

Digital Signal Processing:
Experiment 3

Design of Low Pass Filters by Windowing Method

Rajeswari Mahapatra (15EC10044)

Group : 40

Aim of the experiment:

- Design FIR filters of various orders and cut-offs
- Check whether pass band and stop band frequencies are attenuated by the designed filters.
- Check the designed filter response corresponding to noise contaminated input.

Theory:

Low pass filters actually have an infinite (duration) impulse response which is not practical to implement in real life, hence FIR filters provide a way to practically implement various filters. They are non-recursive digital filters as they do not have a feedback.

Window method is most commonly used method for designing FIR filters. The simplicity of design process makes this method very popular. A window is a finite array consisting of coefficients selected to satisfy the desirable requirements.

Digital FIR Filters are specified by:

- The Windowing function
- The filter order

These two requirements are interrelated and there exists a compromise between

- The sharpness of the filter and selectivity
- Stop band attenuation

The impulse response of a Low-Pass filter is given by:

$$W(n) = \frac{\sin(\omega_c(n-k))}{\pi(n-k)} \quad \text{for } n \neq k$$
$$= \frac{\omega_c}{\pi} \quad \text{for } n = k$$

N = Number of Samples

$$K = \frac{N-1}{2}$$

The various windowing functions in time domain are:

Rectangular window	$W(n) = 1; n = 0, 1 \dots N - 1$ $0; \text{otherwise}$
Triangular window	$W(n) = 1 - 2 \left(\frac{n - (N - 1)/2}{N - 1} \right); n = 0, 1 \dots N - 1$ $0; \text{otherwise}$
Hanning window	$W(n) = 0.5 - 0.5 \cos \left(\frac{2\pi n}{N - 1} \right); n = 0, 1 \dots N - 1$ $0; \text{otherwise}$
Hamming window	$W(n) = 0.54 - 0.46 \cos \left(\frac{2\pi n}{N - 1} \right); n = 0, 1 \dots N - 1$ $0; \text{otherwise}$
Blackmann window	$W(n) = 0.42 - 0.5 \cos \left(\frac{2\pi n}{N - 1} \right)$ $+ 0.08 \cos \left(\frac{4\pi n}{N - 1} \right); n = 0, 1 \dots N - 1$ $0; \text{otherwise}$

Now one of these windowing functions are multiplied to the impulse response of a low pass filter to make an FIR filter.

Once the desired time domain characteristics of the FIR Filter are established, the freqz function is used to find and plot the frequency response of the FIR filter.

Design of filters for different orders

Code Snippets:

```
wc = 0.3*pi;

%%

hd = LPFilt(wc,N);

% plot(hd);

%%

w = rectWindow(N);

B1 = hd .* w;

figure;

freqz(B1,1,-0.9*pi:0.005:pi);

title(['Rectangular Window (N=' num2str(N) ')']);

%%

w = triWindow(N);

B2 = hd .* w;

figure;

freqz(B2,1,-pi:0.005:pi);

title(['Triangular Window (N=' num2str(N) ')']);

%%

w = hannWindow(N);

B3 = hd .* w;

figure;
```

```
freqz(B3,1,-pi:0.005:pi);  
title(['Hanning Window (N=' num2str(N) ')']);  
%%  
w = hamWindow(N);  
B4 = hd .* w;  
figure;  
freqz(B4,1,-pi:0.005:pi);  
title(['Hamming Window (N=' num2str(N) ')']);  
%%  
w = blackWindow(N);  
B5 = hd .* w;  
figure;  
freqz(B5,1,-pi:0.005:pi);  
title(['Blackman Window (N=' num2str(N) ')']);
```

Results:

The results for different orders of specific filters have been tabulated:

N = 9

Window	Transition Width(π)	Peak of First Lobe (dB)	Max Stop Band attenuation (dB)
Rectangle	0.17	-18.75	66
Triangle	0.274	-18.63	23
Hamming	0.5	-48.56	100
Hanning	0.42	-44.55	93
Blackmann	0.65	NA	76

N = 65

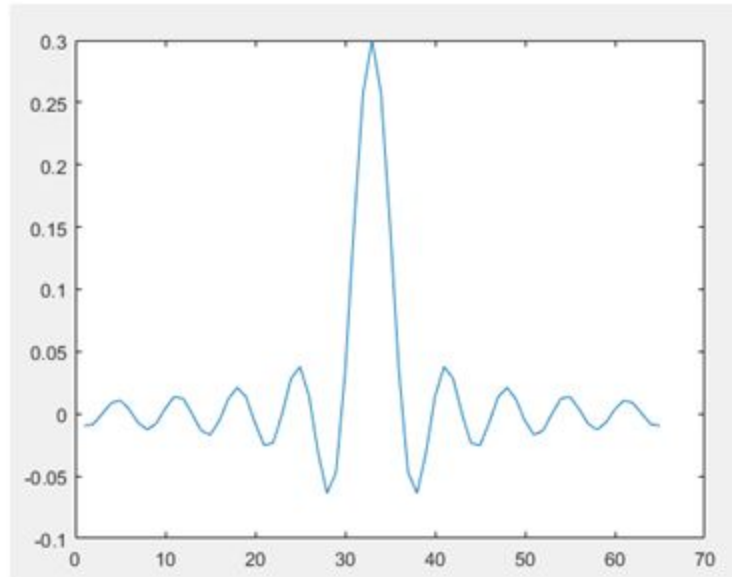
Window	Transition Width(π)	Peak of First Lobe (dB)	Max Stop Band attenuation (dB)
Rectangle	0.021	-20.94	100
Triangle	0.03	-21.26	42
Hamming	0.06	-53.54	107

Hanning	0.062	-44.43	140
Blackmann	0.1	-75.43	149

N = 257

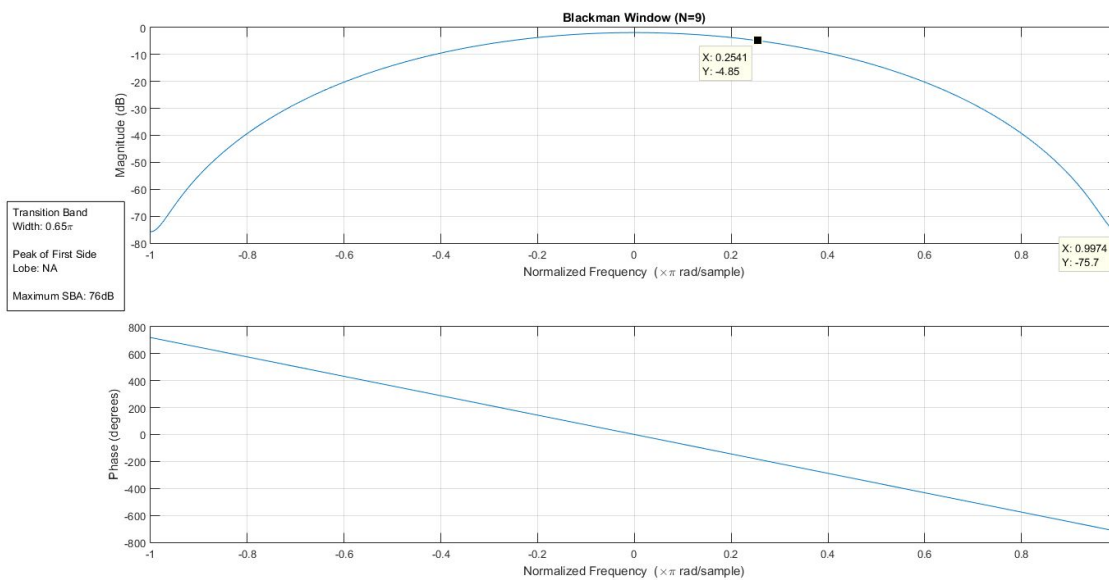
Window	Transition Width(π)	Peak of First Lobe (dB)	Max Stop Band attenuation (dB)
Rectangle	0.009	-20.86	66
Triangle	0.046	NA	51.19
Hamming	0.02	-57.24	124
Hanning	0.016	-44.24	186
Blackmann	0.02	-57.24	194

