magnitude Response');
xlabel(' (a) Normalized Frequency -->');
ylabel(' Gain in dB -->');

an=angle(H);

subplot(2,1,2)
plot((NF/pi),an)
title('Phase Response');
xlabel(' (b) Normalized Frequency -->');
ylabel(' Phase in Radians -->');LUSIO
```

COMMAND WINDOW

```
Enter the passband attenuation in dB: 10
Enter the stopband attenuation in dB: 50
Enter the first digital normalised freq.(in between 0 to 1): .5
Enter the second digital normalised freq.(in between 0 to 1): .6

N = 11

fc = 0.5000    0.6000

b =  Columns 1 through  23

    0.5045    1.7364    8.1330   19.5378   52.1341   95.9104  184.2809  271.5387  407.7672
   492.1554  599.1892  597.4515  599.1892  492.1554  407.7672  271.5387  184.2809
    95.9104   52.1341   19.5378    8.1330    1.7364    0.5045

a =  Columns 1 through  23

     1.0000    3.2289   14.1329   31.8420   79.5318  137.2817  247.2118  342.0046  481.8484
     546.4082  624.6946  585.6308  551.9665  426.5341  332.3406  208.3396  133.0417
     65.2213   33.3745   11.7874    4.6207    0.9302    0.2546
```

MATLAB Figure:



CONCLUSION:

Date: __________

Practical No.: 11

Aim: Write a MATLAB program to design FIR filter using following window.

(a) Rectangular window

(b) Kaiser window

(c) Bartlett window

(d) Blackman window

(e) Hanning window

(f) Hamming window

MATLAB Program.

```
% Matlab program <FIR.m>
%This program gives FIR Low pass filter, High pass filter, Band pass filter, Band stop filter.
%This program gives FIR magnitude response using different windowing techniques.

clc;                                                    %Clear the window
close all;                              `               %Close all files
clear all;                                              %Clear the screen

rp=input('Enter the pasband ripple: ');                 %Enter the passband ripple
rs=input('Enter the stopband ripple: ');                %Enter the stopband ripple
fp=input('Enter the Passband frequency in hertz: ');    %Enter the passband freq.
fs=input('Enter the Stopband frequency in hertz: ');    %Enter the stopband freq.
f=input('Enter the sampling frequency in hertz: ');     %Enter the sampling freq.

wp=2*fp/f;
ws=2*fs/f;

num=-20*log10(sqrt(rp*rs))-13;
dem=14.6*(fs-fp)/f;
n=ceil(num/dem)                                         %To find order

n1=n+1;
if(rem(n,2)~=0)
```

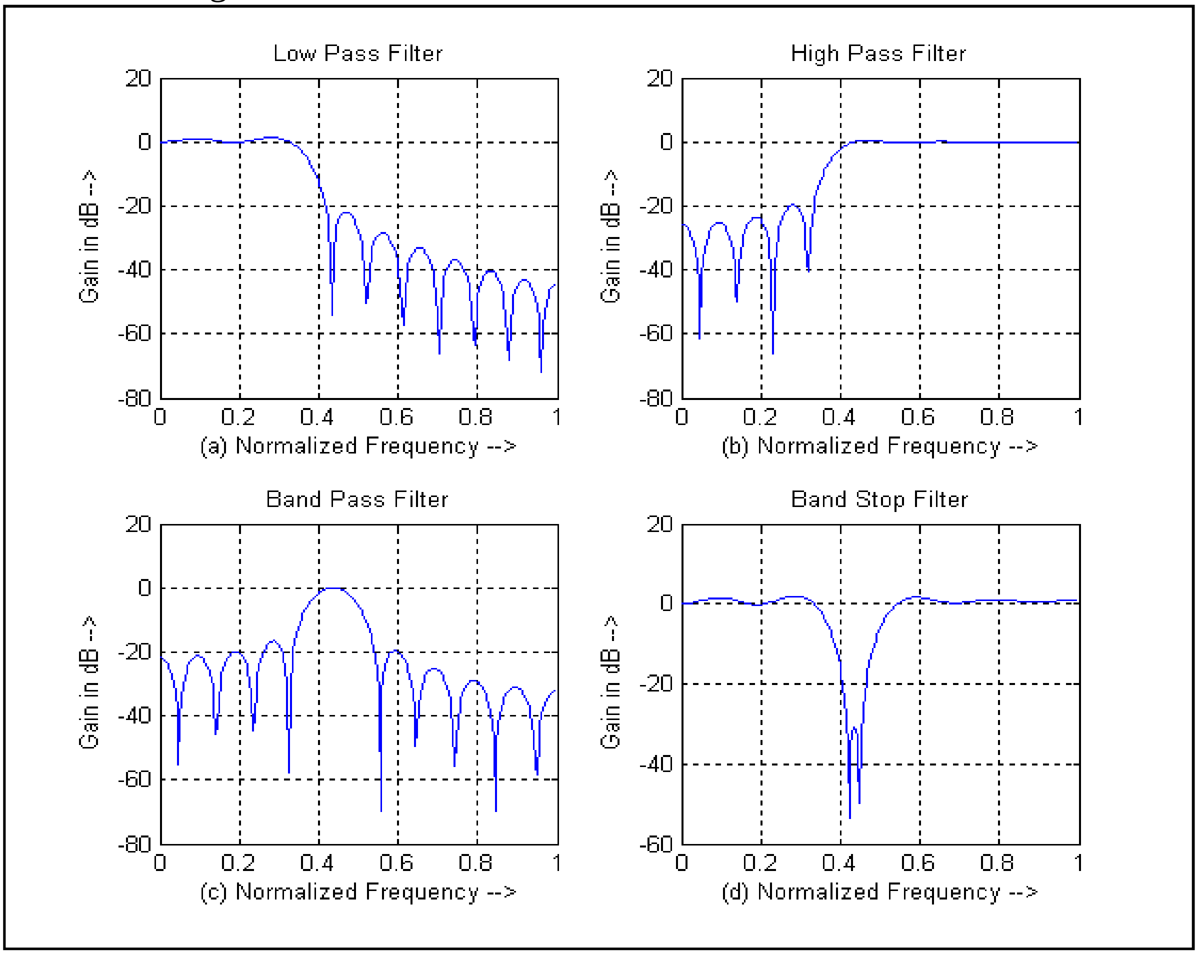
```
   n1=n;
   n=n-1;
end

