Digital Signal Processing Lab

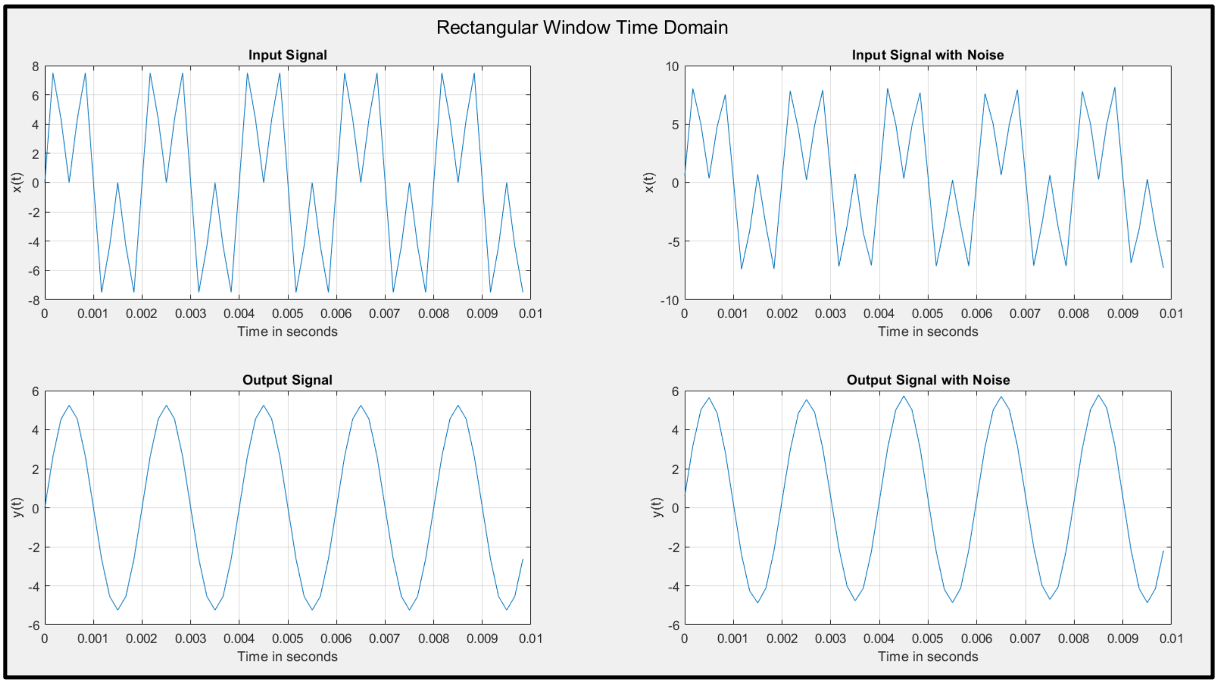
Experiment 2

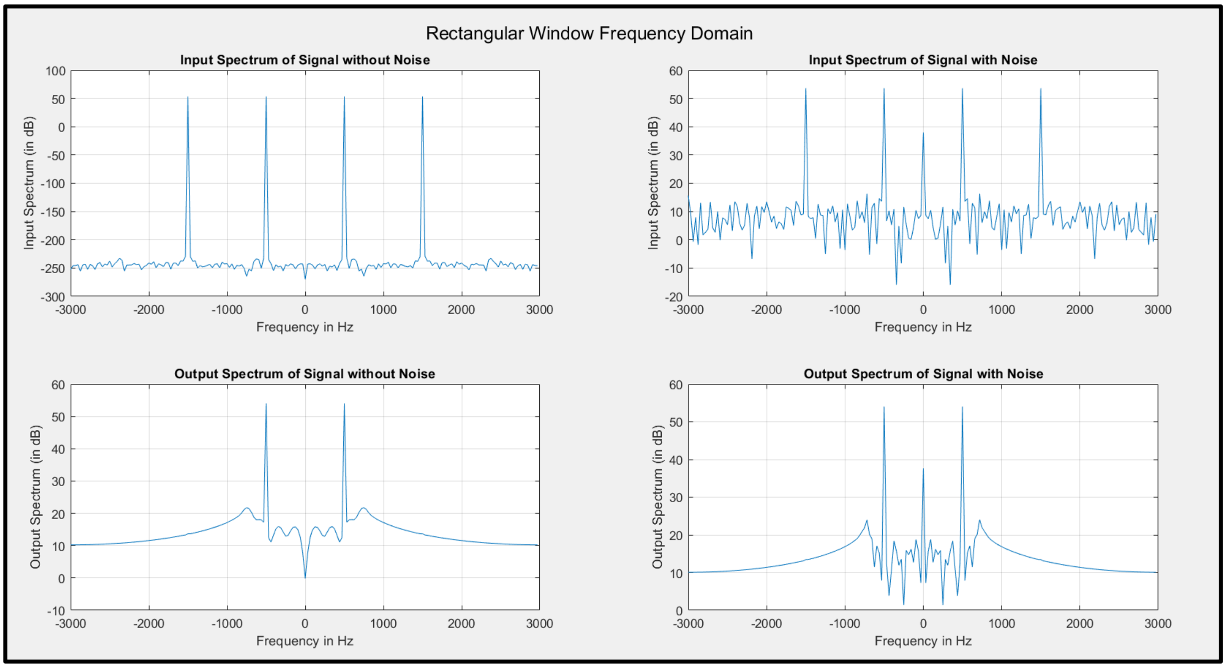
By Hardik Tibrewal (18EC10020)