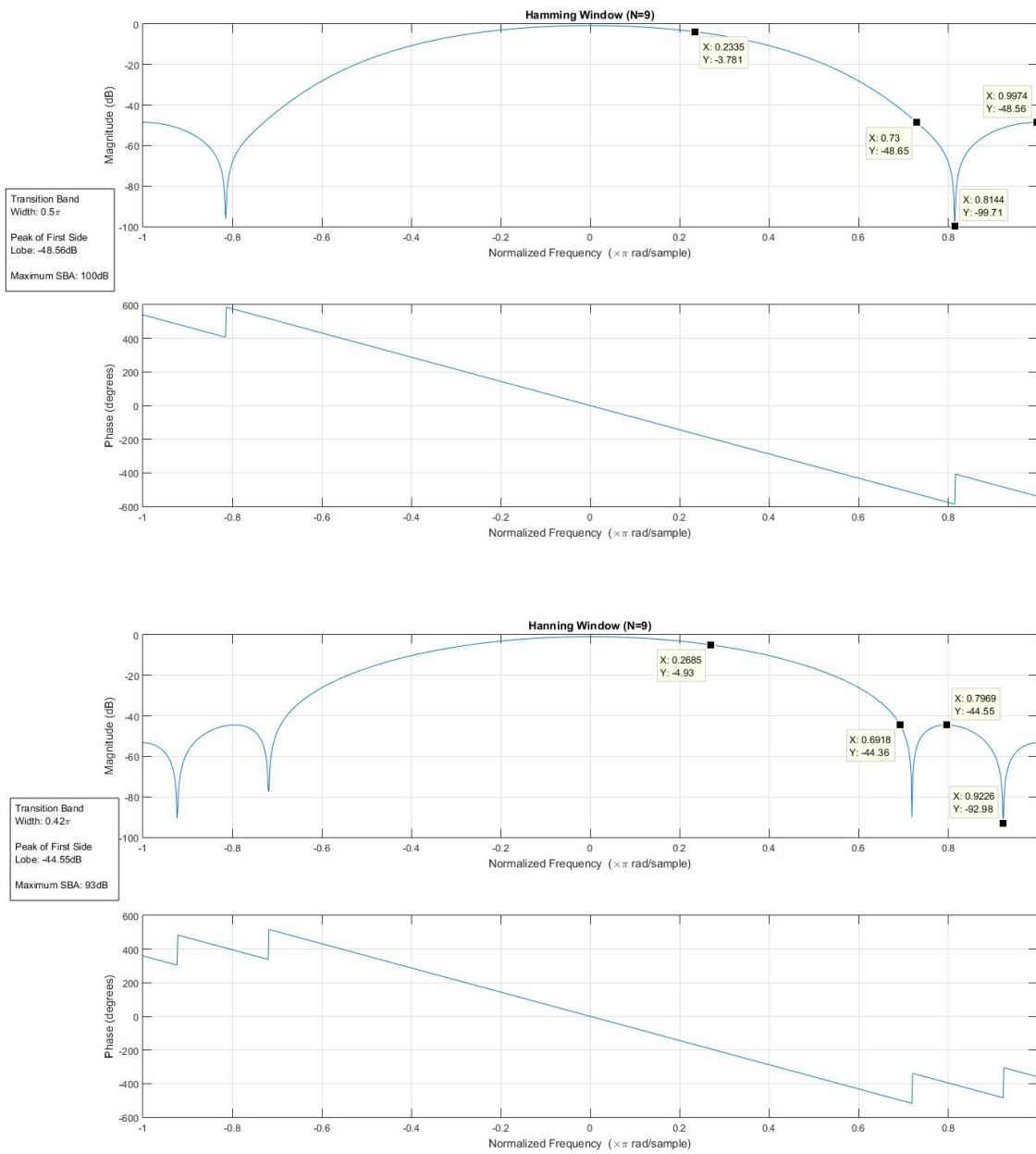
Plots of Frequency Responses:

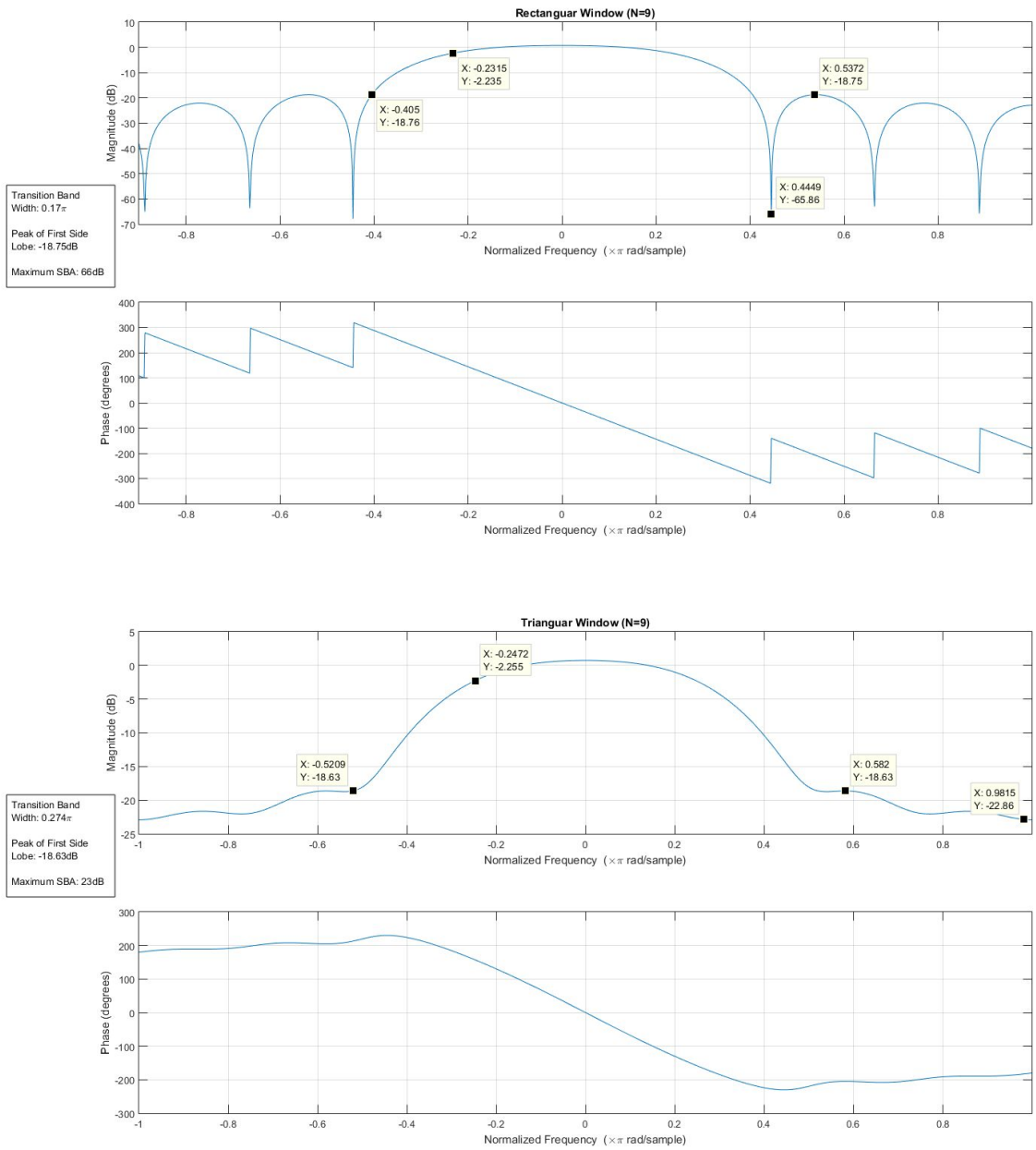


Impulse response of an ideal low-pass filter

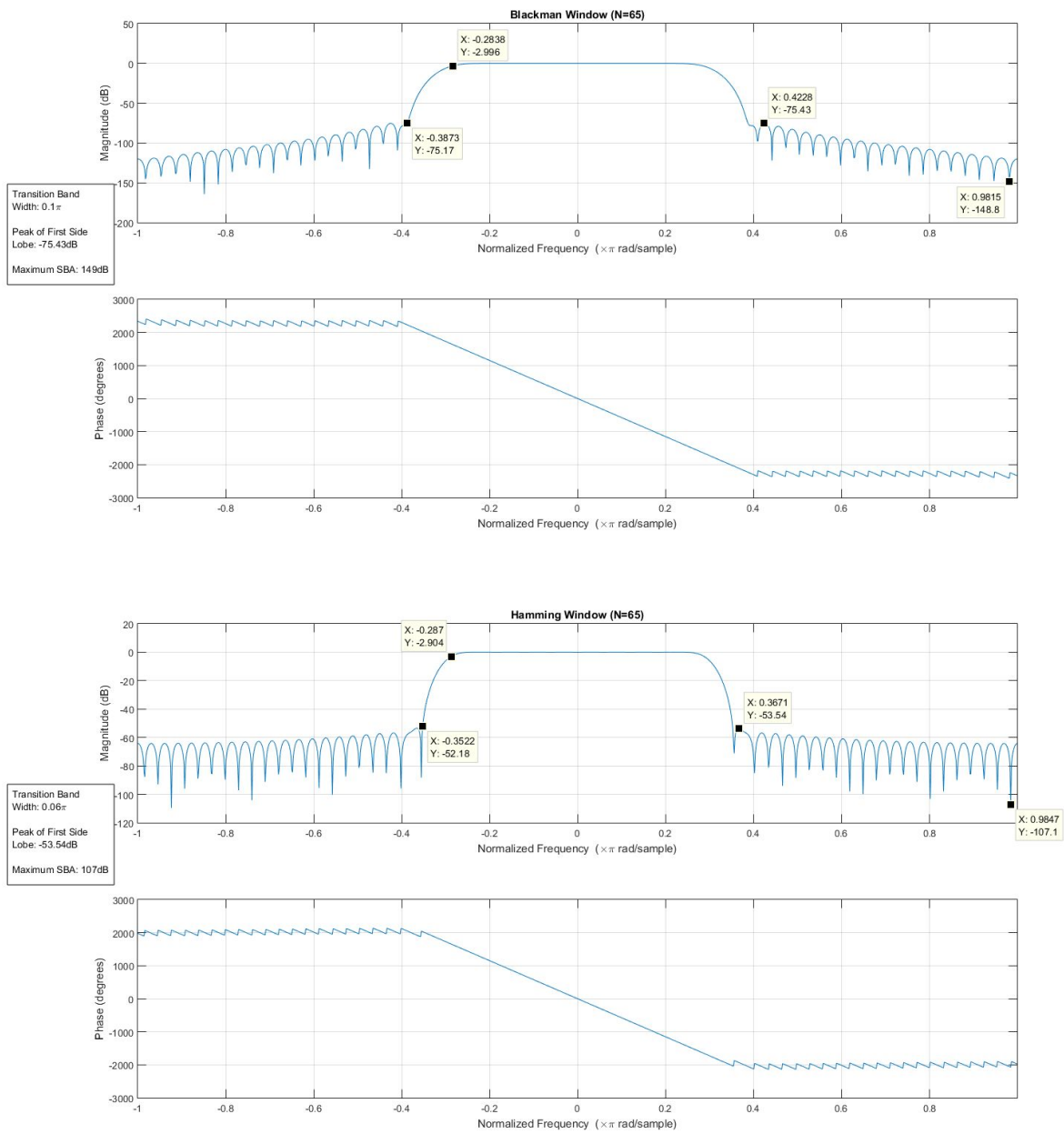
For $N=9$,

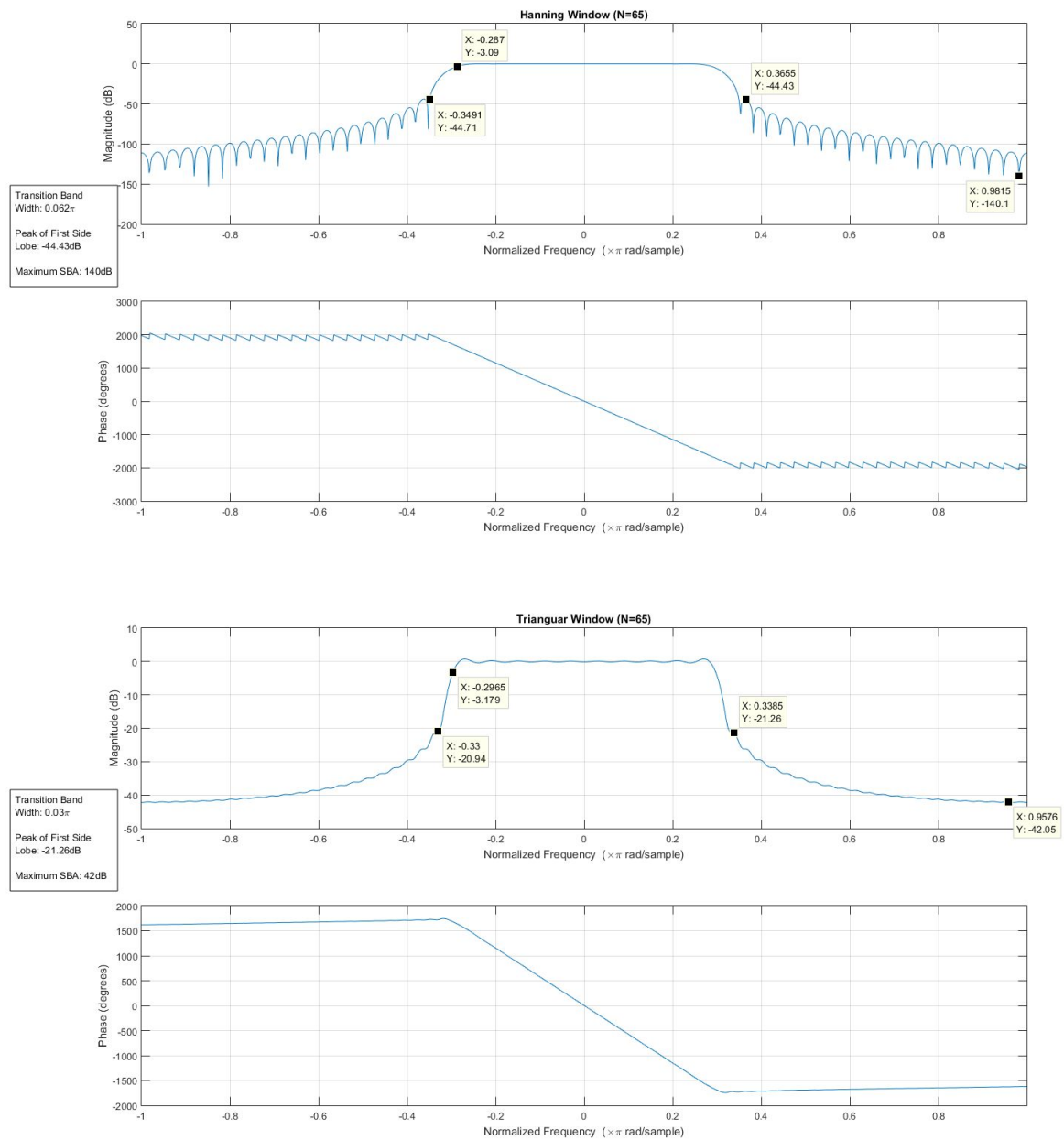


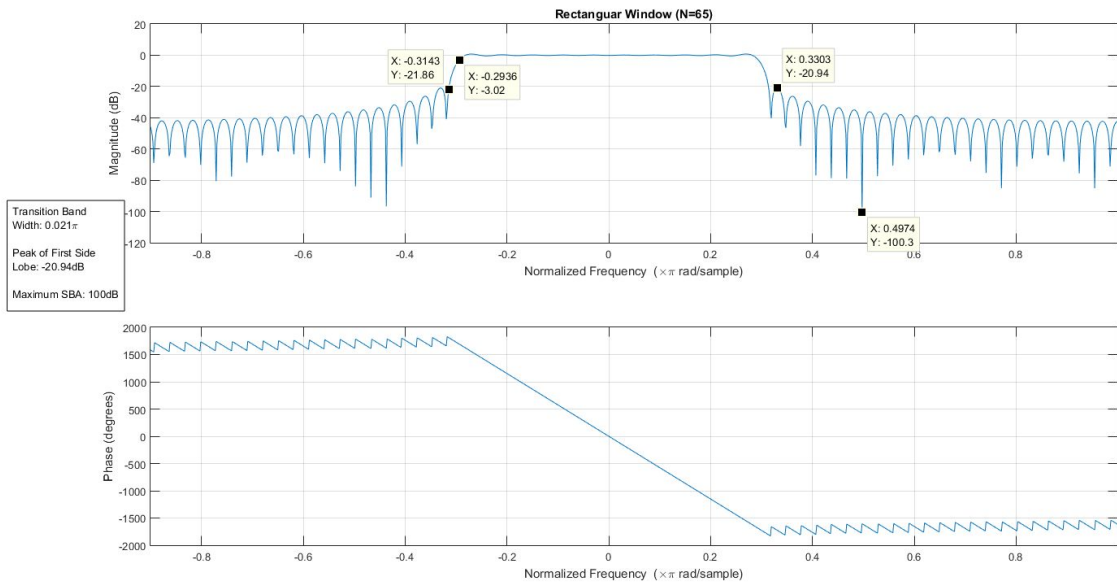




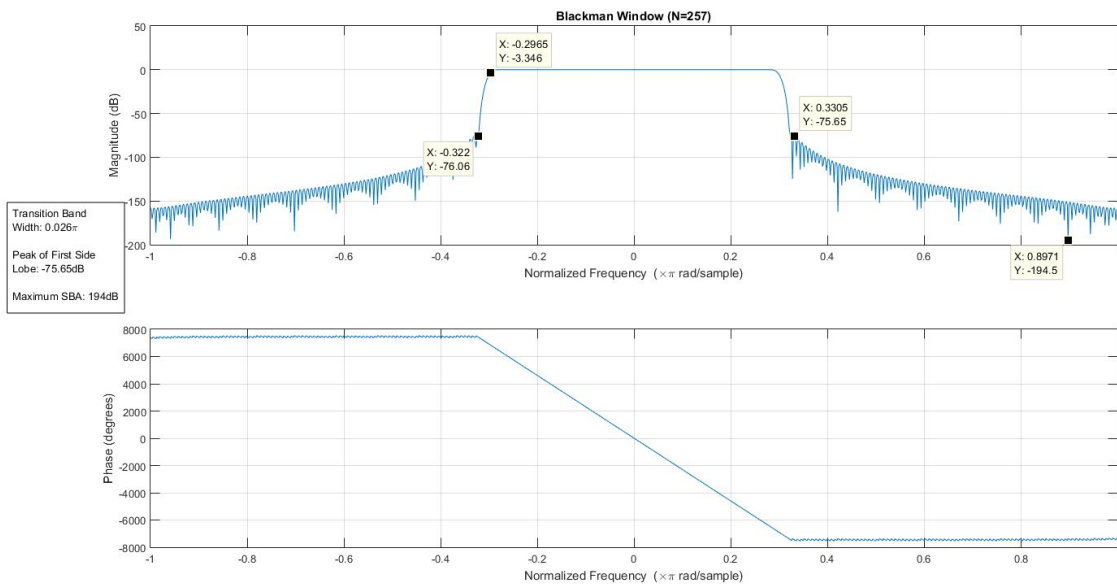
For N=65,

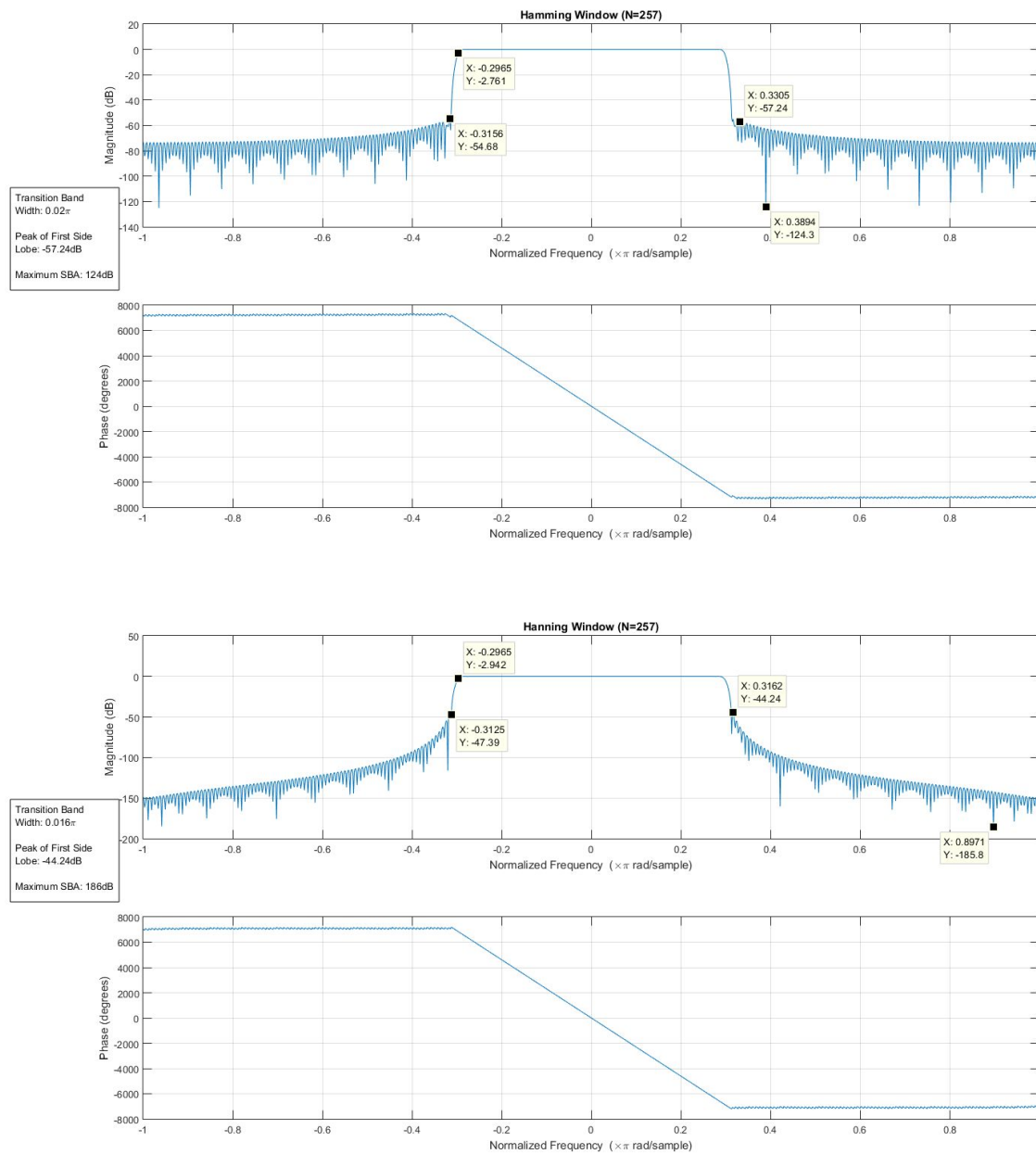


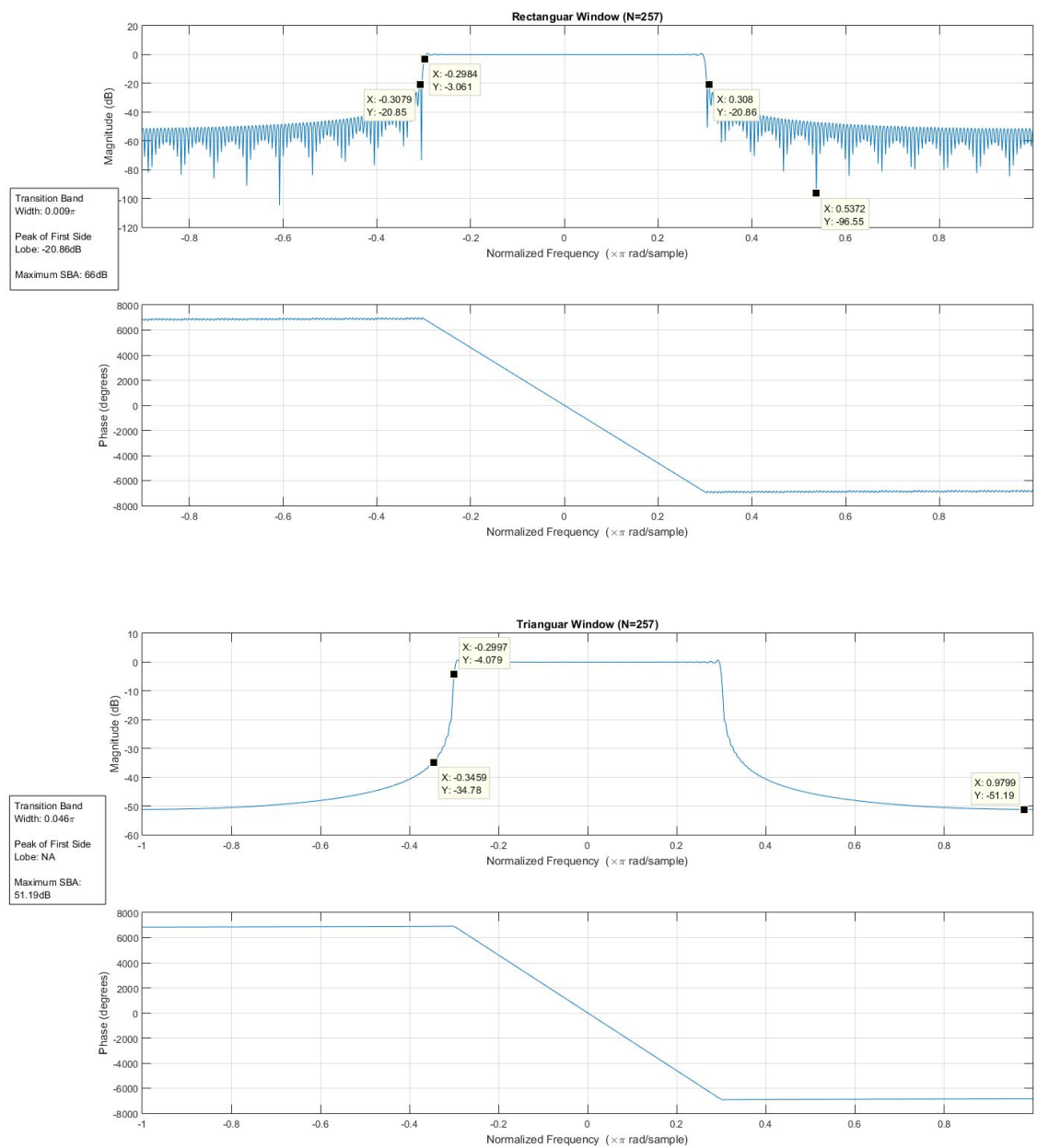




For N=257,





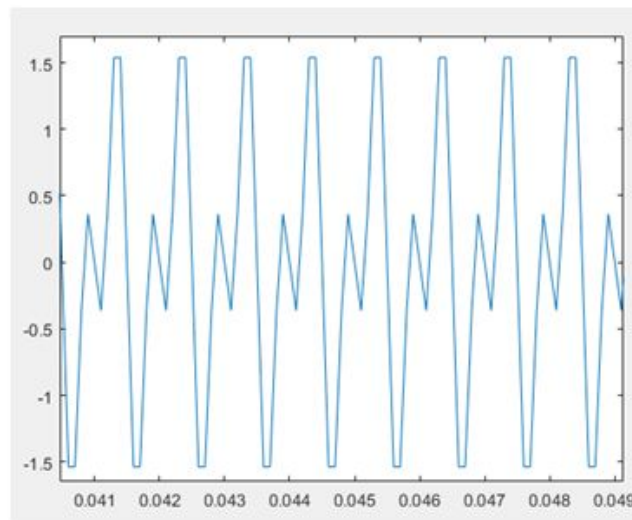


Filtered response for a given input:

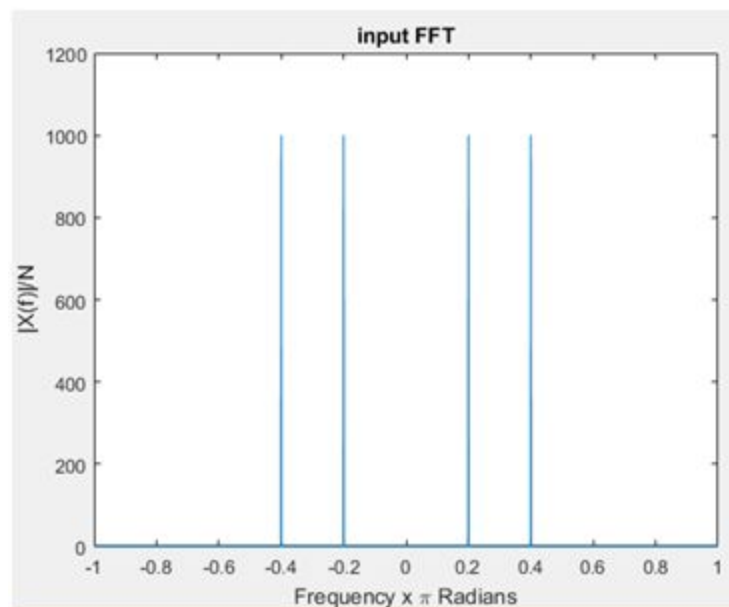
A sinusoid signal is generated such that one component is in the passband and another in the stop band.


$$x = \sin(2\pi \cdot 0.1 \cdot F_s \cdot t) + \sin(2\pi \cdot 0.8 \cdot F_s \cdot t)$$

Generated Signal:



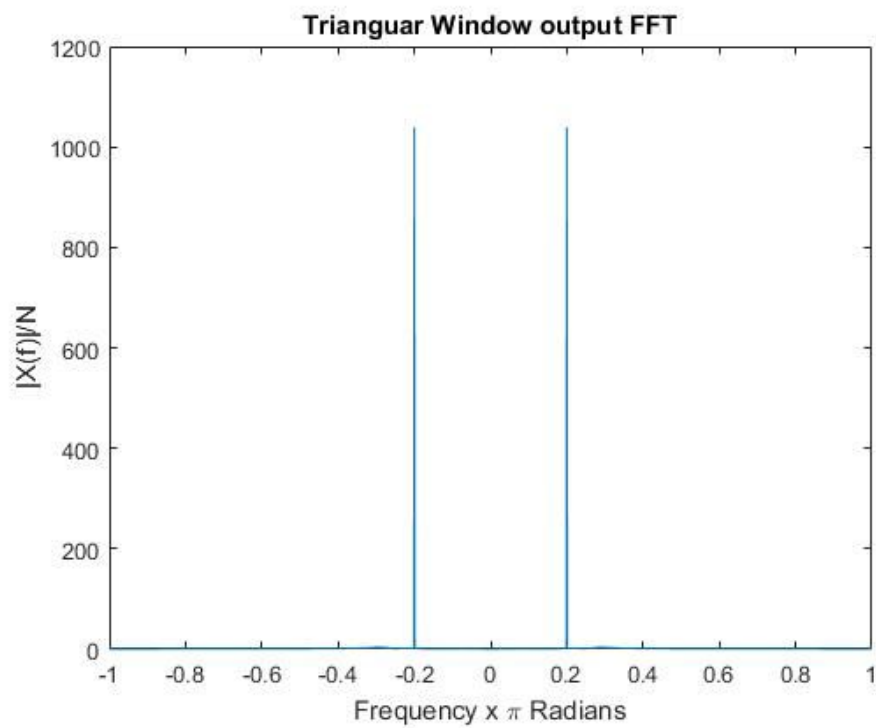
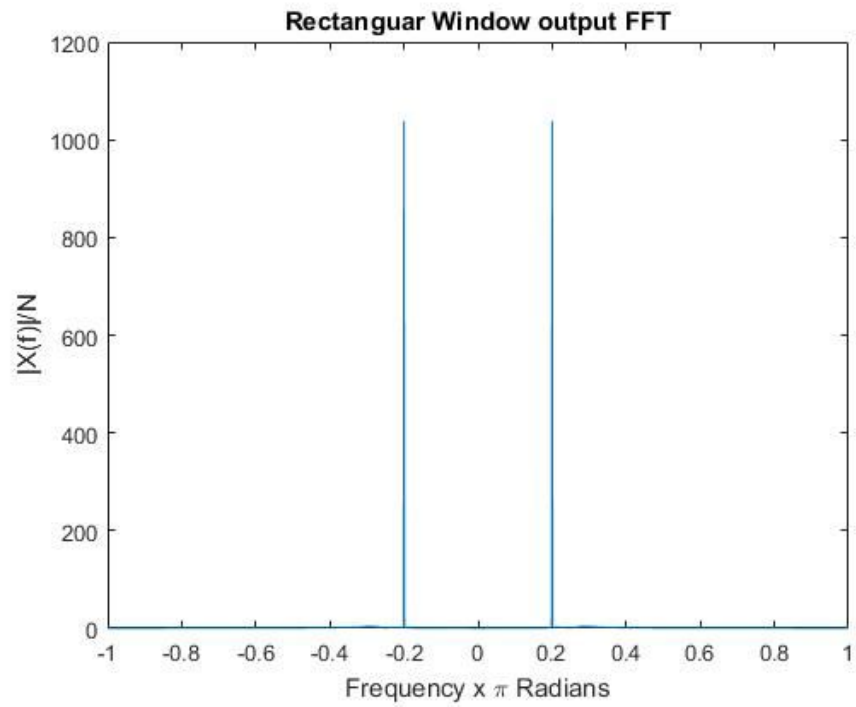
Input Signal DFT:

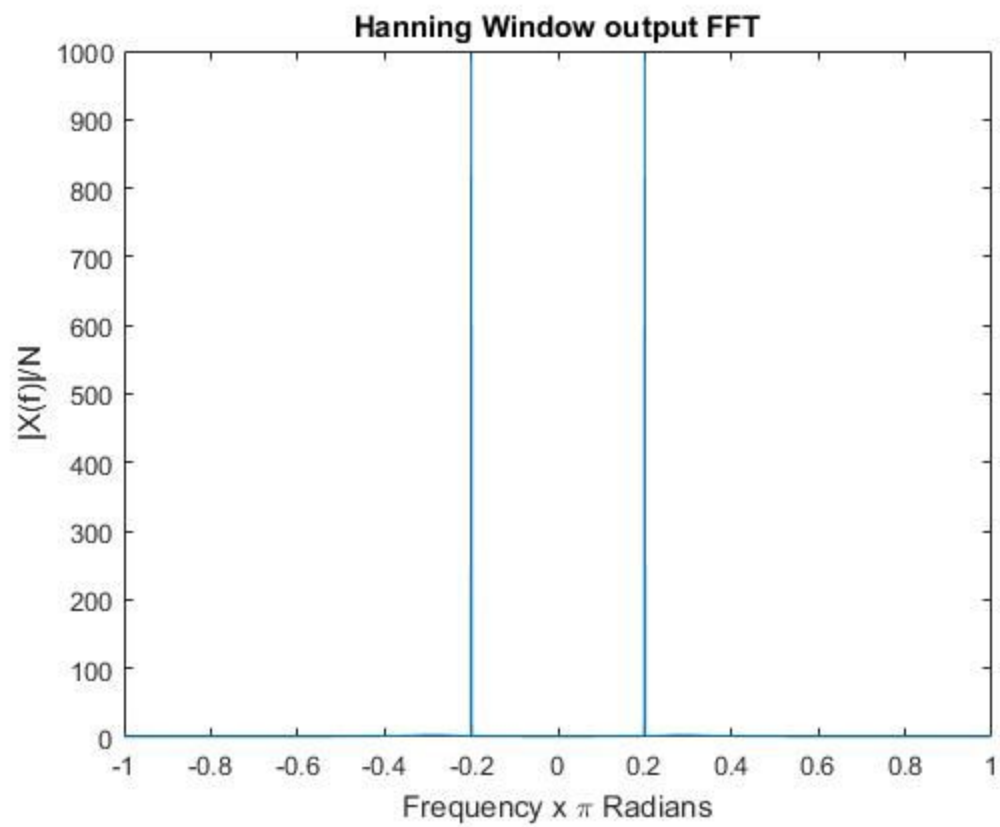
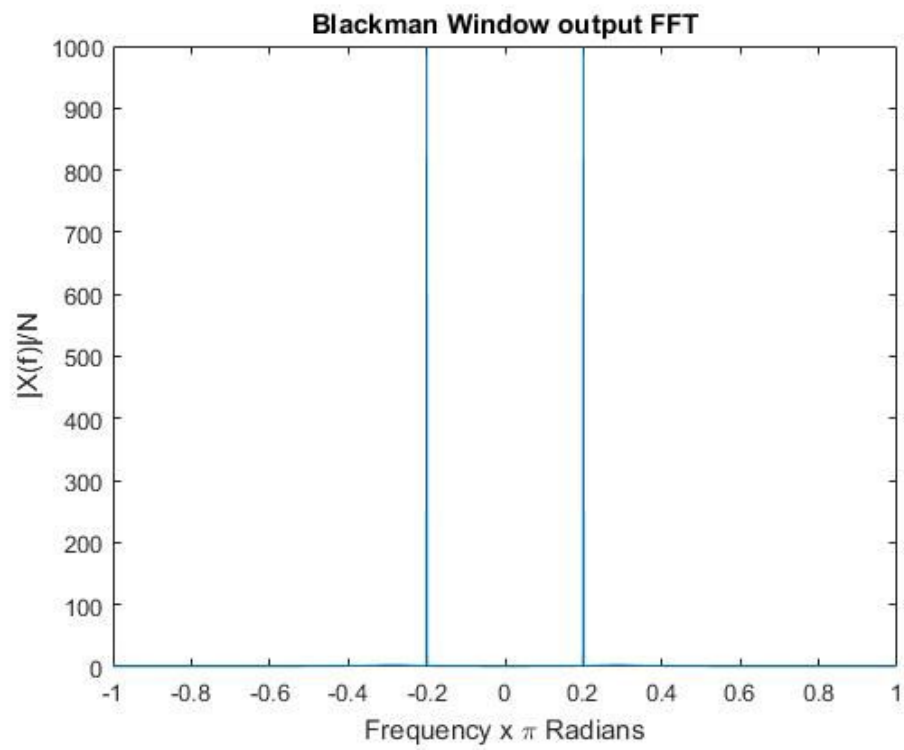


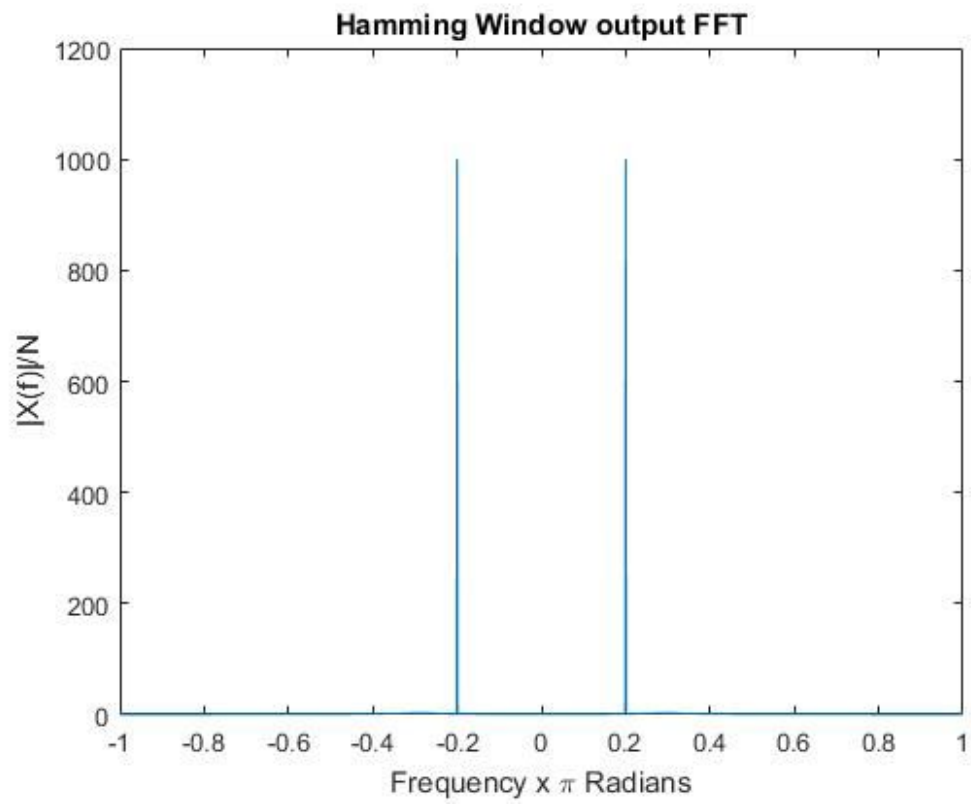


One of the components has a digital frequency of 0.2 while the other is at 0.4. Since the cut-off is at 0.3. One is in the passband while the other is in the stopband.

Filtered Outputs:

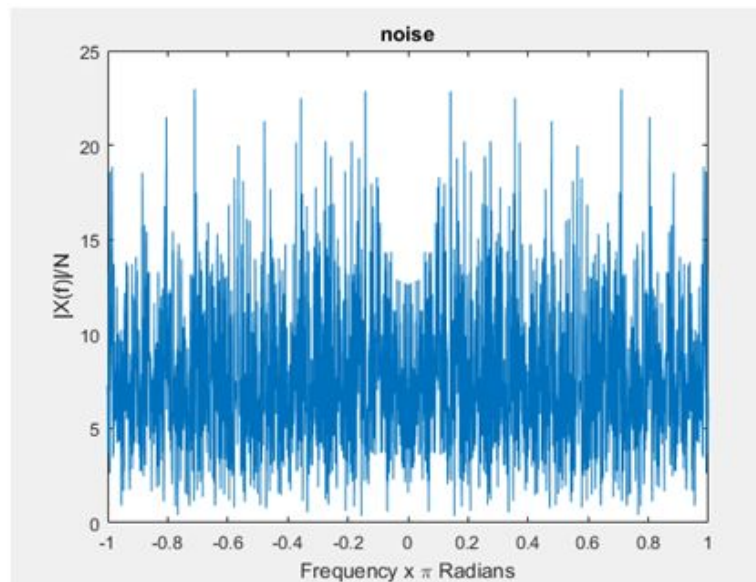




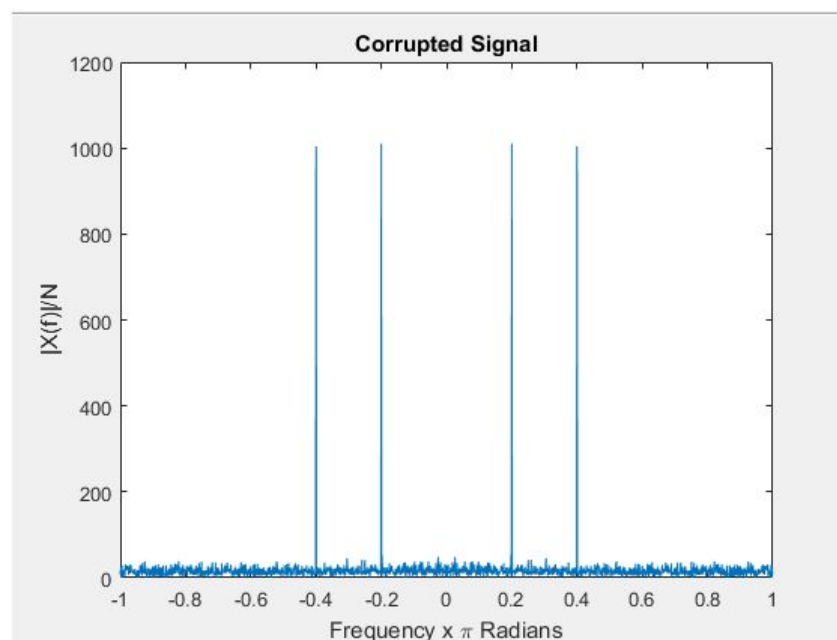


Addition of noise:

AWGN noise of a specific variance is now added to this signal.



Power spectral density of noise



Noise corrupted signal

Code Snippets:

```
%% Generating Input Signal

Fs = 10 * 1e3;           % Sampling frequency
T = 1/Fs;                % Sampling period
L = 2000;                % Length of signal
t = (0:L-1)*T;           % Time vector

x = sin(2*pi*0.1*Fs*t)+sin(2*pi*0.8*Fs*t);

%plot(t,x);

r = rms(x);

plotdft(x,Fs,'input FFT')

%% Filtering using FIR Filters

y = filtfilt(B1,1,x);

plotdft(y,Fs,'Rectanguar Window output FFT')

y = filtfilt(B2,1,x);

plotdft(y,Fs,'Trianguar Window output FFT')

y = filtfilt(B3,1,x);

plotdft(y,Fs,'Hanning Window output FFT')

y = filtfilt(B4,1,x);

plotdft(y,Fs,'Hamming Window output FFT')

y = filtfilt(B5,1,x);
```

```

plotdft(y,Fs,'Blackman Window output FFT')

%% Generation of noise

n = 0.4*r*randn(1,L);

plotdft(n,Fs,'noise')

%%

x1 = x+n;

plotdft(x1,Fs,'Corrupted Signal')

%% Filtering using FIR Filters

y1 = filtfilt(B1,1,x1);

plotdft(y1,Fs,'Rectangular Window output FFT')

snr(y1)

y1 = filtfilt(B2,1,x1);

plotdft(y1,Fs,'Trianguar Window output FFT')

snr(y1)

y1 = filtfilt(B3,1,x1);

plotdft(y1,Fs,'Hanning Window output FFT')

snr(y1)

y1 = filtfilt(B4,1,x1);

plotdft(y1,Fs,'Hamming Window output FFT')

snr(y1)

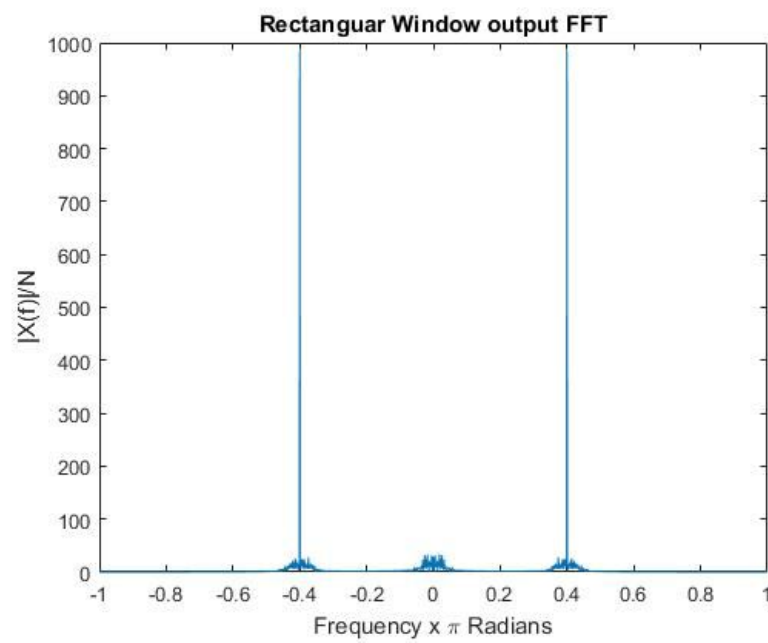
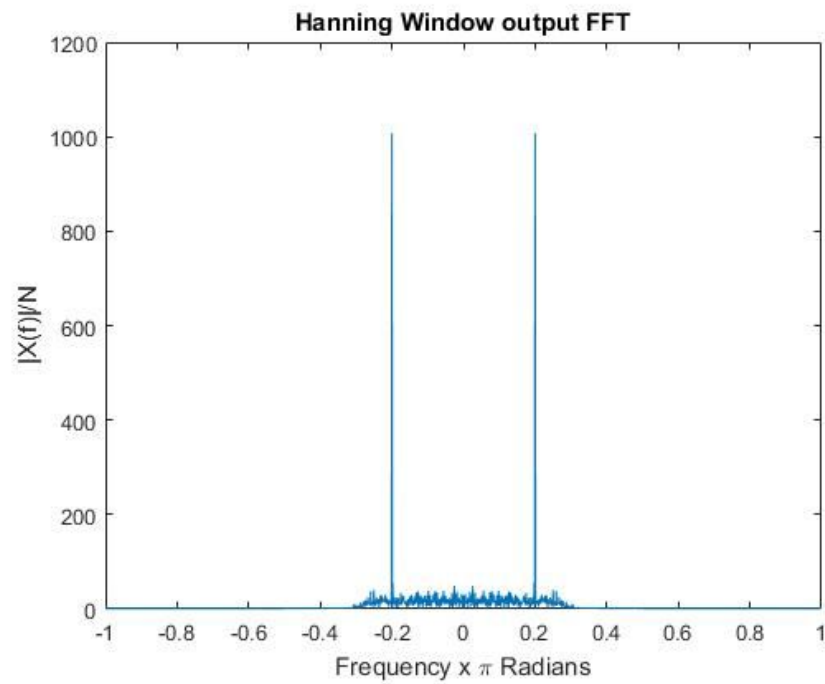
y1 = filtfilt(B5,1,x1);

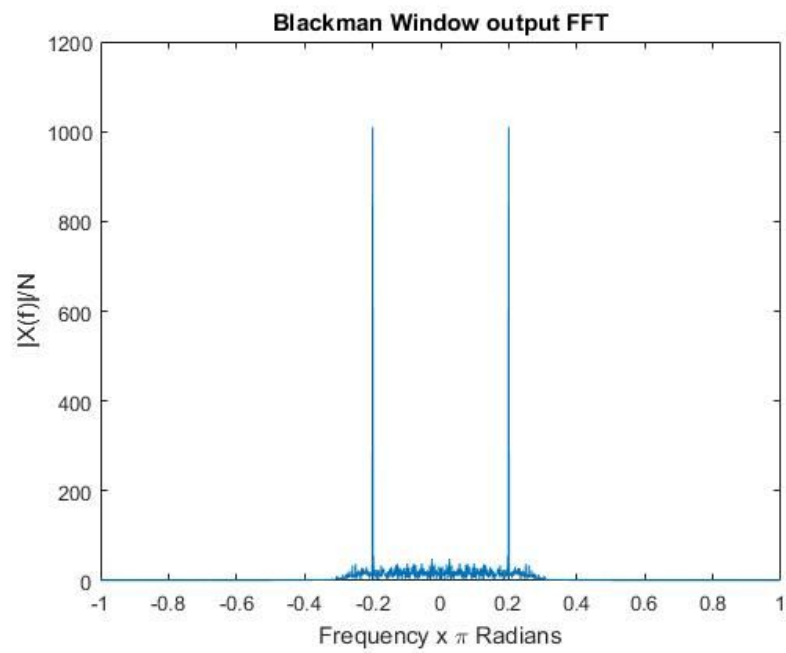
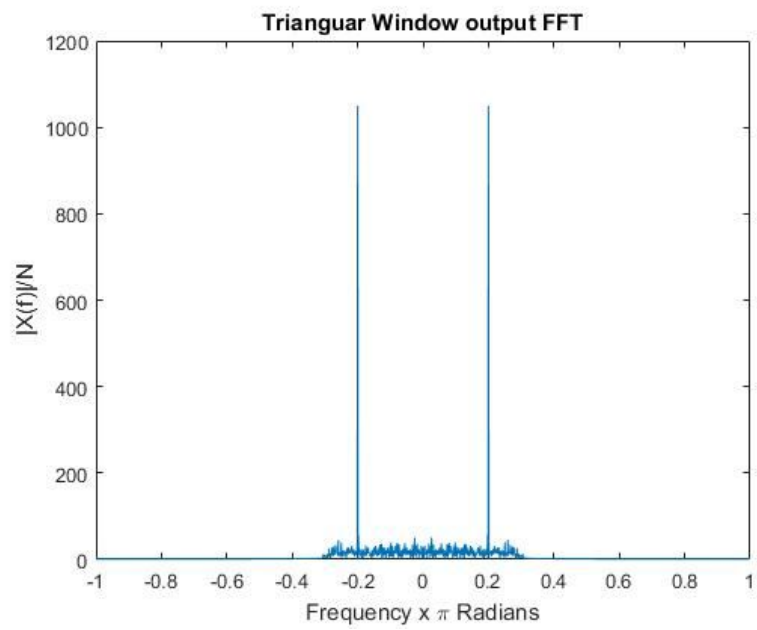
plotdft(y1,Fs,'Blackman Window output FFT')

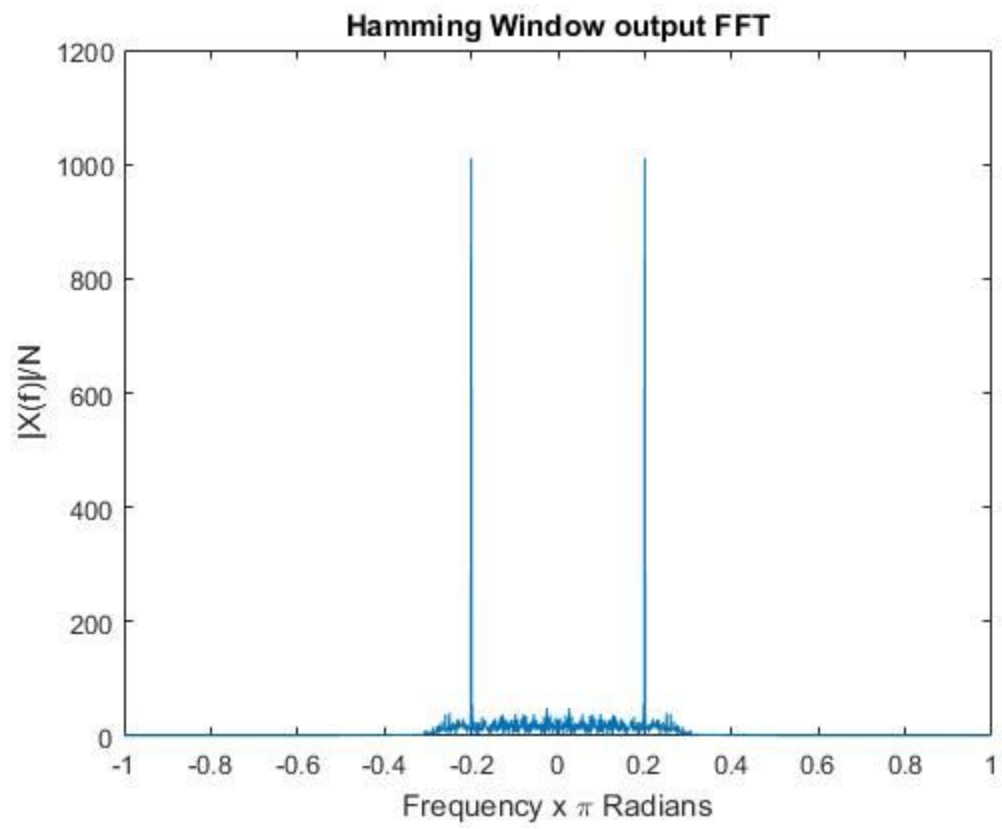
snr(y1)

```

Filtered Output:

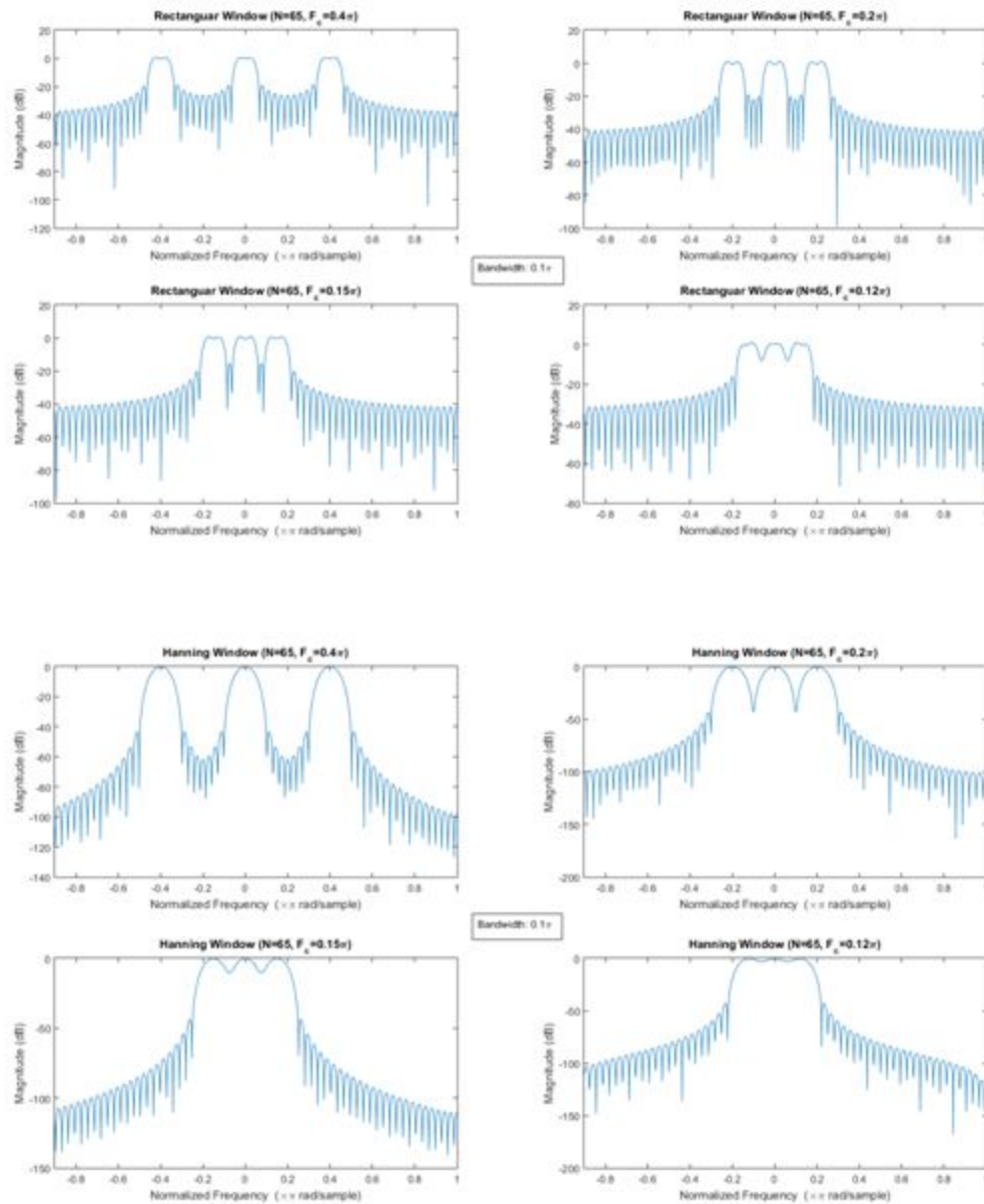






Limitations of Windowing Method:

The windowing method for designing FIR filters is very effective, but is usually used to design filters with only a single passband. For multiple passbands, we have constructive interference in stop bands and degradation of filter sharpness as well as stop band attenuation. This can be seen in the following two screenshots, for the rectangular and hanning windows:



As we can see, in rectangular windowing technique, as the passbands are brought closer, we have an added passband ripple ($>1.5\text{dB}$, which is the accepted standard for a good filter)

The selectivity is also hampered. This is even more evident in the Hanning windowing method. Although any passband ripple has been avoided, due to the slow roll-off of the Hanning FIR filter, we have lesser sharpness and selectivity and even at $F_c=0.15\pi$, we have sufficient interference, keeping us from differentiating the bands clearly.

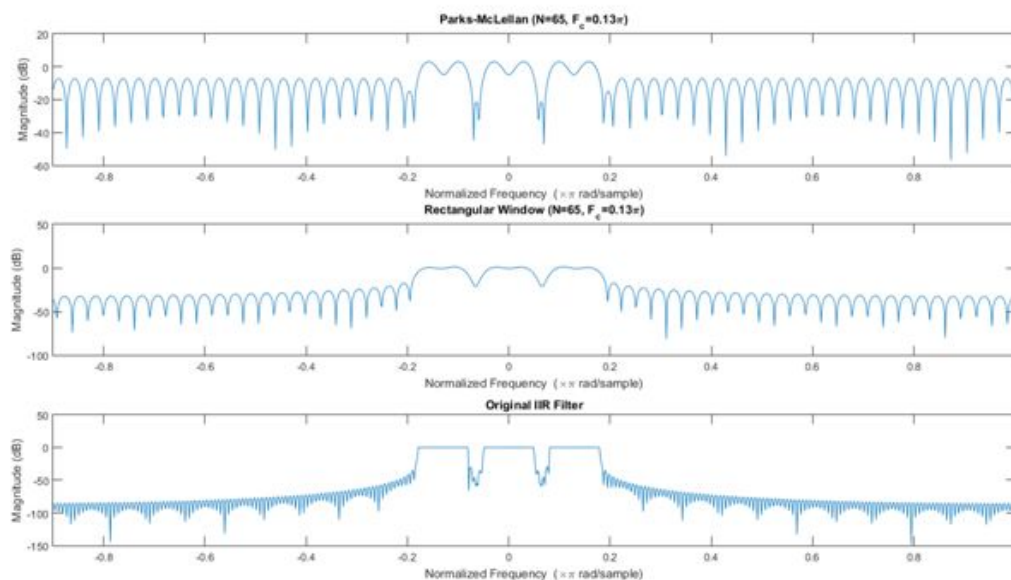
Code Snippets:

```
wc = 0.05*pi; %cutoff of band on either side
Fc = 0.13; %center frequency of pass band
N=65;
h1 = 2*BPFilt(wc,N,0.5*Fc)+LPFilt(wc,N); %compensate for division of power in
sidebands
w = rectWindow(N);
B1 = h1 .* w;
[A,~]=freqz(B1,1,-0.9*pi:0.005*pi);
%subplot(2,2,1)
plot(-0.9:1.9/size(A,2):1-1.9/size(A,2),20*log10(abs(A)));
xlim([-0.9 1]);
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Magnitude (dB)');
title(['Rectangular Window (N=' num2str(N) ', F_c=' num2str(Fc) '\pi)']);
```

Design of FIR filters using Parks-McClellan algorithm:

Parks-McClellan algorithm uses the Remez exchange algorithm and Chebyshev approximation theory to design optimal FIR filter corresponding to a given frequency response. This is optimal in the sense that the maximum error between the desired and actual response is minimized. We do see some extra ripple, which is a characteristic of the Chebyshev approximation.

Here are the results, showing a comparison between the filter response for Parks-McClellan method, Windowing method and IIR filter (desired frequency response):



Code Snippets:

```
h2=2*BPFilt(wc,1001,0.5*Fc)+LPFilt(wc,1001);  
[a,~]=freqz(h2,1,-0.9*pi:0.005*pi);  
a(1:566)=[];  
f=0:1/size(a,2):1-1/size(a,2);
```

```

b=firpm(N,f,abs(a));

subplot(3,1,1)

[A,~]=freqz(b,1,-0.9*pi:0.005*pi);

plot(-0.9:1.9/size(A,2):1-1.9/size(A,2),20*log10(abs(A)));

xlim([-0.9 1]);

xlabel('Normalized Frequency (\times\pi rad/sample)');

ylabel('Magnitude (dB)');

title(['Parks-McLellan (N=' num2str(N) ', F_c=' num2str(Fc) '\pi)']);

subplot(3,1,2)

[A,~]=freqz(B1,1,-0.9*pi:0.005*pi);

plot(-0.9:1.9/size(A,2):1-1.9/size(A,2),20*log10(abs(A)));

xlim([-0.9 1]);

xlabel('Normalized Frequency (\times\pi rad/sample)');

ylabel('Magnitude (dB)');

title(['Rectangular Window (N=' num2str(N) ', F_c=' num2str(Fc) '\pi)']);

subplot(3,1,3)

[A,~]=freqz(h2,1,-0.9*pi:0.005*pi);

plot(-0.9:1.9/size(A,2):1-1.9/size(A,2),20*log10(abs(A)));

xlim([-0.9 1]);

xlabel('Normalized Frequency (\times\pi rad/sample)');

ylabel('Magnitude (dB)');

title('Original IIR Filter');

```



Discussions:

- Windowing method is used to realize Low pass filters by making the impulse response finite, making it possible to be implemented using limited hardware.
- Multiple windowing schemes were used to obtain various FIR Filters and we can see that as the order of the filter increases the number of side lobes increased, at the same time the cut-off was sharper.
- The filters whose windows were narrower in time domain were stretched in the frequency domain.
- Rectangular has a better transition time at the cost of a worse stop band attenuation while Blackmann window had a better stop band attenuation at the cost of cutoff.
- While using windowing method, we have already used a truncated Sinc sequence. This is as MATLAB only takes finite sequences and since we are going to window it anyway, it doesn't matter. So, in a sense, we have already rectangular-windowed it beforehand.
- Following from the above, it is clear that we cannot provide the frequency response of the above to the Parks-McClellan algorithm as we need the desired IIR frequency response to convert to FIR, and the above filter is already FIR.