a=input('Enter the type of windowing techniques(in lower case in single quote):');
switch a
   case 'rectangular'
      y=boxcar(n1);
   case 'kaiser'
      y=kaiser(n1);
   case 'bartlett'
      y=bartlett(n1);
   case 'blackman'
      y=blackman(n1);
   case 'hamming'
      y=Hamming(n1);
   case 'hanning'
      y=Hanning(n1);
otherwise
      disp('no windowing techniques');
end

b=fir1(n,wp,y);                              %Low pass filter
[h,o]=freqz(b,1,256);

m=20*log10(abs(h));                          %Magnitude Response
subplot(2,2,1)
plot((o/pi),m);
title ('Low Pass Filter')
grid on;
xlabel(' (a) Normalized Frequency -->');
ylabel(' Gain in dB -->');

b=fir1(n,wp,'high',y);                       %High pass filter
[h,o]=freqz(b,1,256);

m=20*log10(abs(h));
subplot(2,2,2)
plot((o/pi),m);
title ('High Pass Filter')
grid on;
xlabel(' (b) Normalized Frequency -->');
ylabel(' Gain in dB -->');
```

```
wn=[wp ws]
b=fir1(n,wn,y);                                  %Band pass filter
[h,o]=freqz(b,1,256);

m=20*log10(abs(h));
subplot(2,2,3)
plot((o/pi),m);
title ('Band Pass Filter')
grid on;
xlabel(' (c) Normalized Frequency -->');
ylabel(' Gain in dB -->');

b=fir1(n,wn,'stop',y);                           %Band reject filter
[h,o]=freqz(b,1,256);

m=20*log10(abs(h));
subplot(2,2,4)
plot((o/pi),m);
title ('Band Stop Filter')
grid on;
xlabel(' (d) Normalized Frequency -->');
ylabel(' Gain in dB -->');
```

COMMAND WINDOW (for rectangular window):

```
Enter the passband ripple: .04
Enter the stopband ripple: .02
Enter the Passband frequency in hertz: 1500
Enter the Stopband frequency in hertz: 2000
Enter the sampling frequency in hertz: 8000

n = 20

Enter the type of windowing techniques(in lower case in single quote):'rectangular'