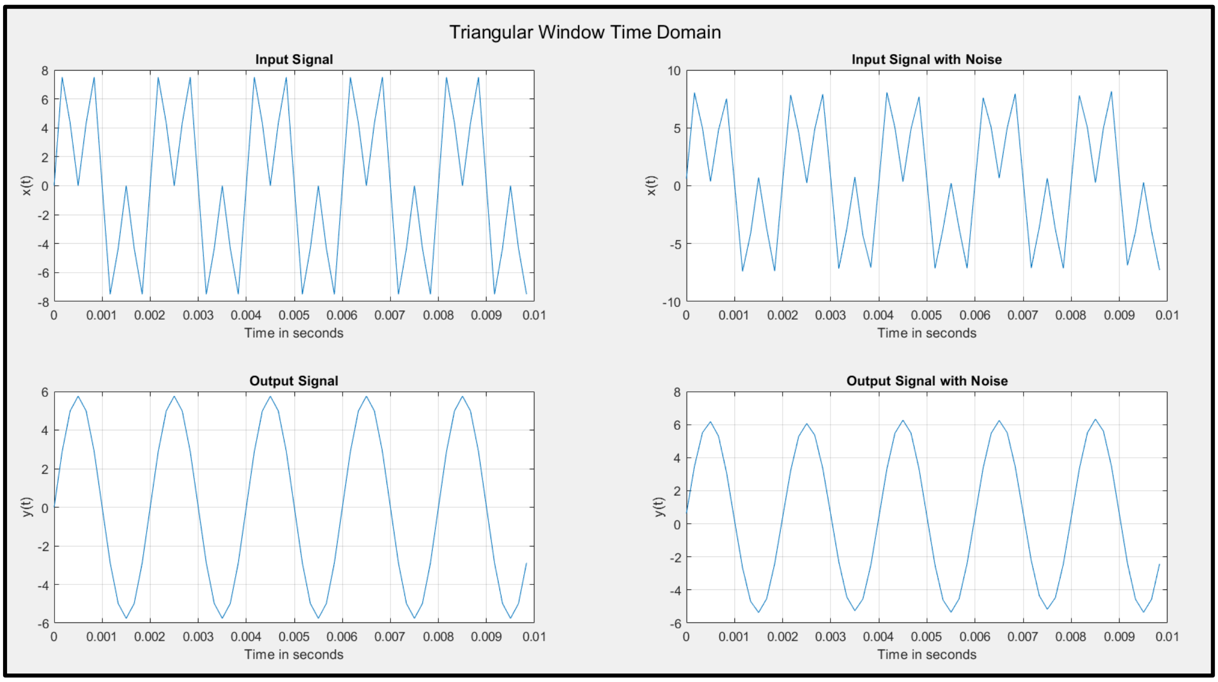
Aim:

To design various FIR Filters using windowing, and test them on noisy signals

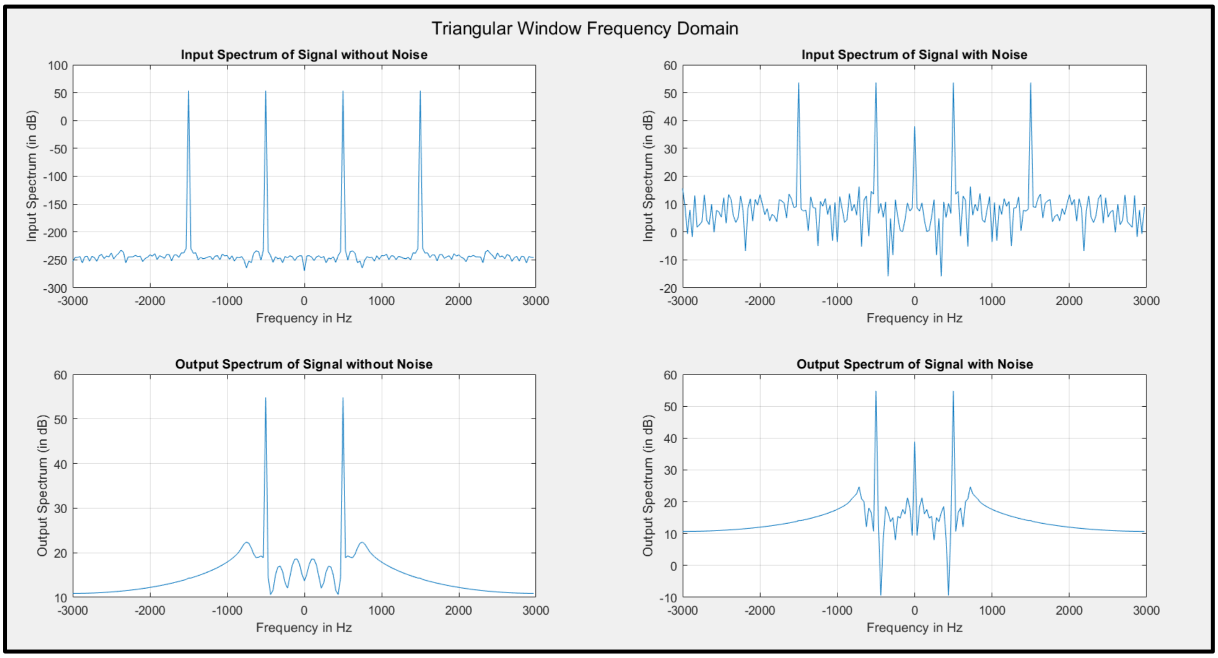
Plots: (N = 64 for filter)



