# Digital Signal Processing Lab

## Expt. No. 2
## Designing of Low Pass Filters by Windowing Method

By Pranit Dalal
Roll no. 16EC10016
Group 22 (Tuesday)

# AIM:

(a)Design FIR filters for various orders and cut-off frequencies.
(b)Check attenuation of pass band frequencies and compare it with stop band frequencies
(c)Response of FIR filters when signals are contaminated by noise.

## (a) Design FIR filters for various orders and cut-off frequencies:

(i)Theory:

In digital signal processing, ideal filters having sharp cut-off are not realizable in practice since their impulse responses extend up to infinity. In order to realize filters practically, we need to truncate the impulse response after a few samples. But this abrupt transition introduces many undesired features in the frequency domain (e.g. a large amount of power lies in the side-lobes). By windowing method, the truncated filter is modified to satisfy certain frequency domain requirements and to perform its task as well.
A few window functions we implemented are:

(1) Rectangular window : $w(n) = 1 \quad ; \quad n=0,1,\ldots,N-1$
$\quad = 0 \quad$ otherwise

(2) Triangular window : $w(n) = 1 - 2*[n - (N-1)/2]\ /(N-1) \quad ; \ n=0,1,\ldots,N-1$
$\quad = 0 \quad$ otherwise

(3) Hanning window: : $w(n) = 0.5 - 0.5 * \cos[2\pi n/(N-1)] \quad ; n=0,1,\ldots N-1$
$\quad = 0 \quad$ otherwise

(4) Hamming window : $w(n) = 0.54 - 0.46 * \cos[2\pi n/(N-1)] \quad ; n=0,1,\ldots N-1$
$\quad = 0 \quad$ otherwise

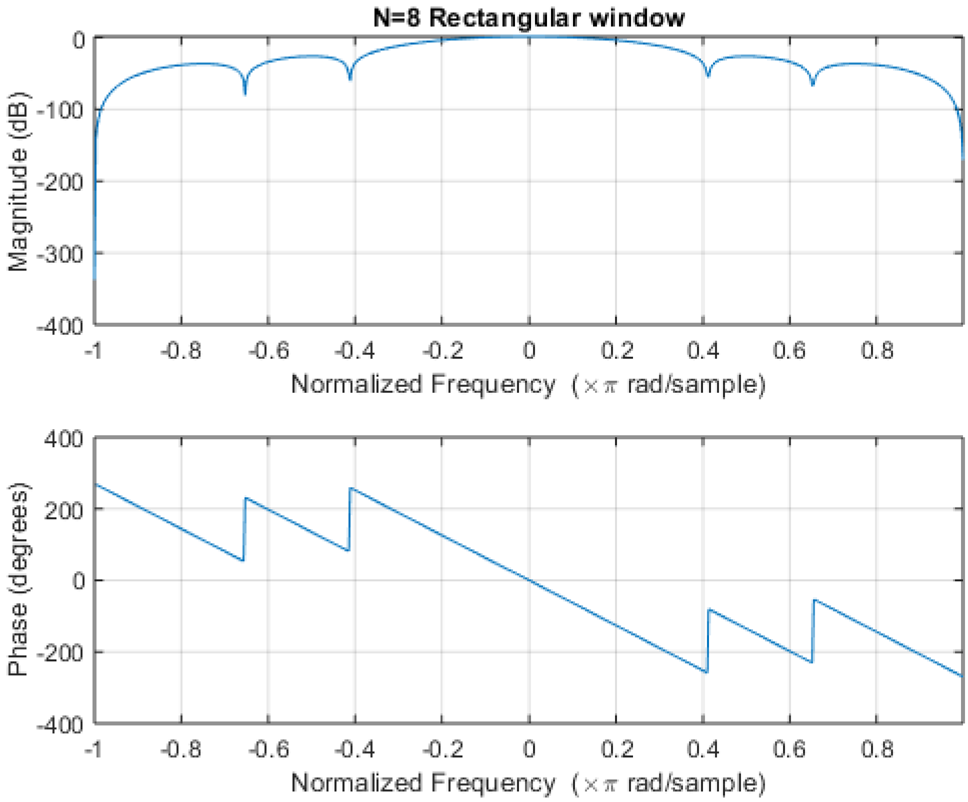
(5) Blackmnan window : $w(n) = 0.42 - 0.5 * \cos[2\pi n/(N-1)] + 0.08 * \cos[4\pi n/(N-1)] \quad ; n=0,1,\ldots N-1$
$\quad =0 \quad$ otherwise

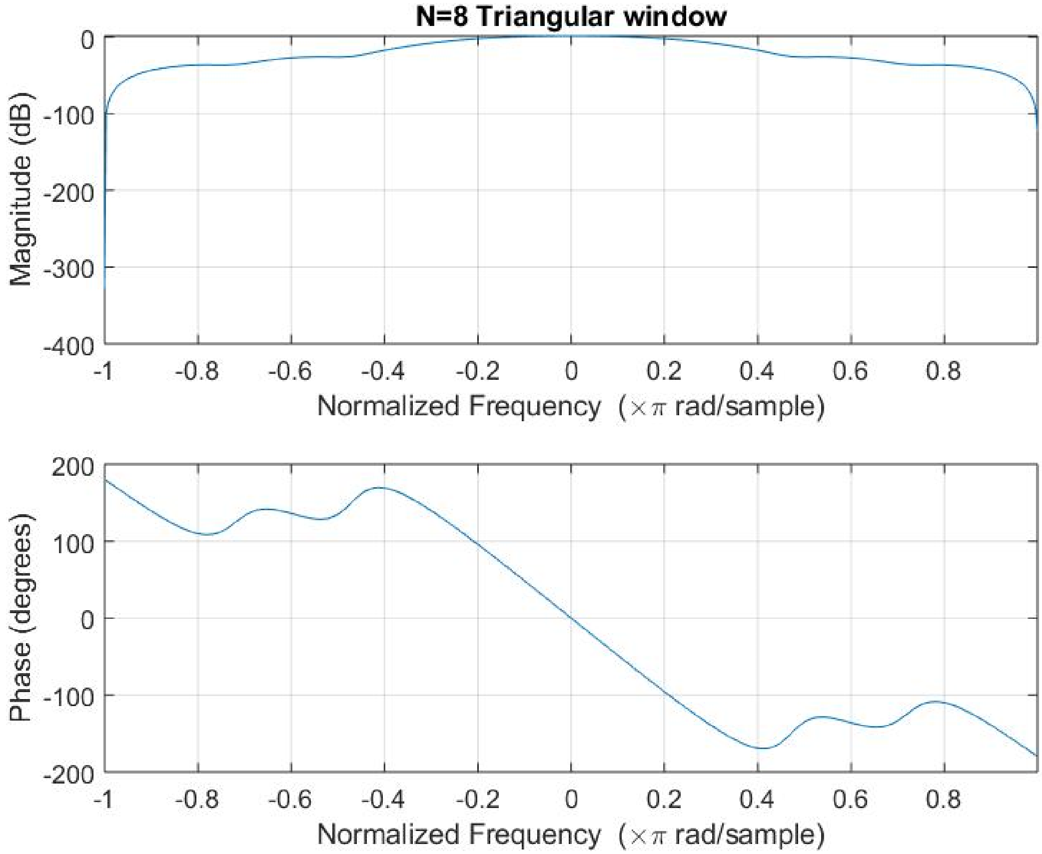
(ii) Code:

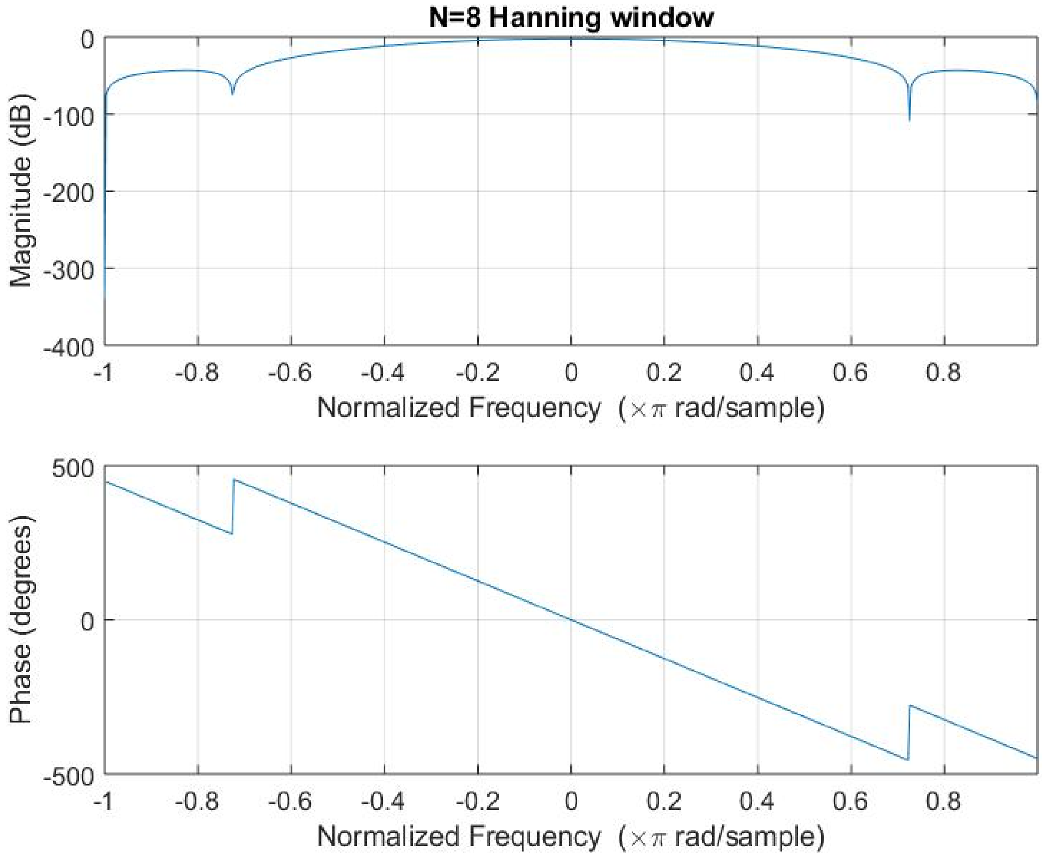
Frequency response of different windows

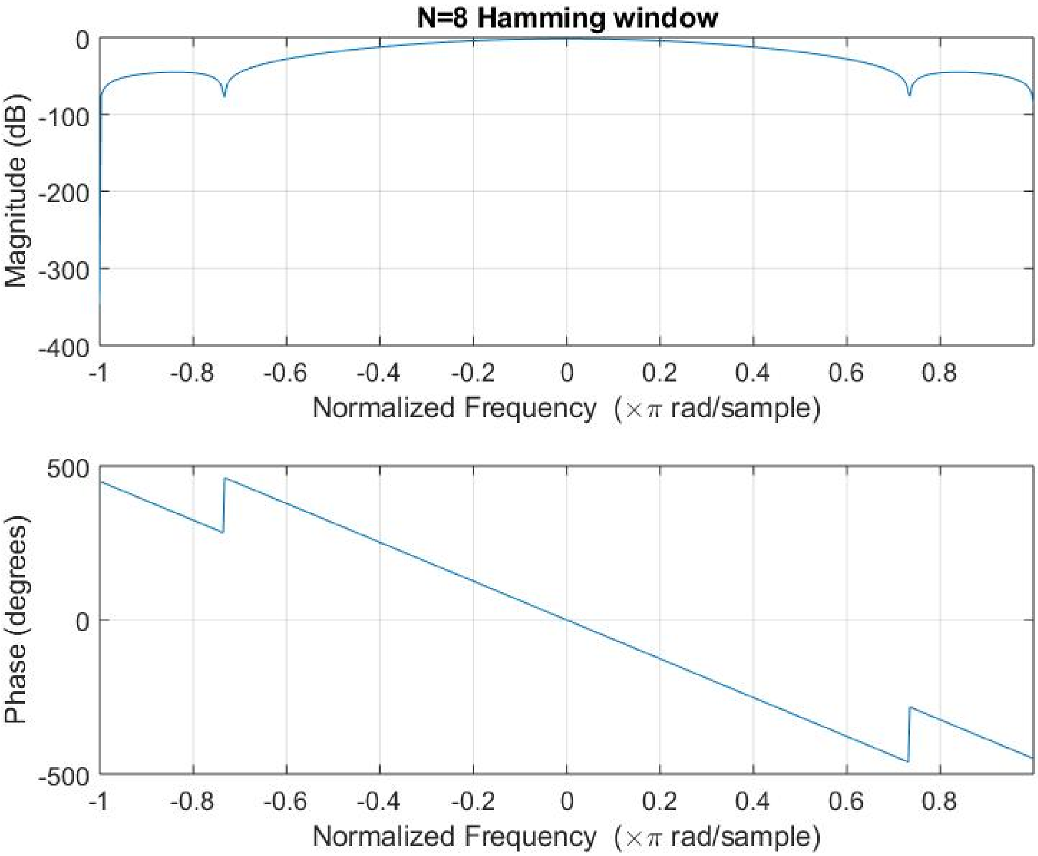
```
N =8;
wc = pi/4;           %cut-off frequency
k = (N-1)/2;
n = 0:1:N-1;
hd = (sin(wc*(n-k)))./(pi*(n-k));        %impulse response
w1 = (n>=0)-(n>=N);                                   %rectangular window
%w2 = ((n>=0)-(n>=N)).*(1-2*(n-(N-1)/2)/(N-1));        %triangular window
%w3 = ((n>=0)-(n>=N)).*(0.5-0.5*cos((2*pi*n)/(N-1)));  %hanning window
%w4 = ((n>=0)-(n>=N)).*(0.54-0.46*cos((2*pi*n)/(N-1)));%hamming window
%w5 = ((n>=0)-(n>=N)).*(0.42-0.5*cos((2*pi*n)/(N-1))+0.08*cos((4*pi*n)/(N-1)));%blackman wind

h = hd.*w1;
c = -pi:0.01:pi;
[h1,w] = freqz(h,1,c);
h2 = abs(h1);
figure, freqz(h,1,c);
title('N=8 Rectangular window');
```

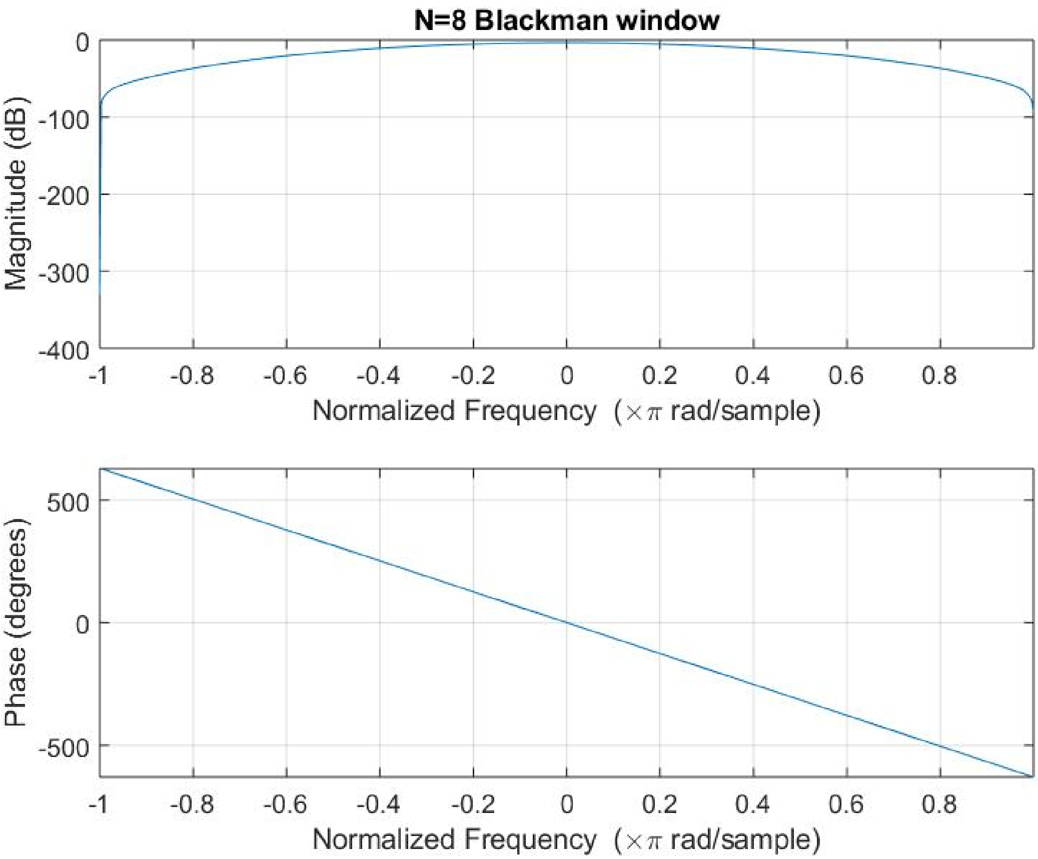
# For N = 8;



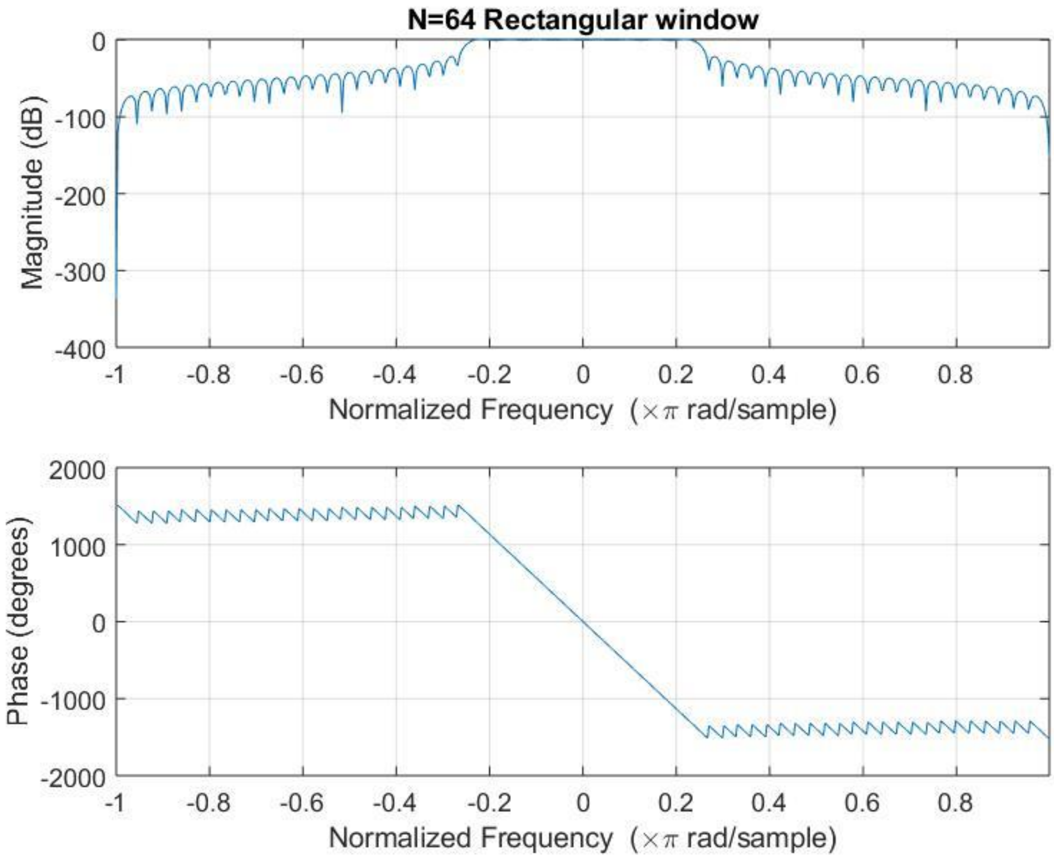

N=8 Triangular window
Magnitude (dB)
Normalized Frequency (×π rad/sample)
Phase (degrees)
Normalized Frequency (×π rad/sample)

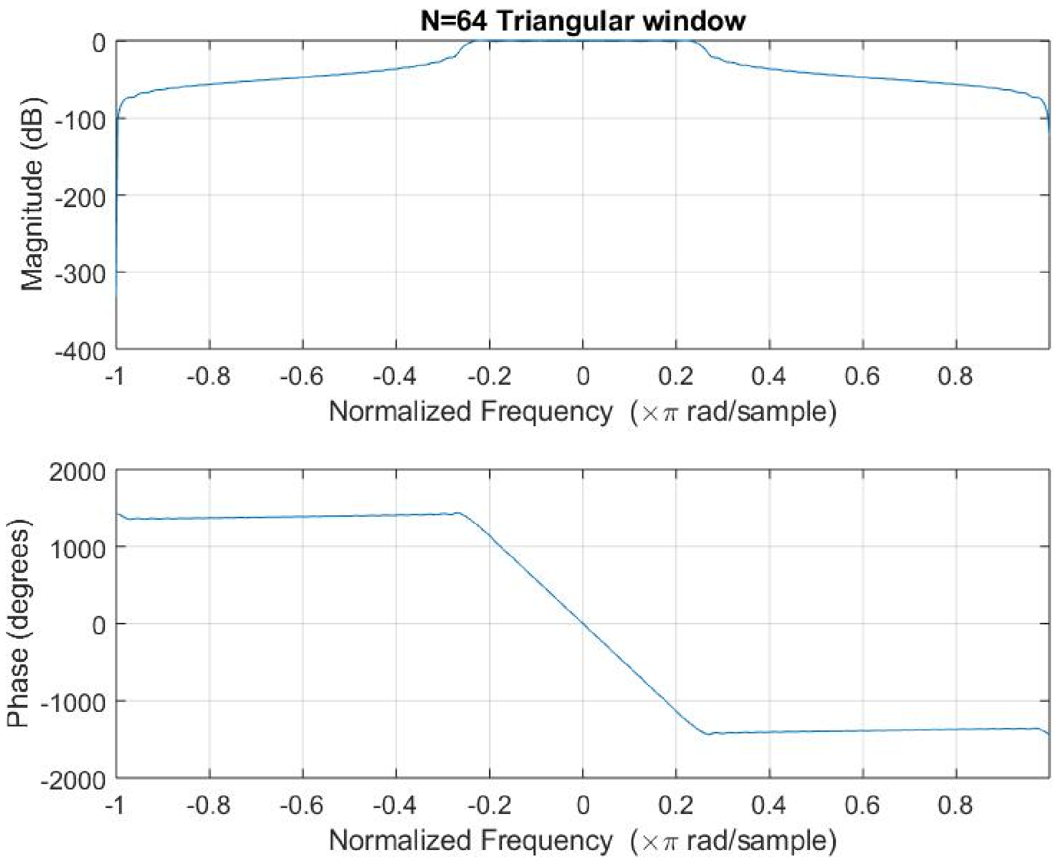

N=8 Hanning window
Magnitude (dB)
Normalized Frequency (×π rad/sample)
Phase (degrees)
Normalized Frequency (×π rad/sample)

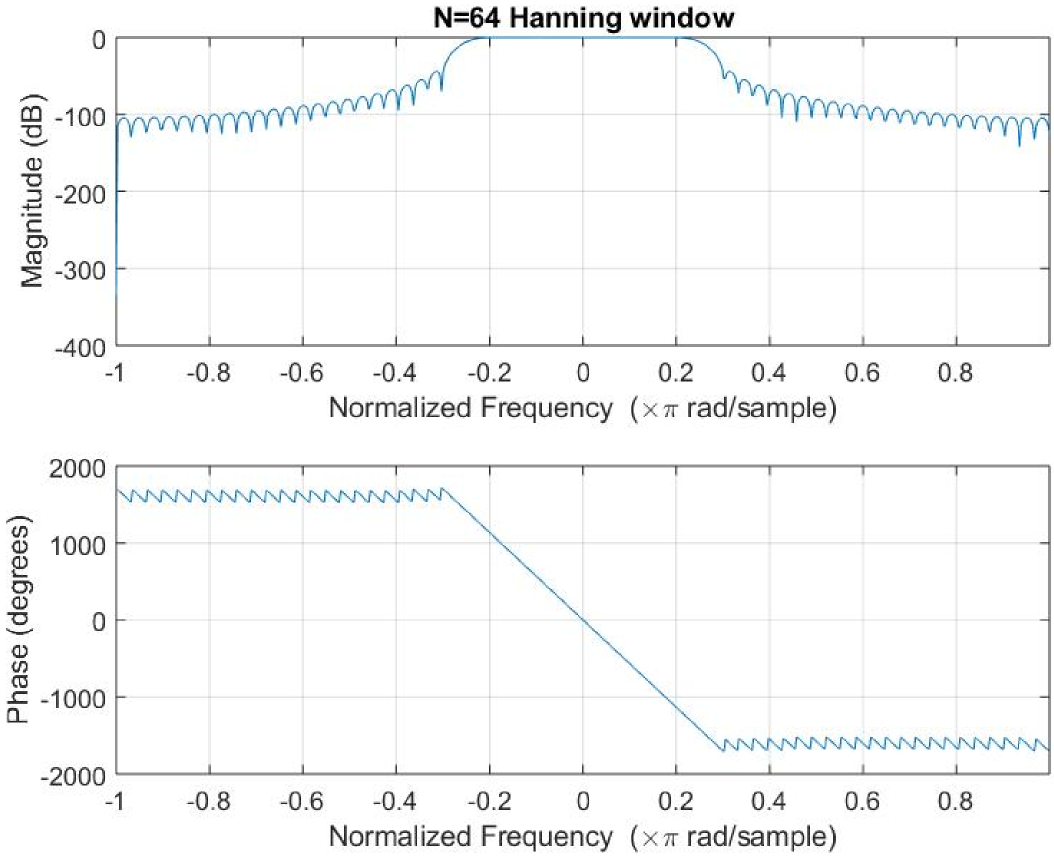

N=8 Hamming window
Magnitude (dB)
0
-100
-200
-300
-400
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)
Phase (degrees)
500
0
-500
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)


N=8 Blackman window
Magnitude (dB)
0
-100
-200
-300
-400
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)
Phase (degrees)
500
0
-500
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)

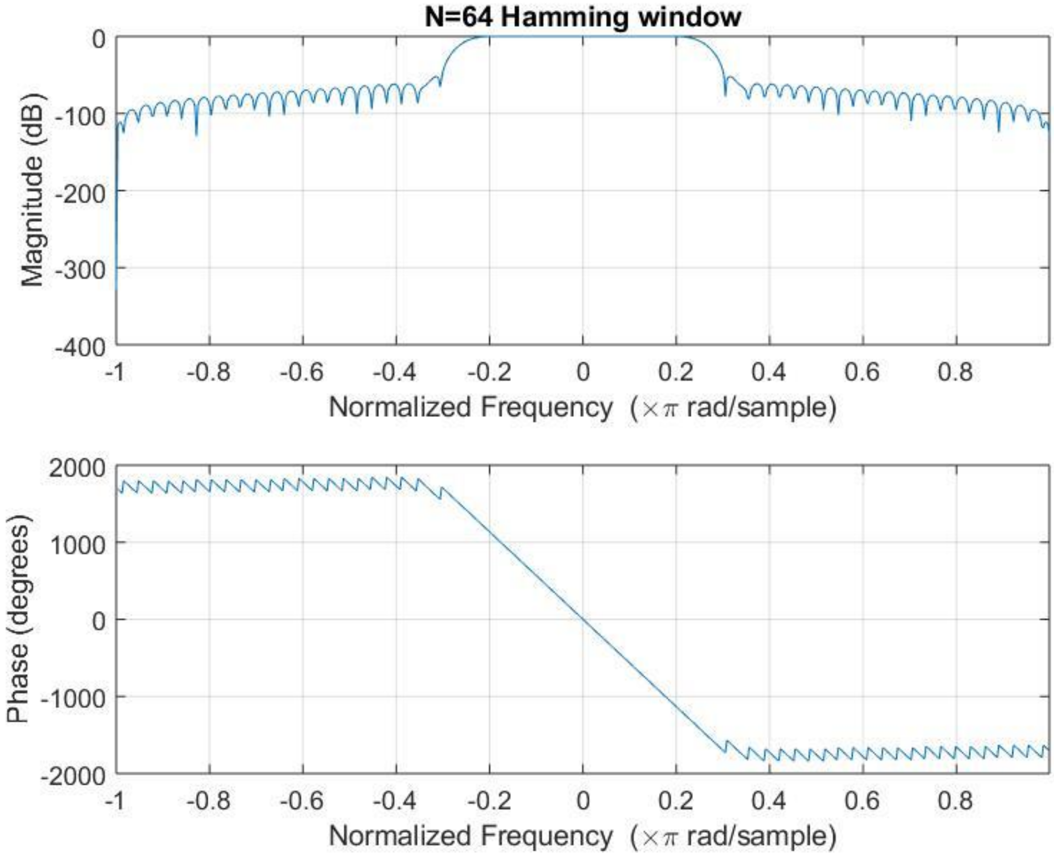
For N = 64:


N=64 Rectangular window
Magnitude (dB)
0
-100
-200
-300
-400
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)
Phase (degrees)
2000
1000
0
-1000
-2000
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)

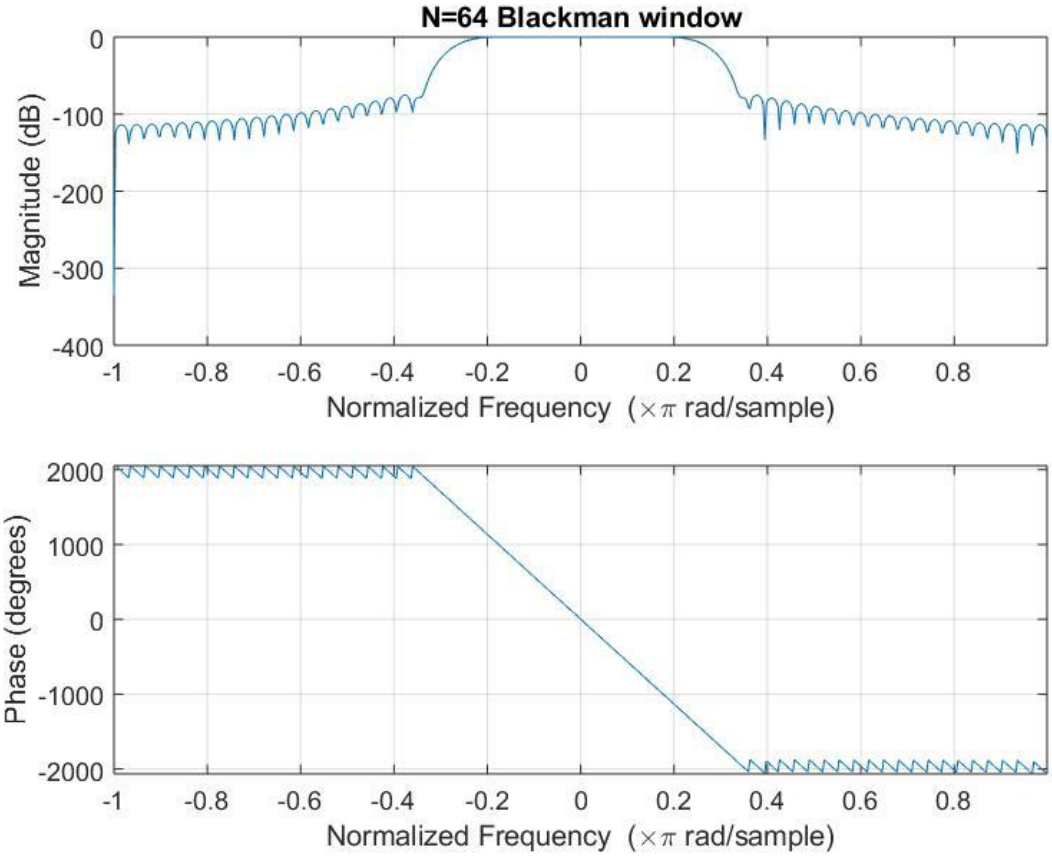

N=64 Triangular window
Magnitude (dB)
0
-100
-200
-300
-400
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)
Phase (degrees)
2000
1000
0
-1000
-2000
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)



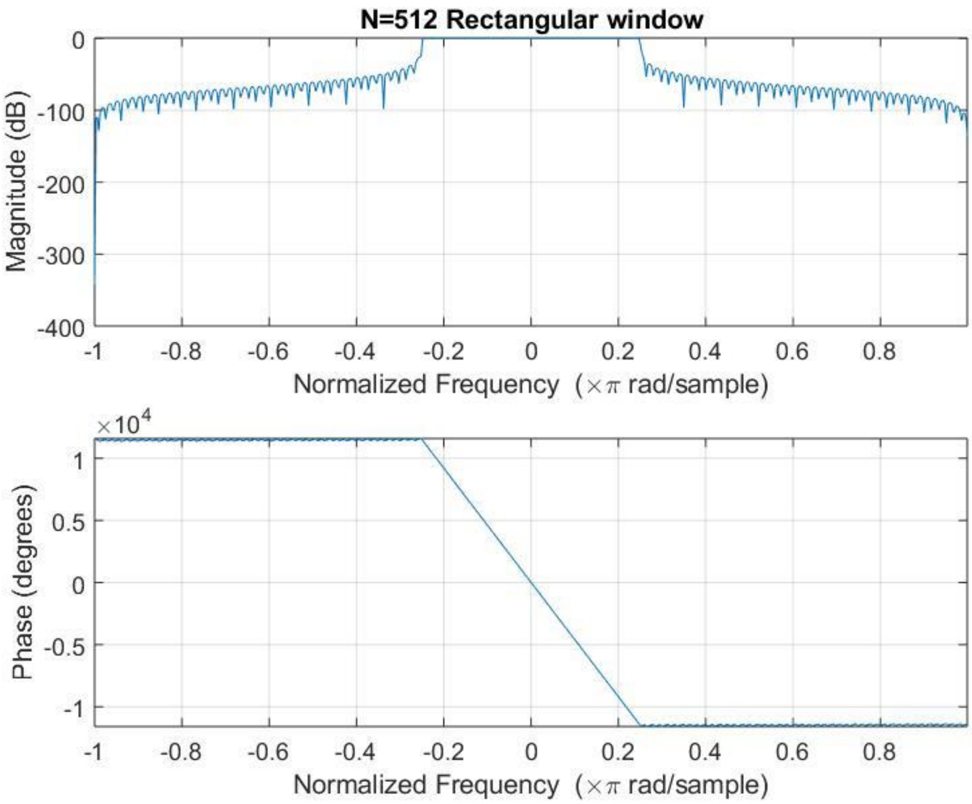
N=64 Hanning window
Magnitude (dB)
Phase (degrees)
Normalized Frequency (×π rad/sample)
Normalized Frequency (×π rad/sample)

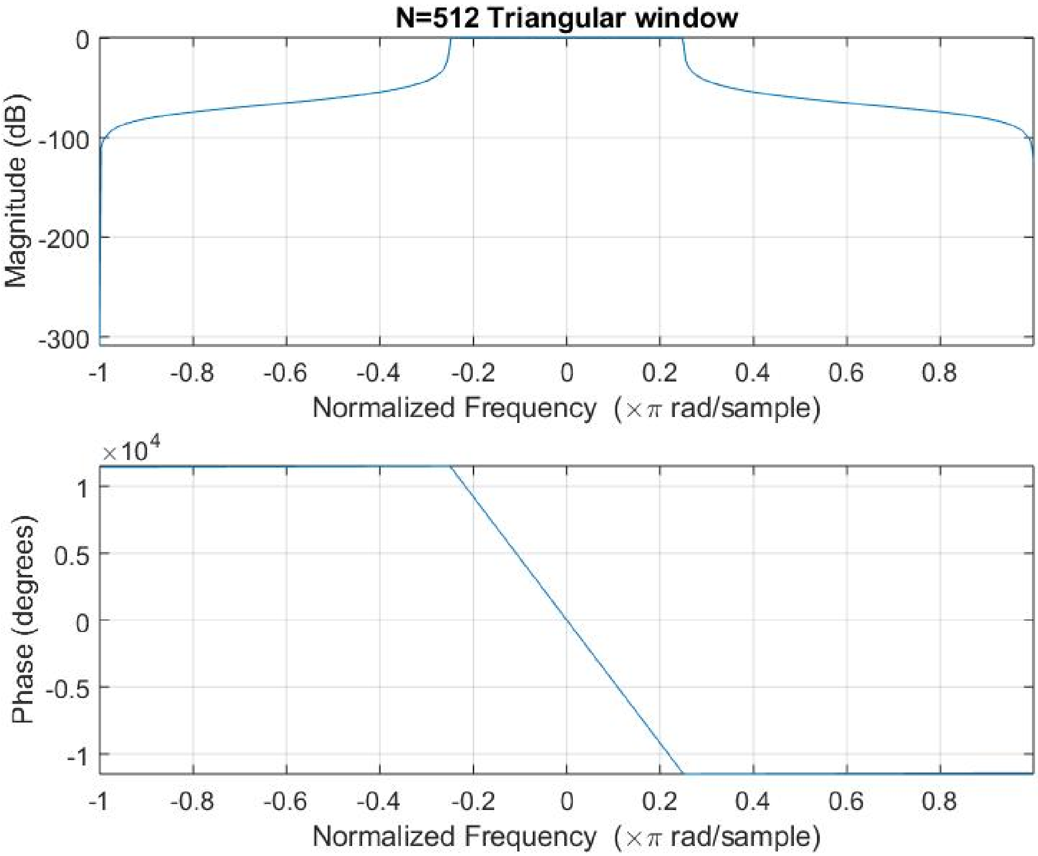



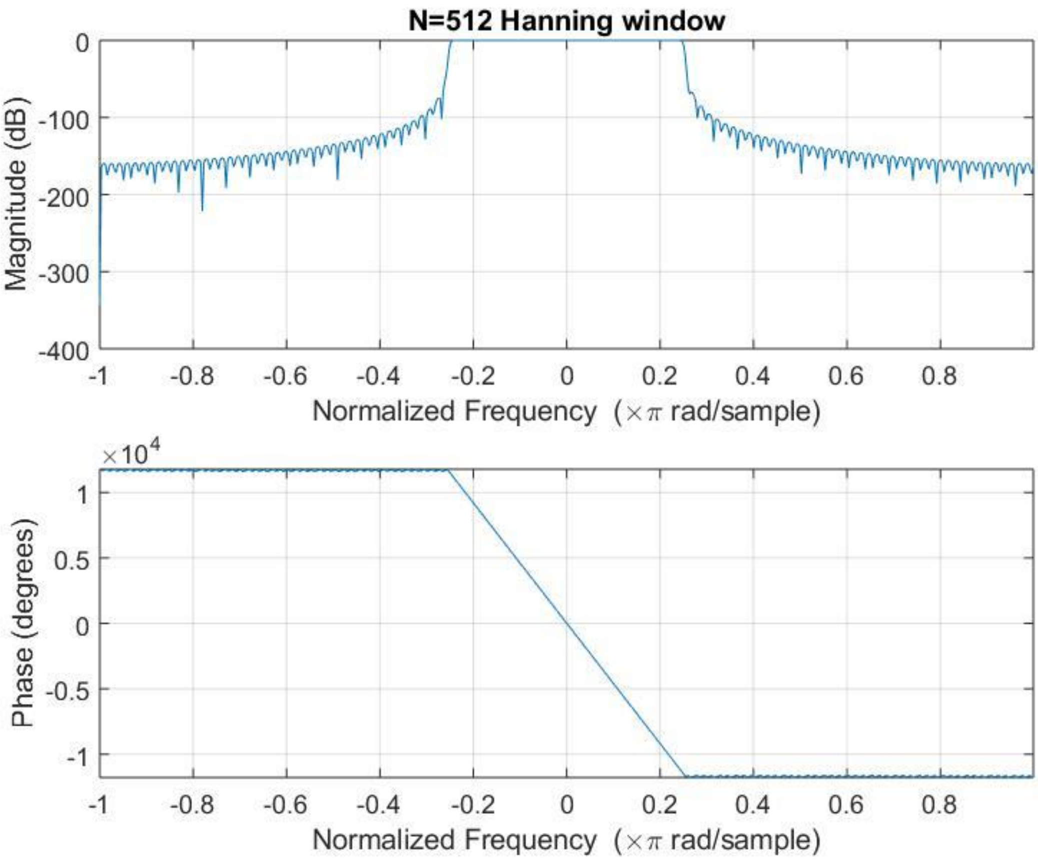
N=64 Hamming window
Magnitude (dB)
Phase (degrees)
Normalized Frequency (×π rad/sample)
Normalized Frequency (×π rad/sample)

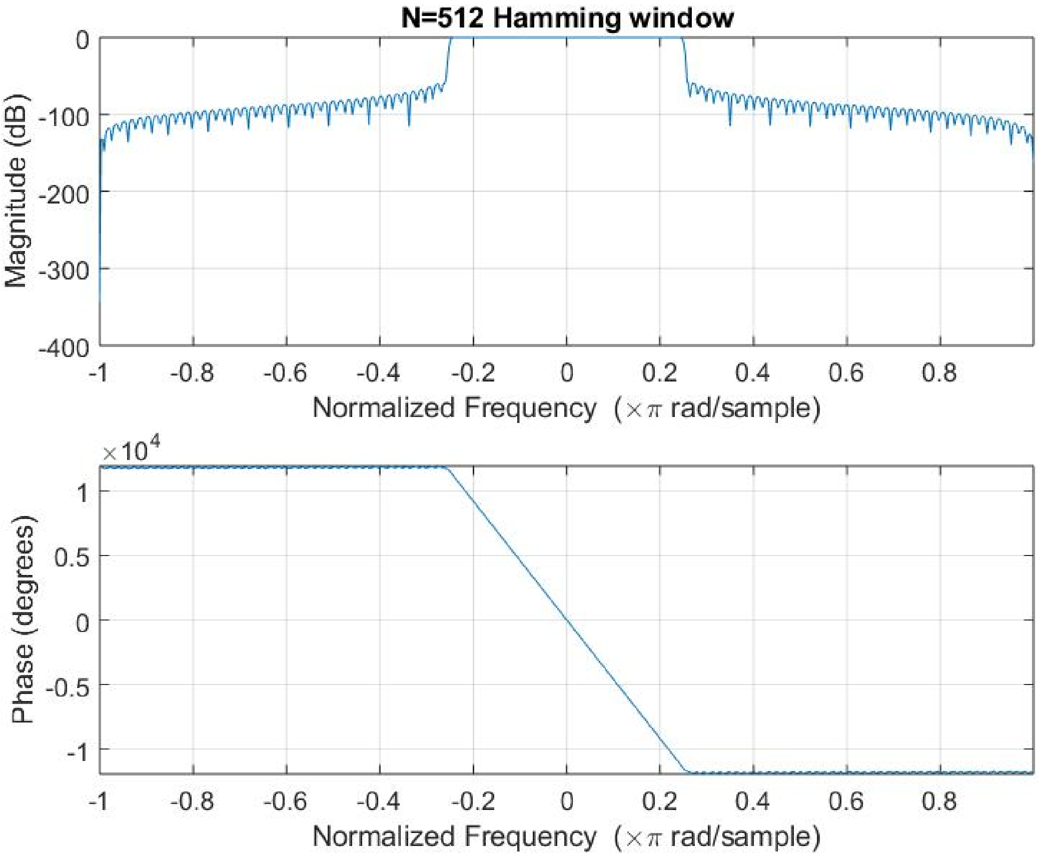

N=64 Blackman window
Magnitude (dB)
Normalized Frequency (×π rad/sample)
Phase (degrees)
Normalized Frequency (×π rad/sample)

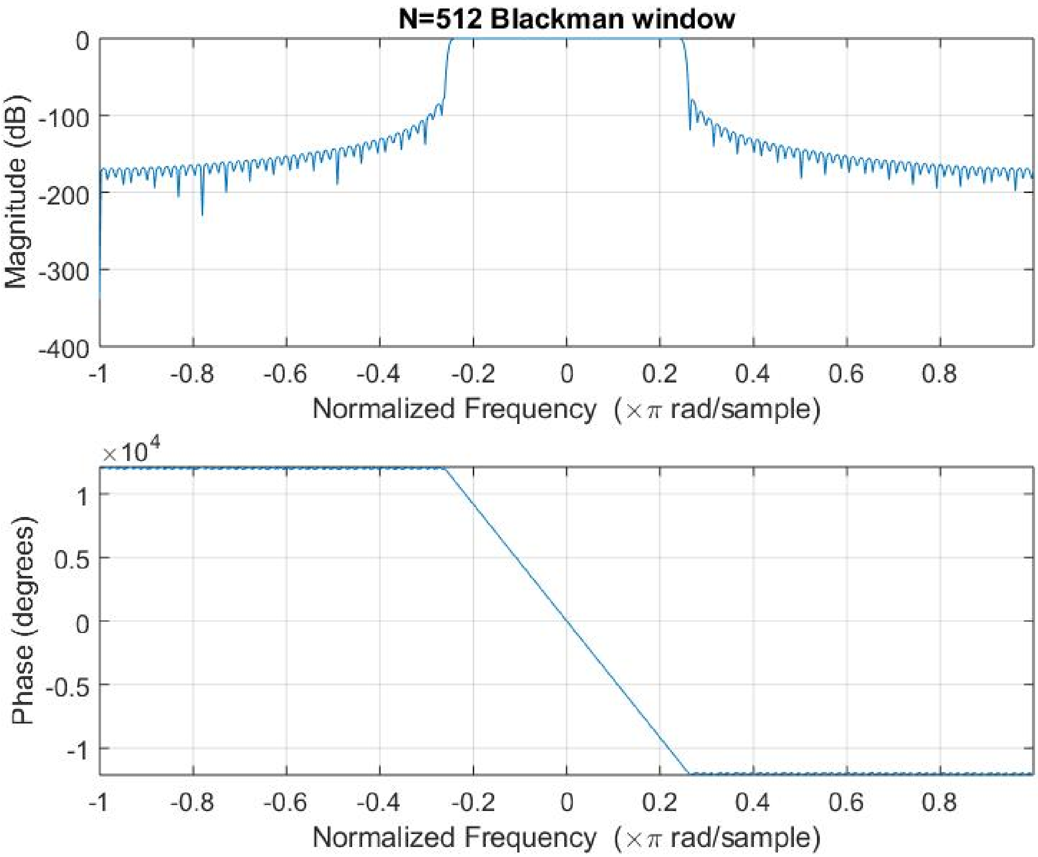
For N = 512:


N=512 Rectangular window
Magnitude (dB)
Normalized Frequency (×π rad/sample)
×10^4
Phase (degrees)
Normalized Frequency (×π rad/sample)


N=512 Triangular window
Magnitude (dB)
0
-100
-200
-300
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)
×10^4
Phase (degrees)
1
0.5
0
-0.5
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)


N=512 Hanning window
Magnitude (dB)
0
-100
-200
-300
-400
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)
×10^4
Phase (degrees)
1
0.5
0
-0.5
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)


N=512 Hamming window
Magnitude (dB)
0
-100
-200
-300
-400
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)
×10^4
Phase (degrees)
1
0.5
0
-0.5
-1
Normalized Frequency (×π rad/sample)


N=512 Blackman window
Magnitude (dB)
0
-100
-200
-300
-400
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)
×10^4
Phase (degrees)
1
0.5
0
-0.5
-1
Normalized Frequency (×π rad/sample)

(iii) Results:

| N | Rectangular Window | | |
|---|---|---|---|
| | Transition width $\pi$ rad/sample | Peak of first lobe (dB) | Maximum Stopband Attenuation(dB) |
| **8** | 0.2217 | -26.05 | -55.02 |
| **64** | 0.0460 | -21.75 | -39.55 |
| **512** | 0.0145 | -35.10 | -53.55 |

| N | Triangular Window | | |
|---|---|---|---|
| | Transition width $\pi$ rad/sample | Peak of first lobe (dB) | Maximum Stopband Attenuation(dB) |
| **8** | 0.2323 | -25.86 | -26.04 |
| **64** | 0.0268 | -21.53 | -21.60 |
| **512** | 0.0402 | -43.41 | -43.41 |

| N | Hanning Window | | |
|---|---|---|---|
| | Transition width $\pi$ rad/sample | Peak of first lobe (dB) | Maximum Stopband Attenuation(dB) |
| **8** | 0.6300 | -43.28 | -109.1 |
| **64** | 0.0644 | -44.02 | -54.93 |
| **512** | 0.0140 | -66.64 | -69.7 |

| N | Hamming Window | | |
|---|---|---|---|
| | Transition width $\pi$ rad/sample | Peak of first lobe (dB) | Maximum Stopband Attenuation(dB) |
| **8** | 0.6096 | -45.06 | -71.76 |
| **64** | 0.0664 | -52.07 | -77.63 |
| **512** | 0.0148 | -59.46 | -77.52 |

| N | Blackman Window | | |
|---|---|---|---|
| | Transition width $\pi$ rad/sample | Peak of first lobe (dB) | Maximum Stopband Attenuation(dB) |
| **8** | 0.7010 | -62.61 | -62.61 |
| **64** | 0.1240 | -75.44 | -90.96 |
| **512** | 0.0158 | -79.77 | -119.5 |

## (b)Check attenuation of pass band frequencies and compare it with stop band frequencies

### (i)Theory:

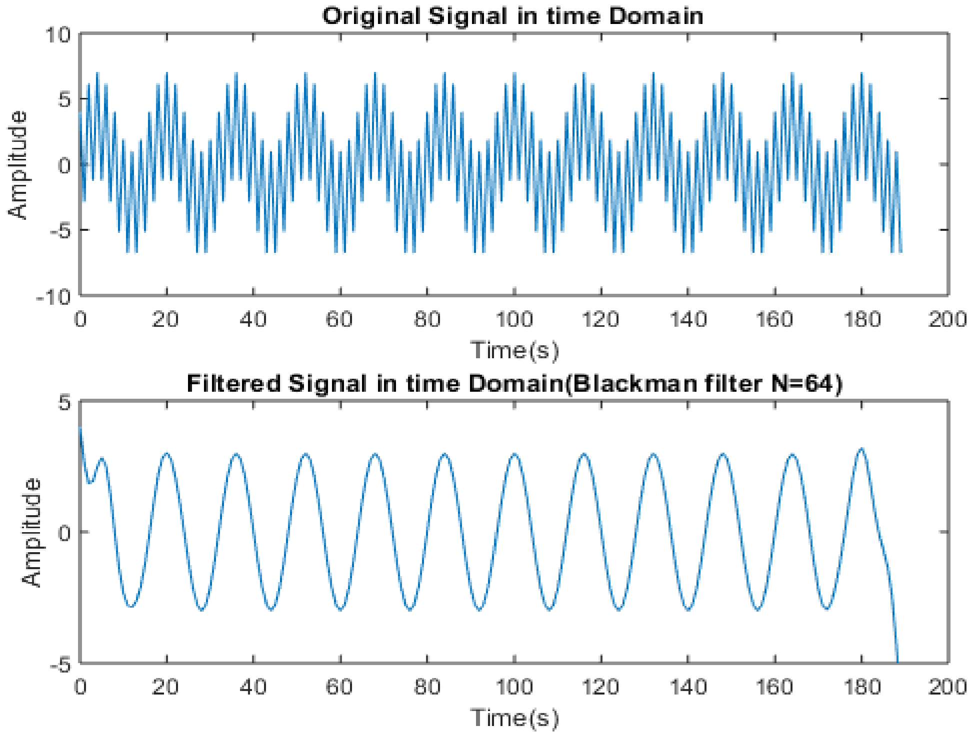
When a signal containing 2 frequencies, one in pass-band and another in the stop-band, we observe the attenuation of the frequency component in the stop-band.
In Fourier Domain the stop-band frequencies don’t pass and hence the amplitude is nearly equal to zero. Certain abnormalities are observed at the start and end in time domain when passed through the filter.

### (ii)Code:

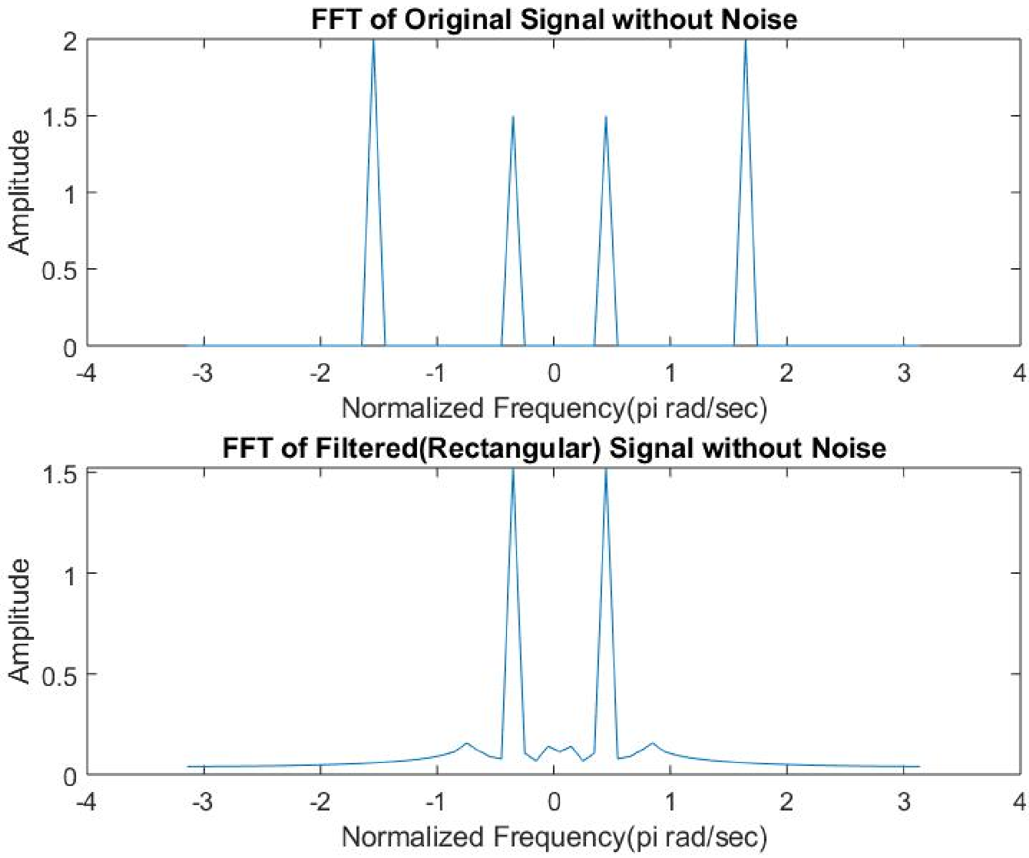
```
N = 64;
wc = pi/3;
k = (N-1)/2;
n = 0:1:N-1;
hd = (sin(wc*(n-k)))./(pi*(n-k));
w5 = ((n>=0)-(n>=N)).*(0.42-0.5*cos((2*pi*n)/(N-1))+0.08*cos((4*pi*n)/(N-1)));
h = hd.*w5;  %For Blackman Window
c = -pi:0.01:pi;
[h1,w] = freqz(h,1,c);
h1 = abs(h1);

t = 0:1:3*(N-1);
x = 3*sin(pi/8*t)+4*cos(pi*t);
grid on

y = filtfilt(h,1,x);
subplot(211)
plot(t,x)
title('Original Signal in time Domain');
xlabel('Time(s)');
ylabel('Amplitude');

subplot(212);
plot(t,y)
ylim([-5 5]);
title('Filtered Signal in time Domain(Blackman filter N=64)');
xlabel('Time(s)');
ylabel('Amplitude');
```

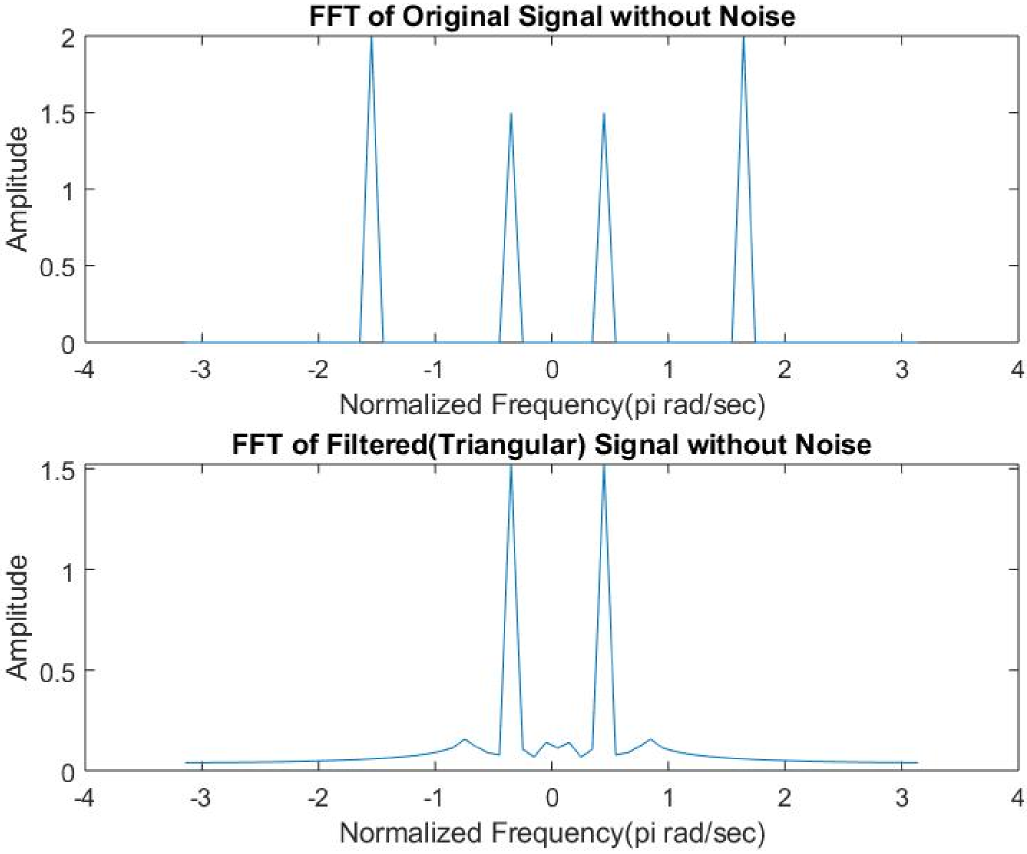
(iii)Results:

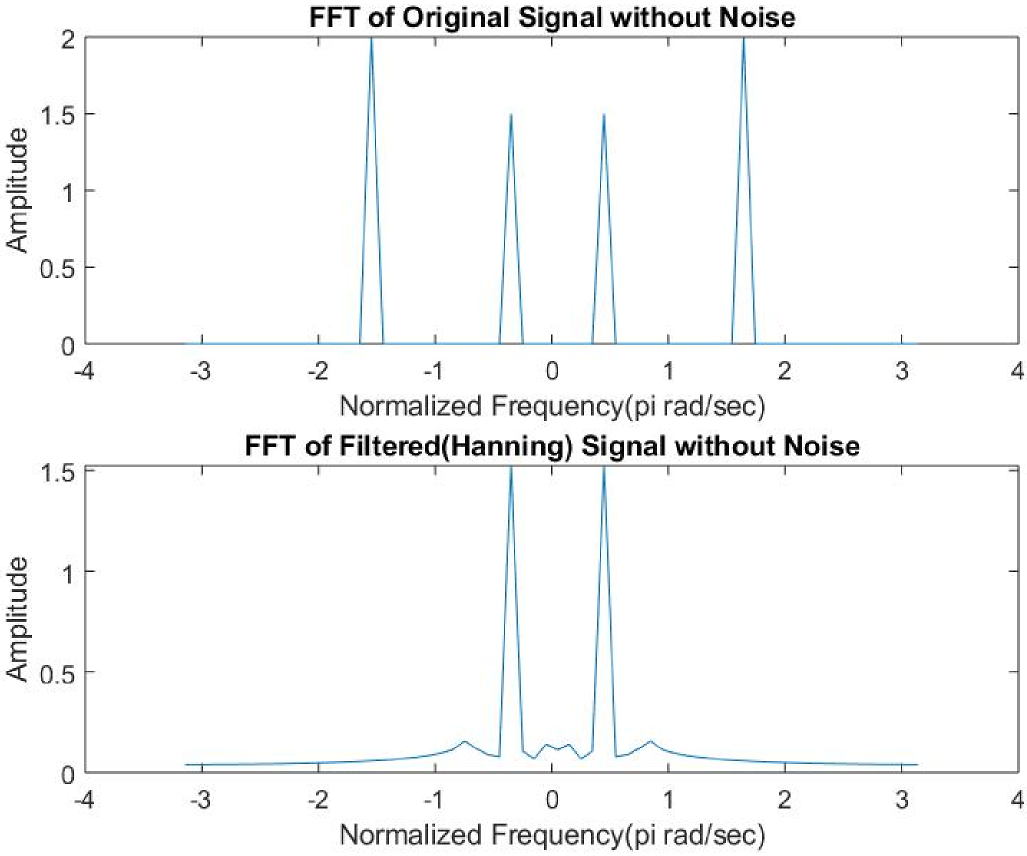
Time Domain response (Blackman Filter)

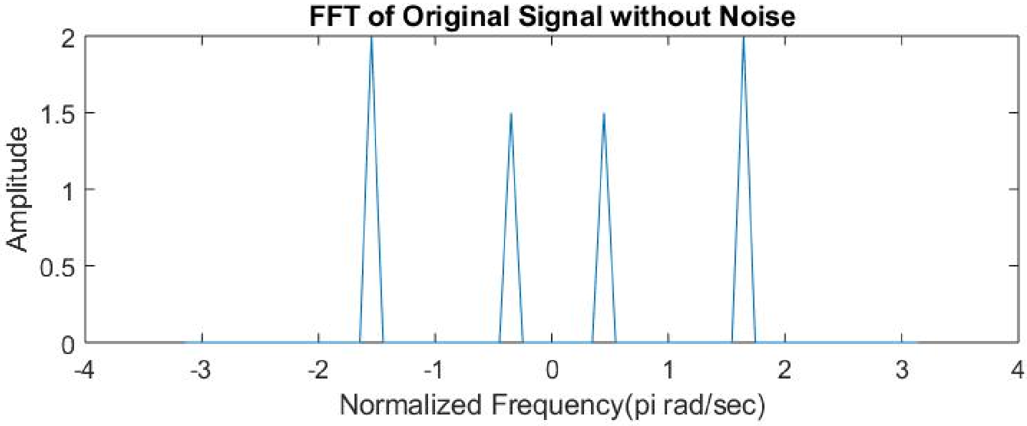


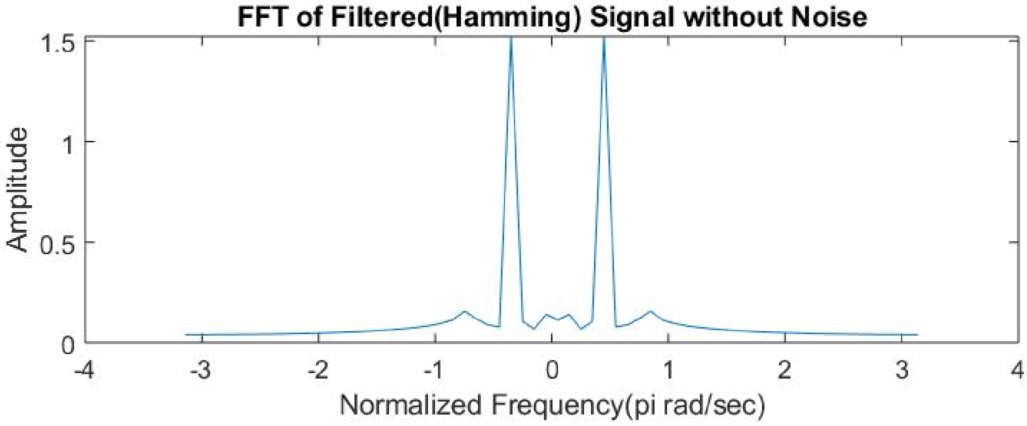

Frequency Domain(fft) for various filters at N = 64

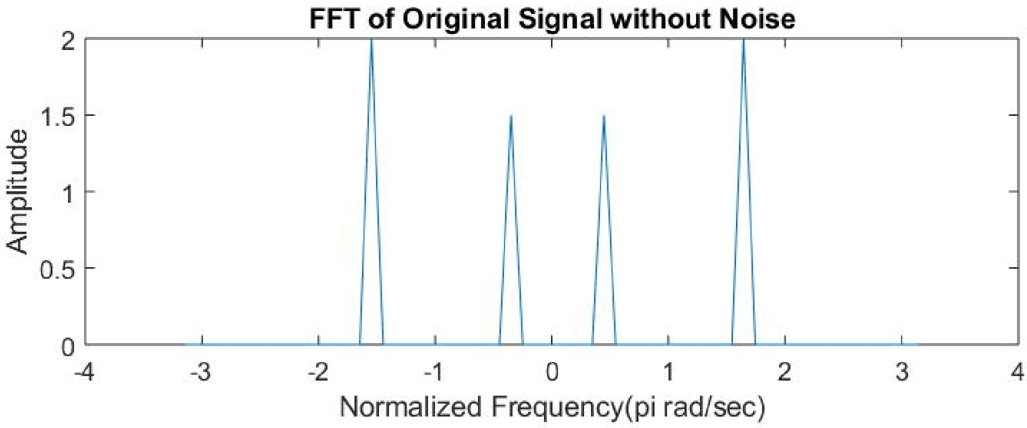


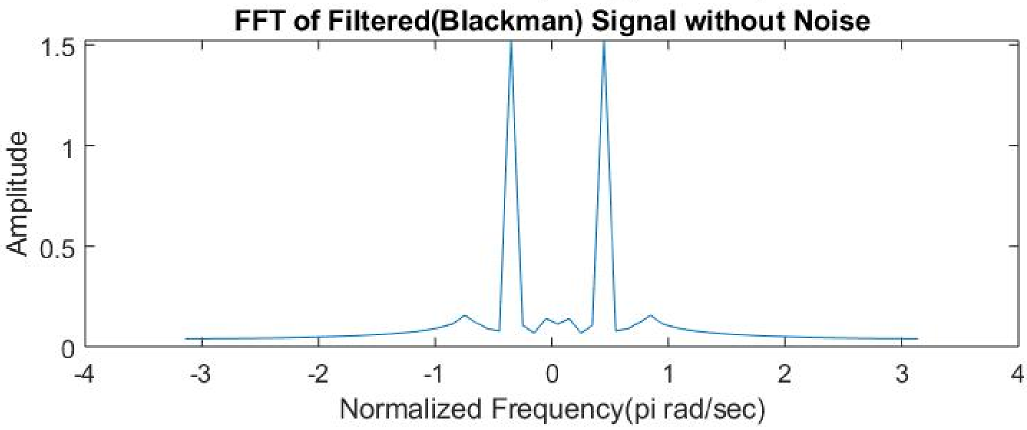

FFT of Original Signal without Noise
Amplitude
Normalized Frequency(pi rad/sec)
FFT of Filtered(Triangular) Signal without Noise
Amplitude
Normalized Frequency(pi rad/sec)


FFT of Original Signal without Noise
Amplitude
Normalized Frequency(pi rad/sec)
FFT of Filtered(Hanning) Signal without Noise
Amplitude
Normalized Frequency(pi rad/sec)


FFT of Original Signal without Noise
Amplitude
Normalized Frequency(pi rad/sec)


FFT of Filtered(Hamming) Signal without Noise
Amplitude
Normalized Frequency(pi rad/sec)


FFT of Original Signal without Noise
Amplitude
Normalized Frequency(pi rad/sec)


FFT of Filtered(Blackman) Signal without Noise
Amplitude
Normalized Frequency(pi rad/sec)

# (c)Response of FIR filters when signals are contaminated by noise.

## (i)Theory:

It is natural for noise to be added up in the signal due to channel or various other reasons. The filter’s capability to reduce the noise is a great parameter for the quality of the filter. Noise is added using randn() function which generates randomly distributed noise.

## (ii)Code:

a) For FFT with noise added:

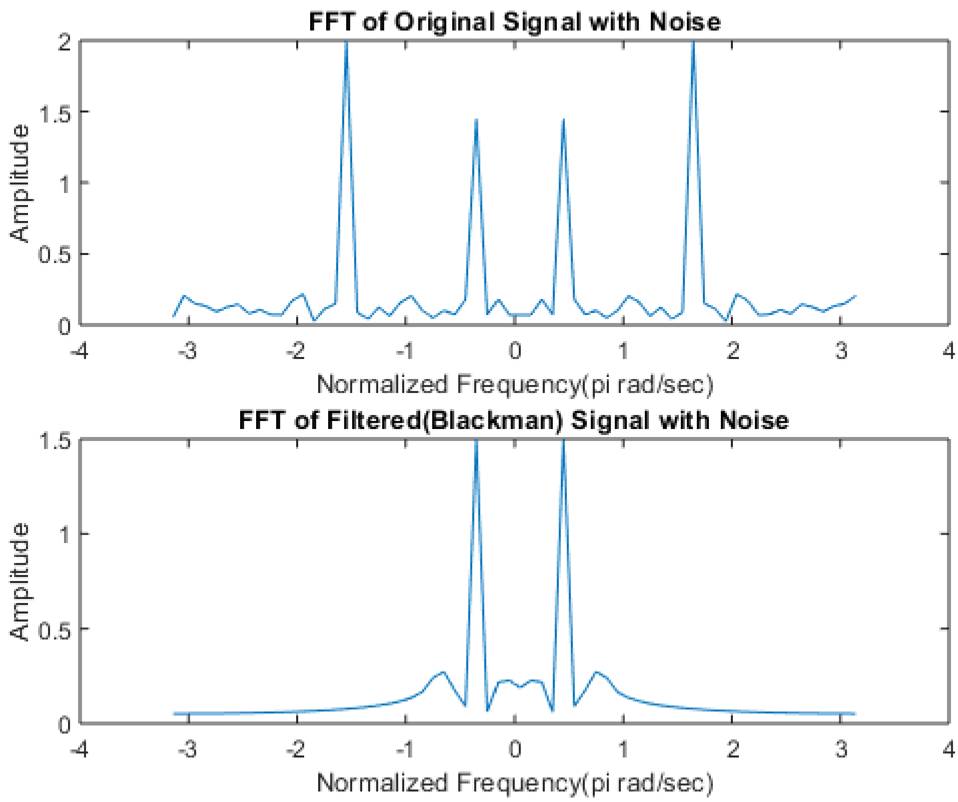
b)
```
N=64;
wc = pi/4;
k = (N-1)/2;
n = 0:1:N-1;
hd = (sin(wc*(n-k)))./(pi*(n-k));
w1 = (n>=0)-(n>=N).*(0.42-0.5*cos((2*pi*n)/(N-1))+0.08*cos((4*pi*n)/(N-1))); %blackman
h = hd.*w1;
c = -pi:0.01:pi;
[h1,w] = freqz(h,1,c);
h2 = abs(h1);

t = 0:1:3*(N-1);
x = 3*cos(pi/8*t)+4*cos(pi/2*t)+randn(size(t));

w2 = -pi:2*pi/(N-1):pi;

y = filtfilt(h,1,x);


x1 = fft(x,N);
x1 = fftshift(x1);
x1a = abs(x1/N);
subplot(211);
plot(w2,x1a);
title('FFT of Original Signal with Noise');
xlabel('Normalized Frequency(pi rad/sec)');
ylabel('Amplitude');

y1 = fft(y,N);
y1 = fftshift(y1);
y1a = abs(y1/N);
subplot(212);
plot(w2,y1a);
title('FFT of Filtered(Blackman) Signal with Noise');
xlabel('Normalized Frequency(pi rad/sec)');
ylabel('Amplitude');
```

# b) For SNR Calculation:

```
N = 64;
wc = pi/3;
k = (N-1)/2;
n = 0:1:N-1;
hd = (sin(wc*(n-k)))./(pi*(n-k));
w1 = (n>=0)-(n>=N);
h = hd.*w1;
c = -pi:0.01:pi;
[h1,w] = freqz(h,1,c);
h2 = abs(h1);


t = 0:1:3*(N-1);
x = 3*cos(pi/8*t)+5*cos(pi/2*t);
y = filtfilt(h,1,x);
x1 = 3*cos(pi/8*t)+5*cos(pi/2*t) + 0.1*randn(size(t));
y1 = filtfilt(h,1,x1);

yn = y1-y;
xn = x1-x;

R1 = snr(y,yn);
R2 = snr(x,xn);

E1 = 0; %y
E2 = 0; %yn
E3 = 0; %x
E4 = 0; %xn

for i=1:1:3*(N-1)
    E1 = E1+y(i)*y(i);
end

for i=1:1:3*(N-1)
    E2 = E2+yn(i)*yn(i);
end

for i=1:1:3*(N-1)
    E3 = E3+x(i)*x(i);
end

for i=1:1:3*(N-1)
    E4 = E4+xn(i)*xn(i);
end

SNRin = 10*log(E3/E4);
SNRout = 10*log(E1/E2);
```

(iii)Results:

FFT of Blackman Filter with Noise:




SNR Calculations for various filters:

N=8

| Window | SNRin(dB) | SNRout(dB) |
|---|---|---|
| Rectangular | 17.7469 | 15.0708 |
| Triangular | 17.8404 | 15.8097 |
| Hanning | 17.7695 | 16.1127 |
| Hamming | 17.8849 | 16.0747 |
| Blackman | 17.8256 | 16.1860 |

N=64

| Window | SNRin(dB) | SNRout(dB) |
|---|---|---|
| Rectangular | 17.7758 | 15.7602 |
| Triangular | 17.8258 | 15.9406 |
| Hanning | 17.8235 | 15.9720 |
| Hamming | 17.9184 | 16.0036 |
| Blackman | 17.9240 | 16.0561 |

N=512

| Window | SNRin(dB) | SNRout(dB) |
|---|---|---|
| Rectangular | 17.9152 | 15.9896 |
| Triangular | 17.8151 | 15.8714 |
| Hanning | 17.9141 | 16.0256 |
| Hamming | 17.8771 | 15.9285 |
| Blackman | 17.8743 | 15.9495 |

# Discussions:

We can see that as we increase the value of N (i.e. the number of samples of IIR response of filter considered) the filter response becomes sharper with faster transitions from pass-band to stop-band and lower side lobes. In general, the rectangular window has much faster transitions but comparatively higher side lobes around -20 dB which is 10% of passband value and hence not a good response. The triangular window has very slow transitions but no significant side lobes. The Hanning, Hamming and Blackman windows have comparatively slower transitions but have much lower side-bands which means a much better stop-band performance.

We observed that higher the order of the filter (value of N), the better is the impulse response i.e. lesser transition width (from pass-band to stop-band), more stop-band attenuation and lower value of side-lobe peaks. It was observed that rectangular window has lower transition width as compared to other windows but has comparatively more energy in the side lobes. No significant side lobes were observed in the triangular window. The other three windows have much better performance in the stop-band but comparatively have larger transition width than the rectangular window.

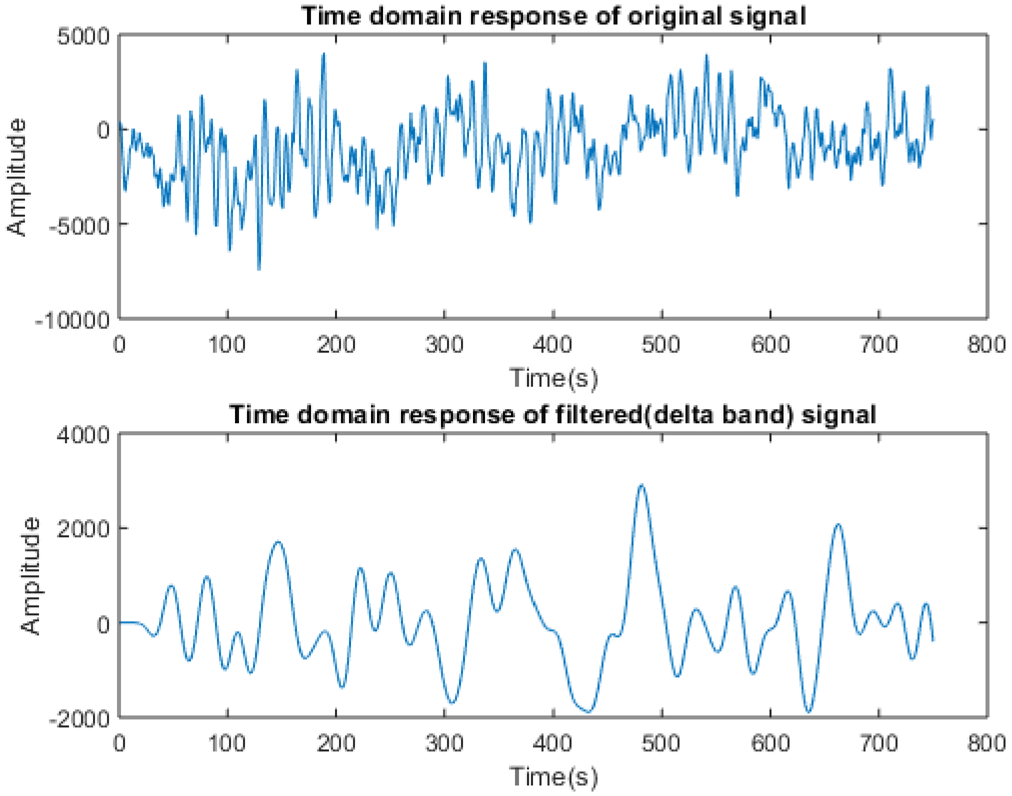
# References:

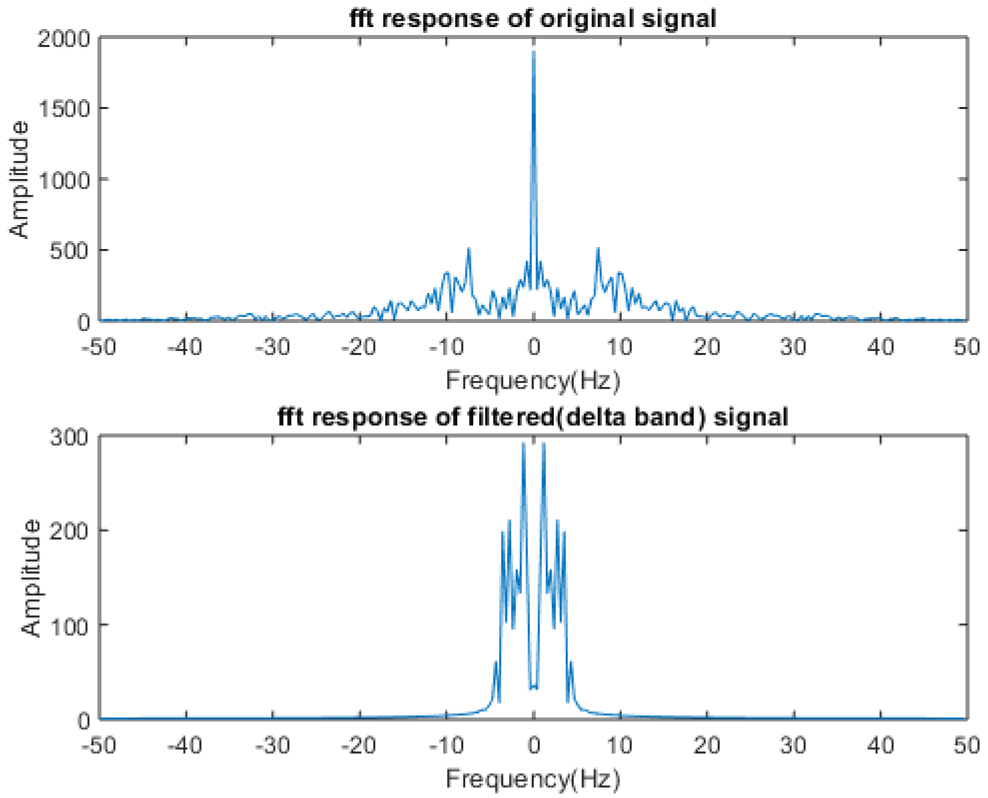
-Alan V. Oppenheim et.al Discrete Time Signal Processing.
-Wikipedia.
-Digital Signal Processing – Proakis  and Manolakis

# Extended Experiment on EEG Signals:

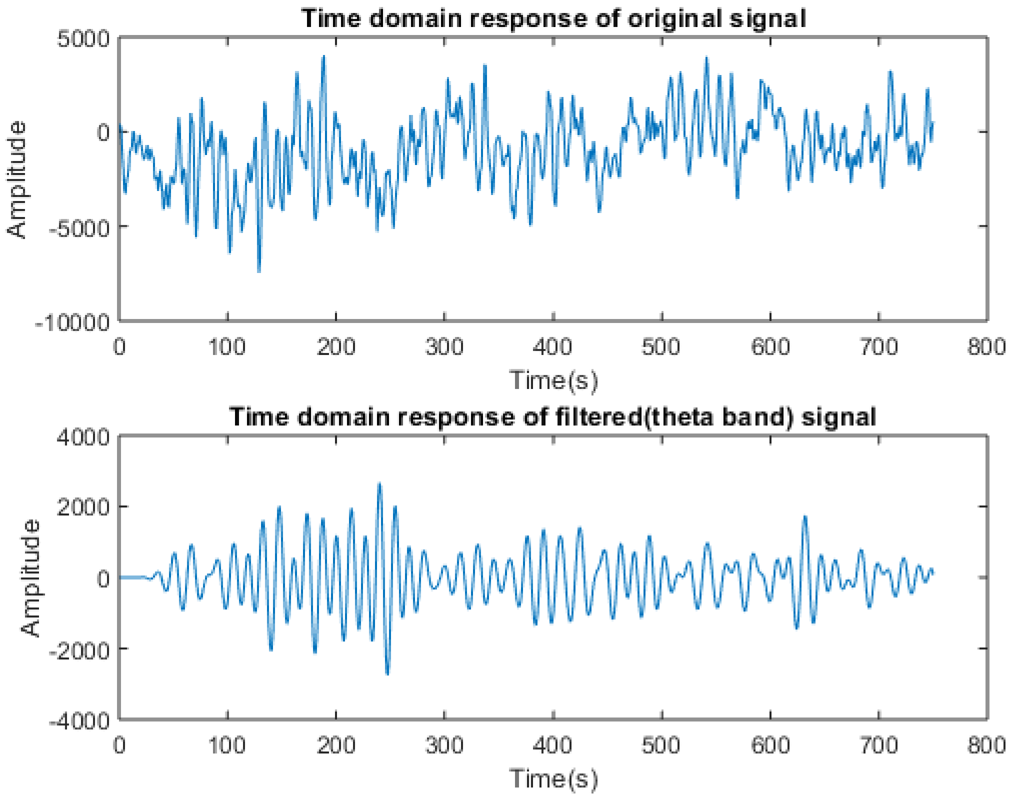
## Code and results:

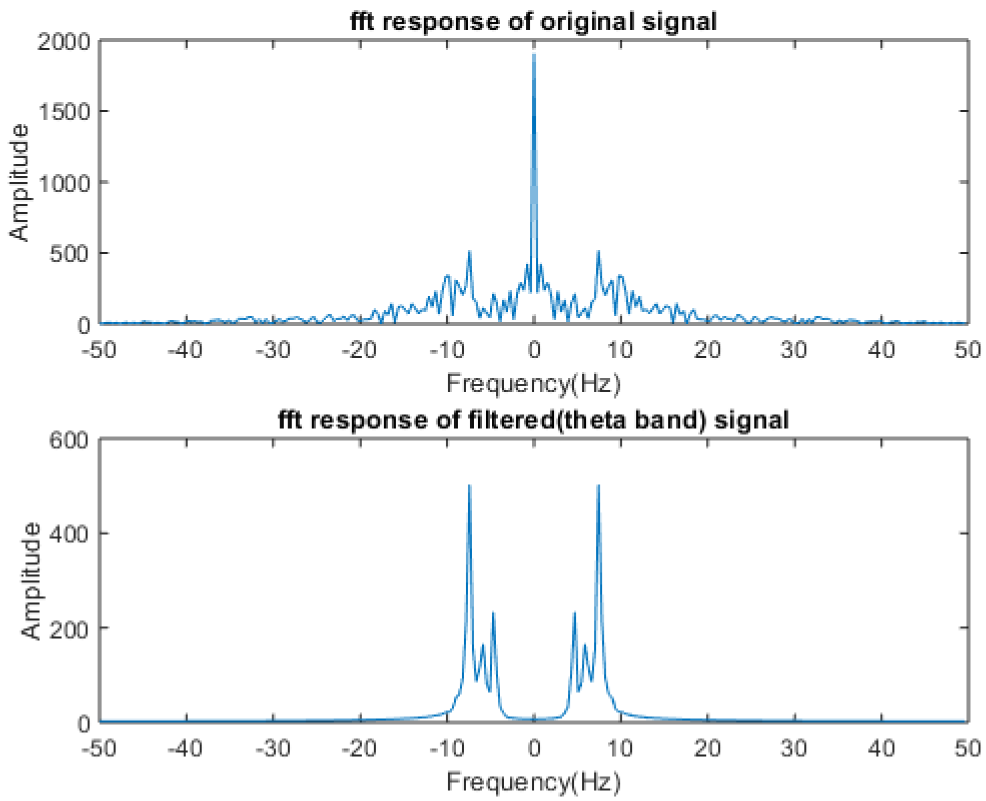
```
x = load ('C:\Users\Pranit Dalal\Downloads\EXP2\signals\eeg1-f3.dat');
fs = 100;
t = 0:1/fs:1;

N = 256;

bpf = designfilt('bandpassiir','FilterOrder',20,'HalfPowerFrequency1',0.5
,'HalfPowerFrequency2',4 ,'SampleRate',100);
y = filter(bpf,x);
figure(1);
subplot(211);
plot(x);
title('Time domain response of original signal');
xlabel('Time(s)');
ylabel('Amplitude');
subplot(212);
plot(y);
title('Time domain response of filtered(delta band) signal');
xlabel('Time(s)');
ylabel('Amplitude');

y1 = fft(x,N);
y1 = fftshift(y1);
m1 = abs(y1/N);
f1 = (-N/2:(N/2-1))*100/N;
figure(2);
subplot(211);
plot(f1,m1);
title('fft response of original signal');
xlabel('Frequency(Hz)');
ylabel('Amplitude');
y2 = fft(y,N);
y2 = fftshift(y2);
m2 = abs(y2/N);
f2 = (-N/2:(N/2-1))*100/N;
subplot(212);
plot(f2,m2);
title('fft response of filtered(delta band) signal');
xlabel('Frequency(Hz)');
ylabel('Amplitude');
```


Time domain response of original signal
Amplitude
Time(s)
5000
0
-5000
-10000
0
100
200
300
400
500
600
700
800
Time domain response of filtered(delta band) signal
Amplitude
Time(s)
4000
2000
0
-2000
0
100
200
300
400
500
600
700
800

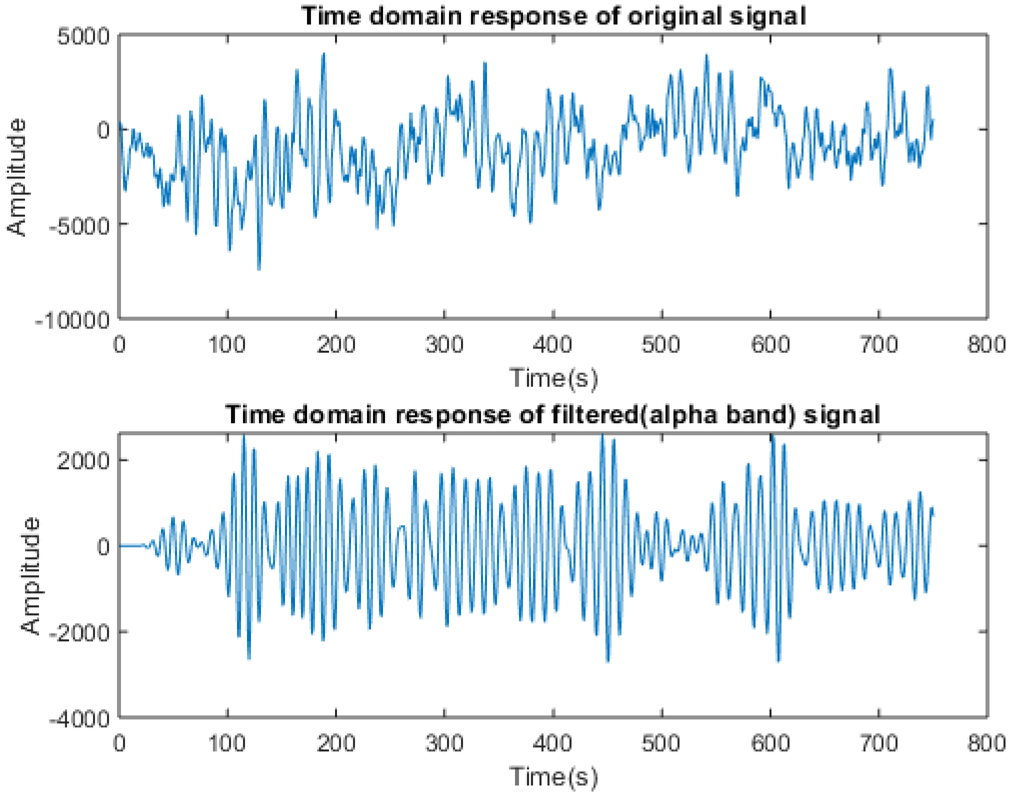

fft response of original signal
Amplitude
Frequency(Hz)
2000
1500
1000
500
0
-50
-40
-30
-20
-10
0
10
20
30
40
50
fft response of filtered(delta band) signal
Amplitude
Frequency(Hz)
300
200
100
0
-50
-40
-30
-20
-10
0
10
20
30
40
50

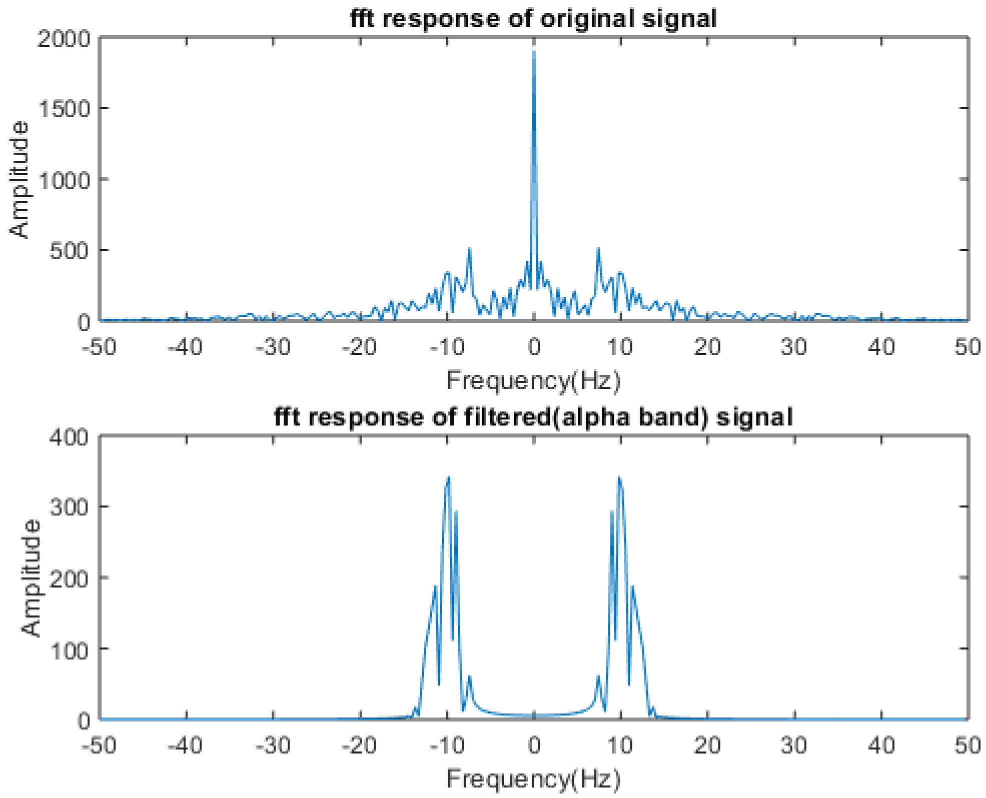
```
x = load ('C:\Users\Pranit Dalal\Downloads\EXP2\signals\eeg1-f3.dat');
fs = 100;
t = 0:1/fs:1;

N = 256;

bpf = designfilt('bandpassiir','FilterOrder',20,'HalfPowerFrequency1',4
,'HalfPowerFrequency2',8 ,'SampleRate',100);
y = filter(bpf,x);
figure(1);
subplot(211);
plot(x);
title('Time domain response of original signal');
xlabel('Time(s)');
ylabel('Amplitude');
subplot(212);
plot(y);
title('Time domain response of filtered(theta band) signal');
xlabel('Time(s)');
ylabel('Amplitude');

y1 = fft(x,N);
y1 = fftshift(y1);
m1 = abs(y1/N);
f1 = (-N/2:(N/2-1))*100/N;
figure(2);
subplot(211);
plot(f1,m1);
title('fft response of original signal');
xlabel('Frequency(Hz)');
ylabel('Amplitude');
y2 = fft(y,N);
y2 = fftshift(y2);
m2 = abs(y2/N);
f2 = (-N/2:(N/2-1))*100/N;
subplot(212);
plot(f2,m2);
title('fft response of filtered(theta band) signal');
xlabel('Frequency(Hz)');
ylabel('Amplitude');
```


Time domain response of original signal
Amplitude
Time(s)
Time domain response of filtered(theta band) signal
Amplitude
Time(s)


fft response of original signal
Amplitude
Frequency(Hz)
fft response of filtered(theta band) signal
Amplitude
Frequency(Hz)

```
x = load ('C:\Users\Pranit Dalal\Downloads\EXP2\signals\eeg1-f3.dat');
fs = 100;
t = 0:1/fs:1;

N = 256;

bpf = designfilt('bandpassiir','FilterOrder',20,'HalfPowerFrequency1',8
,'HalfPowerFrequency2',13 ,'SampleRate',100);
y = filter(bpf,x);
figure(1);
subplot(211);
plot(x);
title('Time domain response of original signal');
xlabel('Time(s)');
ylabel('Amplitude');
subplot(212);
plot(y);
title('Time domain response of filtered(alpha band) signal');
xlabel('Time(s)');
ylabel('Amplitude');

y1 = fft(x,N);
y1 = fftshift(y1);
m1 = abs(y1/N);
f1 = (-N/2:(N/2-1))*100/N;
figure(2);
subplot(211);
plot(f1,m1);
title('fft response of original signal');
xlabel('Frequency(Hz)');
ylabel('Amplitude');
y2 = fft(y,N);
y2 = fftshift(y2);
m2 = abs(y2/N);
f2 = (-N/2:(N/2-1))*100/N;
subplot(212);
plot(f2,m2);
title('fft response of filtered(alpha band) signal');
xlabel('Frequency(Hz)');
ylabel('Amplitude');
```




*Published with MATLAB® R2018a*