wn =  0.3750    0.5000
```

MATLAB Figures:



COMMAND WINDOW (for Kaiser window):

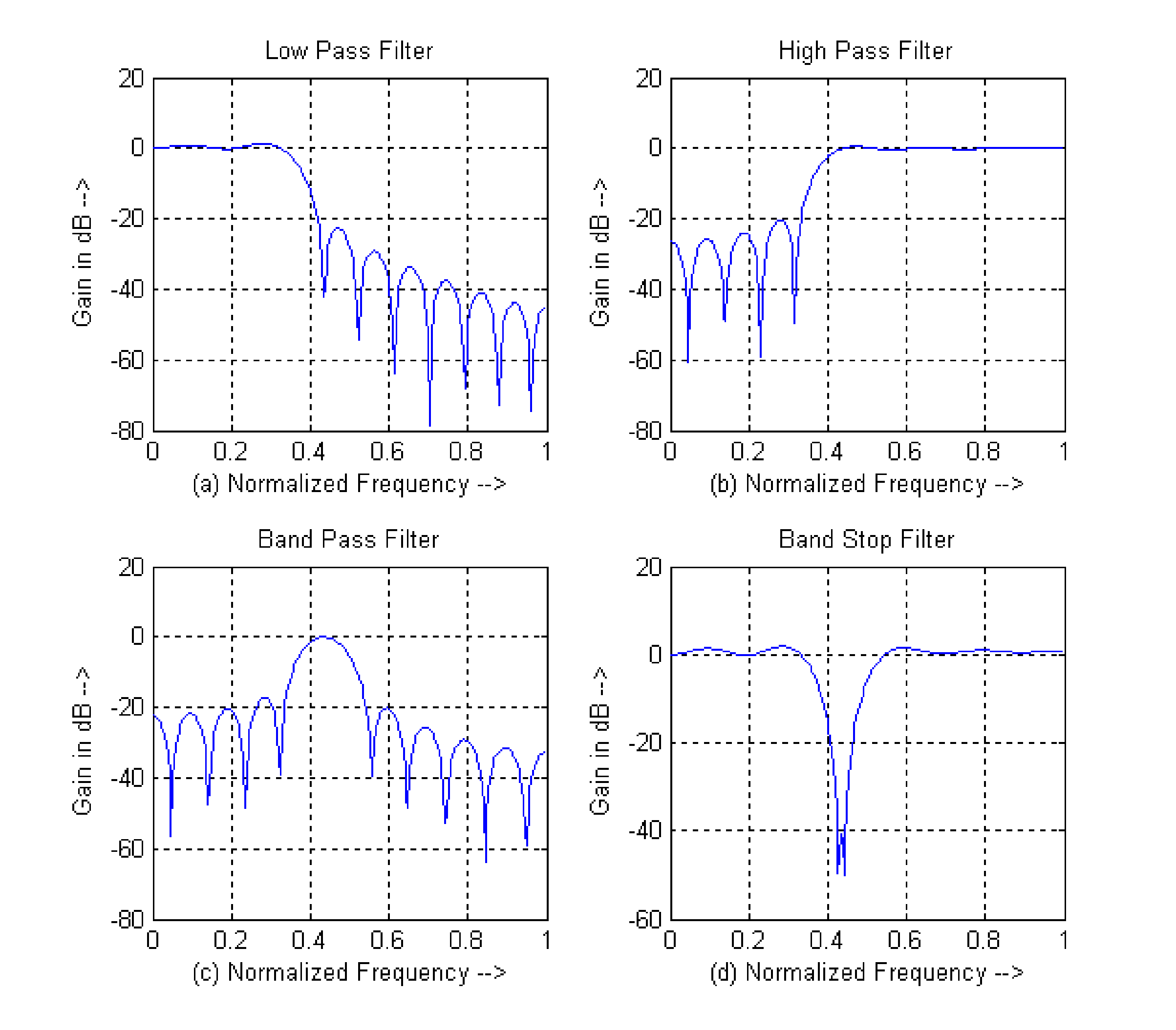
Enter the passband ripple: .04
Enter the stopband ripple: .02
Enter the Passband frequency in hertz: 1500
Enter the Stopband frequency in hertz: 2000
Enter the sampling frequency in hertz: 8000

n = 20

Enter the type of windowing techniques(in lower case in single quote):'kaiser'

wn = 0.3750   0.5000

MATLAB Figures:



COMMAND WINDOW (for bartlett window):

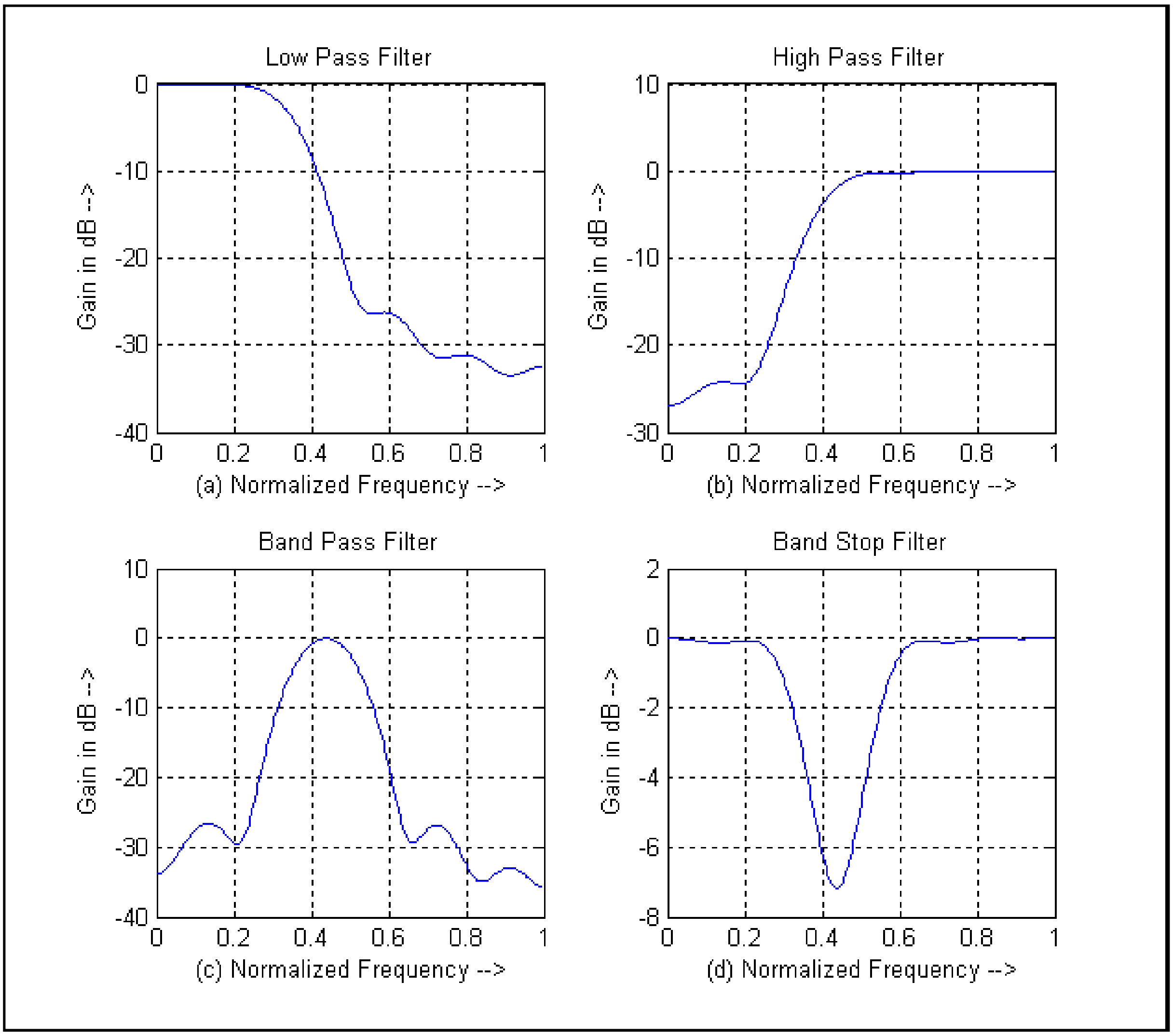
Enter the passband ripple: .04
Enter the stopband ripple: .02
