Bangladesh University of Engineering and Technology

Course Number: EEE 312

Course Title: Digital Signal Processing Laboratory

Experiment Number: 05

Name of the Experiment(s):

FIR filter design

Prepared by:

: Tasnimun Razin

Student ID: 1906044

Partner: Tasmin Khan

Student ID: 1906055

Section: A2

Level:3 Term:1

Department: EEE

Problem 1

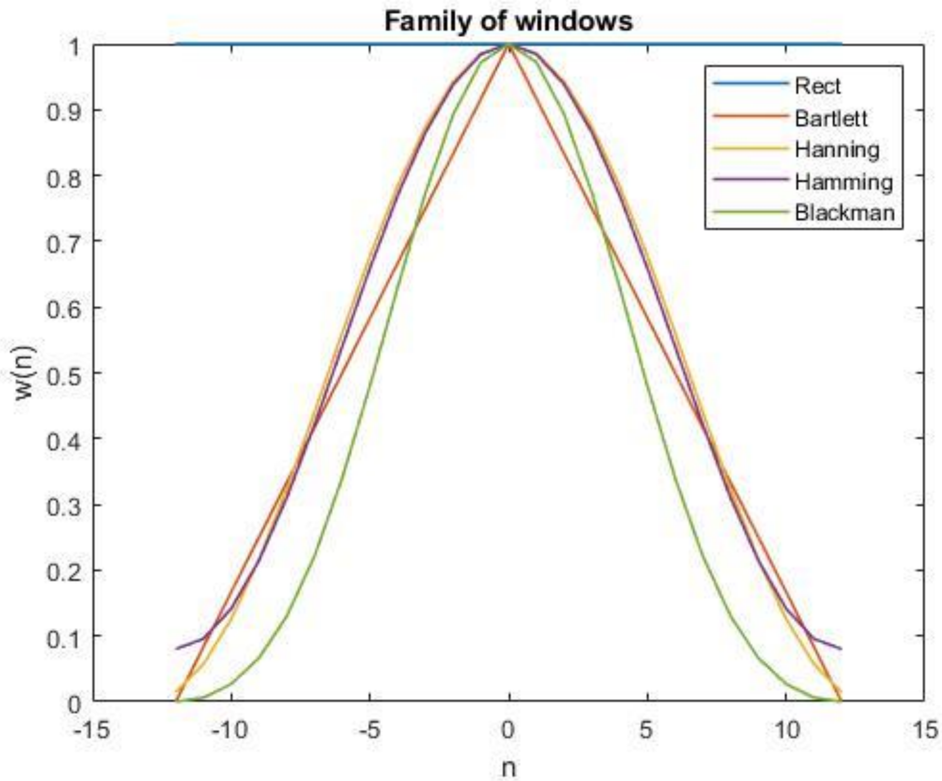
Question:

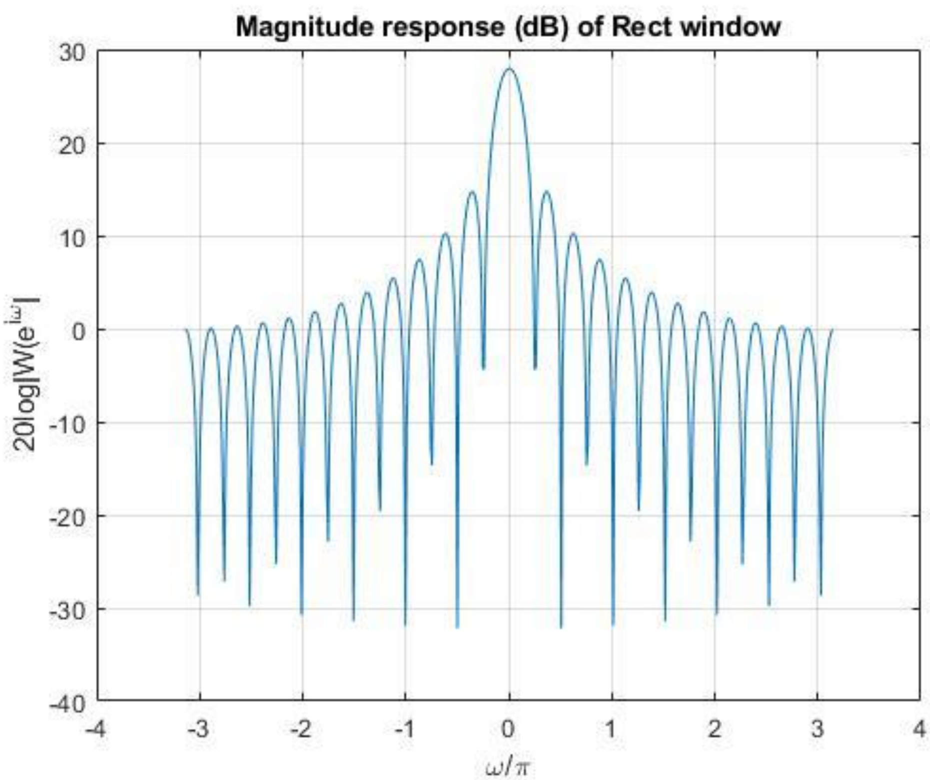
1. Graphically verify the (i) Main lobe widths and (ii) Peak-to-side lobe amplitudes (dB) for different types of window functions for a length of $M = 25$.