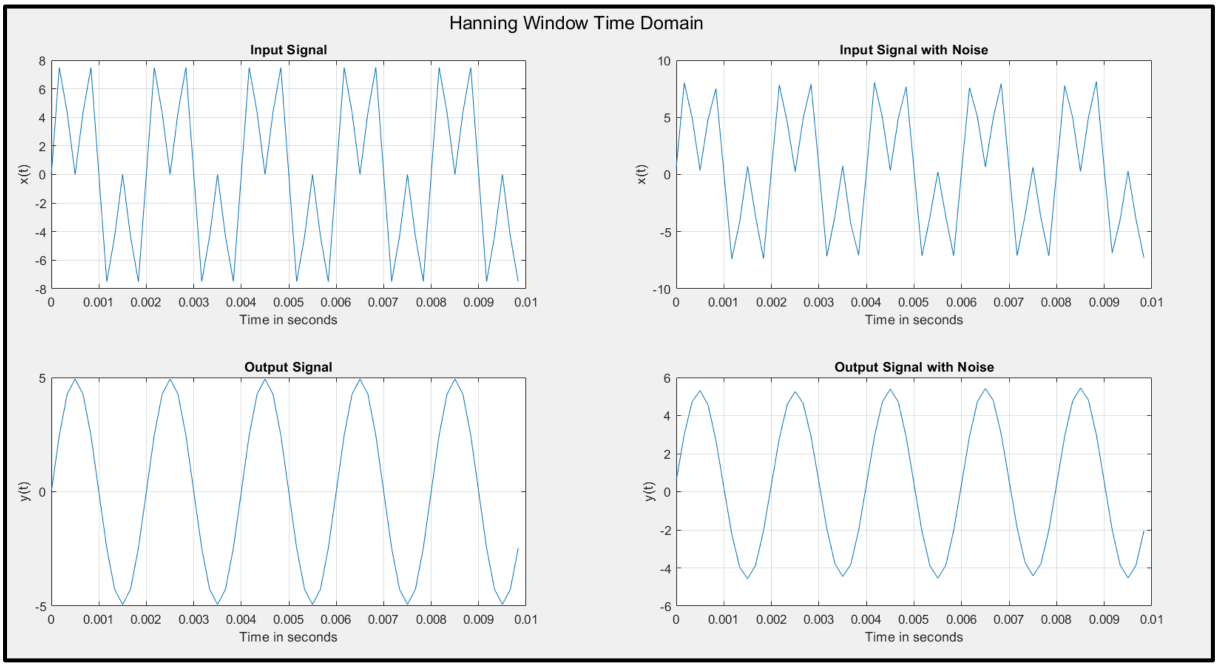


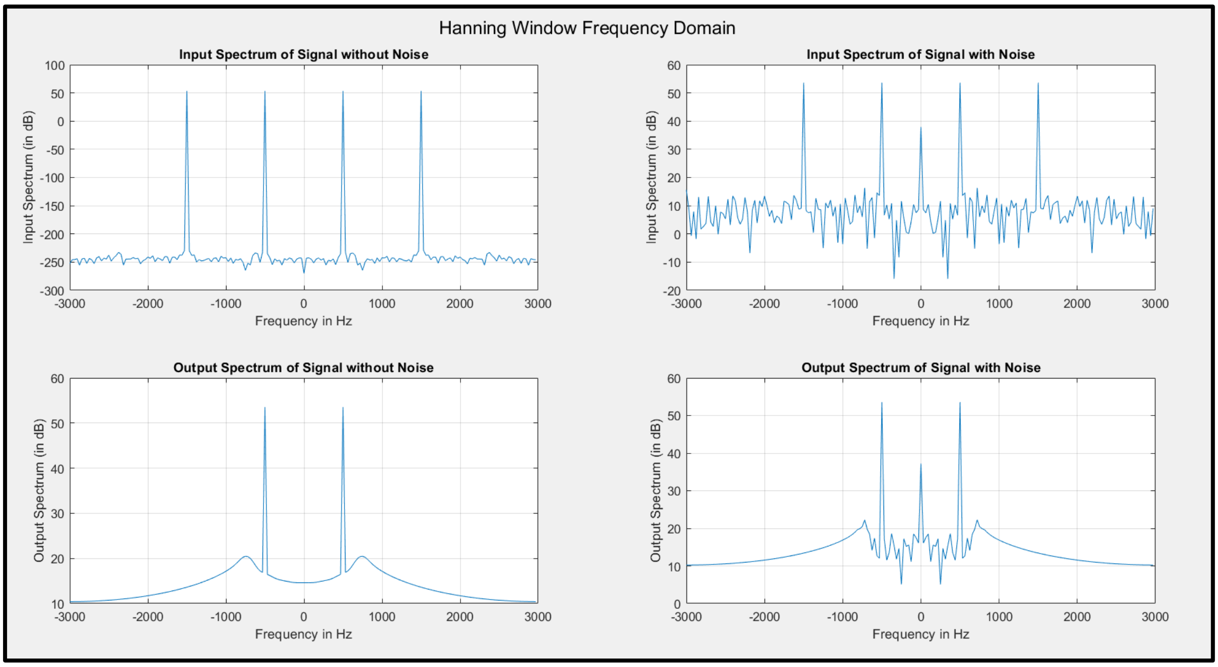


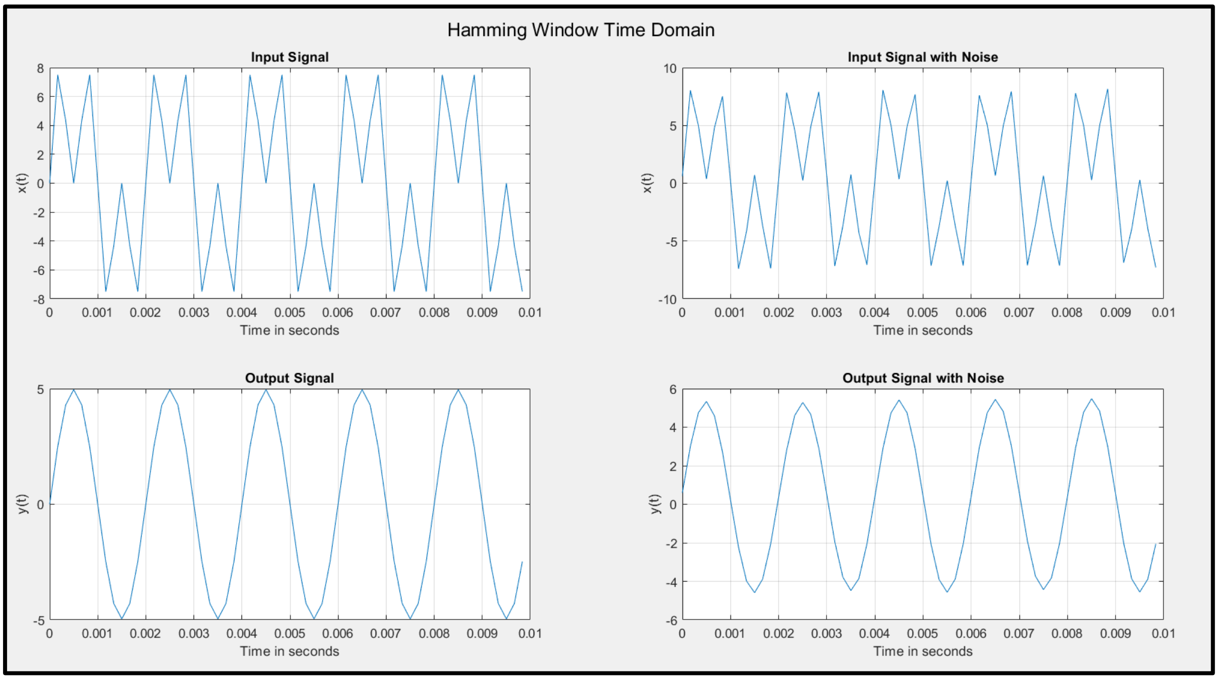
Triangular Window Time Domain
Input Signal
Input Signal with Noise
Output Signal
Output Signal with Noise
x(t)
y(t)
Time in seconds



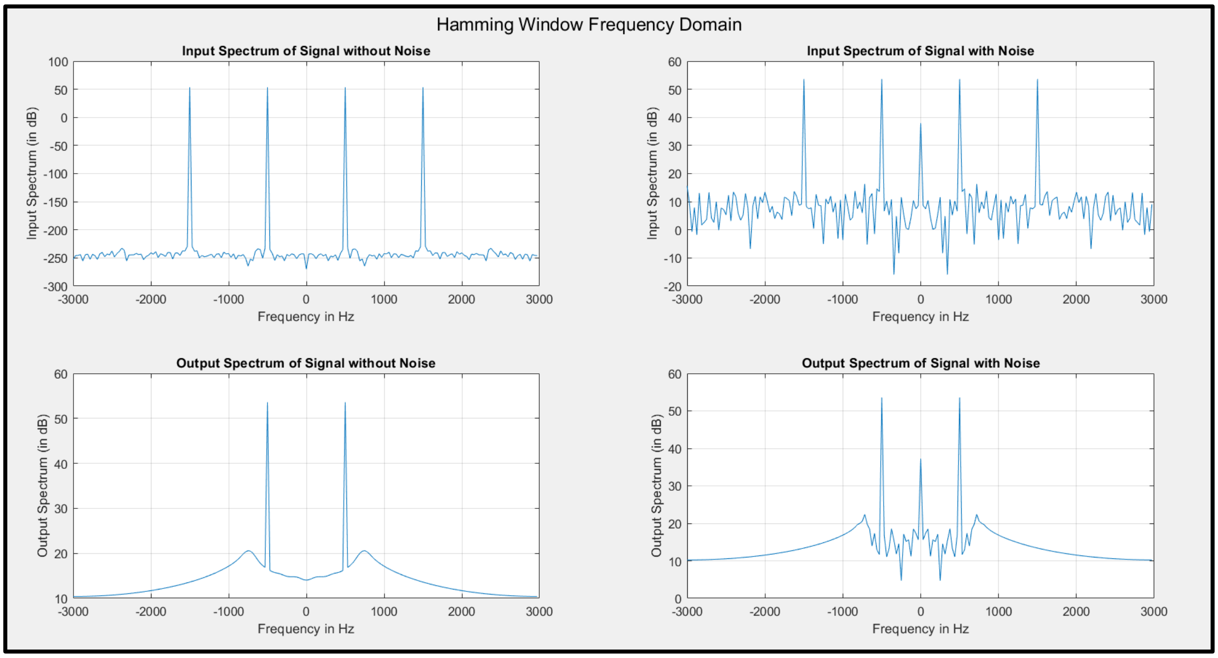
Triangular Window Frequency Domain
Input Spectrum of Signal without Noise
Input Spectrum of Signal with Noise
Output Spectrum of Signal without Noise
Output Spectrum of Signal with Noise
Input Spectrum (in dB)
Output Spectrum (in dB)
Frequency in Hz


Hanning Window Time Domain
Input Signal
Input Signal with Noise
Output Signal
Output Signal with Noise
x(t)
y(t)
Time in seconds

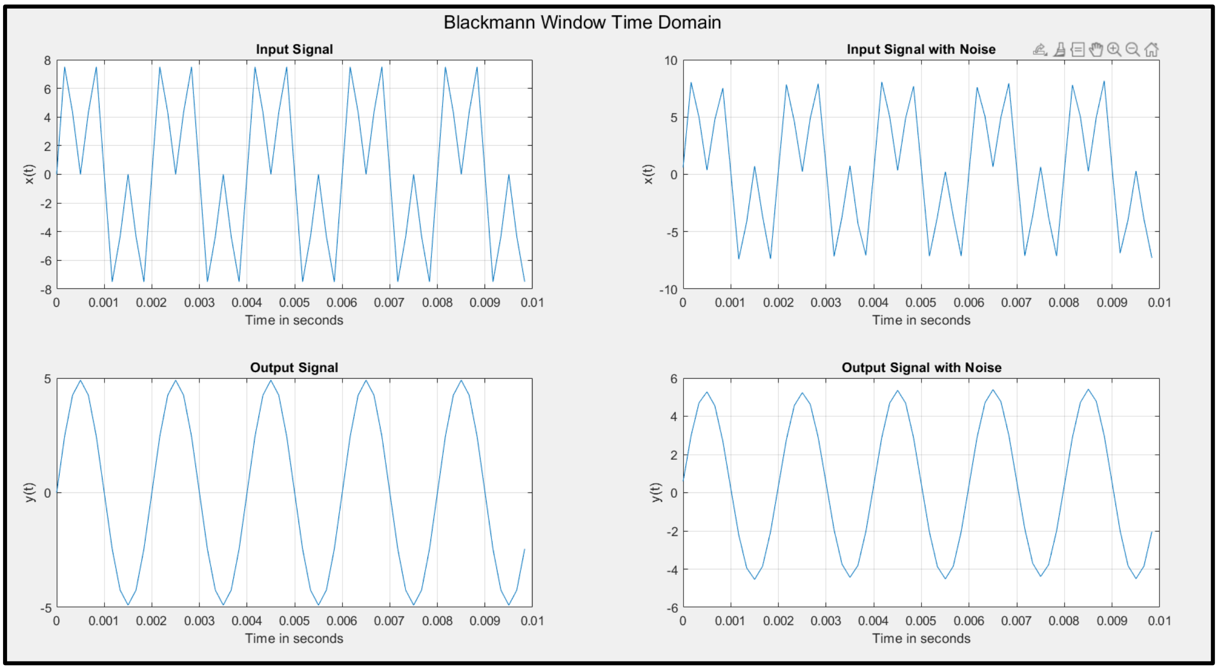

Hanning Window Frequency Domain
Input Spectrum of Signal without Noise
Input Spectrum of Signal with Noise
Output Spectrum of Signal without Noise
Output Spectrum of Signal with Noise
Input Spectrum (in dB)
Output Spectrum (in dB)
Frequency in Hz

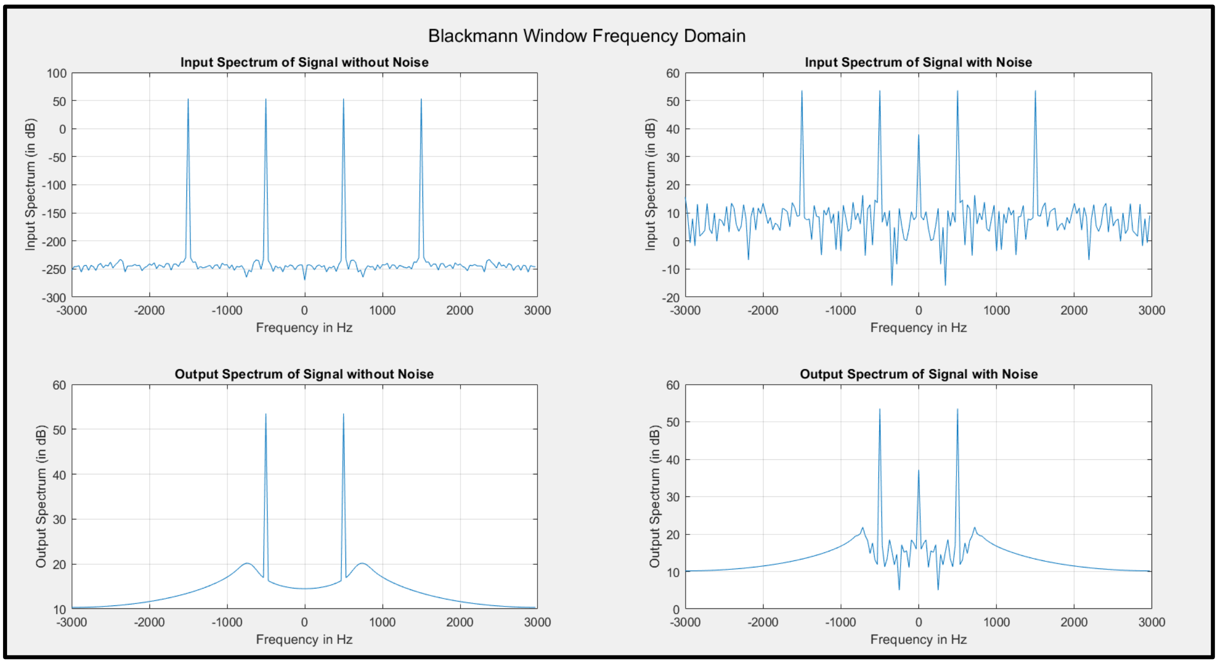


Hamming Window Time Domain
Input Signal
Input Signal with Noise
Output Signal
Output Signal with Noise
x(t)
y(t)
Time in seconds



Hamming Window Frequency Domain
Input Spectrum of Signal without Noise
Input Spectrum of Signal with Noise
Output Spectrum of Signal without Noise
Output Spectrum of Signal with Noise
Input Spectrum (in dB)
Output Spectrum (in dB)
Frequency in Hz


Blackmann Window Time Domain
Input Signal
x(t)
Time in seconds
Input Signal with Noise
Output Signal
y(t)
Output Signal with Noise


Blackmann Window Frequency Domain
Input Spectrum of Signal without Noise
Input Spectrum (in dB)
Frequency in Hz
Input Spectrum of Signal with Noise
Output Spectrum of Signal without Noise
Output Spectrum (in dB)
Output Spectrum of Signal with Noise

```
Command Window
Input SNR(in dB) for Rectangular window = 21.1327
Output SNR(in dB) for Rectangular window = 19.5595
Input SNR(in dB) for Triangular window = 21.1327
Output SNR(in dB) for Triangular window = 19.7095
Input SNR(in dB) for Hanning window = 21.1327
Output SNR(in dB) for Hanning window = 19.0582
Input SNR(in dB) for Hamming window = 21.1327
Output SNR(in dB) for Hamming window = 19.0991
Input SNR(in dB) for Blackmann window = 21.1327
Output SNR(in dB) for Blackmann window = 19.0789
fx >>
```

Code:

```
clc
clear all
close all

N = 64;
k = floor((N-1)/2);
n = 0:1:(N-1);

wc = 0.8;
w = -pi:1/2000:pi;

hd = zeros(1, N);
for ii = 1:N
    if ii == k
        hd(ii) = wc/pi;
    else
        hd(ii) = sin(wc*(ii-k))/(pi*(ii-k));
    end
end

rectangular = ones(1, N);
triangular = 1 - 2*(n-(N-1)/2)/(N-1);
hanning = 0.5 - 0.5*cos((2*pi/(N-1))*n);
hamming = 0.54 - 0.46*cos((2*pi/(N-1))*n);
blackmann = 0.42 - 0.5*cos((2*pi/(N-1))*n) + 0.08*cos((4*pi/(N-1))*n);

h_rect = hd.*rectangular;
h_trig = hd.*triangular;
h_hann = hd.*hanning;
h_hamm = hd.*hamming;
h_black = hd.*blackmann;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%

f_pass = 500;
f_stop = 1500;
fs = 6000;
```

```
t = 0:1/fs:(3*N-1)/fs;
noise = rand(1, 3*N);
x = 5*sin(2*pi*f_pass*t) + 5*sin(2*pi*f_stop*t);
add_noise = (max(x)/10)*noise/abs(max(noise));
noisy_x = x + add_noise;
f_eq = -3000:2000/N:3000-2000/N;

h_matrix = [h_rect; h_trig; h_hann; h_hamm; h_black];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%
for ii=1:5
    if ii == 1
        name = "Rectangular";
    elseif ii == 2
        name = "Triangular";
    elseif ii == 3
        name = "Hanning";
    elseif ii == 4
        name = "Hamming";
    else
        name = "Blackmann";
    end
    y = filtfilt(h_matrix(ii,:), 1, x);
    y_n = filtfilt(h_matrix(ii,:), 1, noisy_x);

    figure();
    sgtitle(name+" Window Time Domain");
    subplot(221);
    plot(t(1:floor(15*fs/f_stop)),x(1:floor(15*fs/f_stop)));
    grid on
    xlabel('Time in seconds'); ylabel('x(t)'); title('Input Signal');
    subplot(222);
    plot(t(1:floor(15*fs/f_stop)),noisy_x(1:floor(15*fs/f_stop)));
    grid on
    xlabel('Time in seconds'); ylabel('x(t)'); title('Input Signal with Noise');
    subplot(223);
    plot(t(1:floor(15*fs/f_stop)),y(1:floor(15*fs/f_stop)));
    grid on
    xlabel('Time in seconds'); ylabel('y(t)'); title('Output Signal');
    subplot(224);
    plot(t(1:floor(15*fs/f_stop)),y_n(1:floor(15*fs/f_stop)));
    grid on
    xlabel('Time in seconds'); ylabel('y(t)'); title('Output Signal with Noise');

    figure();
    sgtitle(name+" Window Frequency Domain");
    subplot(2,2,1)
    plot(f_eq, 20*log10(abs(fftshift(fft(x)))));
    xlabel('Frequency in Hz'); ylabel('Input Spectrum (in dB)'); title('Input Spectrum of Signal without
Noise')
    grid on;

    subplot(2,2,2)
    plot(f_eq, 20*log10(abs(fftshift(fft(noisy_x)))));
```

```
    xlabel('Frequency in Hz'); ylabel('Input Spectrum (in dB)'); title('Input Spectrum of Signal with Noise')
    grid on;
    subplot(2,2,3)
    plot(f_eq, 20*log10(abs(fftshift(fft(y)))));
    xlabel('Frequency in Hz'); ylabel('Output Spectrum (in dB)'); title('Output Spectrum of Signal without Noise')
    grid on;

    subplot(2,2,4)
    plot(f_eq, 20*log10(abs(fftshift(fft(y_n)))));
    xlabel('Frequency in Hz'); ylabel('Output Spectrum (in dB)'); title('Output Spectrum of Signal with Noise')
    grid on;

    disp("Input SNR(in dB) for "+name+" window = "+snr(x, noisy_x-x))
    disp("Output SNR(in dB) for "+name+" window = "+snr(y, y_n-y))
end
```