Enter the Passband frequency in hertz: 1500
Enter the Stopband frequency in hertz: 2000
Enter the sampling frequency in hertz: 8000

n = 20

Enter the type of windowing techniques(in lower case in single quote):'bartlett’

wn = 0.3750 0.5000

MATLAB Figures:



COMMAND WINDOW (for blackman window):

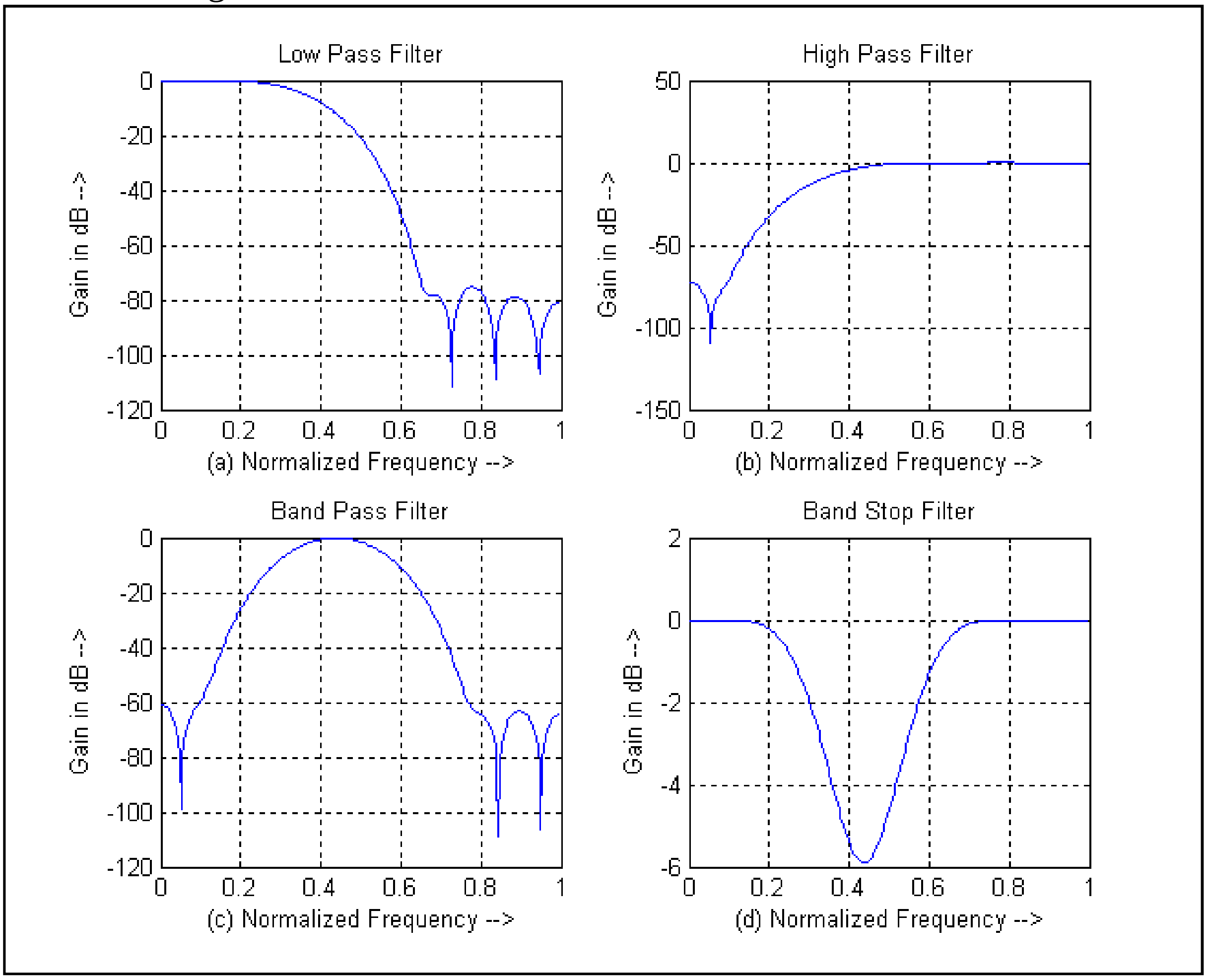
Enter the passband ripple: .04
Enter the stopband ripple: .02
Enter the Passband frequency in hertz: 1500
Enter the Stopband frequency in hertz: 2000
Enter the sampling frequency in hertz: 8000

n = 20

