**EXPERIMENT 1**

To write a MATLAB script to generate the following waveform (Discrete-time signal)

1. Unit Impulse function

2. Unit Step function

3. Ramp sequence

4. Exponential function

5. Sinusoidal sequence

6. Random Sequence

**Code:**

clc; clear; close all;

t = (-1:0.1:1);

x = (-3:0.1:3);

impulse = t==0;

unitstep = t>=0;

ramp = t.\*unitstep;

exp = exp(t);

sine = sin(x);

rand = randi([0,1],1,21);

subplot(2,3,1);

stem(t,impulse)

xlabel('time')

ylabel('amplitude')

title('impulse function')

subplot(2,3,2);

stem(t,unitstep)

xlabel('time')

ylabel('amplitude')

title('unitstep function')

subplot(2,3,3);

stem(t,ramp)

xlabel('time')

ylabel('amplitude')

title('ramp function')

subplot(2,3,4);

stem(t,exp)

xlabel('time')

ylabel('amplitude')

title('exponential function')

subplot(2,3,5);

stem(x,sine)

xlabel('time')

ylabel('amplitude')

title('sine wave')

subplot(2,3,6);

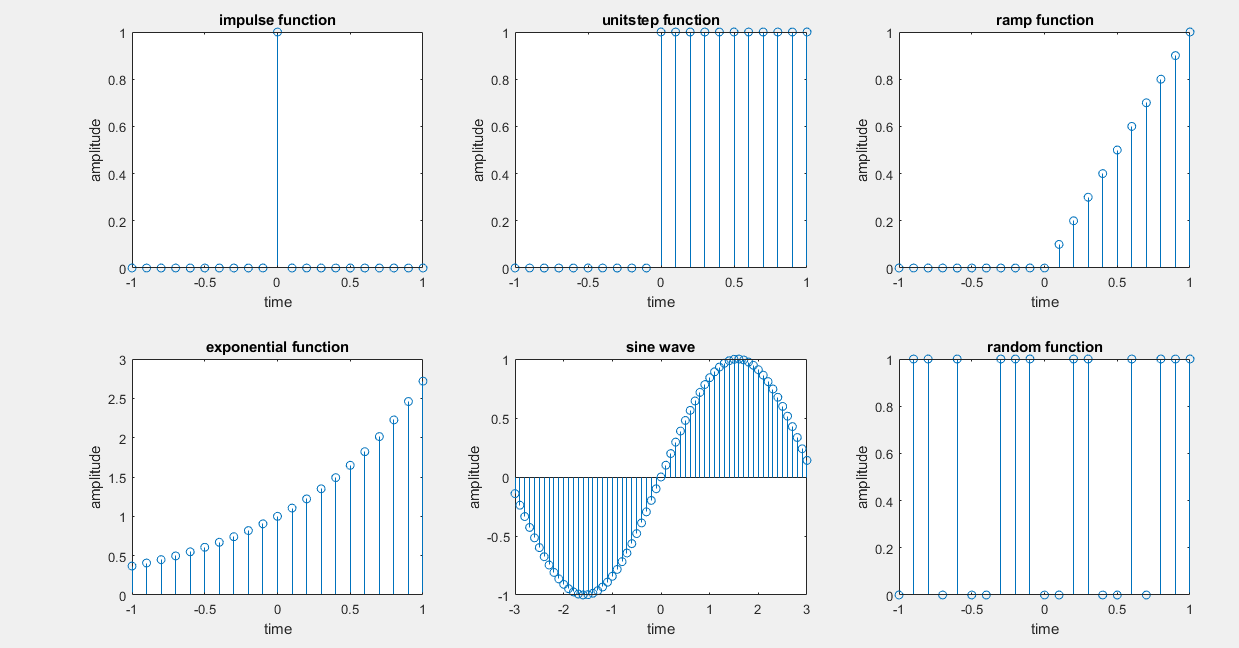
stem(t,rand)

xlabel('time')

ylabel('amplitude')

title('random function')

**Output:**



**EXPERIMENT 2**

To write a MATLAB script to study the basic operations on the Discrete–time signals.

1) Amplitude manipulation

i) Addition of two sequences

ii) Multiplication of two sequences

iii) Amplitude scaling

2) Time manipulation

i) Time scaling

ii) Time shifting

iii) Time reversal

**Code:**

clc; %clearing the screen

clear; %clearing the variables

close all; %closing all the previous windows

x1 = randi([0,9],1,randi(10));%generating the 1st sequence randomly

x2 = randi([0,9],1,randi(10));%generating the 2nd sequence randomly

n1 = length(x1); %length of the sequence

n2 = length(x2); %length of the sequence

n = max(n1,n2); % to have same length

if n1~=n2

x1 = [x1, zeros(1,n-n1)]; %add zeros if needed

x2 = [x2, zeros(1,n-n2)]; %add zeros if needed

end

subplot(5,2,1); %1st subplot

stem(0:n-1,x1); %discrete values of n length

xlabel('time');

ylabel('amplitude');

title('1st sequence'); %raw input sequence 1

subplot(5,2,2); %2nd subplot

stem(0:n-1,x2); %discrete values of n length

xlabel('time');

ylabel('amplitude');

title('2nd sequence'); %raw input sequence 2

% Amplitude Modulation

add = x1+ x2; %addition of two arrays

subplot(5,2,3); %3rd subplot

stem(0:n-1,add); %discrete values of n length

xlabel('time');

ylabel('amplitude');

title('addition of two sequence'); %addition of arrays

mul = x1 .\* x2; %multiplication of two arrays

subplot(5,2,4); %3rd subplot

stem(0:n-1,mul); %discrete values of n length

xlabel('time');

ylabel('amplitude');

title('multiplication of two sequence'); %multiplication of arrays

atten = x1 .\* .25; %attenuation of 1st array

subplot(5,2,5); %5th subplot

stem(0:n-1,atten); %discrete values of n length

xlabel('time');

ylabel('amplitude');

title('attenuation of 1st sequence'); %attenuation of 1st array

ampl = x1 .\* 5; %amplification of 1st array

subplot(5,2,6); %5th subplot

stem(0:n-1,ampl); %discrete values of n length

xlabel('time');

ylabel('amplitude');

title('amplification of 1st sequence'); %amplification of 1st array

subplot(5,2,7); %7th subplot

stem(0:n-1,-x1); %discrete values of n length

xlabel('time');

ylabel('amplitude');

title('amplitude inversion of sequence 1'); %amplitude inversion of sequence 1

% Time Modulation

subplot(5,2,8); %8th subplot

stem((0:n-1)/10,x2); %discrete values of n/10 length

xlabel('time');

ylabel('amplitude');

title('time scaling of 2nd sequence'); %time scaling of 2nd sequence

subplot(5,2,9); %9th subplot

stem(-n+1:0,x2); %discrete values of n length

xlabel('time');

ylabel('amplitude');

title('time shifting of 2nd sequence'); %time shifting of 2nd sequence

subplot(5,2,10); %10th subplot

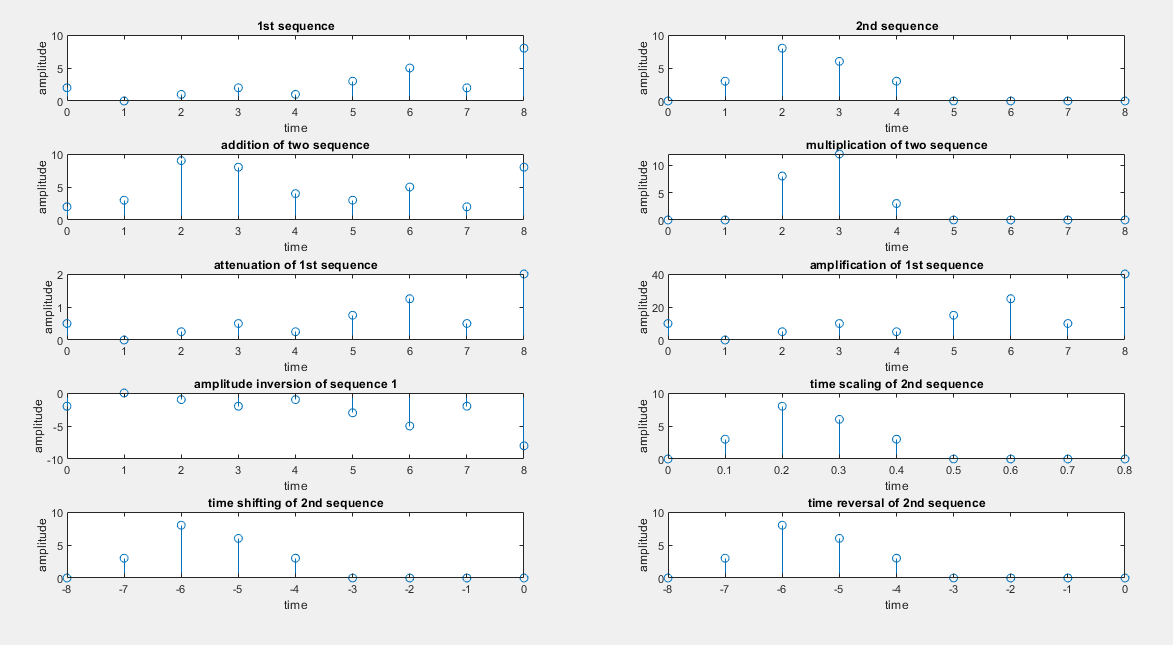
stem(-n+1:0,x2); %time reversed discrete values of n length

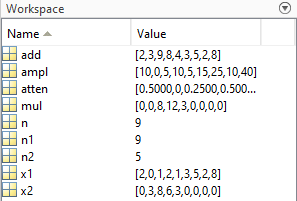
xlabel('time');

ylabel('amplitude');

title('time reversal of 2nd sequence'); %time reversal of 2nd sequence

**Output:**





**EXPERIMENT 4**

Write a MATLAB script to perform the convolution of sequences.