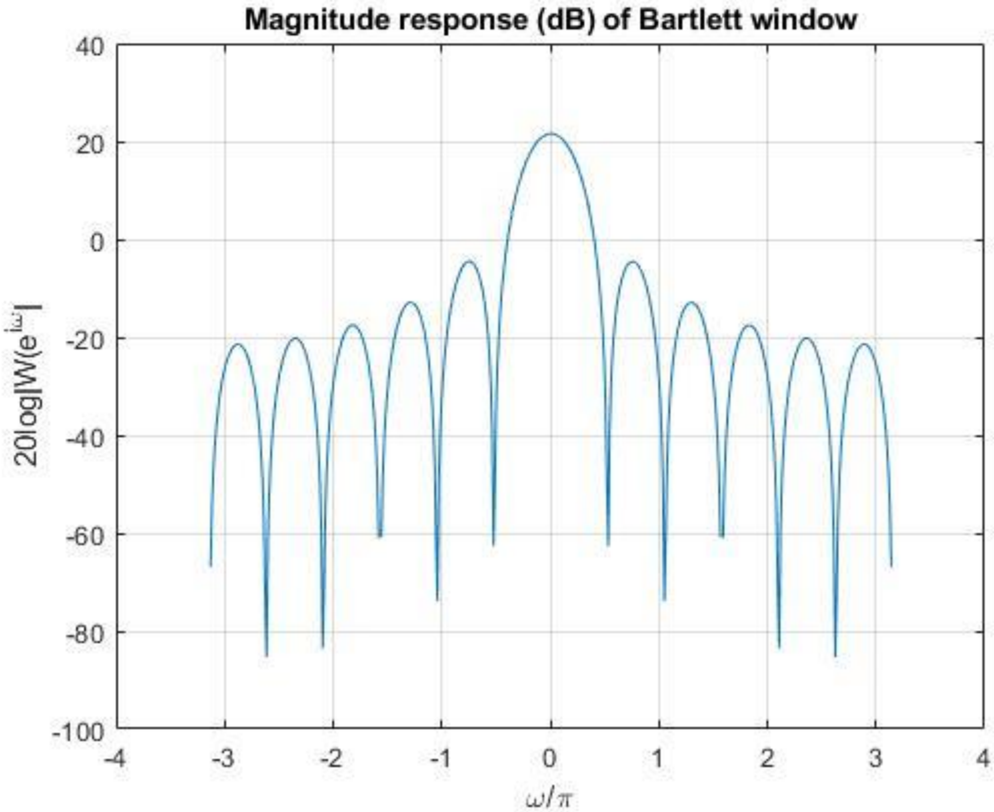
MATLAB code:

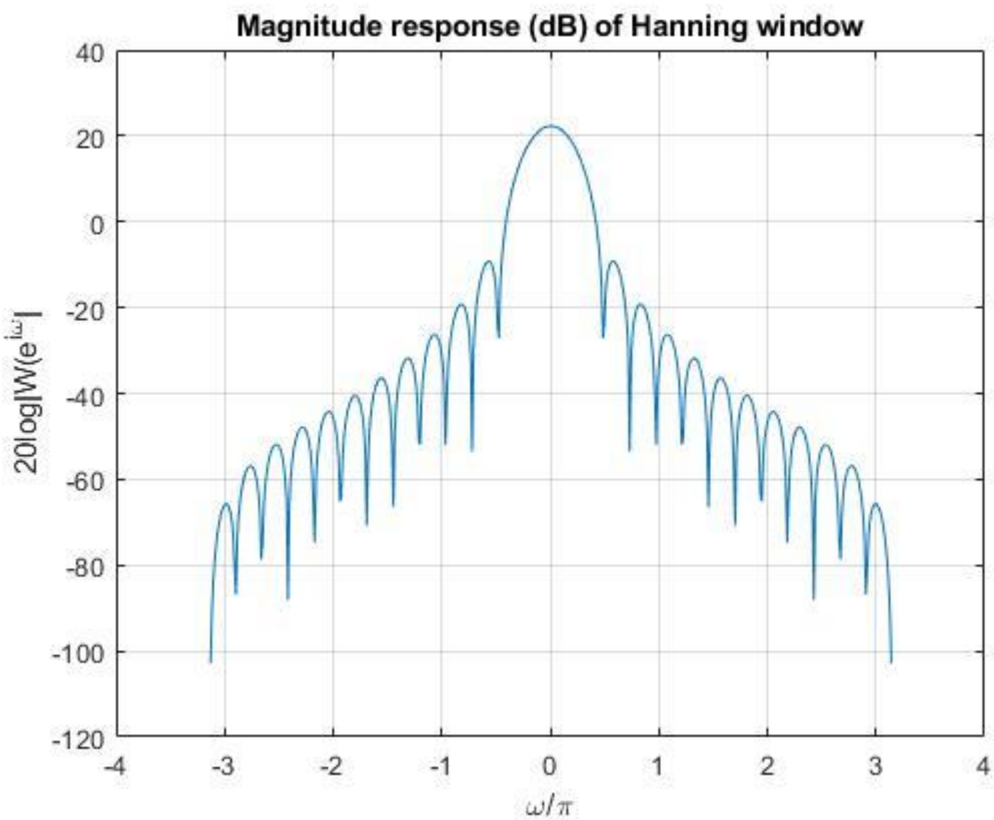
```
clc
clear all
close all

%% building the windows

M=25;
n=-(M-1)/2:(M-1)/2;
Z=512;
titles = {'Rect','Bartlett','Hanning','Hamming','Blackman'};
win = { rectwin(M) , bartlett(M) , hanning(M) , hamming(M) , blackman(M) };

%% plot on a single figure

figure(1)
for i=1:5
    plot(n,win{i},'Linewidth',1.1)
    hold on
end
xlabel('n')
ylabel('w(n)')
title('Family of windows');
legend(titles);

%% magnitude spectrums in dB

for i=1:5
    W=abs(fftshift(fft(win{i},Z)));
    w=linspace(-pi,pi,Z);
    figure(i+1)
    plot(w,20*log10(W))
    title(sprintf('Magnitude response (dB) of %s window', titles{i}))
    xlabel('\omega/\pi')
    ylabel ('20log|W(e^{i\omega}|') ;
    grid on
end
```