Enter the type of windowing techniques(in lower case in single quote):'blackman'

wn = 0.3750 0.5000

MATLAB Figures:



COMMAND WINDOW (for Hanning window):

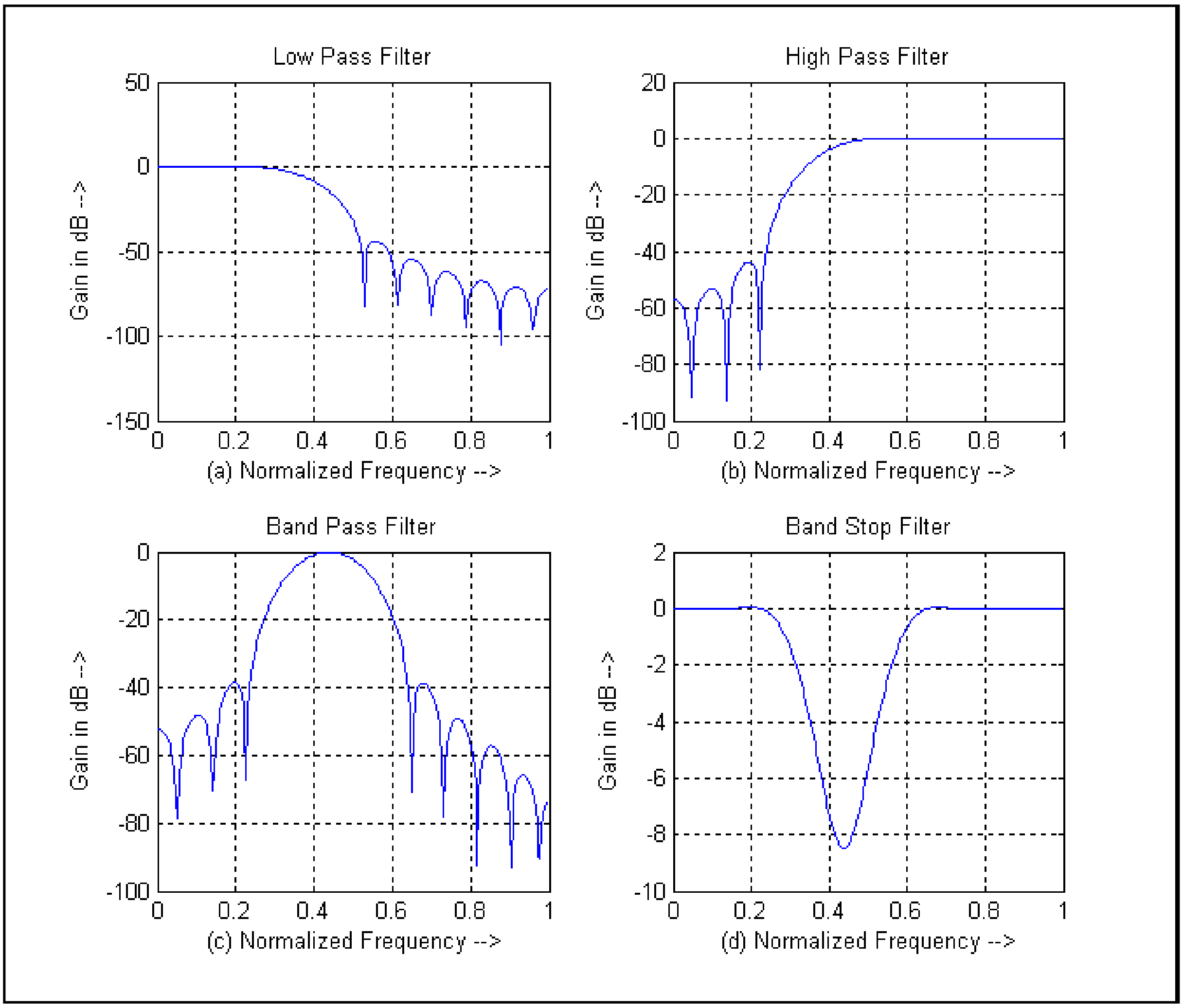
Enter the passband ripple: .04
Enter the stopband ripple: .02
Enter the Passband frequency in hertz: 1500
Enter the Stopband frequency in hertz: 2000
Enter the sampling frequency in hertz: 8000

n = 20

Enter the type of windowing techniques(in lower case in single quote):'Hanning'

wn = 0.3750   0.5000

MATLAB Figures:



COMMAND WINDOW (for Hamming window):

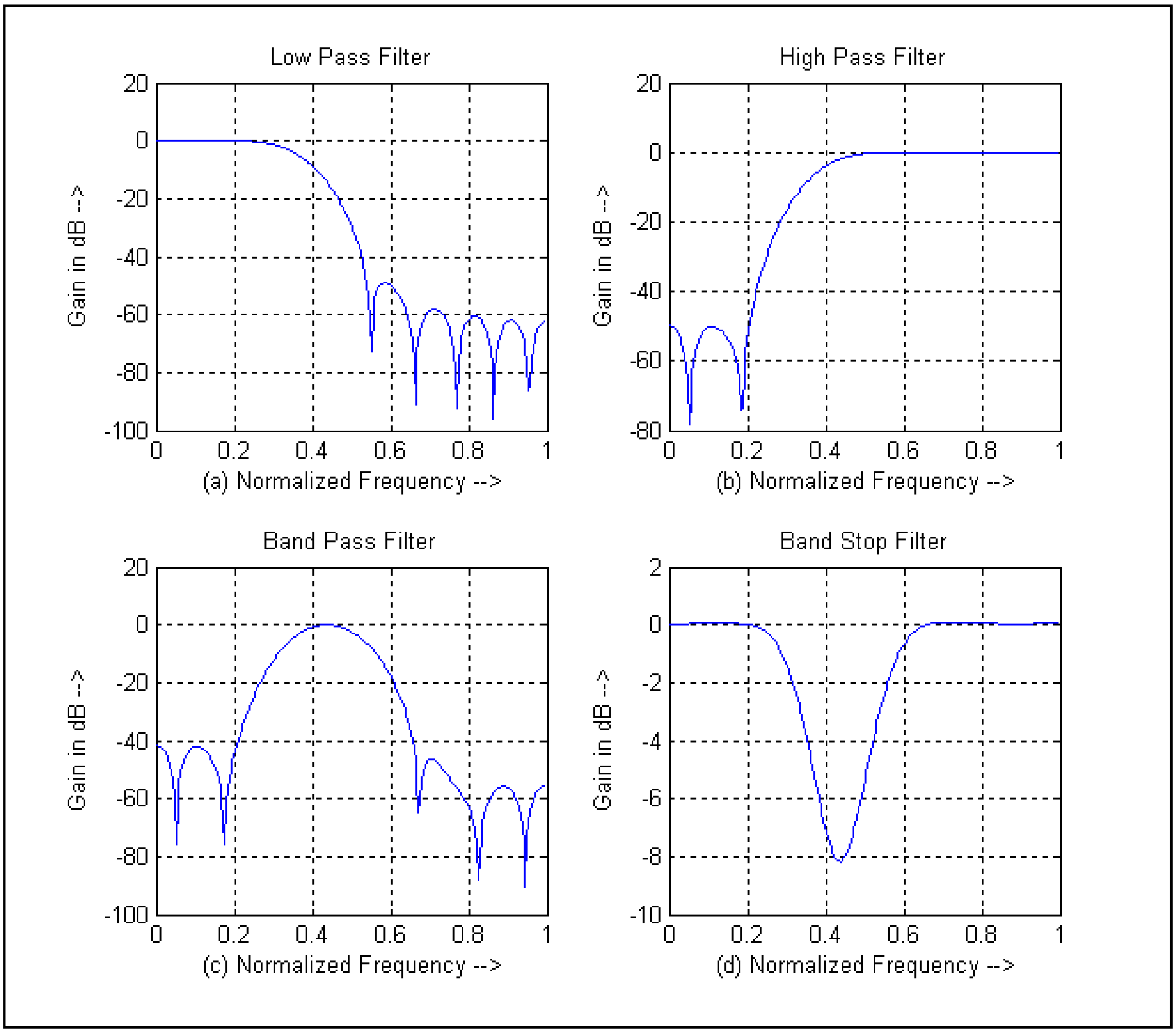
Enter the passband ripple: .04
Enter the stopband ripple: .02
Enter the Passband frequency in hertz: 1500
Enter the Stopband frequency in hertz: 2000
Enter the sampling frequency in hertz: 8000

n = 20

Enter the type of windowing techniques(in lower case in single quote):'hamming'

wn = 0.3750 0.5000

MATLAB Figures:



CONCLUSION:

Date: ___________

Practical No.: 13

Aim: Write a MATLAB program to design Fast convolution method.

a) Find linear convolution using overlap add method.

b) Find linear convolution using overlap save method.

MATLAB Program.