1. Circular convolution / periodic convolution  
2. Linear convolution / aperiodic convolution  
3. Linear convolution using circular convolutions

**Code:**

clc; %clearing the screen

clear; %clearing the variables

close all; %closing all the previous windows

x1 = randi([0,9],1,randi(10));%generating the 1st sequence randomly

x2 = randi([0,9],1,randi(10));%generating the 2nd sequence randomly

subplot(3,2,1); %1st subplot

stem(x1,'filled'); %discrete values of n length

xlabel('time');

ylabel('amplitude');

title('1st sequence'); %raw input sequence 1

subplot(3,2,2); %2nd subplot

stem(x2,'filled'); %discrete values of n length

xlabel('time');

ylabel('amplitude');

title('2nd sequence'); %raw input sequence 2

clin = conv(x1,x2);

subplot(3,2,3); %2nd subplot

stem(clin,'filled'); %discrete values of n length

xlabel('time');

ylabel('amplitude');

title('Linear Convolution of x1 and x2'); %Linear Convolution

ccirc = cconv(x1,x2);

subplot(3,2,4); %2nd subplot

stem(ccirc,'filled'); %discrete values of n length

xlabel('time');

ylabel('amplitude');

title('Circular Convolution of x1 and x2'); %Circular Convolution

n1 = length(x1); %length of the sequence

n2 = length(x2); %length of the sequence

n = max(n1,n2); % to have same length

if n1~=n2

x1 = [x1, zeros(1,n-n1)]; %add zeros if needed

x2 = [x2, zeros(1,n-n2)]; %add zeros if needed

end

ccirc1 = cconv(x1,x2,n1+n2-1);

subplot(3,2,5); %2nd subplot

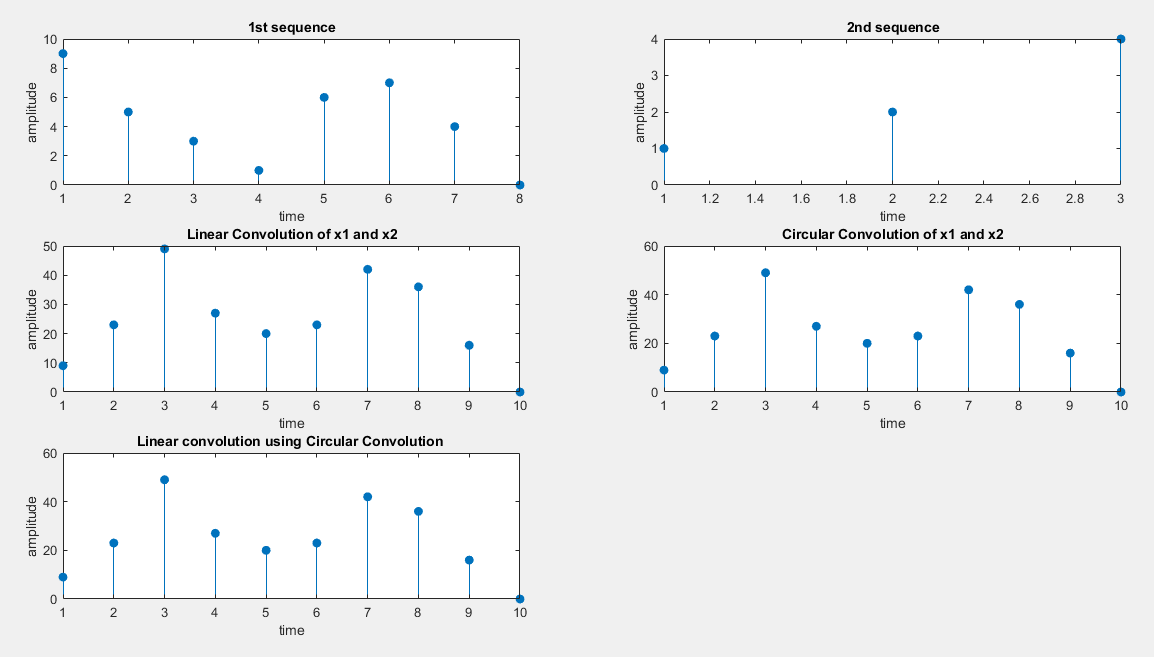
stem(ccirc1,'filled'); %discrete values of n length

xlabel('time');

ylabel('amplitude');

title('Linear convolution using Circular ');%Circular Convolution

**Output:**



**EXPERIMENT 5**

**Write a MATLAB script to perform the convolution of sequences using:  
1. Overlap-Add Method  
2. Overlap-Save Method**

**Code:**

clc;

clear;

close all;

disp('Enter the method through which you want to do linear convolution-');

c = input('(Overlap-Add Method = 1; Overlap-Save Method=2): ');

x1 = input('Enter the input sequence: ');

x2 = input('Enter the impulse sequence: ');

n1 = length(x1);

n2 = length(x2);

b = n2-1;

Ls = n1+n2-1;

L = floor(n1/n2);

N = L+n2-1;

if c == 1

x1 = [x1 zeros(1,N)];

sum = 0;

for k=0:N-1

xk=x1(k\*L+1:k\*L+L);

xk=[xk zeros(1,L-1)];

yk=cconv(xk,x2,N);

yk=[yk zeros(1,n1-1)];

yk=circshift(yk,L\*k);

sum=sum+yk;

end

y=sum(1:Ls);

y = round(y,0);

disp('Convolution using Overlap-Add Method-');

disp(y);

elseif c ==2

K = floor((n1+b-1)/L);

x1 = [zeros(1,b) x1 zeros(1,N-1)];

for i = 0:K

xi = x1(i\*L+1:i\*L+N);

y(i+1,:) = cconv(xi,x2,N);

end

y = round(y,0);

y = y(:,n2:N)';

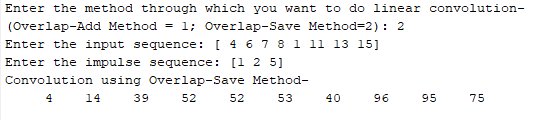
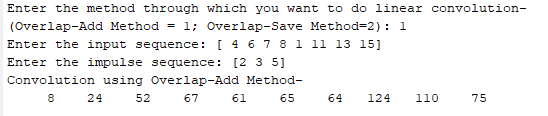
y = y(:)';

disp('Convolution using Overlap-Save Method-');

disp(y);

end

**Output :**



**EXPERIMENT 6**

Write a MATLAB Script to design the FIR filter (low pass, high pass, band pass, band stop) using Window Method.

1. Rectangular

2. Hamming

3. Hann

4. Kaiser

**Code:**

clc;%clear console

clear;%clear variables

close all;%close all figures

n = 20;%order of filter

fp = 200;%pass band frequency

fs = 600;%stop band frequency

f = 2000;%sampling frequency

wp = 2\*(fp/f);%pass band frequency in rads

ws = 2\*(fs/f);%stop band frequency in rads

window\_1 = boxcar(n+1);%rectangular window

window\_2 = hamming(n+1);%hamming windoe

window\_3 = hanning(n+1);%hanning window

window\_4 = kaiser(n+1); %kaiser window

%attaching all the doubles to use in loop conveniently

y = [window\_1 ,window\_2 ,window\_3 ,window\_4];

%to be used in titles

name = ["rectangular","hamming","hanning","kaiser"];

%ploting loops

for i = 1:4

window = y(1:n+1,i);%takes one window at a time

t = ["high","low","bandpass","stop"];%four different types of filter

for x = 1:length(t)

if x < 3

wn = wp;%single frequency for high and low pass filters

else

wn = [wp,ws];%two frequencies for band filters

end

b = fir1(n,wn,t(x),window);%filter function

[H,w] = freqz(b,1);%transfer function

figure (i)%individual figures for each window

subplot(4,2,2\*x-1)

plot(w/pi,20\*log(abs(H)));%magnitude

xlabel('normalized frequency');

ylabel('mag in db');

title ('mag response for ' + name(i) + ' as '+ t(x) +' pass filter ');

subplot(4,2,2\*x)

plot(w/pi,angle(H));%phase

xlabel('normalized frequency');

ylabel('angle');

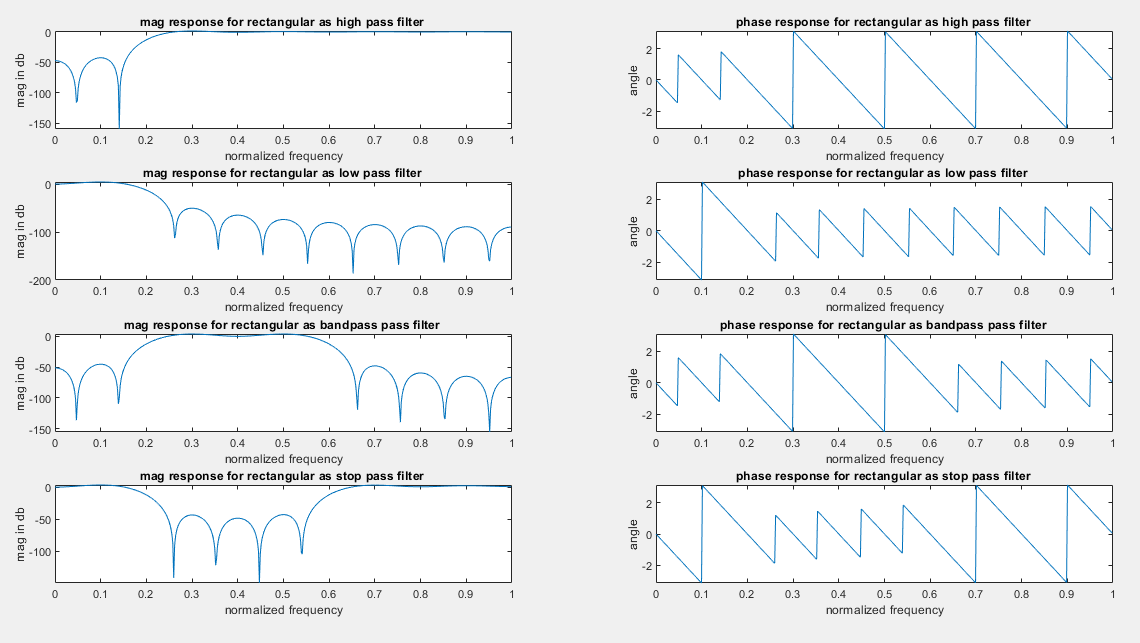
title ('phase response for ' + name(i) +' as '+ t(x) +' pass filter ');