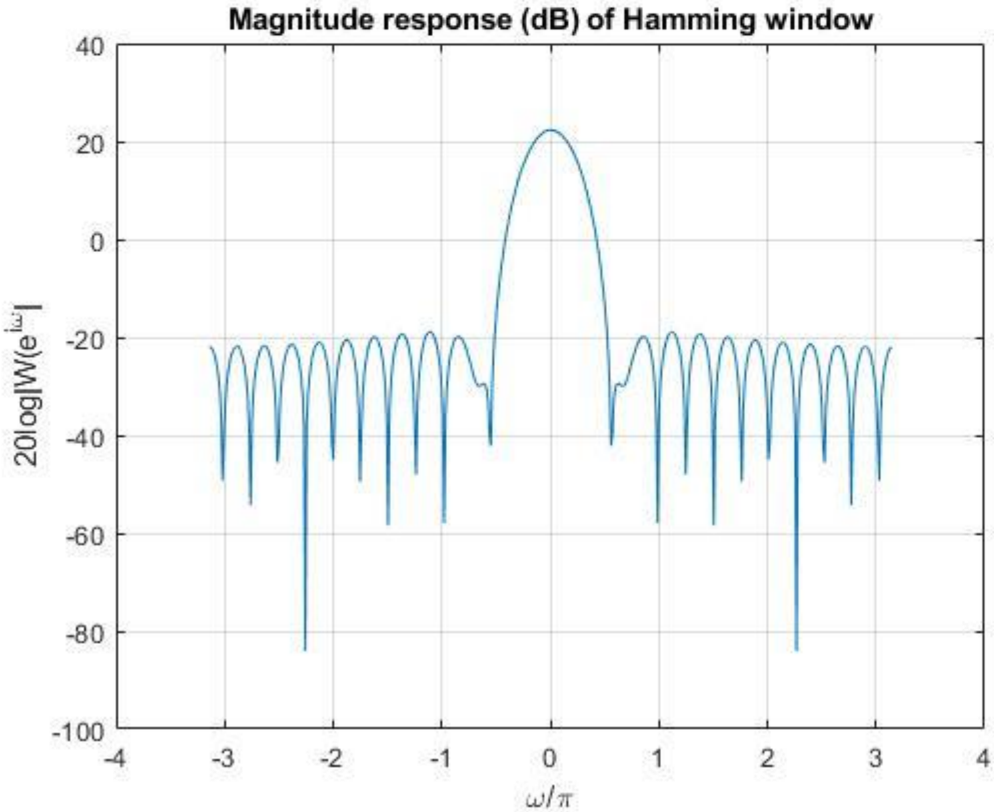
Plot:


Family of windows
Rect
Bartlett
Hanning
Hamming
Blackman
w(n)
n

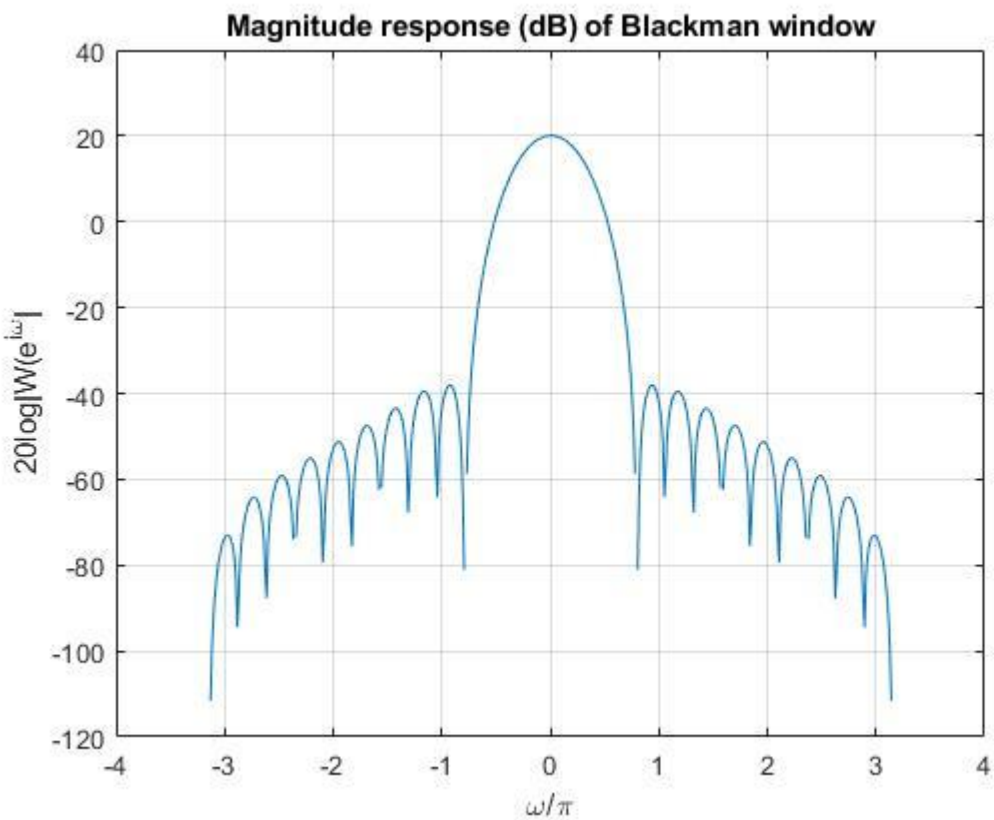

Magnitude response (dB) of Rect window
20log|W(e^{jω})|
ω/π


Magnitude response (dB) of Bartlett window
20log|W(e^{jω})|
ω/π
40
20
0
-20
-40
-60
-80
-100
-4
-3
-2
-1
0
1
2
3
4


Magnitude response (dB) of Hanning window
20log|W(e^{jω})|
ω/π
40
20
0
-20
-40
-60
-80
-100
-120
-4
-3
-2
-1
0
1
2
3
4



Magnitude response (dB) of Hamming window
40
20
0
-20
-40
-60
-80
-100
-4
-3
-2
-1
0
1
2
3
4
20log|W(e^{jω})|
ω/π



Magnitude response (dB) of Blackman window
40
20
0
-20
-40
-60
-80
-100
-120
-4
-3
-2
-1
0
1
2
3
4
20log|W(e^{jω})|
ω/π

Discussion:

The family of windows comparison reveals distinct characteristics in both time and frequency domains. Rectangular window has the narrowest main lobe but highest side lobes, while Blackman window shows the widest main lobe with lowest side lobes. Hanning and Hamming windows provide intermediate performance with good trade-off between main lobe width and side lobe suppression. Bartlett window exhibits moderate side lobe attenuation but wider main lobe than rectangular. The choice of window depends on the specific filtering requirements - rectangular for sharp transitions, Blackman for maximum side lobe suppression.