```
% Matlab program <Overlap_Add.m>
%This program gives convolved sequence using Overlap Add method.

clc;                                          %Clear the window
close all;                          `         %Close all files
clear all;                                    %Clear the screen

x=input('Enter the x(n): ');                  % Input the sequence x(n)
h=input('Enter the h(n): ');                  % Input the sequence h(n)

n1=length(x);
n2=length(h);
N=n1+n2-1;                                    % Length of convolved sequence

y=zeros(1,N);
h1=[h zeros(1,n2-1)];
n3=length(h1);
y=zeros(1,N+n3-n2);
H=fft(h1);

for i=1:n2:n1                                 % Looping for block making
    if i<=(n1+n2-1)
        x1=[x(i:i+n3-n2) zeros(1,n3-n2)];
   else
        x1=[x(i:n1) zeros(1,n3-n2)];
   end
   x2=fft(x1);
   x3=x2.*H;
   x4=round(ifft(x3));
   if (i==1)
      y(1:n3)=x4(1:n3);
   else
      y(i:i+n3-1)=y(i:i+n3-1)+x4(1:n3);
   end
end
```

```
disp('The output sequence y(n): ');
disp(y(1:N));
stem(y(1:N));
title('Overlap Add Method');
xlabel('n');
ylabel('y(n)');
```

COMMAND WINDOW

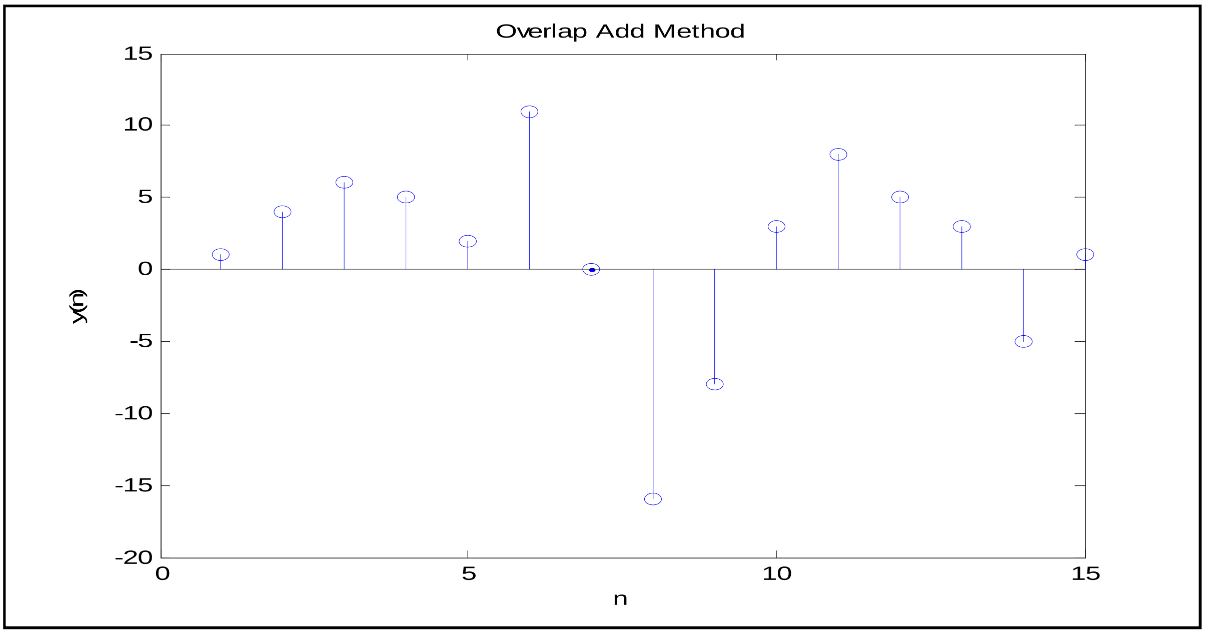