end

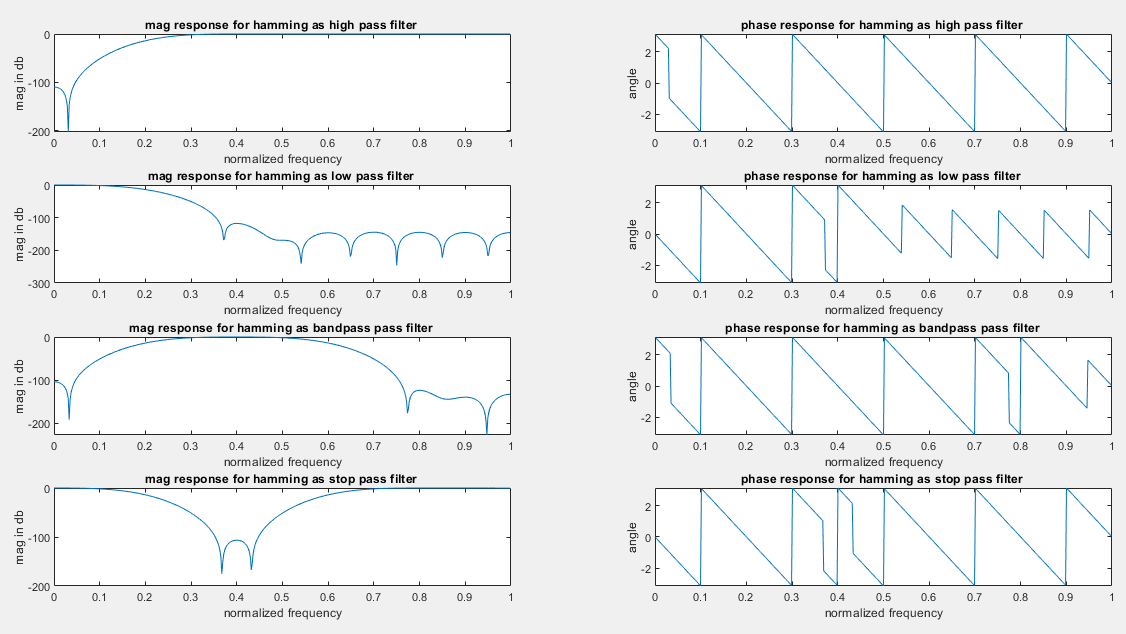
end

**Output:**

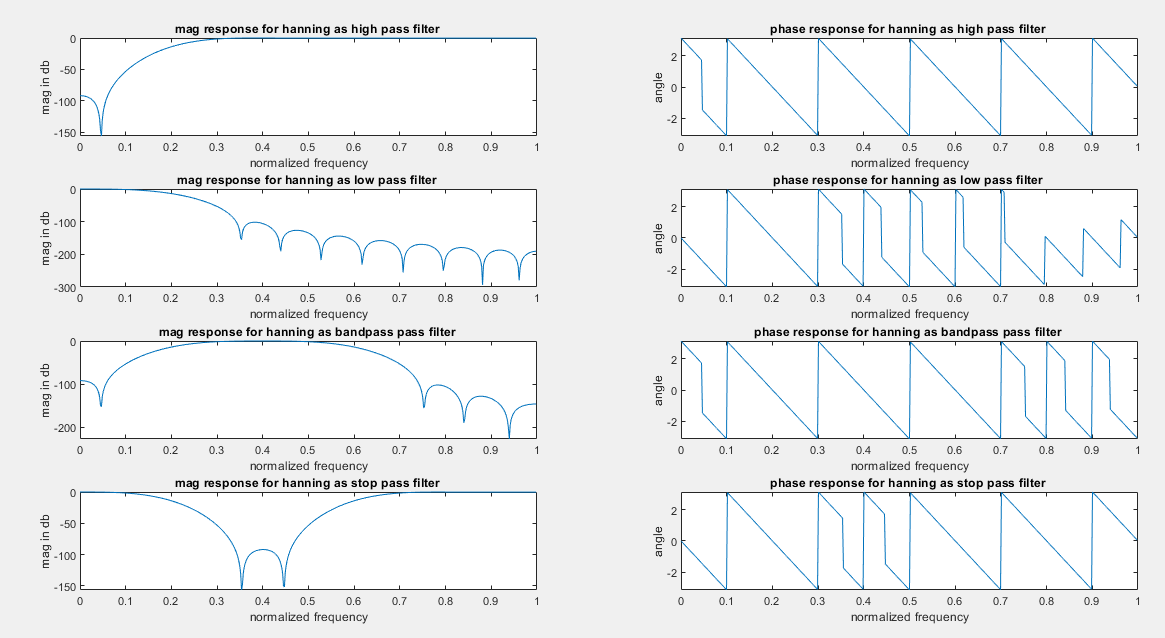
Rectangular:

****

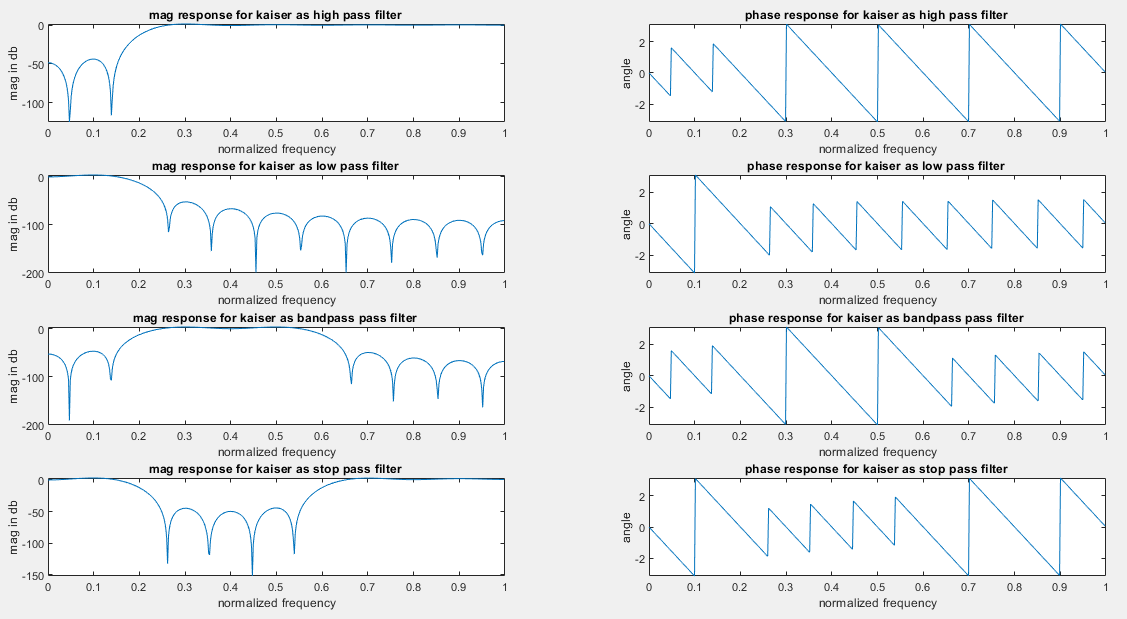
Hamming:



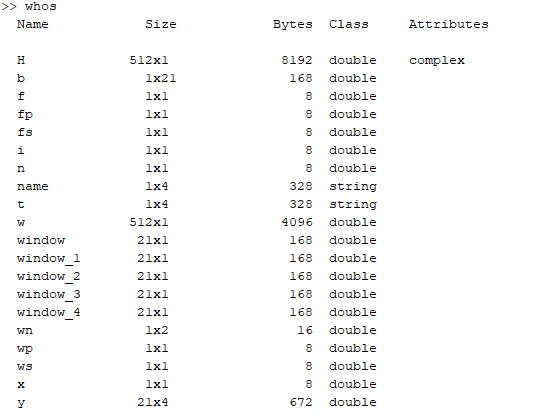
Hanning:



Kaiser:



List of variables:



**EXPERIMENT 7**

Write a MATLAB Script to design the Butterworth, Chebyshev and Elliptic filters based on  
1. Bilinear Transformation  
2. Impulse Invariant Transformation

**Code:**

%Write a MATLAB Script to design the Butterworth, Chebyshev type-1, Chebyshev type-2 and Elliptic filters based on

%1. Bilinear Transformation

%2. Impulse Invariant Transformation

clc;

clear;

close all;

type = input('Enter the type of filter(Low Pass= 1, High Pass= 2,Band Pass= 3, Band Stop= 4): ');

g = input('Enter the design of filter( Butterworth= 1, Chebyshev type1= 2, Chebyshev type2= 3, Elliptic= 4 ): ');

k = input('Enter the type of Filter discretization functions(Bilinear= 1, Impulse Invariant= 2 ): ');

rp = input('Enter the pass band ripple: ');

rs = input('Enter the stop band attenuation: ');

wp = input('Enter the pass band frequency(Hz): ');

ws = input('Enter the stop band frequency(Hz): ');

fs = input('Enter the sampling frequency(Hz): ');

wp = wp/(fs/2);

ws = ws/(fs/2);

if(type == 1)

if(g == 1)

[n,wn] = buttord(wp,ws,rp,rs);

[b,a] = butter(n,wn,'low');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Low Pass Butterworth Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Low Pass Butterworth Filter',n));

elseif(g == 2)

[n,wp] = cheb1ord(wp,ws,rp,rs);

[b,a] = cheby1(n,rp,wp,'low');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Low Pass Chebyshev Type 1 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Low Pass Chebyshev Type 1 Filter',n));

elseif(g == 3)

[n,ws] = cheb2ord(wp,ws,rp,rs);

[b,a] = cheby2(n,rs,ws,'low');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Low Pass Chebyshev Type 2 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Low Pass Chebyshev Type 2 Filter',n));

elseif(g == 4)

[n,wp] = ellipord(wp,ws,rp,rs);

[b,a] = ellip(n,rp,rs,wp,'low');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Low Pass Elliptic Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Low Pass Elliptic Filter',n));

end

elseif(type == 2)

if(g == 1)

[n,wn] = buttord(wp,ws,rp,rs);

[b,a] = butter(n,wn,'high');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital High Pass Butterworth Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog High Pass Butterworth Filter',n));

elseif(g == 2)

[n,wp] = cheb1ord(wp,ws,rp,rs);

[b,a] = cheby1(n,rp,wp,'high');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital High Pass Chebyshev Type 1 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog High Pass Chebyshev Type 1 Filter',n));

elseif(g == 3)

[n,ws] = cheb2ord(wp,ws,rp,rs);

[b,a] = cheby2(n,rs,ws,'high');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital High Pass Chebyshev Type 2 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog High Pass Chebyshev Type 2 Filter',n));

elseif(g == 4)

[n,wp] = ellipord(wp,ws,rp,rs);

[b,a] = ellip(n,rp,rs,wp,'high');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital High Pass Elliptic Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog High Pass Elliptic Filter',n));

end

elseif(type == 3)

if(g == 1)

[n,wn] = buttord(wp,ws,rp,rs);

[b,a] = butter(n,wn,'bandpass');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandpass Butterworth Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandpass Butterworth Filter',n));

elseif(g == 2)

[n,wp] = cheb1ord(wp,ws,rp,rs);

[b,a] = cheby1(n,rp,wp,'bandpass');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandpass Chebyshev Type 1 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandpass Chebyshev Type 1 Filter',n));

elseif(g == 3)

[n,ws] = cheb2ord(wp,ws,rp,rs);

[b,a] = cheby2(n,rs,ws,'bandpass');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandpass Chebyshev Type 2 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandpass Chebyshev Type 2 Filter',n));

elseif(g == 4)

[n,wp] = ellipord(wp,ws,rp,rs);

[b,a] = ellip(n,rp,rs,wp,'bandpass');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandpass Elliptic Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandpass Elliptic Filter',n));

end

elseif(type == 4)

if(g == 1)

[n,wn] = buttord(wp,ws,rp,rs);

[b,a] = butter(n,wn,'stop');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandstop Butterworth Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandstop Butterworth Filter',n));

elseif(g == 2)

[n,wp] = cheb1ord(wp,ws,rp,rs);

[b,a] = cheby1(n,rp,wp,'stop');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandstop Chebyshev Type 1 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandstop Chebyshev Type 1 Filter',n));

elseif(g == 3)

[n,ws] = cheb2ord(wp,ws,rp,rs);

[b,a] = cheby2(n,rs,ws,'stop');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandstop Chebyshev Type 2 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandstop Chebyshev Type 2 Filter',n));

elseif(g == 4)

[n,wp] = ellipord(wp,ws,rp,rs);

[b,a] = ellip(n,rp,rs,wp,'stop');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandstop Elliptic Filter',n));

figure(2);

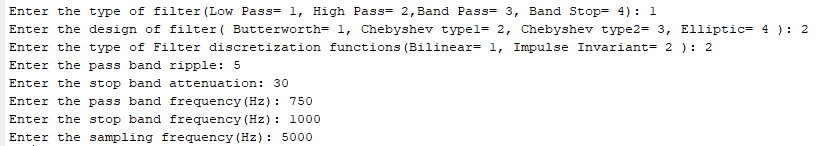
freqz(b,a,1024,fs);

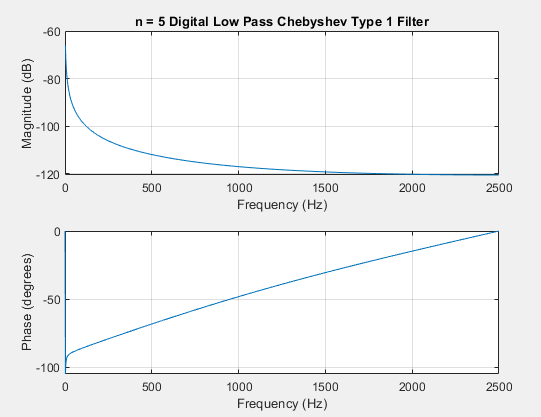
title(sprintf('n = %d Analog Bandstop Elliptic Filter',n));

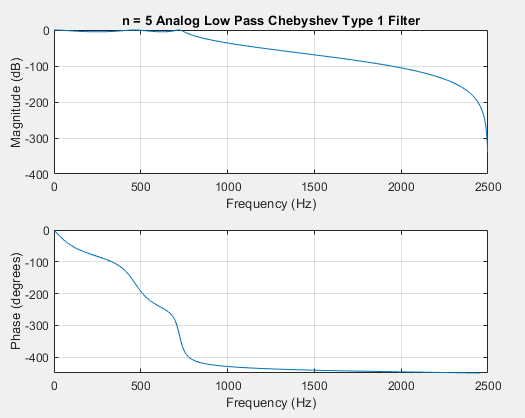
end

end

**Output :**

****

****

****

**EXPERIMENT 8**

Write a MATLAB Script to find the impulse response and step response of the designed FIR and IIR filters( in Lab6 and Lab7) .

%Write a MATLAB Script to design the FIR filter using Window Method. %1. Rectangular (rectwin) 2. Hamming 3. Hann 4. Kaiser

**Code:**

clc;

clear;

close all;

type = input('Enter the type of filter-1.Low Pass,2. High Pass,3.Band Pass,4. Band Stop: ');

win = input('Enter the type of window- 1.Rectangular, 2.Hamming, 3.Hann,4. Kaiser: ');

n = input('Enter the order of the filter: ');

if(type == 1)

n1 = input('Enter the cutoff frequency(in multiples of pi) : ');

if(win == 1)

fil = fir1(n,n1,'low',rectwin(n+1));

freqz(fil,1);

title('Low Pass Filter with Rectangular Window');

elseif(win == 2)

fil = fir1(n,n1,'low',hamming(n+1));

freqz(fil,1);

title('Low Pass Filter with Hamming Window');

elseif(win == 3)

fil = fir1(n,n1,'low',hann(n+1));

freqz(fil,1);

title('Low Pass Filter with Hann Window');

elseif(win == 4)

fil = fir1(n,n1,'low',kaiser(n+1));

freqz(fil,1);

title('Low Pass Filter with Kaiser Window');

end

elseif(type == 2)

n1 = input('Enter the cutoff frequency(in multiples of pi) : ');

if(mod(n,2))

n=n+1;

end

if(win == 1)

fil = fir1(n,n1,'high',rectwin(n+1));

freqz(fil,1);

title('High Pass Filter with Rectangular Window');

elseif(win == 2)

fil = fir1(n,n1,'high',hamming(n+1));

freqz(fil,1);

title('High Pass Filter with Hamming Window');

elseif(win == 3)

fil = fir1(n,n1,'high',hann(n+1));

freqz(fil,1);

title('High Pass Filter with Hann Window');

elseif(win == 4)

fil = fir1(n,n1,'high',kaiser(n+1));

freqz(fil,1);

title('High Pass Filter with Kaiser Window');

end

elseif(type == 3)

n1 = input('Enter the first cutoff frequency(in multiples of pi): ');

n2 = input('Enter the second cutoff frequency(in multiples of pi): ');

if(win == 1)

fil = fir1(n,[n1 n2],'bandpass',rectwin(n+1));

freqz(fil,1);

title('Band Pass Filter with Rectangular Window');

elseif(win == 2)

fil = fir1(n,[n1 n2],'bandpass',hamming(n+1));

freqz(fil,1);

title('Band Pass Filter with Hamming Window');

elseif(win == 3)

fil = fir1(n,[n1 n2],'bandpass',hann(n+1));

freqz(fil,1);

title('Band Pass Filter with Hann Window');

elseif(win == 4)

fil = fir1(n,[n1 n2],'bandpass',kaiser(n+1));

freqz(fil,1);

title('Band Pass Filter with Kaiser Window');

end

elseif(type == 4)

n1 = input('Enter the first cutoff frequency(in multiples of pi): ');

n2 = input('Enter the second cutoff frequency(in multiples of pi): ');

if(mod(n,2))

n=n+1;

end

if(win == 1)

fil = fir1(n,[n1 n2],'stop',rectwin(n+1));

freqz(fil,1);

title('Band Stop Filter with Rectangular Window');

elseif(win == 2)

fil = fir1(n,[n1 n2],'stop',hamming(n+1));

freqz(fil,1);

title('Band Stop Filter with Hamming Window');

elseif(win == 3)

fil = fir1(n,[n1 n2],'stop',hann(n+1));

freqz(fil,1);

title('Band Stop Filter with Hann Window');

elseif(win == 4)

fil = fir1(n,[n1 n2],'stop',kaiser(n+1));

freqz(fil,1);

title('Band Stop Filter with Kaiser Window');

end

end

figure(2);

impz(fil);

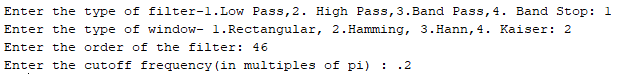
title('Impulse Response of the designed filter');

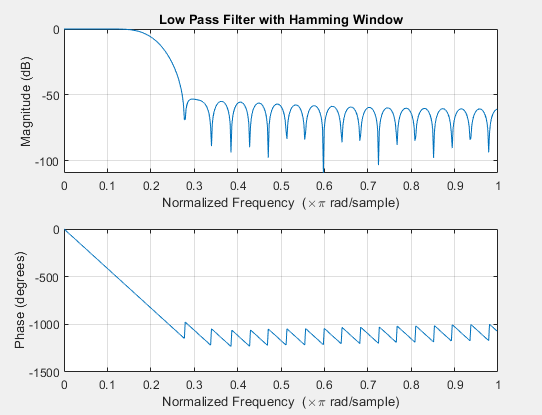
figure(3);

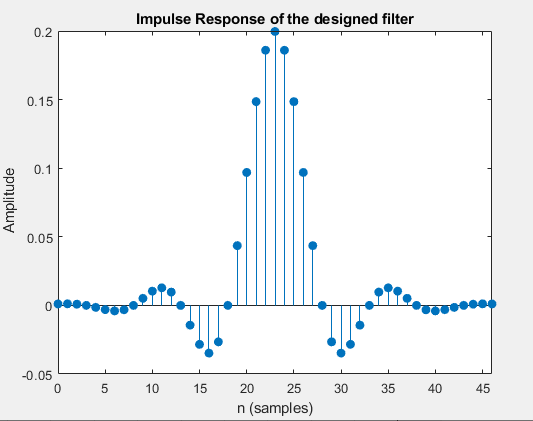
stepz(fil);

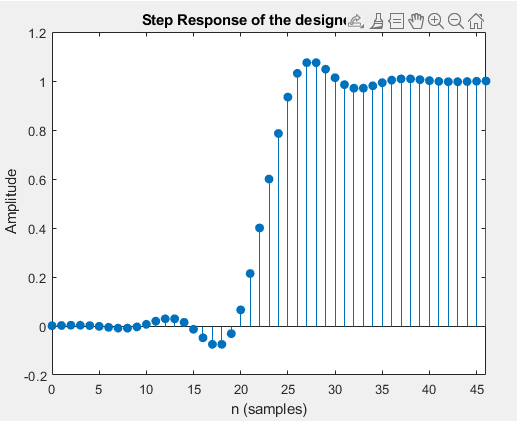
title('Step Response of the designed filter');

**Output:**

****

****

****

****

**Write a MATLAB Script to design the Butterworth, Chebyshev and Elliptic filters based on 1. Bilinear Transformation 2. Impulse Invariant Transformation (Lab 7)**

**Code:**

%Write a MATLAB Script to design the Butterworth, Chebyshev and Elliptic filters based on

% Bilinear Transformation and Impulse Invariant Transformation

clc;

clear;

close all;

type = input('Enter the type of filter- 1.Low Pass, 2.High Pass, 3.Band Pass, 4.Band Stop: ');

g = input('Enter the design of filter- 1.Butterworth, 2.Chebyshev type1, 3.Chebyshev type2, 4.Elliptic: ');

k = input('Enter the type of Filter discretization functions1.Bilinear, 2.Impulse Invariant: ');

rp = input('Enter the pass band ripple: ');

rs = input('Enter the stop band attenuation: ');

wp = input('Enter the pass band frequency(Hz): ');

ws = input('Enter the stop band frequency(Hz): ');

fs = input('Enter the sampling frequency(Hz): ');

wp = wp/(fs/2);

ws = ws/(fs/2);

if(type == 1)

if(g == 1)

[n,wn] = buttord(wp,ws,rp,rs);

[b,a] = butter(n,wn,'low');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Low Pass Butterworth Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Low Pass Butterworth Filter',n));

elseif(g == 2)

[n,wp] = cheb1ord(wp,ws,rp,rs);

[b,a] = cheby1(n,rp,wp,'low');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Low Pass Chebyshev Type 1 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Low Pass Chebyshev Type 1 Filter',n));

elseif(g == 3)

[n,ws] = cheb2ord(wp,ws,rp,rs);

[b,a] = cheby2(n,rs,ws,'low');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Low Pass Chebyshev Type 2 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Low Pass Chebyshev Type 2 Filter',n));

elseif(g == 4)

[n,wp] = ellipord(wp,ws,rp,rs);

[b,a] = ellip(n,rp,rs,wp,'low');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Low Pass Elliptic Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Low Pass Elliptic Filter',n));

end

elseif(type == 2)

if(g == 1)

[n,wn] = buttord(wp,ws,rp,rs);

[b,a] = butter(n,wn,'high');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital High Pass Butterworth Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog High Pass Butterworth Filter',n));

elseif(g == 2)

[n,wp] = cheb1ord(wp,ws,rp,rs);

[b,a] = cheby1(n,rp,wp,'high');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital High Pass Chebyshev Type 1 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog High Pass Chebyshev Type 1 Filter',n));

elseif(g == 3)

[n,ws] = cheb2ord(wp,ws,rp,rs);

[b,a] = cheby2(n,rs,ws,'high');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital High Pass Chebyshev Type 2 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog High Pass Chebyshev Type 2 Filter',n));

elseif(g == 4)

[n,wp] = ellipord(wp,ws,rp,rs);

[b,a] = ellip(n,rp,rs,wp,'high');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital High Pass Elliptic Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog High Pass Elliptic Filter',n));

end

elseif(type == 3)

if(g == 1)

[n,wn] = buttord(wp,ws,rp,rs);

[b,a] = butter(n,wn,'bandpass');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandpass Butterworth Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandpass Butterworth Filter',n));

elseif(g == 2)

[n,wp] = cheb1ord(wp,ws,rp,rs);

[b,a] = cheby1(n,rp,wp,'bandpass');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandpass Chebyshev Type 1 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandpass Chebyshev Type 1 Filter',n));

elseif(g == 3)

[n,ws] = cheb2ord(wp,ws,rp,rs);

[b,a] = cheby2(n,rs,ws,'bandpass');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandpass Chebyshev Type 2 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandpass Chebyshev Type 2 Filter',n));

elseif(g == 4)

[n,wp] = ellipord(wp,ws,rp,rs);

[b,a] = ellip(n,rp,rs,wp,'bandpass');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandpass Elliptic Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandpass Elliptic Filter',n));

end

elseif(type == 4)

if(g == 1)

[n,wn] = buttord(wp,ws,rp,rs);

[b,a] = butter(n,wn,'stop');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandstop Butterworth Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandstop Butterworth Filter',n));

elseif(g == 2)

[n,wp] = cheb1ord(wp,ws,rp,rs);

[b,a] = cheby1(n,rp,wp,'stop');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandstop Chebyshev Type 1 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandstop Chebyshev Type 1 Filter',n));

elseif(g == 3)

[n,ws] = cheb2ord(wp,ws,rp,rs);

[b,a] = cheby2(n,rs,ws,'stop');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandstop Chebyshev Type 2 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandstop Chebyshev Type 2 Filter',n));

elseif(g == 4)

[n,wp] = ellipord(wp,ws,rp,rs);

[b,a] = ellip(n,rp,rs,wp,'stop');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandstop Elliptic Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandstop Elliptic Filter',n));

end

end

figure(3);

impz(bz,az,25);

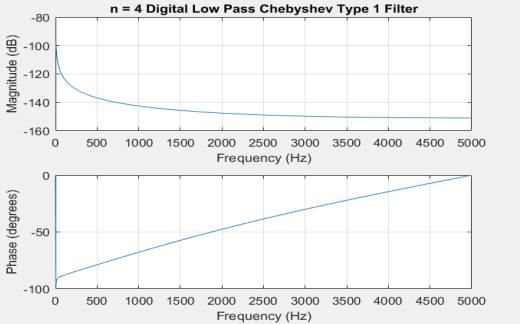
title('Impulse Response of the designed filter');

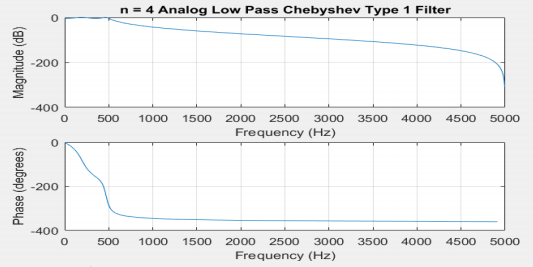
figure(4);

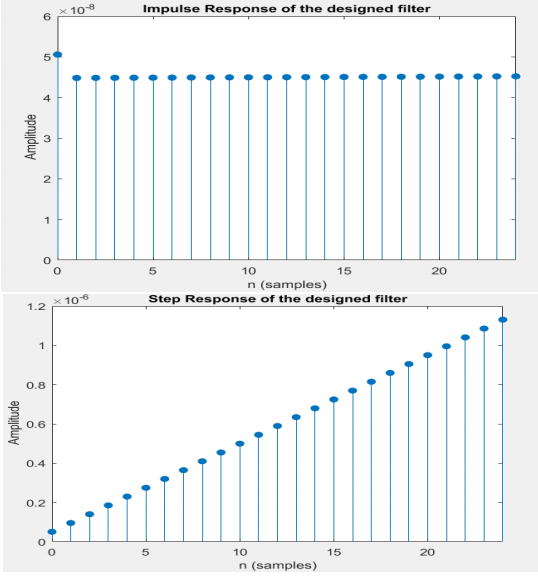
stepz(bz,az,25);

title('Step Response of the designed filter');

**OUTPUT:**

****

****

****

**EXPERIMENT 9**

**Write a MATLAB Script to find the phase angle, phase delay and phase response of the designed IIR filters Lab6.**

**Code:**

clc

clear

disp('The type of Windows are: 1.Rectangular 2.Hamming 3.Hann 4.Kaiser')

c=input('Choose the type of Window: ');

disp('Enter the type of filter')

disp('1. Low pass 2.High Pass 3. Bandpass filter 4. Bandstop filter')

d=input('Choose the filter type: ');

n=input('Enter the order of the filter: ');

wn=input('Enter the frequency: ');

n1=input('Enter the number of evaluation points: ');

nm=mod(n,2);

if nm==1

n=n+1;

end

if c==1

if d==1

b = fir1(n ,wn,'low',rectwin(n+1));

freqz(b,1,n1);

title('Low Pass Filter with Rectangular Window');

elseif d==2

b = fir1(n ,wn,'high',rectwin(n+1));

freqz(b,1,n1);

title('High Pass Filter with Rectangular Window');

theta = angle(z);

elseif d==3

b = fir1(n ,wn,'pass',rectwin(n+1));

freqz(b,1,n1);

title('Bandpass Filter with Rectangular Window');

elseif d==4

b = fir1(n ,wn,'stop',rectwin(n+1));

freqz(b,1,n1);

title('Bandstop Filter with Rectangular Window');

end

elseif c==2

if d==1

b= fir1(n,wn,'low',hamming(n+1));

freqz(b,1,n1);

title('Low Pass Filter with Hamming Window');

elseif d==2

b= fir1(n,wn,'high',hamming(n+1));

freqz(b,1,n1);

title('High Pass Filter with Hamming Window');

elseif d==3

b= fir1(n,wn,'pass',hamming(n+1));

freqz(b,1,n1);

title('Bandpass Filter with Hamming Window');

elseif d==4

b= fir1(n,wn,'stop',hamming(n+1));

freqz(b,1,n1);

title('Bandstop Filter with Hamming Window');

end

elseif c==3

if d==1

b= fir1(n,wn,'low',hann(n+1));

freqz(b,1,n1);

title('Low Pass Filter with Hann Window');

elseif d==2

b= fir1(n,wn,'high',hann(n+1));

freqz(b,1,n1);

title('High Pass Filter with Hann Window');

elseif d==3

b= fir1(n,wn,'pass',hann(n+1));

freqz(b,1,n1);

title('Bandpass Filter with Hann Window');

elseif d==4

b= fir1(n,wn,'stop',hann(n+1));

freqz(b,1,n1);

title('Bandstop Filter with Hann Window');

end

elseif c==4

if d==1

b= fir1(n,wn,'low',kaiser(n+1));

freqz(b,1,n1);

title('Low Pass Filter with Kaiser Window');

elseif d==2

b= fir1(n,wn,'high',kaiser(n+1));

freqz(b,1,n1);

title('High Pass Filter with Kaiser Window');

elseif d==3

b= fir1(n,wn,'pass',kaiser(n+1));

freqz(b,1,n1);

title('Bandpass Filter with Kaiser Window');

elseif d==4

b= fir1(n,wn,'stop',kaiser(n+1));

freqz(b,1,n1);

title('Bandstop Filter with Kaiser Window');

else

disp('Invalid input');

end

end

figure(2);

phasedelay(b,512);

title('Phase delay of the designed filter');

figure(3);

phasez(b);

title('Phase Response of the designed filter');

z = freqz(b,1);

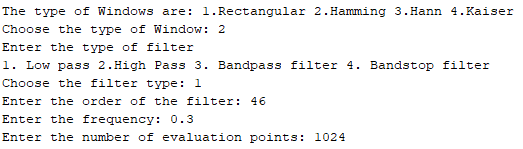
theta = angle(z);

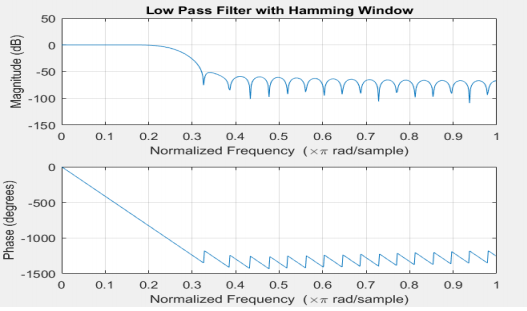
theta = theta';

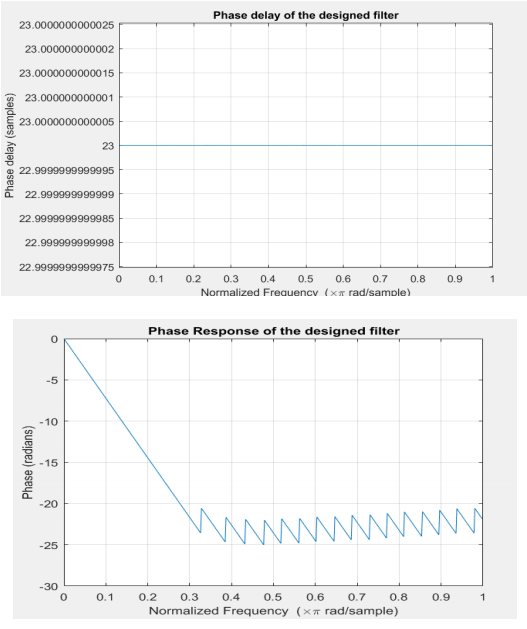
disp('Phase of designed filter: ');

disp(theta);

**Output :**

****

******

****

**EXPERIMENT 9**

**Write a MATLAB Script to find the phase angle, phase delay and phase response of the designed IIR filters Lab7.**

**Code:**

clc;

clear;

close all;

type = input('Enter the type of filter(1. Low Pass, 2. High Pass, 3. Band Pass, 4. Band Stop): ');

g = input('Enter the design of filter(1.Butterworth , 2.Chebyshev type1 , 3. Chebyshev type2 , 4. Elliptic): ');

k = input('Enter the type of Filter discretization functions(1. Bilinear, 2. Impulse Invariant): ');

rp = input('Enter the pass band ripple: ');

rs = input('Enter the stop band attenuation: ');

wp = input('Enter the pass band frequency(Hz): ');

ws = input('Enter the stop band frequency(Hz): ');

fs = input('Enter the sampling frequency(Hz): ');

wp = wp/(fs/2);

ws = ws/(fs/2);

if(type == 1)

if(g == 1)

[n,wn] = buttord(wp,ws,rp,rs);

[b,a] = butter(n,wn,'low');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Low Pass Butterworth Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Low Pass Butterworth Filter',n));

elseif(g == 2)

[n,wp] = cheb1ord(wp,ws,rp,rs);

[b,a] = cheby1(n,rp,wp,'low');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Low Pass Chebyshev Type 1 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Low Pass Chebyshev Type 1 Filter',n));

elseif(g == 3)

[n,ws] = cheb2ord(wp,ws,rp,rs);

[b,a] = cheby2(n,rs,ws,'low');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Low Pass Chebyshev Type 2 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Low Pass Chebyshev Type 2 Filter',n));

elseif(g == 4)

[n,wp] = ellipord(wp,ws,rp,rs);

[b,a] = ellip(n,rp,rs,wp,'low');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Low Pass Elliptic Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Low Pass Elliptic Filter',n));

end

elseif(type == 2)

if(g == 1)

[n,wn] = buttord(wp,ws,rp,rs);

[b,a] = butter(n,wn,'high');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital High Pass Butterworth Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog High Pass Butterworth Filter',n));

elseif(g == 2)

[n,wp] = cheb1ord(wp,ws,rp,rs);

[b,a] = cheby1(n,rp,wp,'high');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital High Pass Chebyshev Type 1 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog High Pass Chebyshev Type 1 Filter',n));

elseif(g == 3)

[n,ws] = cheb2ord(wp,ws,rp,rs);

[b,a] = cheby2(n,rs,ws,'high');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital High Pass Chebyshev Type 2 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog High Pass Chebyshev Type 2 Filter',n));

elseif(g == 4)

[n,wp] = ellipord(wp,ws,rp,rs);

[b,a] = ellip(n,rp,rs,wp,'high');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital High Pass Elliptic Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog High Pass Elliptic Filter',n));

end

elseif(type == 3)

if(g == 1)

[n,wn] = buttord(wp,ws,rp,rs);

[b,a] = butter(n,wn,'bandpass');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandpass Butterworth Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandpass Butterworth Filter',n));

elseif(g == 2)

[n,wp] = cheb1ord(wp,ws,rp,rs);

[b,a] = cheby1(n,rp,wp,'bandpass');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandpass Chebyshev Type 1 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandpass Chebyshev Type 1 Filter',n));

elseif(g == 3)

[n,ws] = cheb2ord(wp,ws,rp,rs);

[b,a] = cheby2(n,rs,ws,'bandpass');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandpass Chebyshev Type 2 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandpass Chebyshev Type 2 Filter',n));

elseif(g == 4)

[n,wp] = ellipord(wp,ws,rp,rs);

[b,a] = ellip(n,rp,rs,wp,'bandpass');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandpass Elliptic Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandpass Elliptic Filter',n));

end

elseif(type == 4)

if(g == 1)

[n,wn] = buttord(wp,ws,rp,rs);

[b,a] = butter(n,wn,'stop');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandstop Butterworth Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandstop Butterworth Filter',n));

elseif(g == 2)

[n,wp] = cheb1ord(wp,ws,rp,rs);

[b,a] = cheby1(n,rp,wp,'stop');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandstop Chebyshev Type 1 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandstop Chebyshev Type 1 Filter',n));

elseif(g == 3)

[n,ws] = cheb2ord(wp,ws,rp,rs);

[b,a] = cheby2(n,rs,ws,'stop');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandstop Chebyshev Type 2 Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandstop Chebyshev Type 2 Filter',n));

elseif(g == 4)

[n,wp] = ellipord(wp,ws,rp,rs);

[b,a] = ellip(n,rp,rs,wp,'stop');

if(k ==1)

[bz,az] = bilinear(b,a,fs);

elseif(k==2)

[bz,az] = impinvar(b,a,fs);

end

freqz(bz,az,1024,fs);

title(sprintf('n = %d Digital Bandstop Elliptic Filter',n));

figure(2);

freqz(b,a,1024,fs);

title(sprintf('n = %d Analog Bandstop Elliptic Filter',n));

end

end

figure(3);

phasedelay(bz,az,512);

title('Phase delay of the designed filter');

figure(4);

phasez(bz,az);

title('Phase Response of the designed filter');

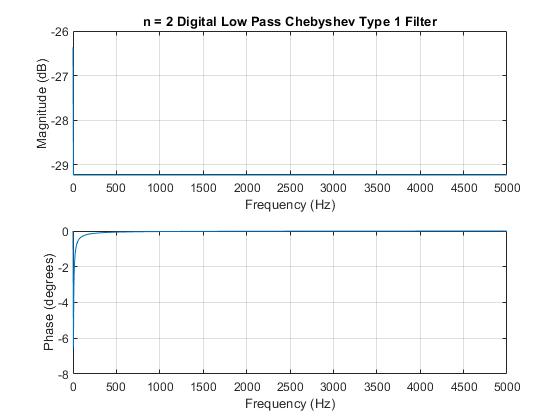
z = freqz(b,a,1024,fs);

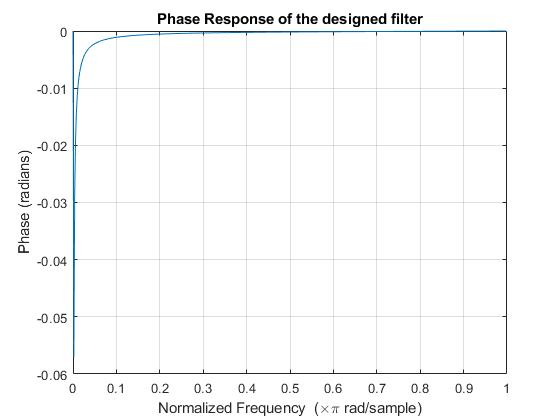
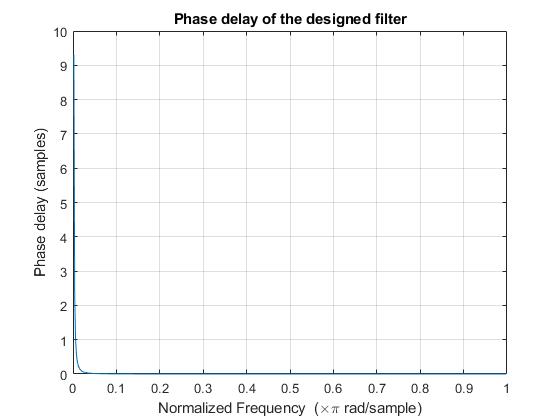
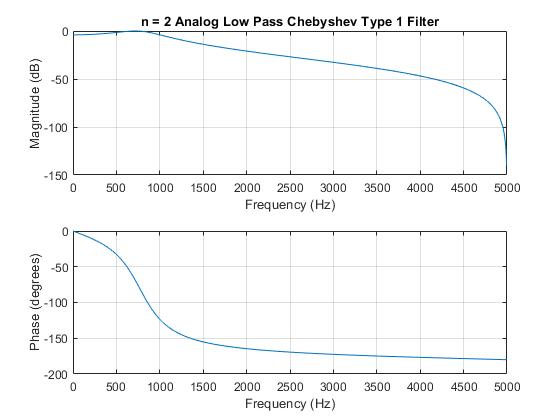
theta = angle(z);

theta = theta';

disp('Phase of designed filter: ');

disp(theta);

****

****

**EXPERIMENT 10**

**Write a MALTAB script to generate and filter the ECG/PPG.**

**Code:**

clc;

clear all;

close all;

ecg=load('C:\Users\Aadhithan\Downloads\PPG\2.txt');

ppg=load('C:\Users\Aadhithan\Downloads\PPG\2.txt');

f\_s=250;

N=length(ecg);

t=[0:N-1]/f\_s; %time period(total sample/Fs )

figure

plot(t,ecg); title(' ECG Data plotting ')

xlabel('time')

ylabel('amplitude')

z=ppg(200:950,1); % PPG signal

figure

plot(z,'r');

title('PPG Data Ploting');

xlabel('time');

ylabel('amplitude');

w=50/(250/2);

bw=w;

[num,den]=iirnotch(w,bw); % notch filter implementation

ecg\_notch=filter(num,den,ecg);

figure,

N1=length(ecg\_notch);

t1=[0:N1-1]/f\_s;

plot(t1,ecg\_notch,'b'); title('Filtered ECG signal ')

xlabel('time')

ylabel('amplitude')

w=50/(250/2);

bw=w;

fs\_ppg=700

[a,b]=iirnotch(w,bw); % notch filter implementation

ppg\_notch=filter(a,b,ppg);

N2=length(ppg);

t1=[0:N2-1]/fs\_ppg

figure

plot(t1,ppg\_notch,'r'); title('Filtered PPG signal ')

xlabel('time')

ylabel('amplitude')

%% Task 2-a

figure, subplot 211% study useage of subplot under help section

plot(t,ecg); title('ECG Data plotting ')

xlabel('time')

ylabel('amplitude')

legend(' ORIGINAL ECG SIGNAL')

subplot 212

plot(t1,ecg\_notch,'r'); title('Filtered ECG signal ')

xlabel('time')

ylabel('amplitude')

legend(' Flitered ECG SIGNAL')

figure

subplot 211% study useage of subplot under help section

plot(z); title('PPG Data plotting ')

xlabel('time')

ylabel('amplitude')

legend(' ORIGINAL PPG SIGNAL')

subplot 212

plot(t1,ppg\_notch,'r'); title('Filtered ECG signal ')

xlabel('time')

ylabel('amplitude')

legend(' Flitered PPG SIGNAL')

%% plot the Both signal Original and FILter signal

figure, plot(t(1:201),ecg(1:201),'r');

title('Data plotting for 0 to 0.804 time frame')

xlabel('time')

ylabel('amplitude')

hold on

figure

plot(t1(1:201),ecg\_notch(1:201),'g');

legend('ORIGINAL ECG SIGNAL',' Flitered ECG SIGNAL')

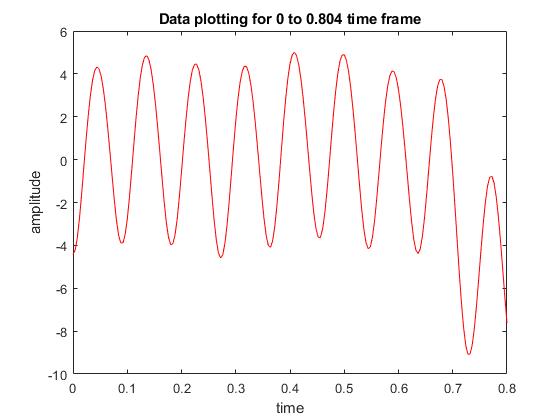
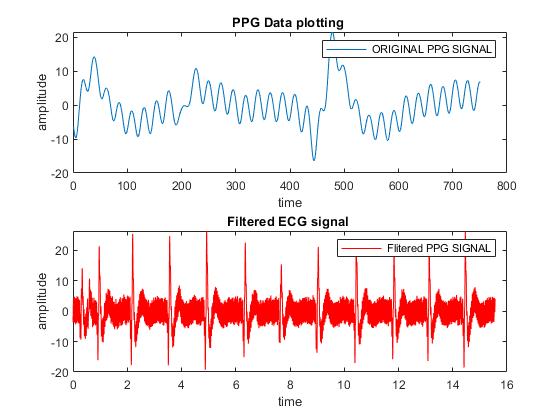
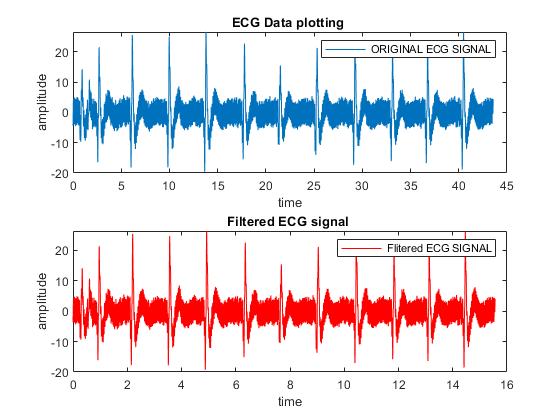
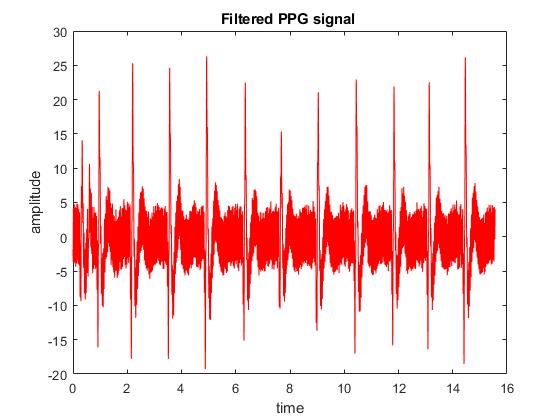
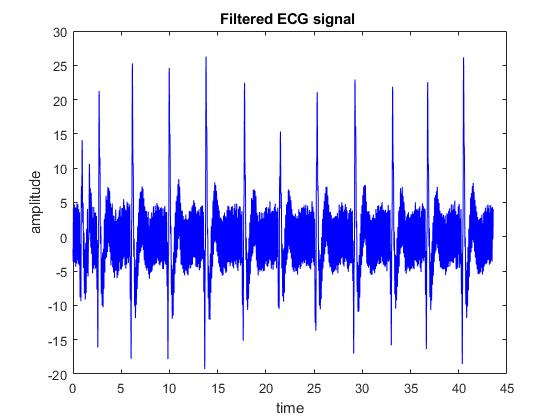
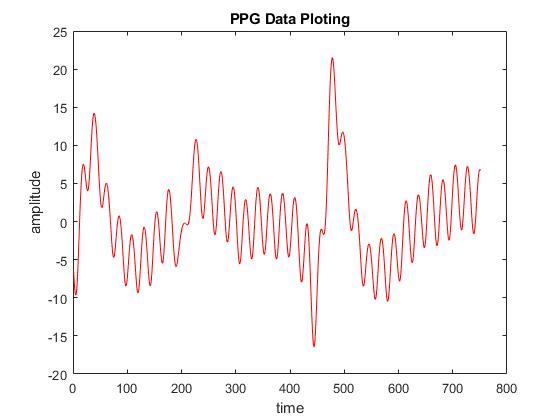
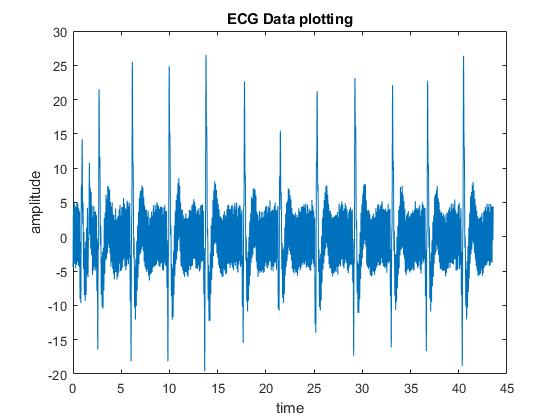
plot(t1(1:201),z(1:201),'y');

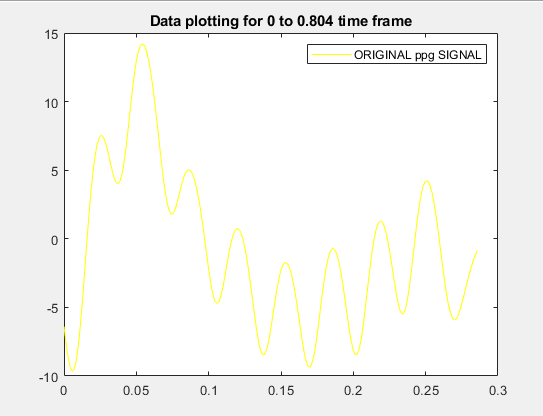
title('Data plotting for 0 to 0.804 time frame')

legend('ORIGINAL ppg SIGNAL',' Flitered ppg SIGNAL')

hold off

**OUTPUT:**

****

****

**EXPERIMENT 11**

**Write a MALTAB script for (time-domain parameters)  
1. ECG/PPG peak detection  
2. ECG/PPG feature extraction**

**Code:**

clc;

clear all;

clear;

fps=100;

vec=importdata('2.txt');

[X Y Z]=pca(vec);

sig = Y;

figure(1)

plot(sig)

title('Raw Data');

sig1 = sig - mean (sig ); % cancel DC conponents

sig1 = sig1/ max( abs(sig1 )); % normalize to one

% LPF (1-z^-6)^2/(1-z^-1)^2

b=[1 0 0 0 0 0 -2 0 0 0 0 0 1];

a=[1 -2 1];

h\_LP=filter(b,a,[1 zeros(1,12)]); % transfer function of LPF

x2p = conv (sig1 ,h\_LP);

%x2 = x2 (6+[1: N]); %cancel delay

x2p = x2p/ max( abs(x2p )); % normalize , for convenience .

figure(2);

plot(x2p);

title('Filtered Data');

nFrames=length(sig1); % Signal length

t = [0:nFrames-1]/fps;

[peaks, peak\_pos,foots, foot\_pos] = peakdetect(fps, nFrames, sig1);

peaks=peaks(2:end);

foots=foots(2:end);

peak\_pos=peak\_pos(2:end);

foot\_pos=foot\_pos(2:end);

npks=length(peak\_pos);

figure(3)

plot(t,sig1, peak\_pos/fps,peaks,'\*r',foot\_pos/fps,sig1(foot\_pos),'\*m')

title('Peak Detection');

[pp,ff,fp,pf,ppbyff,ppbyfp,fpbyff,fpbypf,plht,crti,sarea,darea,totalarea,ratioarea,deti,AI,RI,npks]=timeDomainParameters(sig1,fps);

% else

% [pp,ff,fp,pf,ppbyff,ppbyfp,fpbyff,fpbypf,plht,crti,sarea,darea,totalarea,ratioarea,deti,AI,RI]=timeDomainParameters(SR(c(i):c(i)+124),fps);

%

% end

time=length(sig1)/fps;

bpm=(npks/time)\*60

name='2';

DATA1(1)=pp;

DATA1(2)=ff;

DATA1(3)=fp;

DATA1(4)=pf;

DATA1(5)=ppbyff;

DATA1(6)=ppbyfp;

DATA1(7)=fpbyff;

DATA1(8)=fpbypf;

DATA1(9)=plht;

DATA1(10)=crti;

DATA1(11)=bpm;

DATA1(12)=sarea;

DATA1(13)=darea;

DATA1(14)=totalarea;

DATA1(15)=ratioarea;

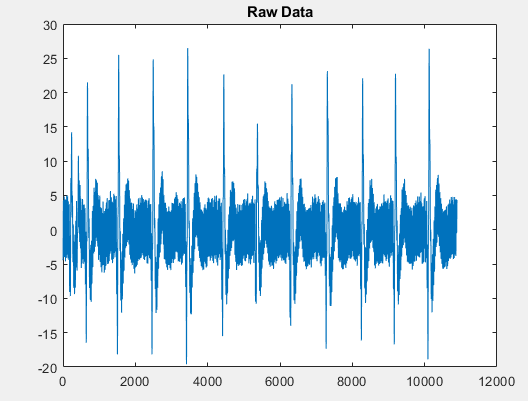
DATA1(16)=deti;

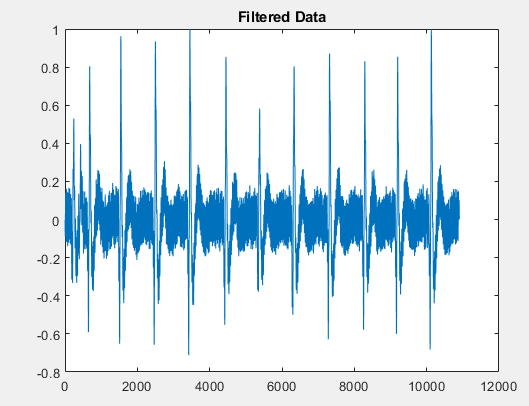
DATA1(17)=AI;

DATA1(18)=RI;

dlmwrite('NEWD.txt',DATA1,'-append')

**OUTPUT:**

****

****

**EXPERIMENT 12**

**Write a MALTAB script for (frequency-domain parameters)  
1. ECG/PPG peak detection  
2. ECG/PPG feature extraction**

**CODE:**

clc;

clear all;

clear;

fps=100;

vec=importdata('2.txt');

[X Y Z]=pca(vec);

sig = Y;

figure(1)

plot(sig)

title('Raw Data');

sig1 = sig - mean (sig ); % cancel DC conponents

sig1 = sig1/ max( abs(sig1 )); % normalize to one

% LPF (1-z^-6)^2/(1-z^-1)^2

b=[1 0 0 0 0 0 -2 0 0 0 0 0 1];

a=[1 -2 1];

h\_LP=filter(b,a,[1 zeros(1,12)]); % transfer function of LPF

x2p = conv (sig1 ,h\_LP);

%x2 = x2 (6+[1: N]); %cancel delay

x2p = x2p/ max( abs(x2p )); % normalize , for convenience .

figure(2);

plot(x2p);

title('Filtered Data');

nFrames=length(sig1); % Signal length

t = [0:nFrames-1]/fps;

[peaks, peak\_pos,foots, foot\_pos] = peakdetect(fps, nFrames, sig1);

peaks=peaks(2:end);

foots=foots(2:end);

peak\_pos=peak\_pos(2:end);

foot\_pos=foot\_pos(2:end);

npks=length(peak\_pos);

figure(3)

plot(t,sig1, peak\_pos/fps,peaks,'\*r',foot\_pos/fps,sig1(foot\_pos),'\*m')

title('Peak Detection');

time=length(sig1)/fps;

bpm=(npks/time)\*60;

name= '2' ;

fps=100;

[psd,domFreq]=FreqDomainParameters( sig1, fps,name);

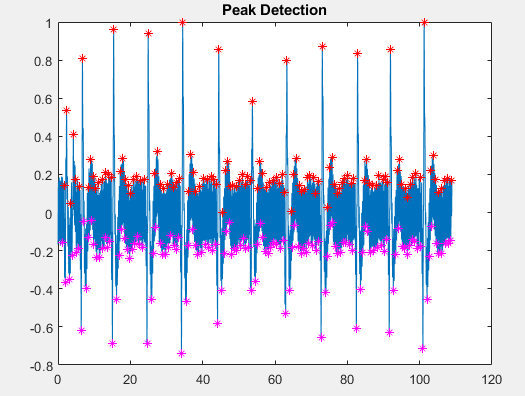
DATA1(1)=bpm;

DATA1(2)=psd;

DATA1(3)=domFreq(1);

dlmwrite('NEWD.txt',DATA1,'-append');

**OUTPUT:**

****