Experiment 2:

Designing Low Pass filter by windowing

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

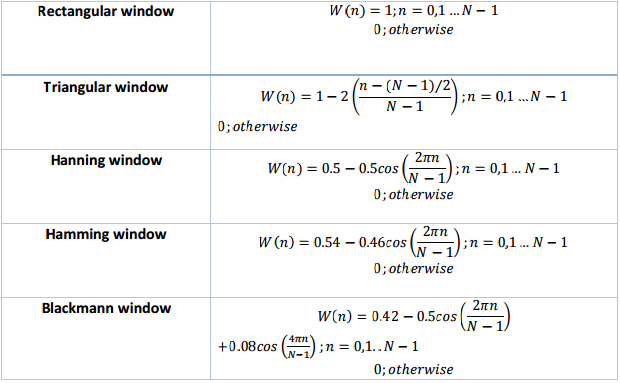
* The sharpness of the filter and selectivity
* Stop band attenuation

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

N = Number of Samples

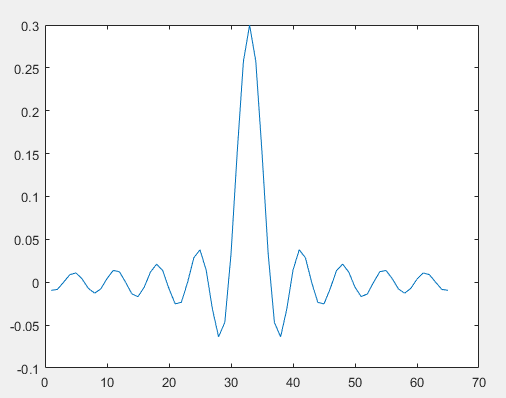
K =

The various windowing functions in time domain are:



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

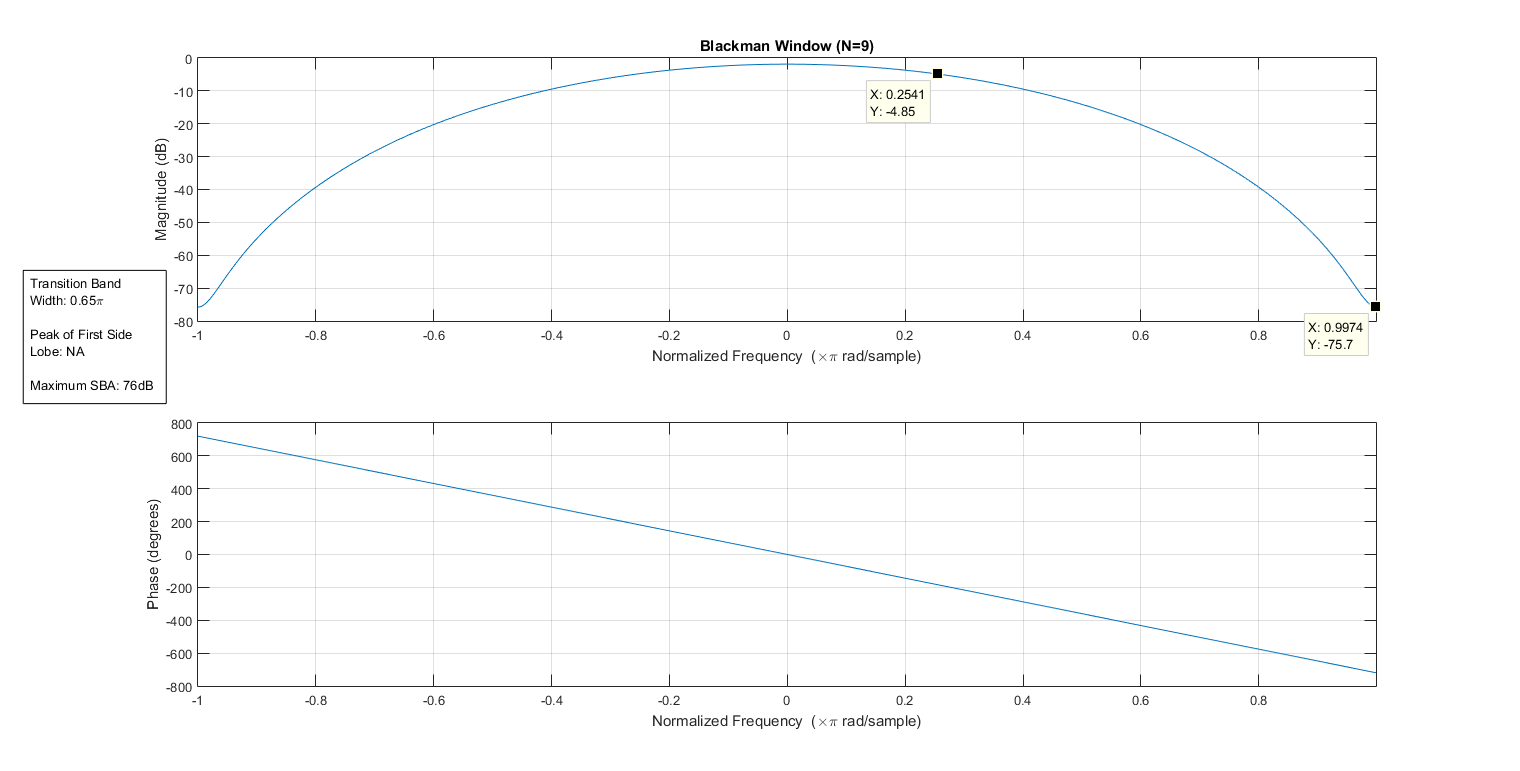
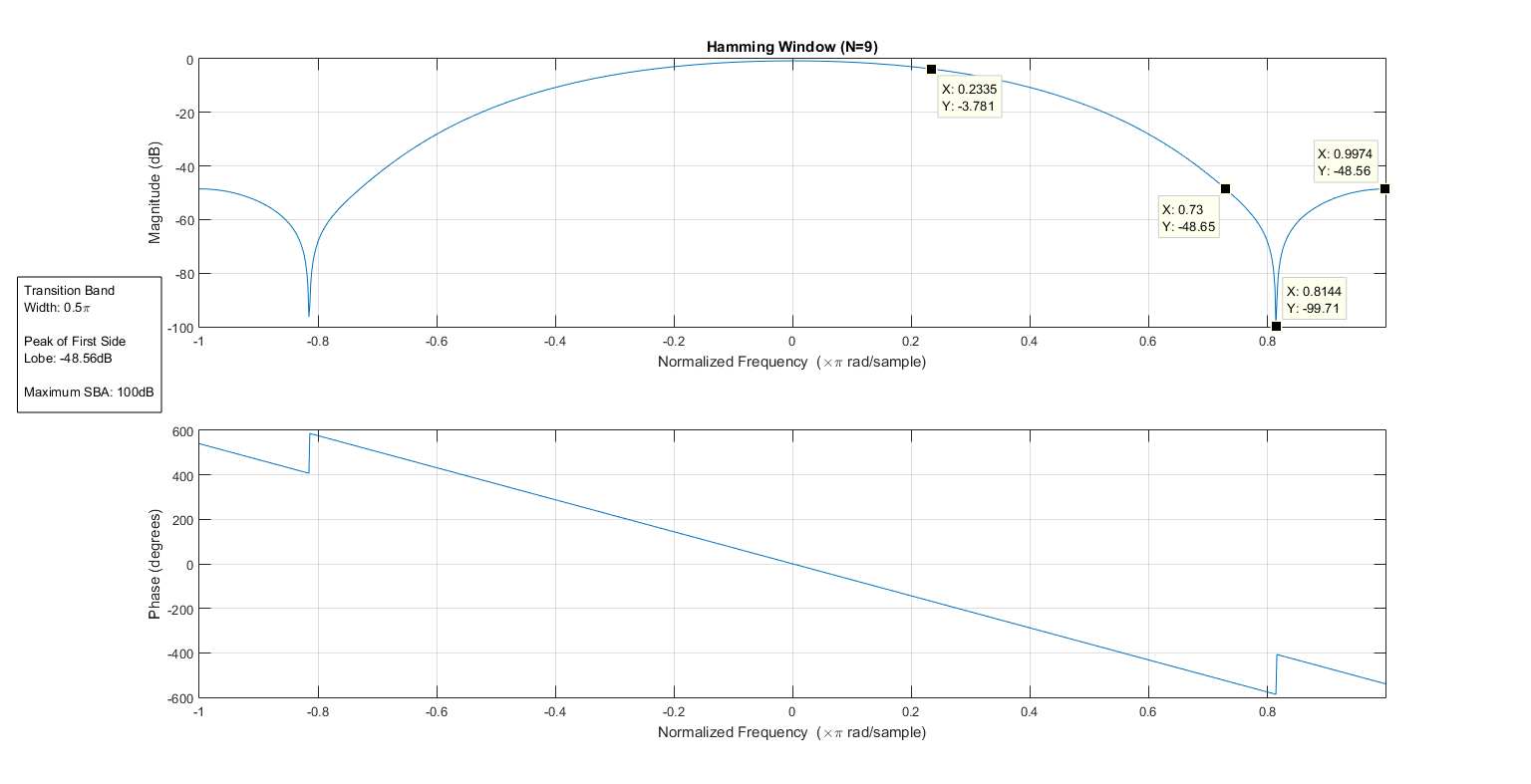
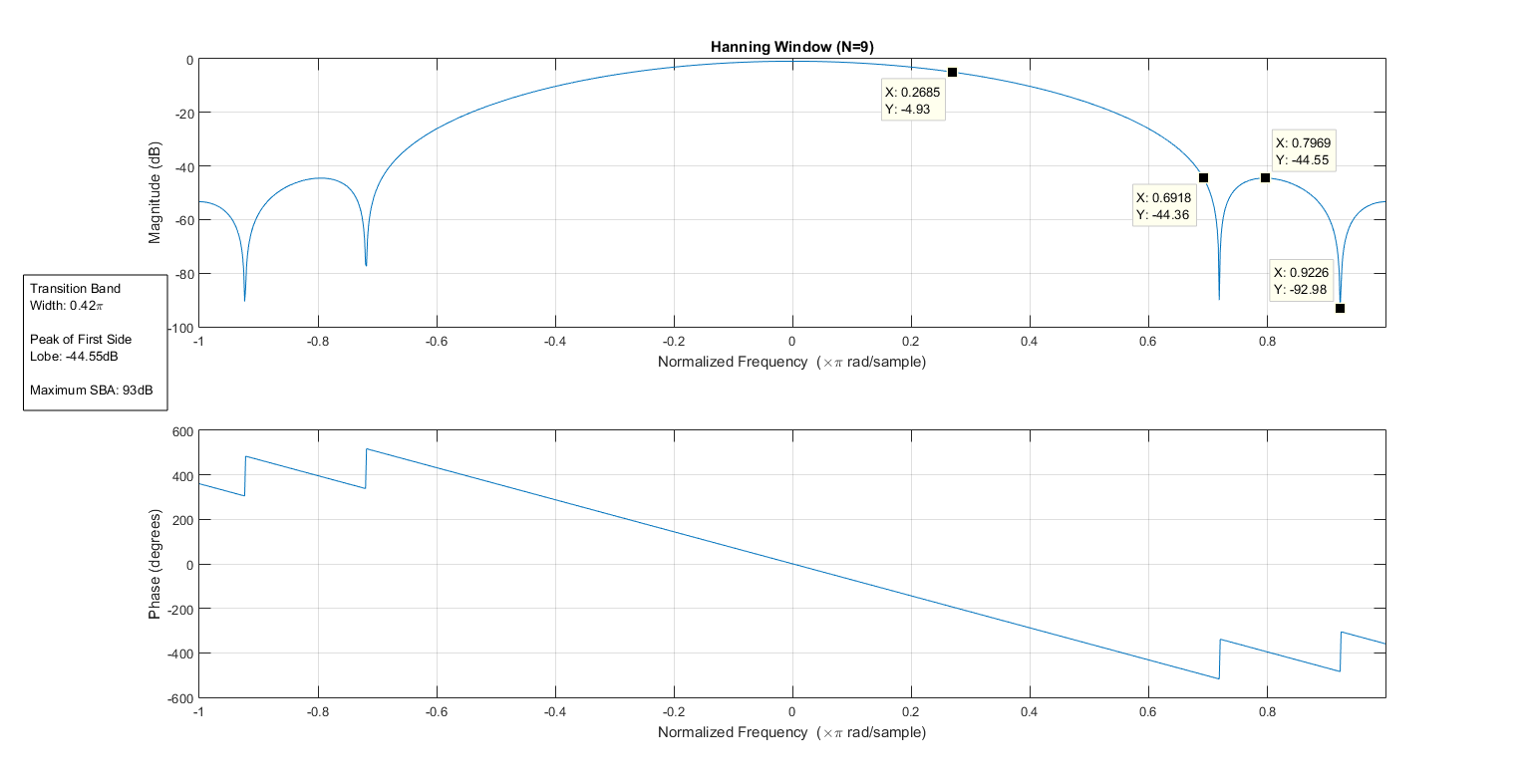
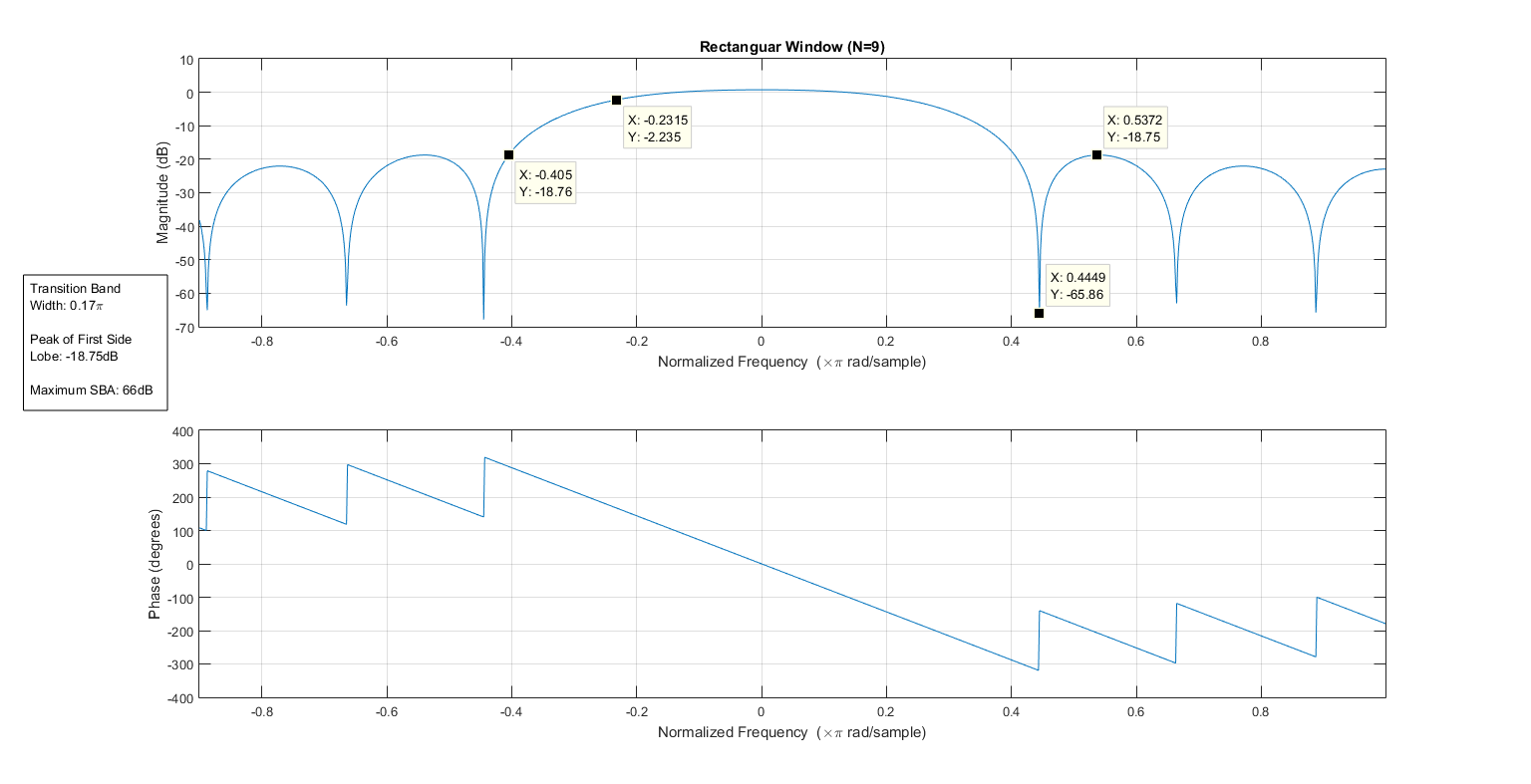
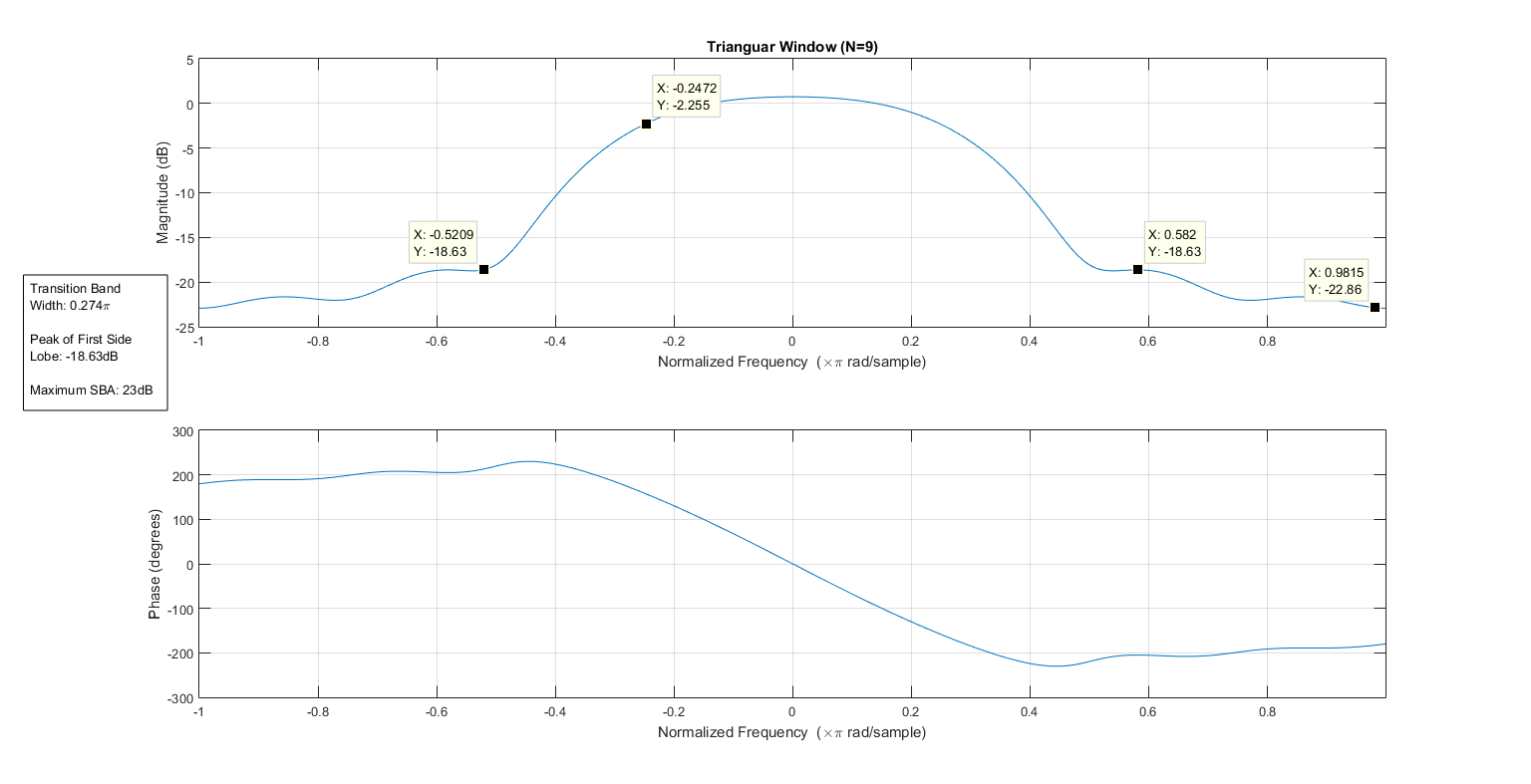
Once the desired time domain characteristics of the FIR Filter are established, the freqz function is used to find and plot the frequency response of the FIR filter.



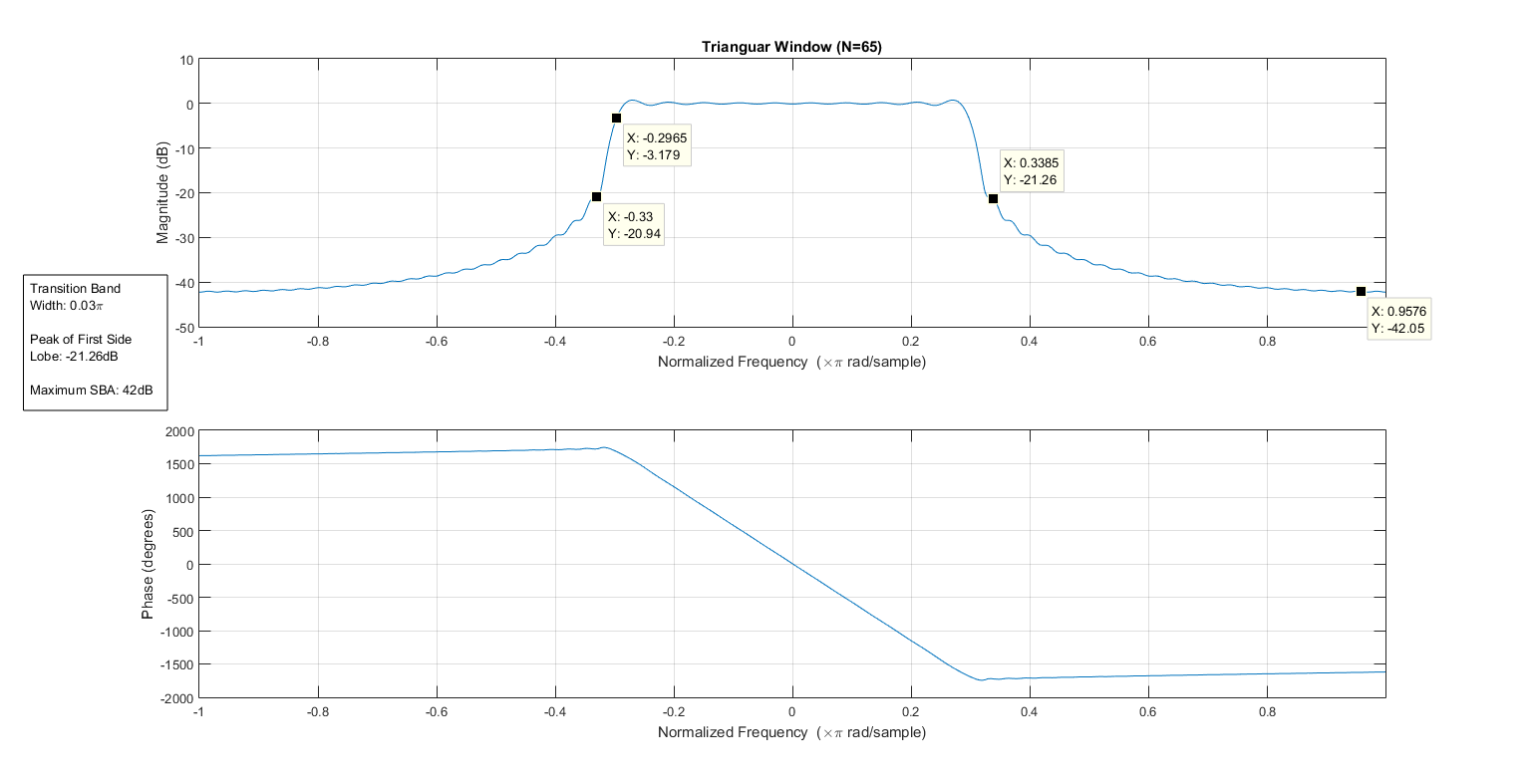
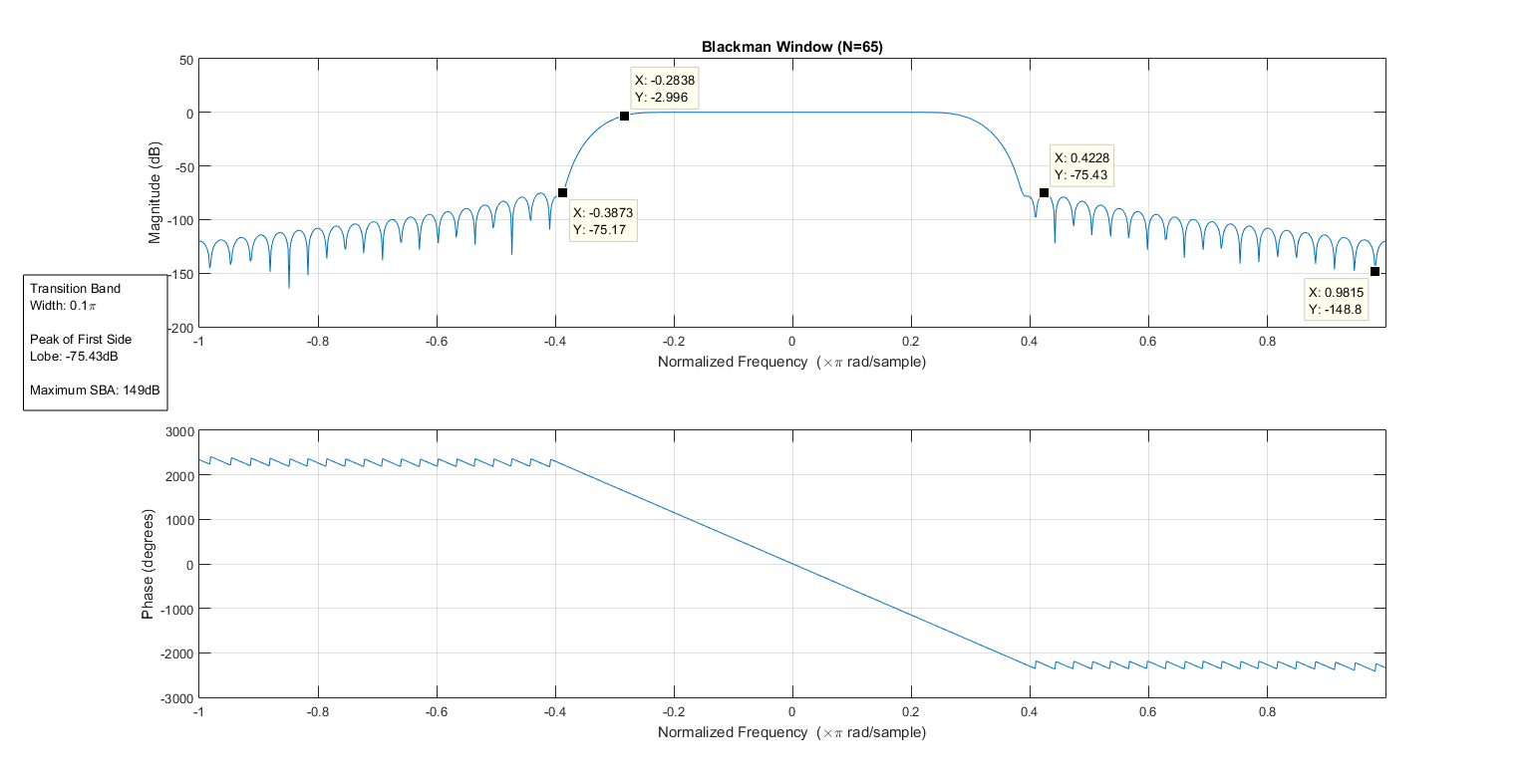
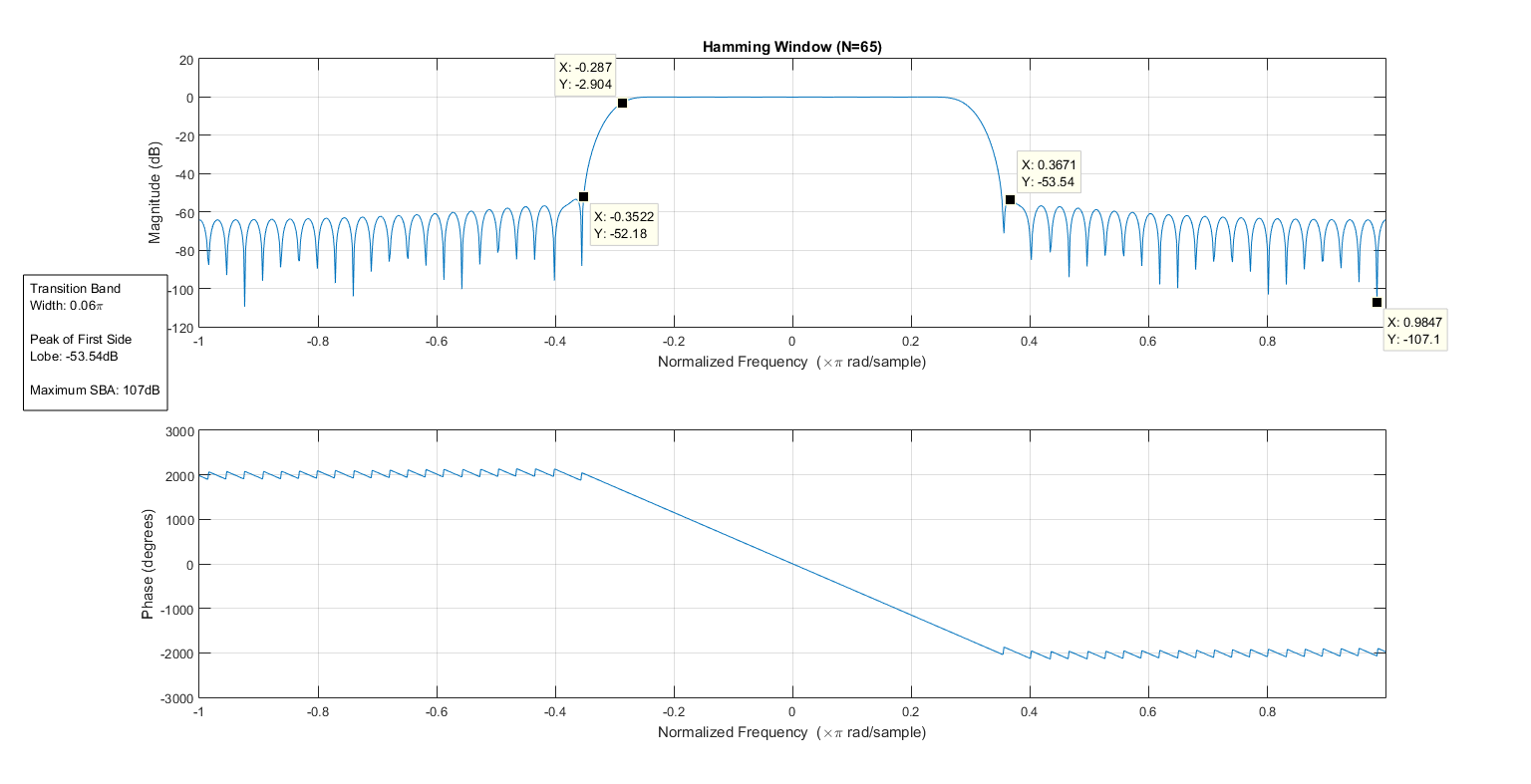
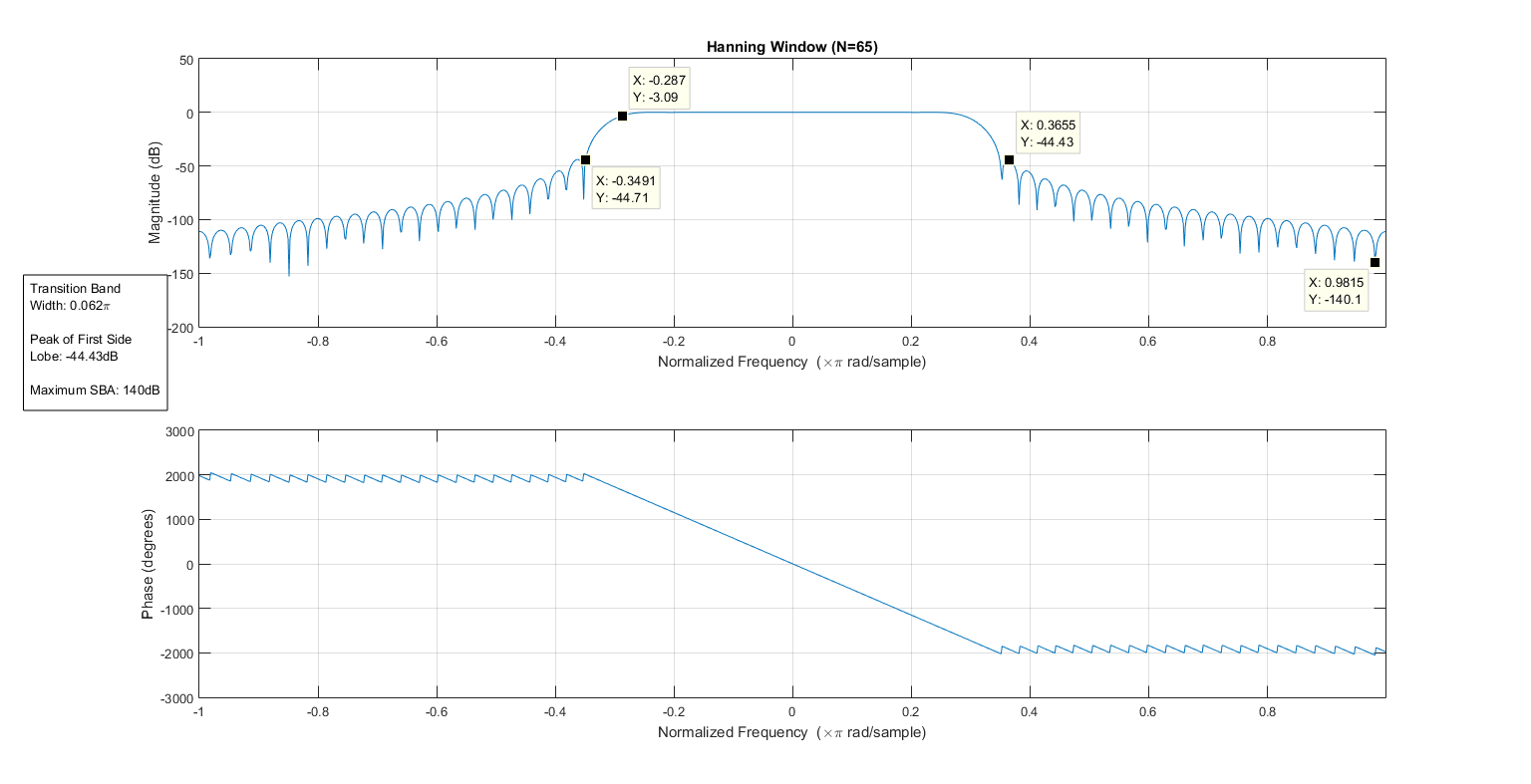
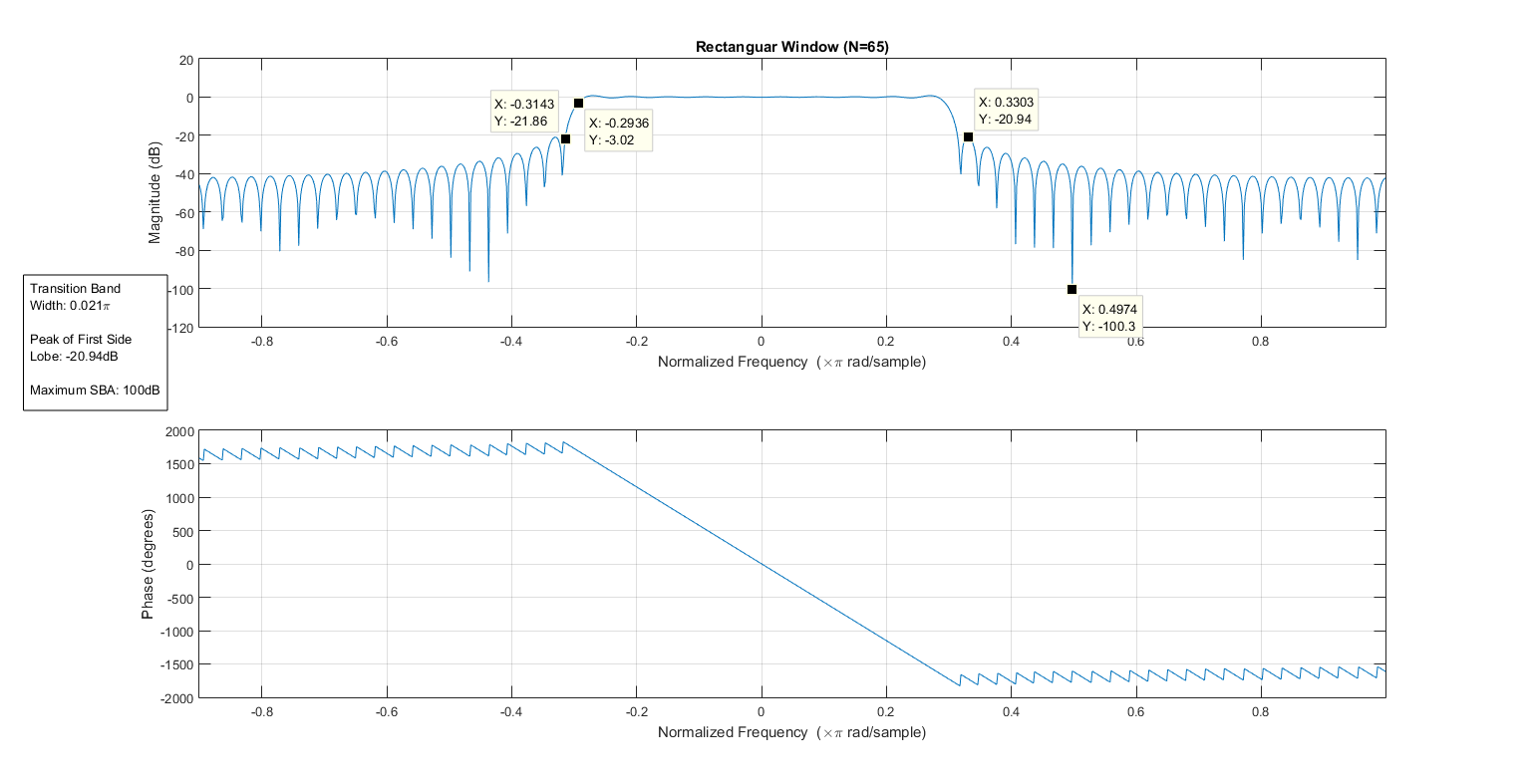
Impulse response of an ideal low-pass filter

# Frequency Reponses:

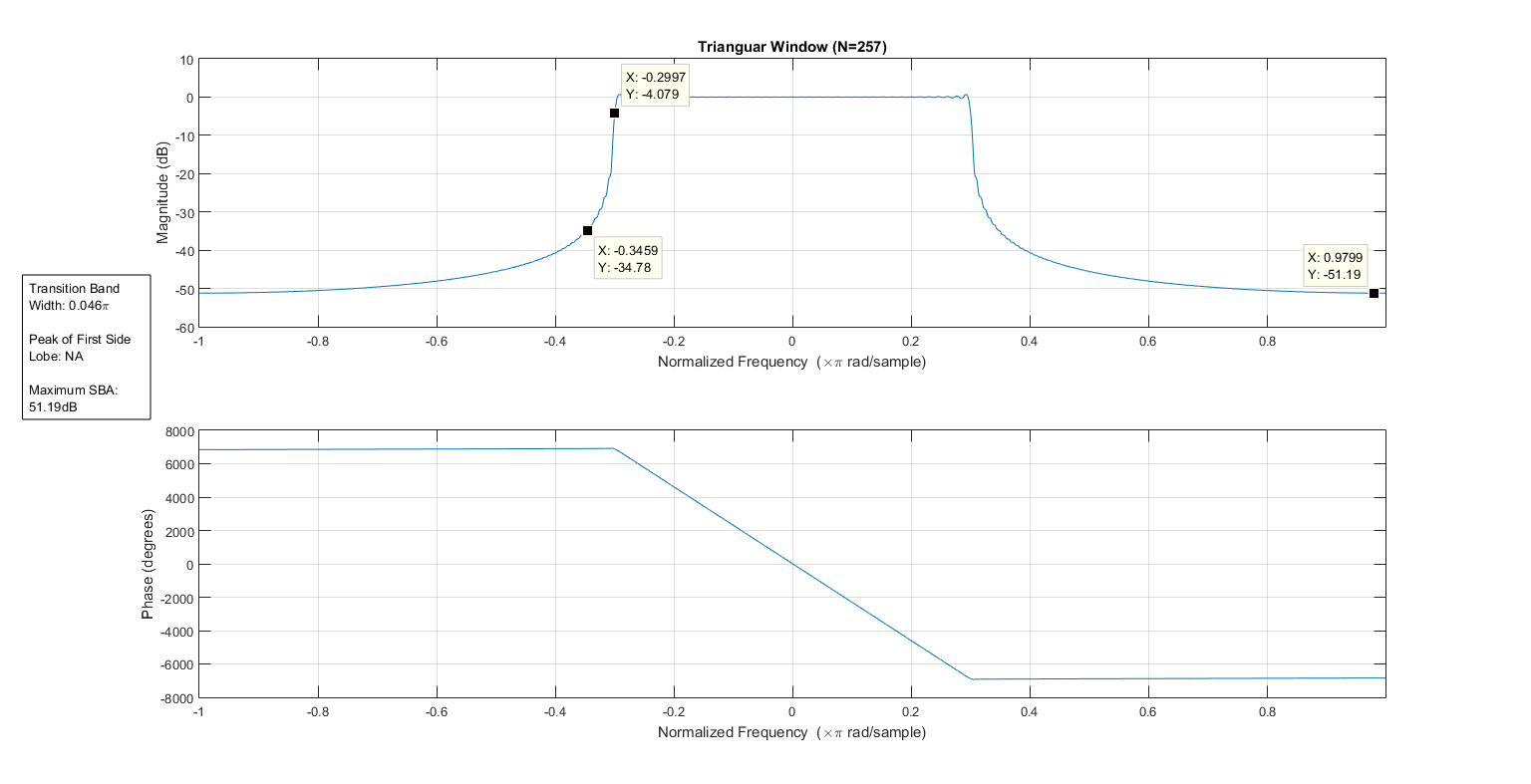
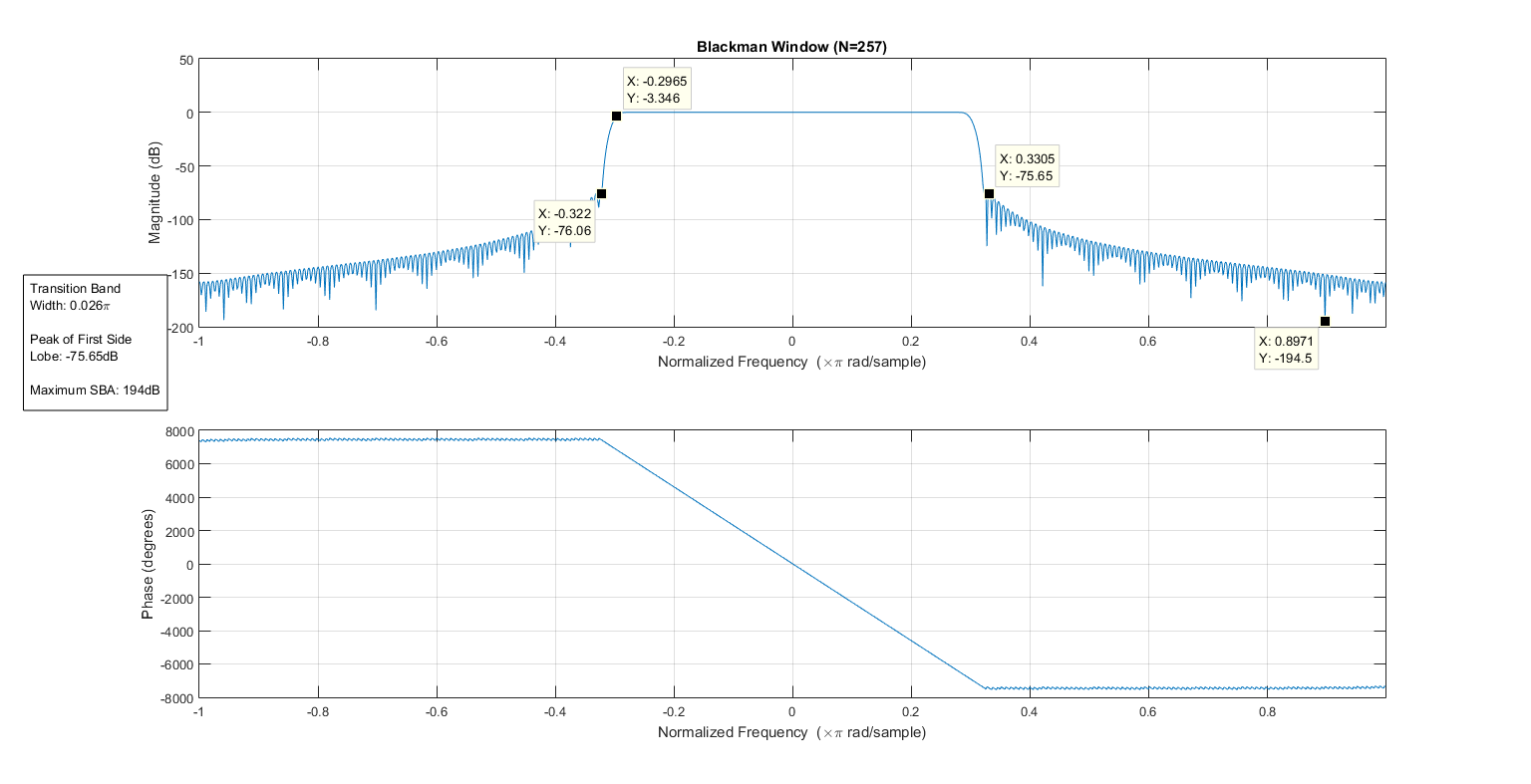
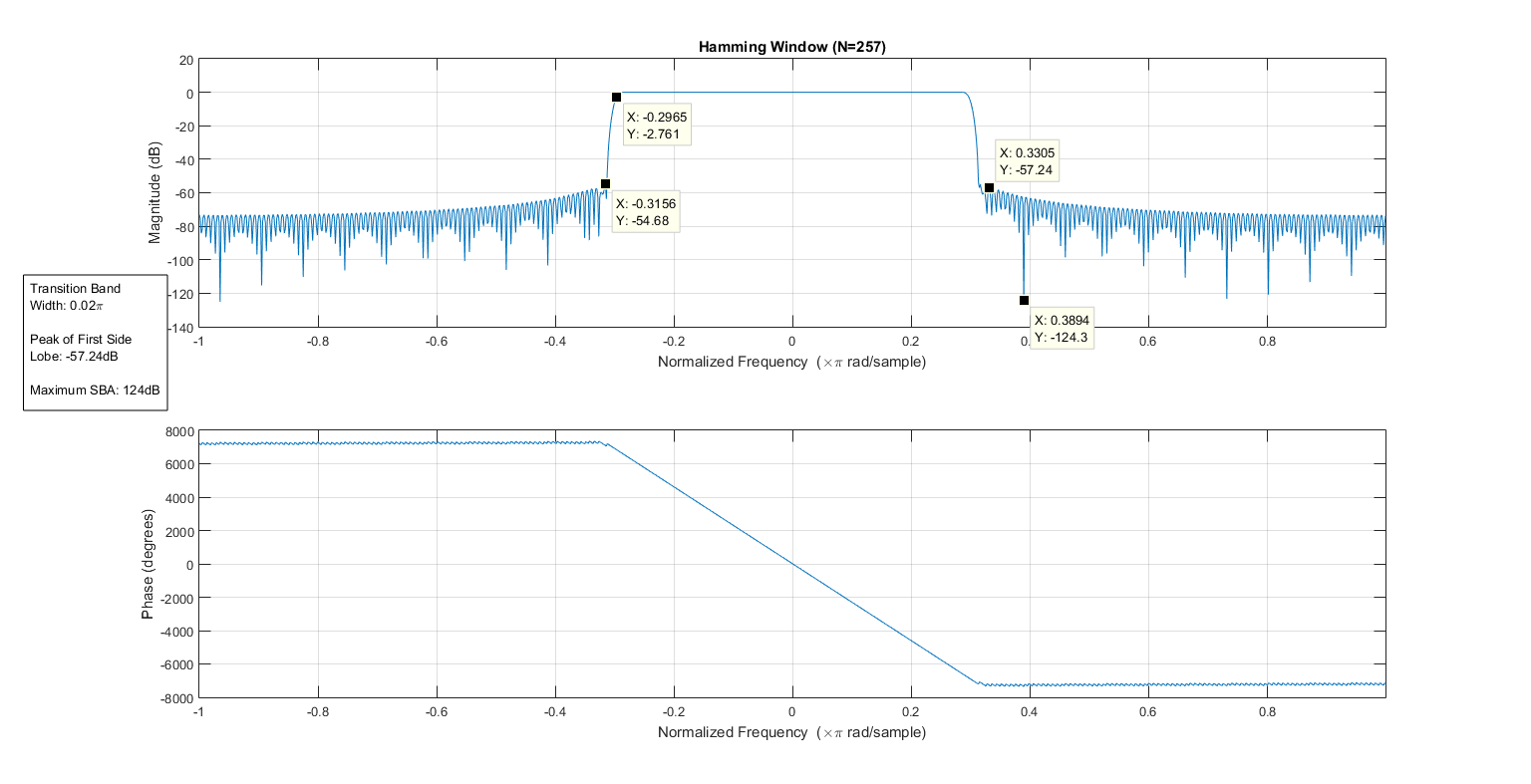
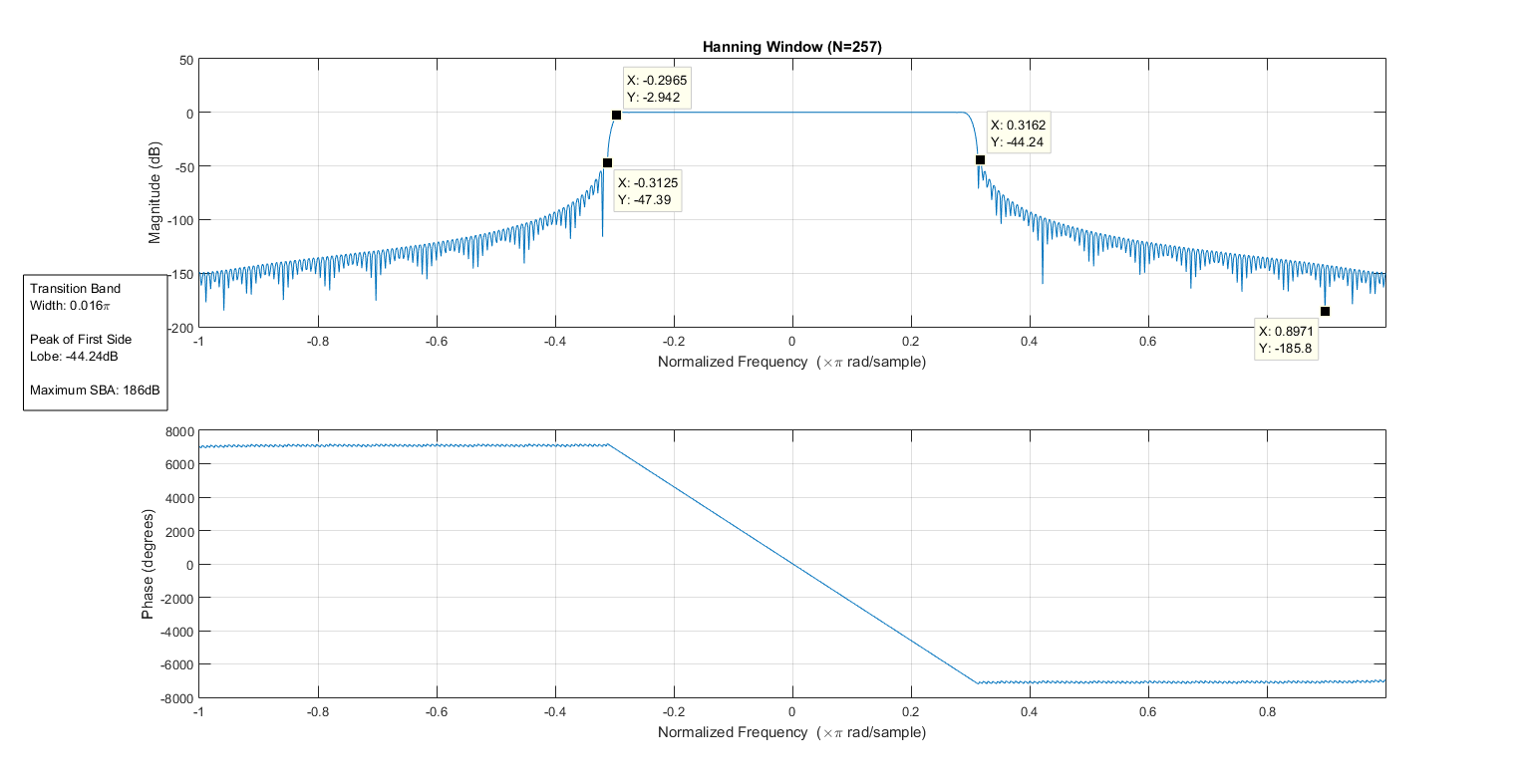
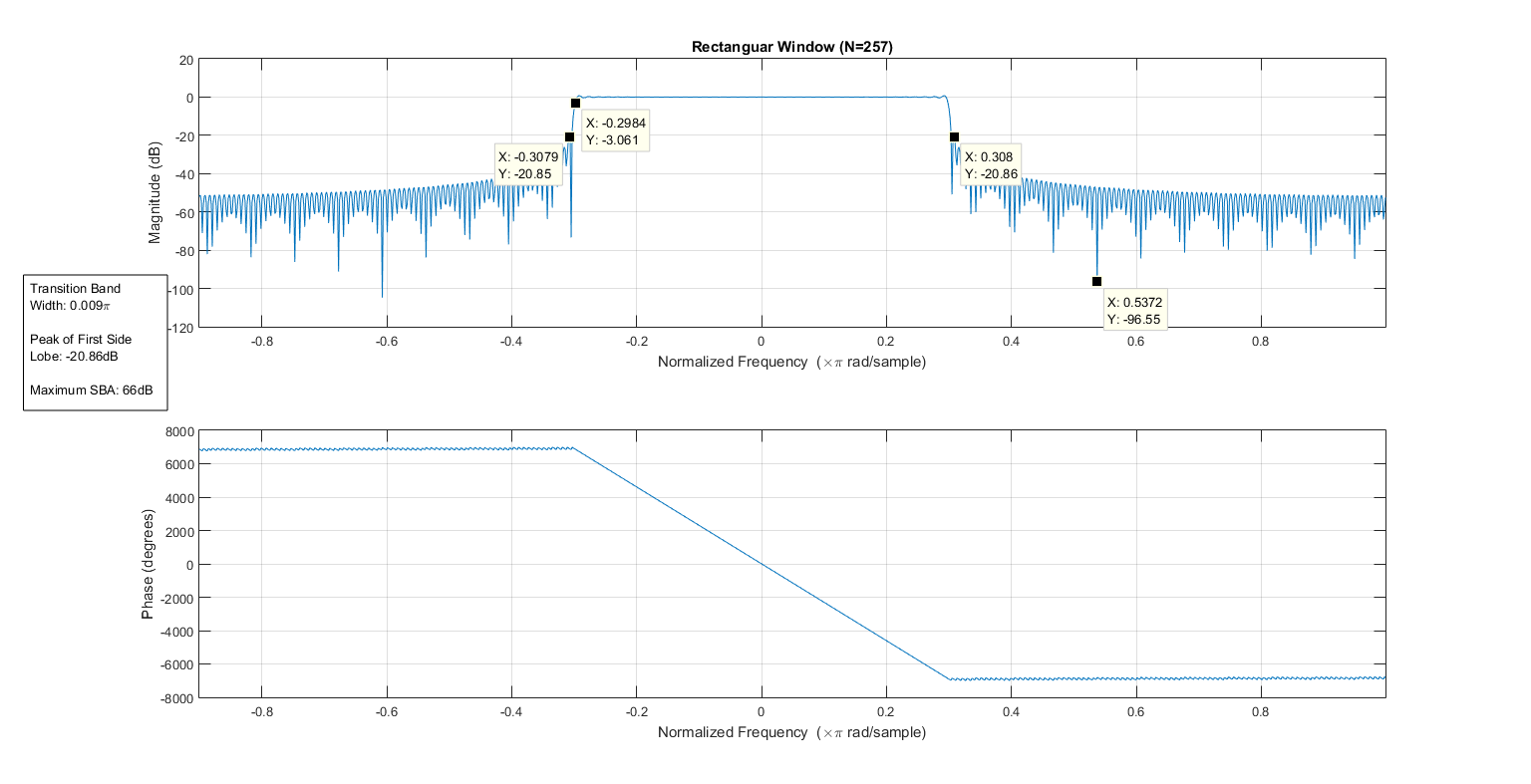
## N = 9



## N = 65



### N = 257



## 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 |

## Code:

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(['Rectanguar Window (N=' num2str(N) ')']);

%%

w = triWindow(N);

B2 = hd .\* w;

figure;

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

title(['Trianguar 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 = hammWindow(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;

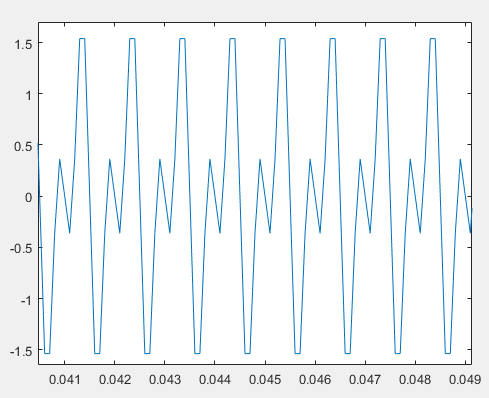
freqz(B5,1,-pi:0.005:pi);

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

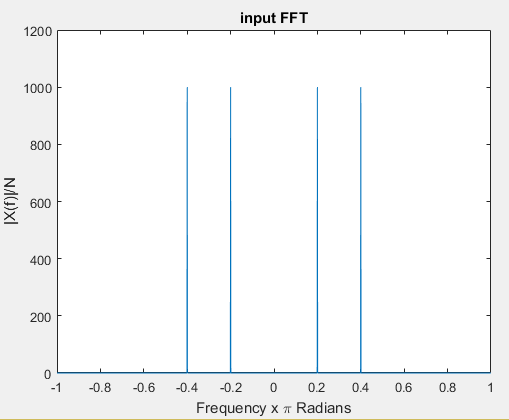
## Part 2:

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

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



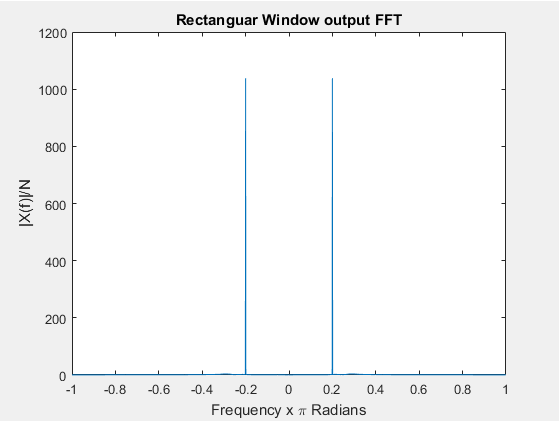
Generated Signal

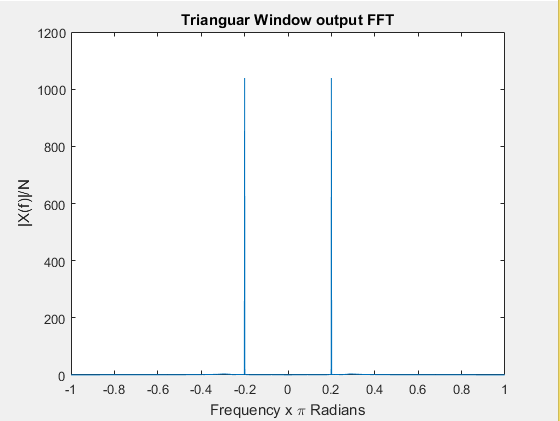


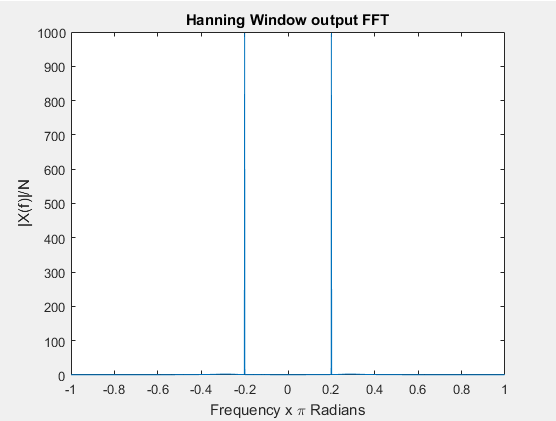
Input Signal DFT

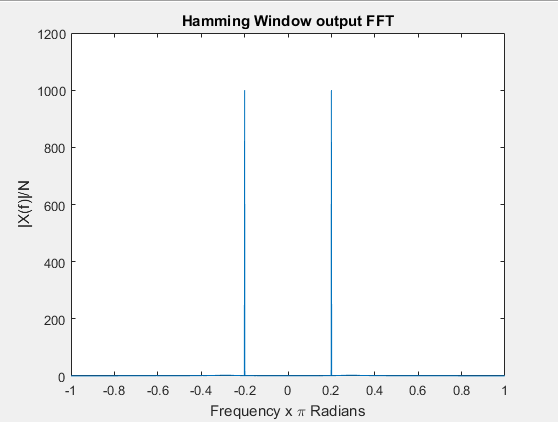
One of the components has a digital frequency of 0.2 while the other is at 0.4. Since the cut-off is at 0.3. One is in the passband while the other is in the stopband.

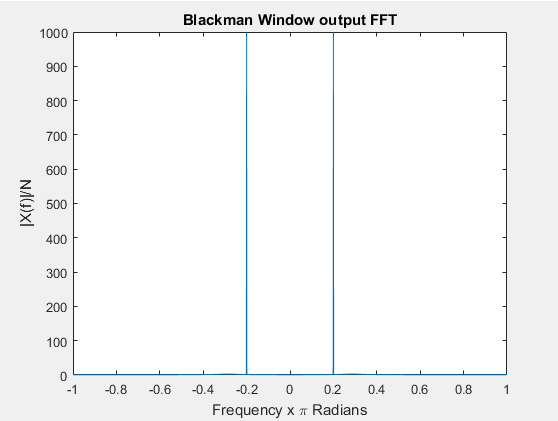
### Filtered Output:





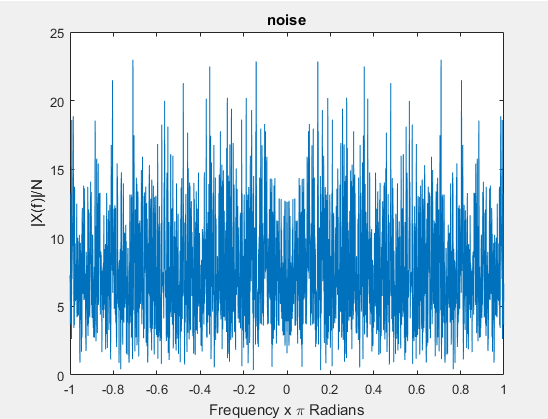




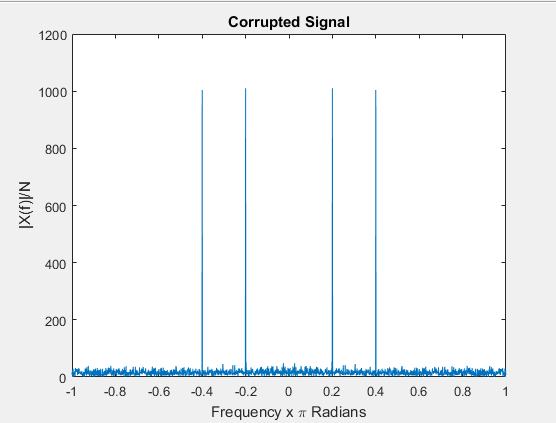


### Addition of Noise:

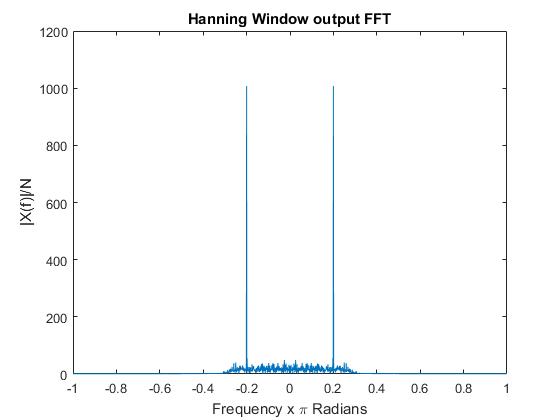
AWGN noise of a specific variance is now added to this signal.

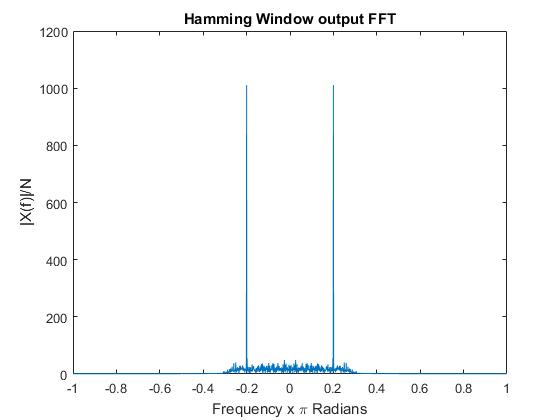
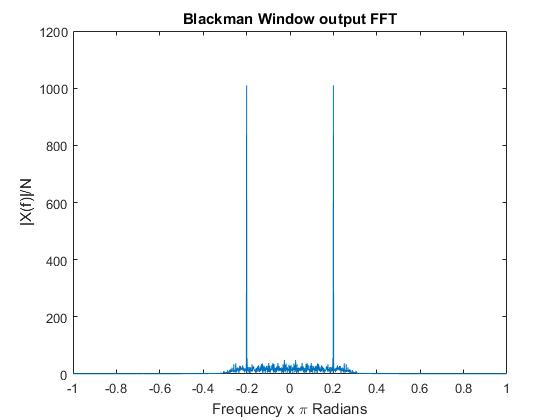


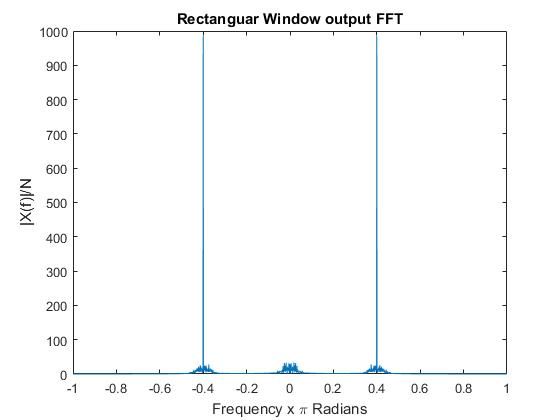
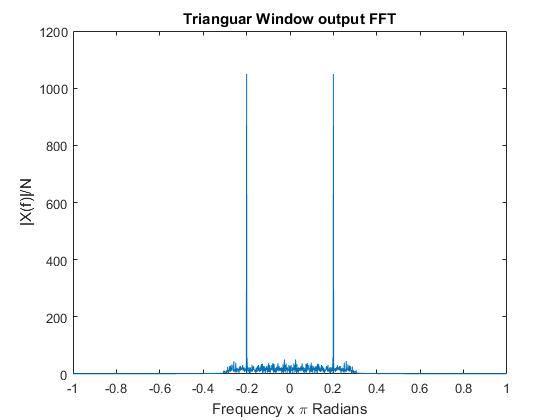
Power Spectral Density of Noise



## Filtered Output:







SNR Noisy input signal = 7.95dB

|  |  |
| --- | --- |
| Window | SNR (dB) |
| Rectangular | 17.69 |
| Triangular | 10.51 |
| Hamming | 10.47 |
| Hanning | 10.48 |
| Blackmann | 10.57 |

## Code:

%% 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 Fiters

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 Fiters

y1 = filtfilt(B1,1,x1);

plotdft(y1,Fs,'Rectanguar 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)