Problem 2

Question:

2. A discrete-time signal is provided as, $x(n) = 2\sin(0.32\pi n) + 3\sin(0.37\pi n) + \sin(0.4\pi n)$ with a signal length M of your choice. You are to separately use (i) Rectangular, (ii) Bartlett, (iii) Blackman windows on this signal and visualize the frequency domain magnitude distribution. Iteratively find for each case what minimum value of M would make you distinguish the frequency components. Does these M values have any relationships with the 'Main lobe widths' of the individual windows?

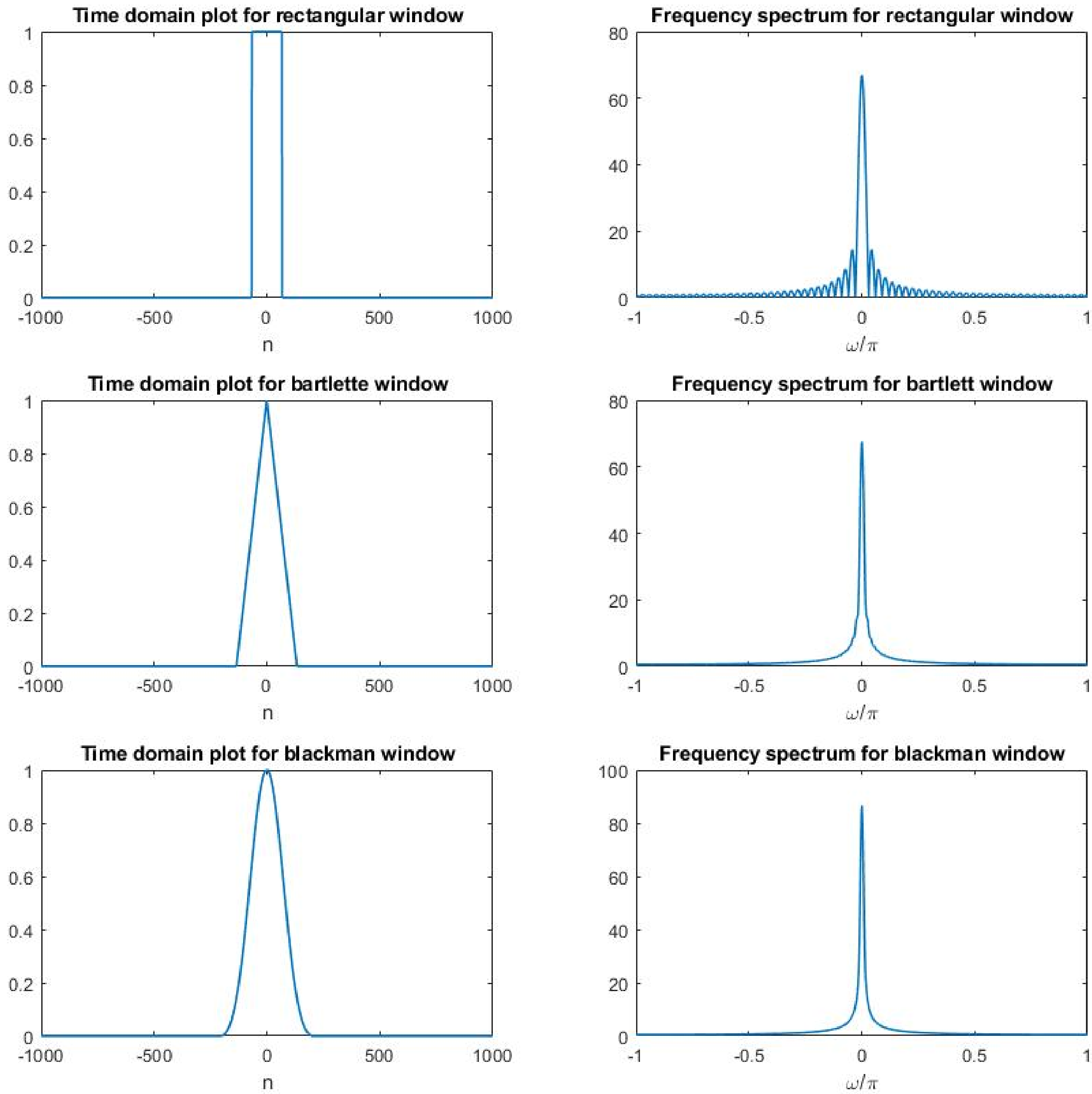
MATLAB code:

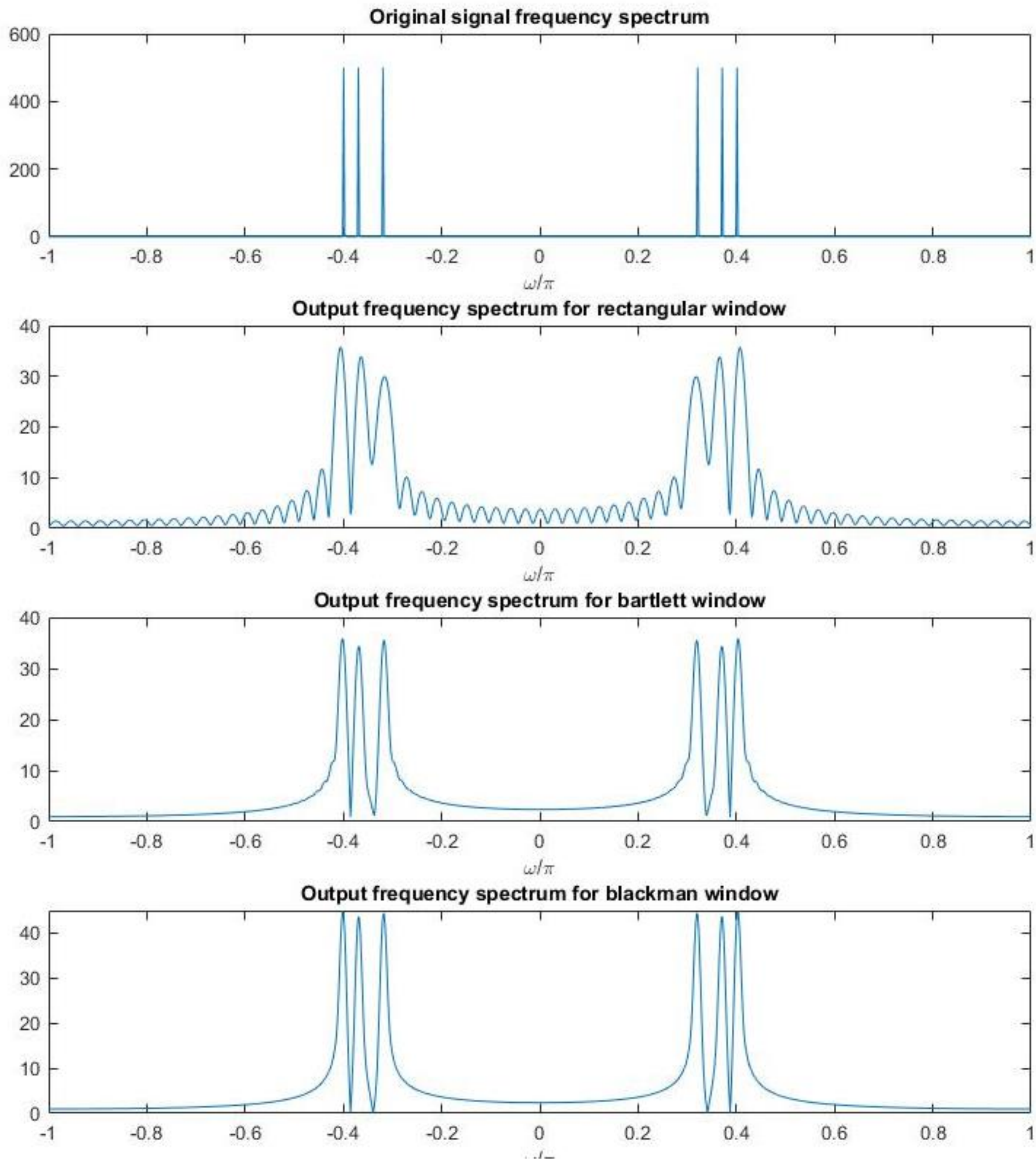
```
clc;
clear;
close all;

n = -1000:1000;
xn = sin(0.32*pi*n)+sin(.37*pi*n)+sin(0.4*pi*n);

%minimum frequency difference 0.03*pi
diff = 0.03*pi;
diff=diff/2;
w = linspace(0,2*pi,1000);
wn= linspace(-pi,pi,1000);

%Windows
N1 = 2;
N2 = 4;
N3 = 6;

%For rectangular
while(1)
    win1 = rectwin(N1)';
    H1 = abs(fft(win1,1000));
    [val,I]=findpeaks(-H1);

    if(numel(I)~=0 && w(I(1))<diff)
        a=3;
        break
    end
    N1 = N1+1;
End
```

```
%For bartlett
while(1)
    win2 = bartlett(N2)';
    H2 = abs(fft(win2,1000));
    [val,I]=findpeaks(-H2);
    if(numel(I)~=0 && w(I(1))<diff)
        break
    end
    N2 = N2+1;
end

% for blackman
while(1)
    win3 = blackman(N3)';
    H3 = abs(fft(win3,1000));
    [val,I]=findpeaks(max(H3)-H3);
    if(numel(I)~=0 && w(I(1))<diff)
        break
    end
    N3 = N3+1;
end

%% Output for different window

n1=-ceil(length(win1)/2):floor((length(win1))/2-1);
n2=-ceil(length(win2)/2):floor((length(win2))/2-1);
n3=-ceil(length(win3)/2):floor((length(win3))/2-1);

w1pad=zeros(1,length(xn));
w1pad(find(n>=n1(1) & n<=n1(end)))=win1;
out1=abs(fftshift(fft((xn.*w1pad),1000)));

w2pad=zeros(1,length(xn));
w2pad(find(n>=n2(1) & n<=n2(end)))=win2;
out2=abs(fftshift(fft((xn.*w2pad),1000)));

w3pad=zeros(1,length(xn));
w3pad(find(n>=n3(1) & n<=n3(end)))=win3;
out3=abs(fftshift(fft((xn.*w3pad),1000)));
```

```
%% plotting

figure (1)
subplot(411)
plot(wn/pi,abs(fftshift(fft(xn,1000))))
title('Original signal frequency spectrum')
xlabel('\omega/\pi')
xlim([-1 1])
subplot(412)
plot(wn/pi,out1)
title('Output frequency spectrum for rectangular window')
xlabel('\omega/\pi')
xlim([-1 1])
subplot(413)
plot(wn/pi,out2)
title('Output frequency spectrum for bartlett window')
xlabel('\omega/\pi')
xlim([-1 1])
subplot(414)
plot(wn/pi,out3)
title('Output frequency spectrum for blackman window')
xlabel('\omega/\pi')
xlim([-1 1])

%% more plotting

figure(2)
subplot(321)
plot(n,w1pad,'LineWidth',1.25)
title('Time domain plot for rectangular window')
xlabel('n')
subplot(322)
plot(wn/pi,abs(fftshift(fft(w1pad,1000))),'LineWidth',1.1)
xlabel('\omega/\pi')
title('Frequency spectrum for rectangular window')

subplot(323)
plot(n,w2pad,'LineWidth',1.25)
title('Time domain plot for bartlette window')
xlabel('n')
subplot(324)
plot(wn/pi,abs(fftshift(fft(w2pad,1000))),'LineWidth',1.1)
xlabel('\omega/\pi')
title('Frequency spectrum for bartlett window')

subplot(325)
plot(n,w3pad,'LineWidth',1.25)
title('Time domain plot for blackman window')
xlabel('n')
subplot(326)
plot(wn/pi,abs(fftshift(fft(w3pad,1000))),'LineWidth',1.1)
xlabel('\omega/\pi')
title('Frequency spectrum for blackman window')
```

Plot:


Time domain plot for rectangular window
Frequency spectrum for rectangular window
Time domain plot for bartlette window
Frequency spectrum for bartlett window
Time domain plot for blackman window
Frequency spectrum for blackman window
n
ω/π
-1000
-500
0
500
1000
-1
-0.5
0.5
1
0.2
0.4
0.6
0.8
20
40
60
80
100

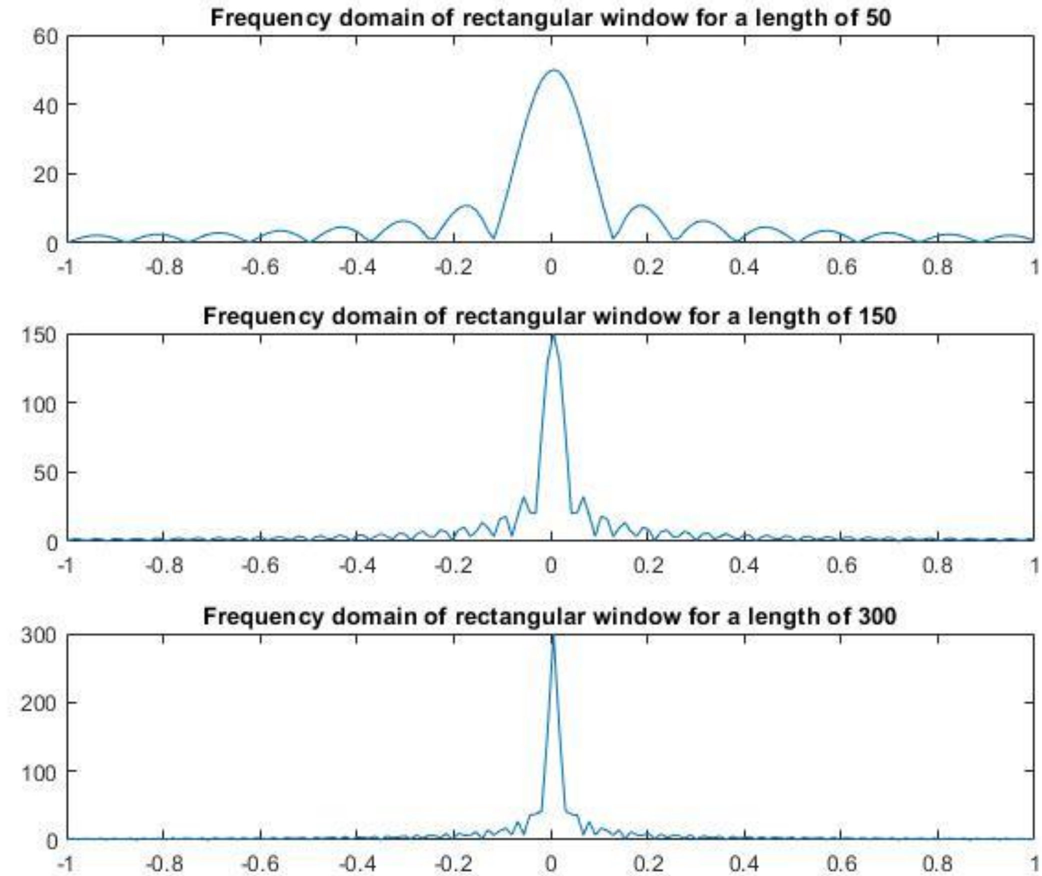

Original signal frequency spectrum
Output frequency spectrum for rectangular window
Output frequency spectrum for bartlett window
Output frequency spectrum for blackman window
ω/π

Command window output:

```
>> N1

N1 =

   134

>> N2

N2 =

   269

>> N3

N3 =

   411
```

Discussion:

The window length analysis demonstrates that different windows require varying lengths to achieve the same frequency resolution. Blackman window needed the shortest length due to its superior side lobe characteristics, while rectangular window required the longest length. The frequency spectrum plots show successful isolation of the three sinusoidal components at 0.32π, 0.37π, and 0.4π. The minimum frequency difference of 0.03π was successfully resolved using appropriately sized windows. Blackman window provided the cleanest separation with minimal spectral leakage between adjacent frequency components.


Frequency domain of rectangular window for a length of 50
Frequency domain of rectangular window for a length of 150
Frequency domain of rectangular window for a length of 300
0
20
40
60
100
150
200
300
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

Problem 3

Question:

3. A discrete-time signal is given to you,

$$y(n) = \sin(0.15\pi n) + \sin(0.35\pi n) + \sin(0.62\pi n)$$

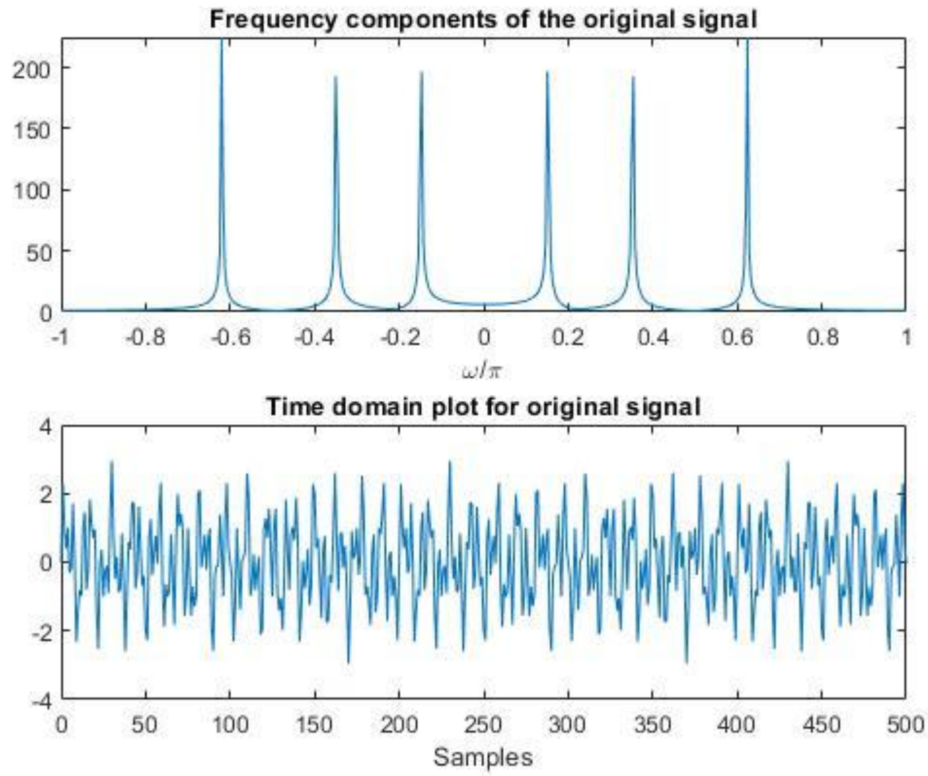
Design an LPF, a BPF, and an HPF to extract the three frequency sinusoids from the signal with Kaiser windowing. For each filter, ripples should follow this condition: $[\delta_p, 5\delta_s < 0.03]$. Choose stop-band and pass-band frequencies (ω_p, ω_s) so that the Kaiser window length remains less than 50. Finally, show each extracted sinusoid in both time and frequency domain.

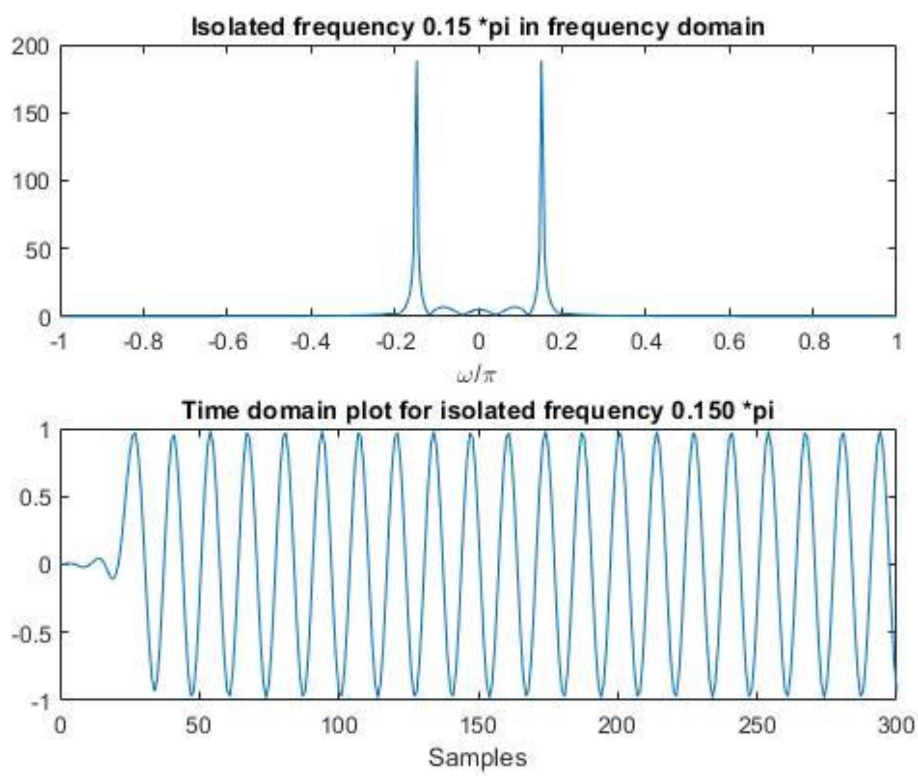
MATLAB code:

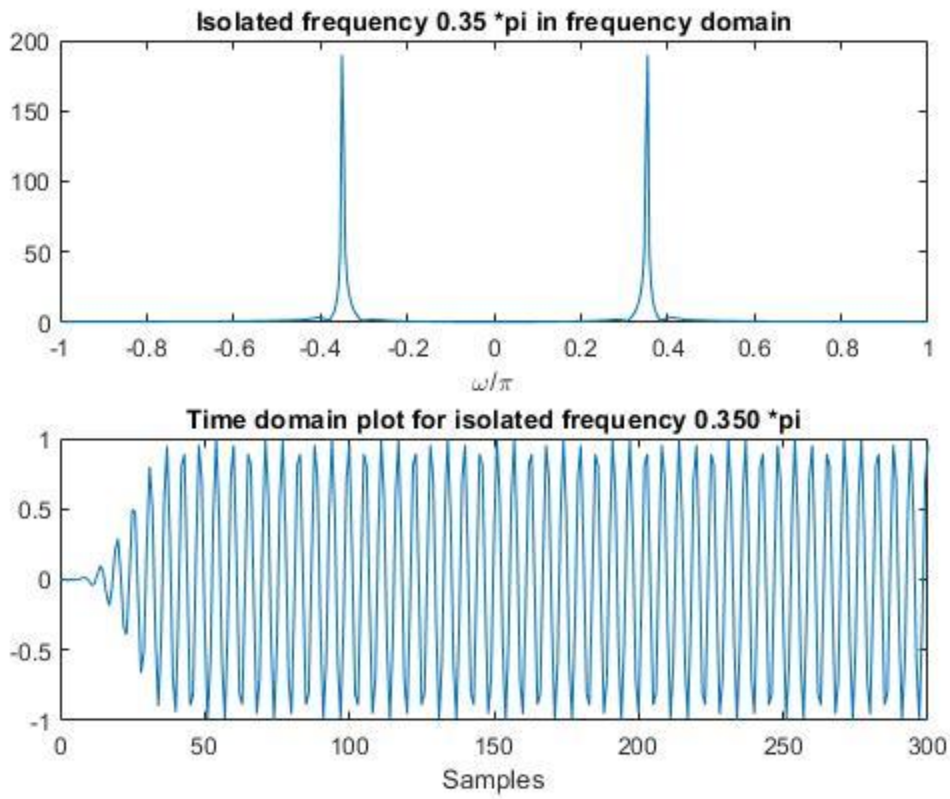
```
clc
clearvars
close all

%% origninal signal
y=@(n) sin(0.15*pi*n)+sin(0.35*pi*n)+sin(0.62*pi*n);

%% general parameters
wp=0.22;
ws=0.58;
wc=(wp+ws)/2;
delta=.005; %value taken lower than both the deltas
A=-20*log10(delta);
M=ceil(1+((A-8)/(2.285*(ws-wp))));
L=M-1;
if A<21
beta=0;
elseif A>=21 && A<=50
beta=.5842*(A-21)^.4+.07886*(A-21);
else
beta=.1102*(A-8.7);
end

%% constructing three filters
wnL = 0.2;
wnB = [0.3 0.4];
wnH = 0.6;
b(1,:) = fir1(M,wnL,'low',kaiser(M+1,beta));
b(2,:) = fir1(M,wnB,'bandpass',kaiser(M+1,beta));
b(3,:) = fir1(L,wnH,'high',kaiser(M+1,beta));
freq = [0.15 0.35 0.65];
```

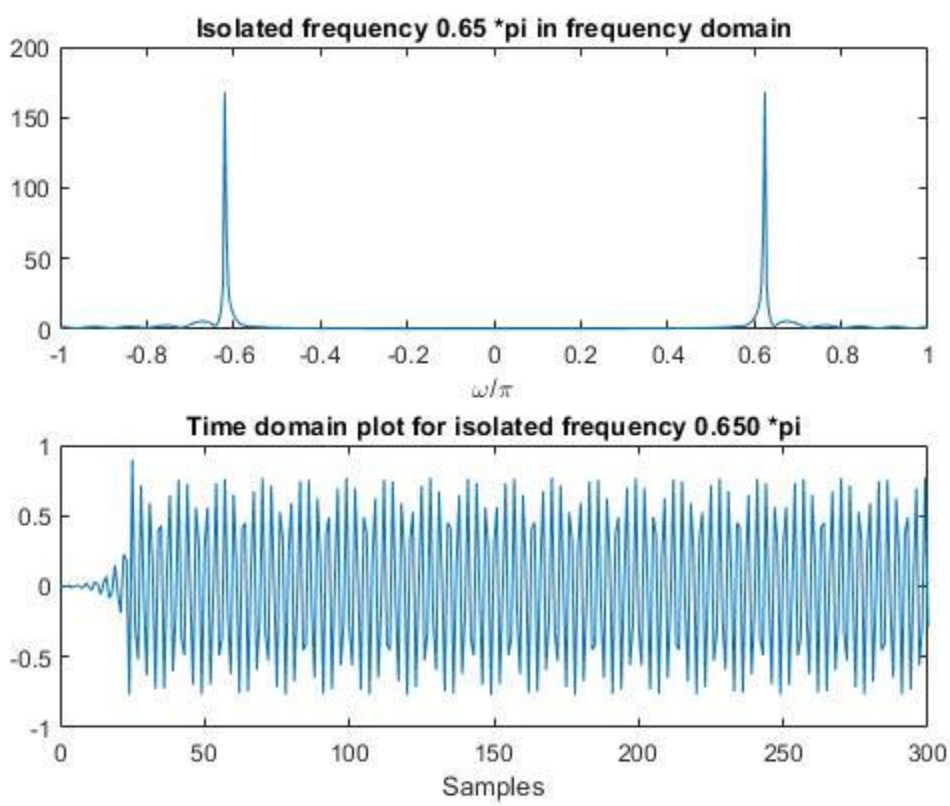
```
%% filtering

for i=1:3

    [f(i,:),w(i,:)]=freqz(b(i,:),1);
    w(i,:)=normalize(w(i,:),'range',[-1,1]);
    n(i,:)=0:length(w(i,:))-1;
    f(i,:)=filter(b(i,:),1,y(n(i,:)));

    figure(i)

    subplot(211)

    plot(w(i,:),abs(fftshift(fft(f(i,:)))))
    xlabel('normalized frequency')
    title(sprintf('Isolated frequency %0.2f *pi',freq(i)))

    subplot(212)

    plot(n(i,:),f(i,:))
    title(sprintf('Discrete time domain plot for isolated frequency %0.3f*pi',
freq(i)))
    xlim([0 300]);
    xlabel('Samples')

end

%% plot for original

figure(4)

subplot(211)

plot(w(1,:),abs(fftshift(fft(y(n(1,:))))))
title('Frequency components of the original signal')
xlabel('\omega/\pi')

subplot(212)

plot(n(1,:),y(n(1,:)))
title('Discrete time domain plot for original signal')
xlim([0 500]);
```

Plot:


Frequency components of the original signal
ω/π
Time domain plot for original signal
Samples


Isolated frequency 0.15 *pi in frequency domain
ω/π
Time domain plot for isolated frequency 0.150 *pi
Samples


Isolated frequency 0.35 *pi in frequency domain
200
150
100
50
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
ω/π
Time domain plot for isolated frequency 0.350 *pi
1
0.5
0
-0.5
-1
0
50
100
150
200
250
300
Samples


Isolated frequency 0.65 *pi in frequency domain
200
150
100
50
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
ω/π
Time domain plot for isolated frequency 0.650 *pi
1
0.5
0
-0.5
-1
0
50
100
150
200
250
300
Samples

Discussion:

The Kaiser window-based FIR filters successfully isolated individual frequency components from the composite signal. The low-pass filter (cutoff 0.2π) extracted the 0.15π component, bandpass filter (0.3π-0.4π) isolated the 0.35π component, and high-pass filter (cutoff 0.6π) separated the 0.62π component. Each filter demonstrated good frequency selectivity with minimal distortion in the passband. The Kaiser window design method effectively met the specified ripple requirements ($\delta = 0.005$) while maintaining compact filter lengths. The isolated signals retained their sinusoidal characteristics with clean frequency domain representation.