Enter the x(n): [1 2 -1 2 3 -2 -3 -1 1 1 2 -1]

Enter the h(n): [1 2 3 -1]

The output sequence y(n):

1 4 6 5 2 11 0 -16 -8 3 8 5 3 -5 1

MATLAB Figure



MATLAB Program.

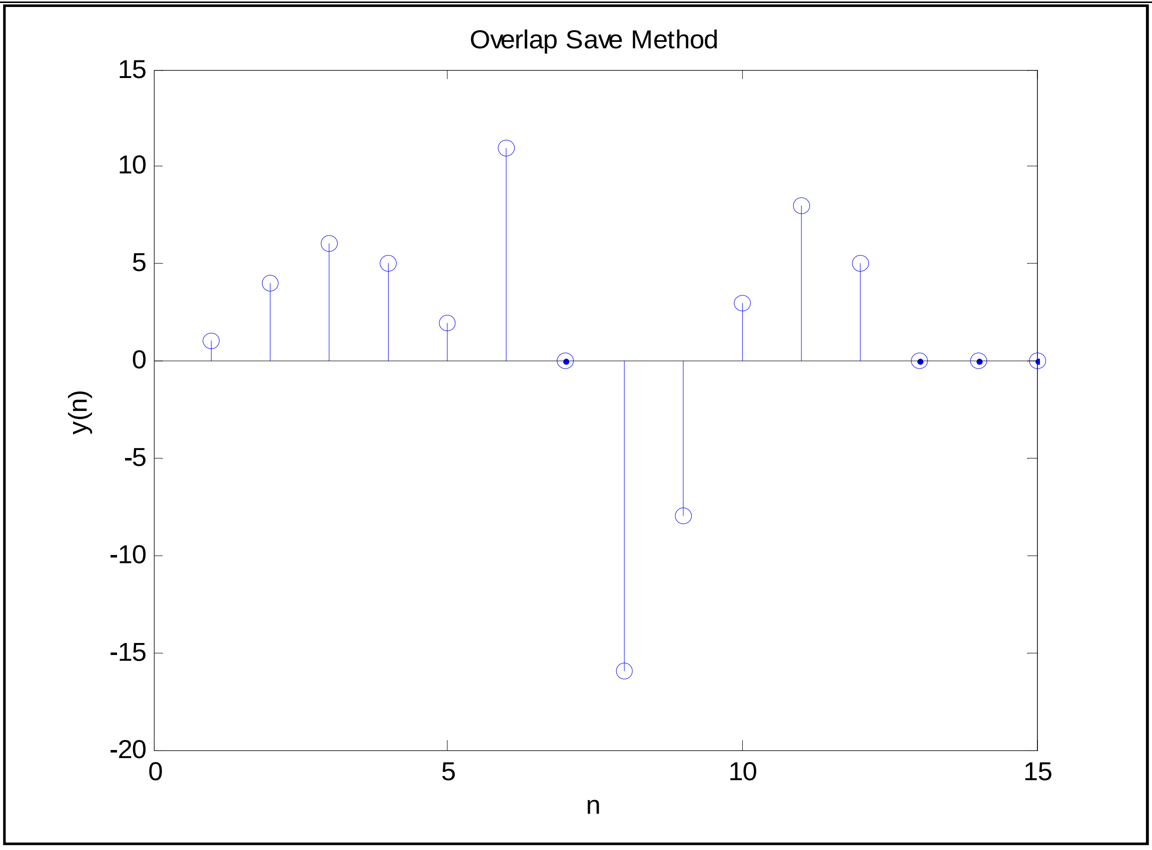
```
% Matlab program <Overlap_Save.m>
%This program gives convolved sequence using Overlap Save method.

clc;                                    %Clear the window
close all;                      `       %Close all files
clear all;                              %Clear the screen
```

```
x=input('Enter the x(n): ');
h=input('Enter the h(n): ');

n1=length(x);
n2=length(h);
N=n1+n2-1;

h1=[h zeros(1,n2-1)];
n3=length(h1);
y=zeros(1,N);
x1=[zeros(1,n3-n2) x zeros(1,n3)];
H=fft(h1);

for i=1:n2:n1
    y1=x1(i:i+(2*(n3-n2)));
    y2=fft(y1);
    y3=y2.*H;
    y4=round(ifft(y3));
    y(i:(i+n3-n2))=y4(n2:n3);
end

disp('The output sequence y(n): ');
disp(y(1:N));
stem(y(1:N));
title('Overlap Save Method');
xlabel('n');
ylabel('y(n)');
```

COMMAND WINDOW

```
Enter the x(n): [1 2 -1 2 3 -2 -3 -1 1 1 2 -1]

Enter the h(n): [1 2 3 -1]

The output sequence y(n):
   1    4    6    5    2   11    0  -16   -8    3    8    5    3   -5    1
```

MATLAB Figure



CONCLUSION: