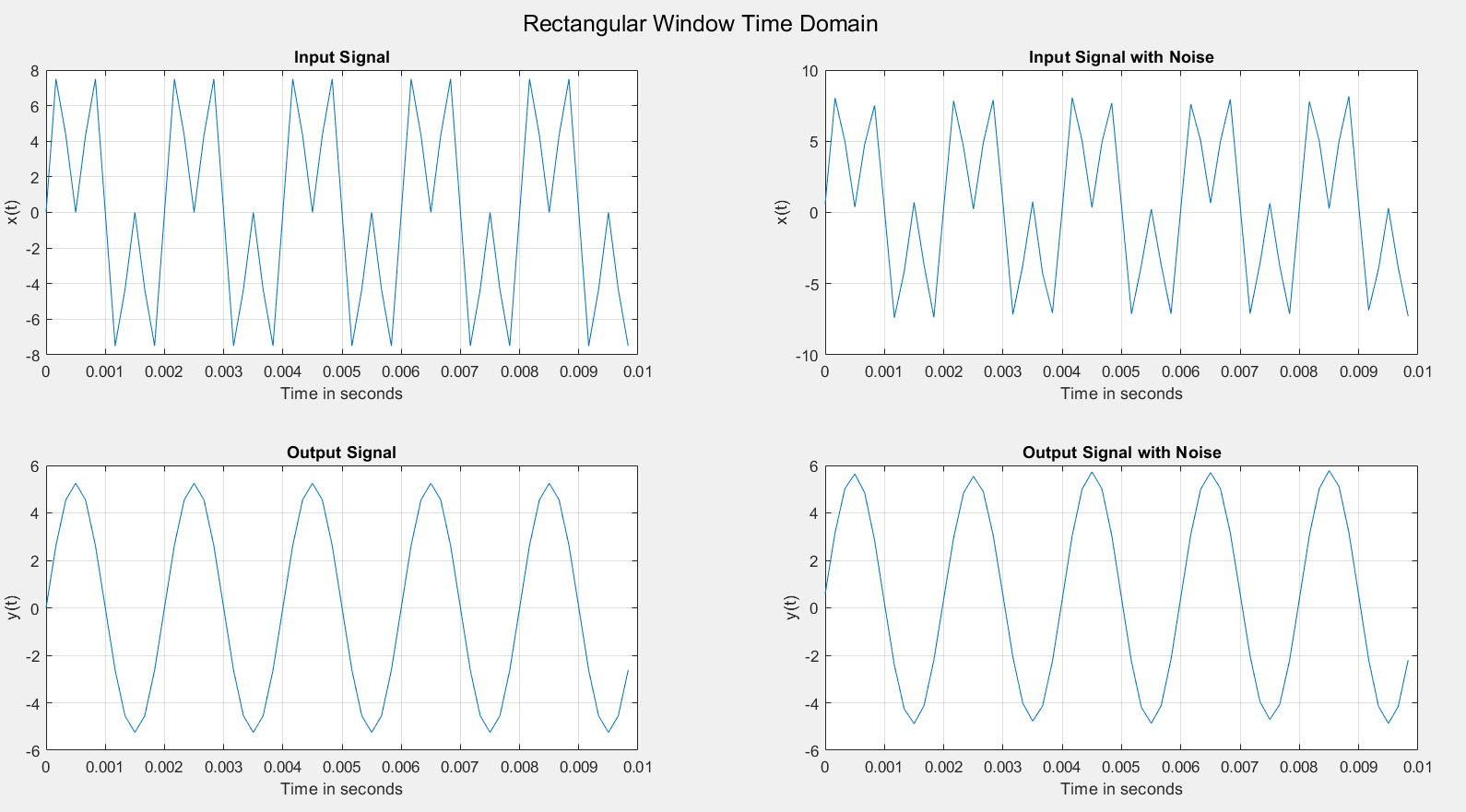
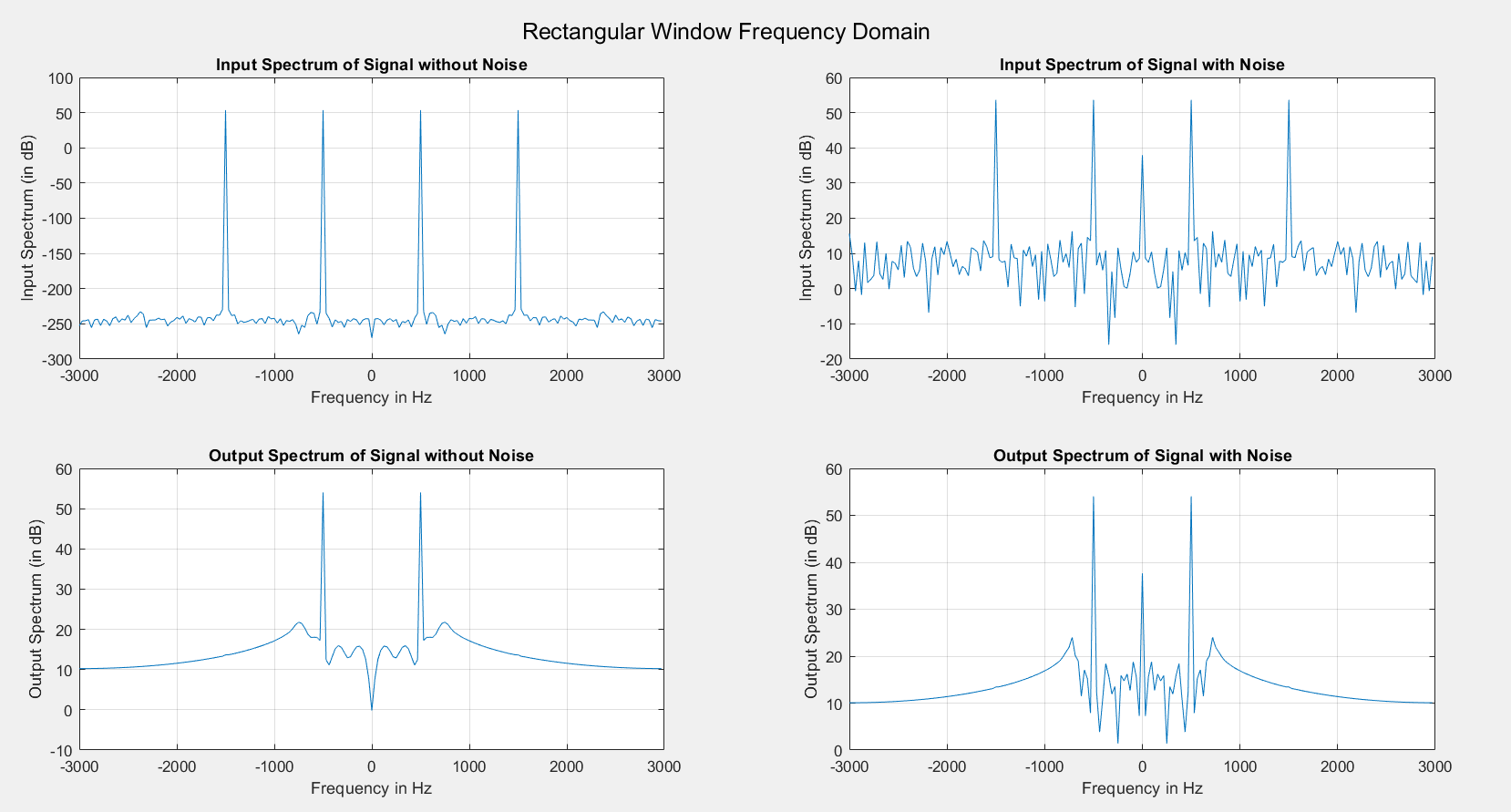
Digital Signal Processing Lab

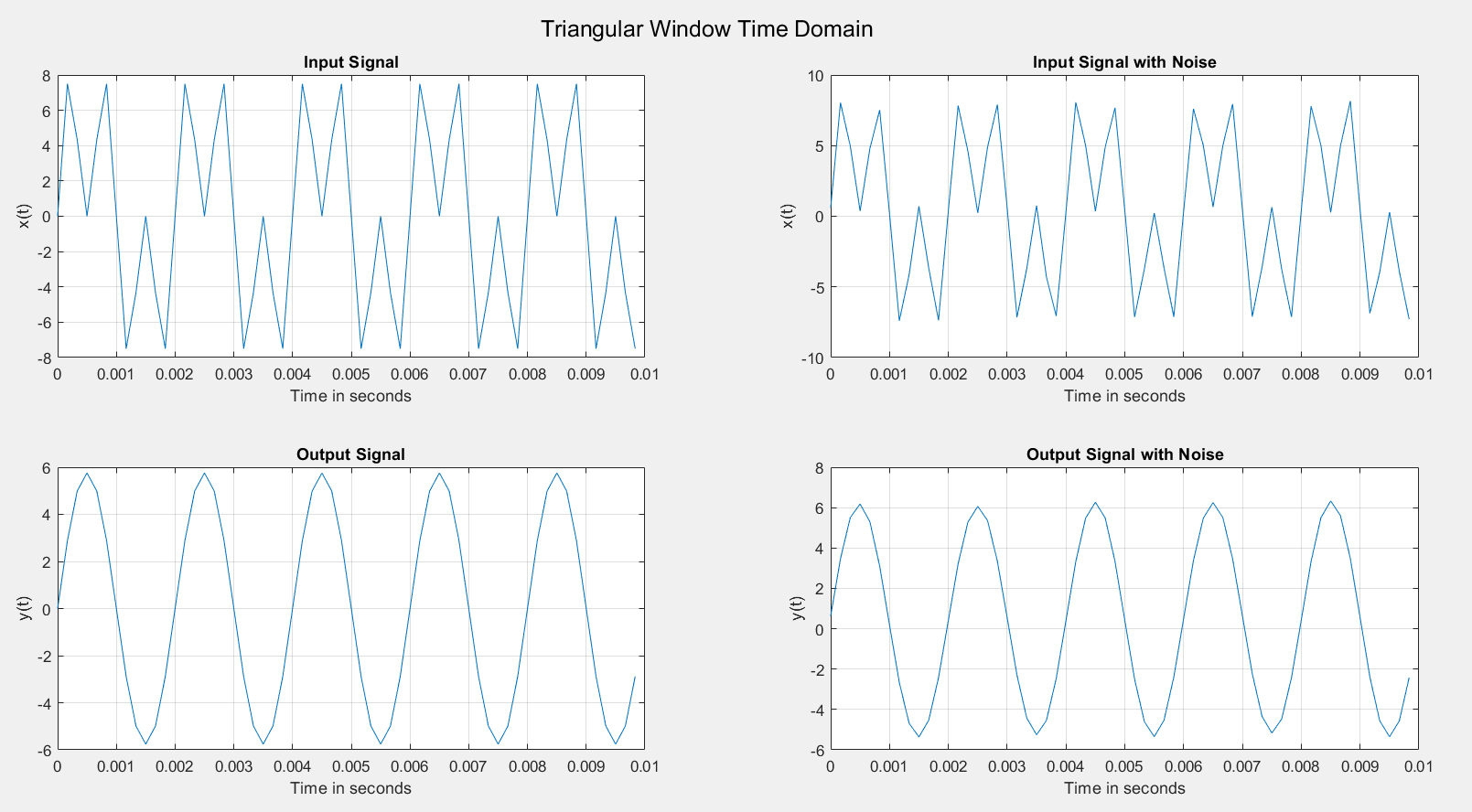
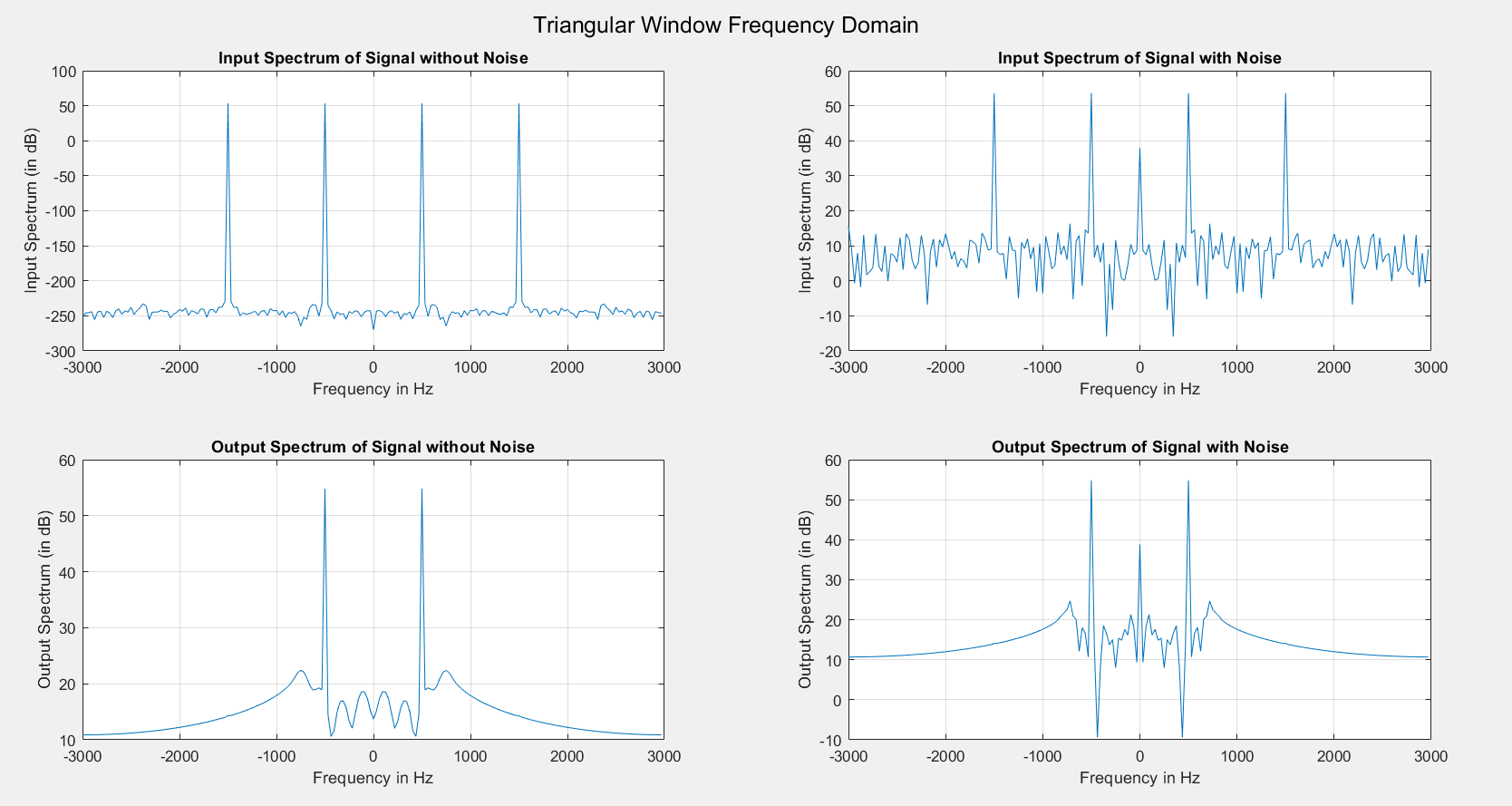
Experiment 2

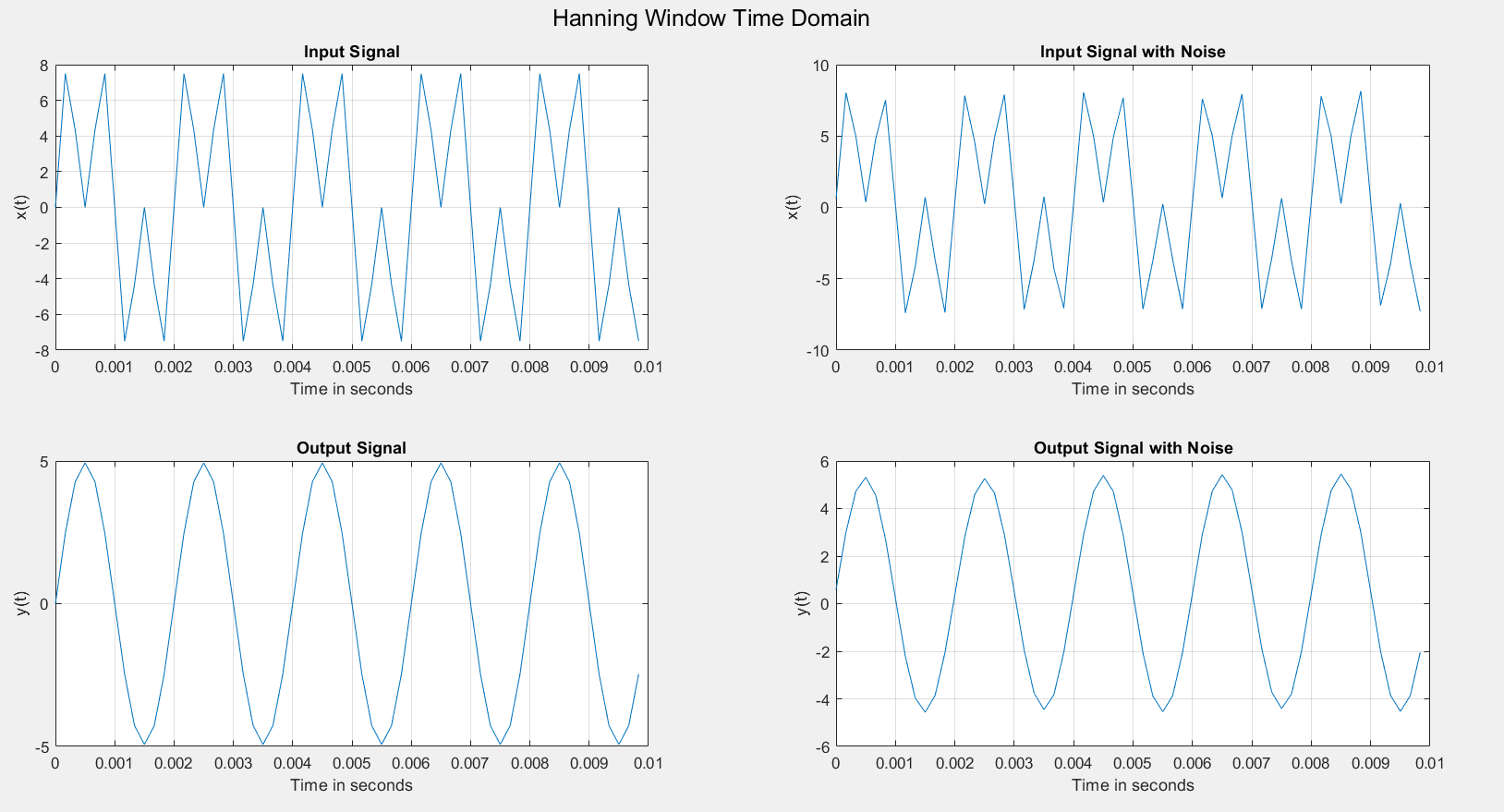
By Hardik Tibrewal (18EC10020)

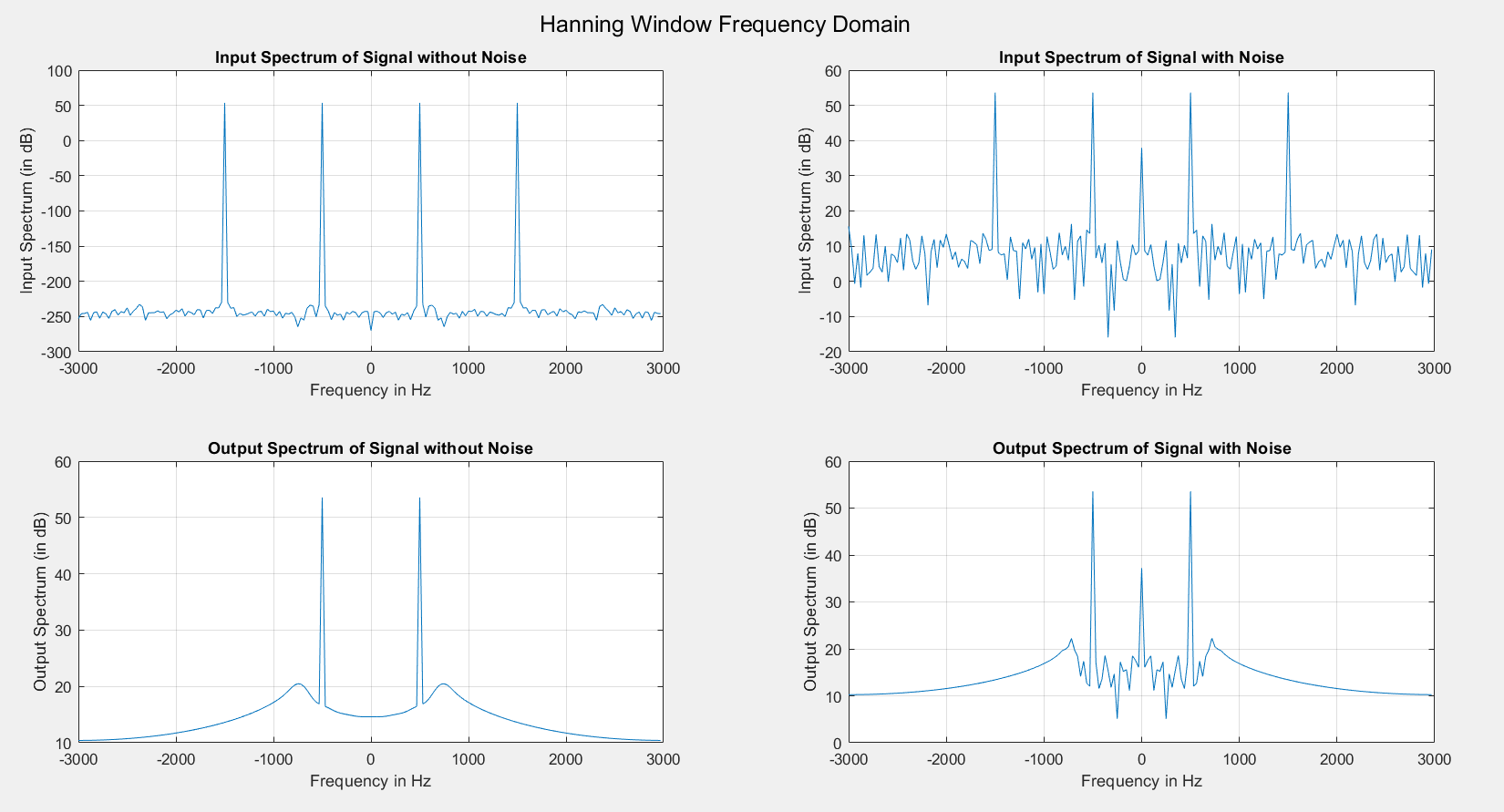
**Aim:**

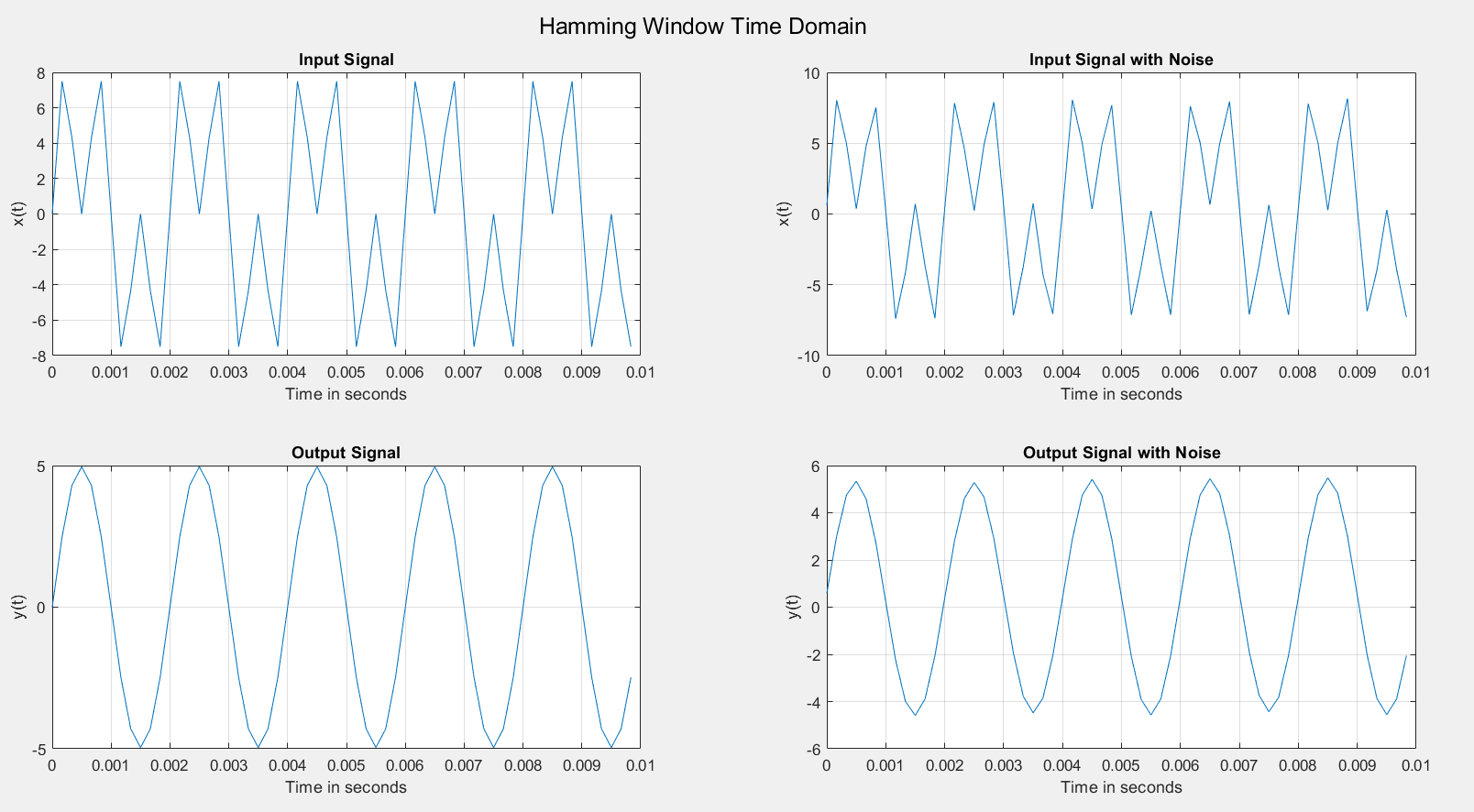
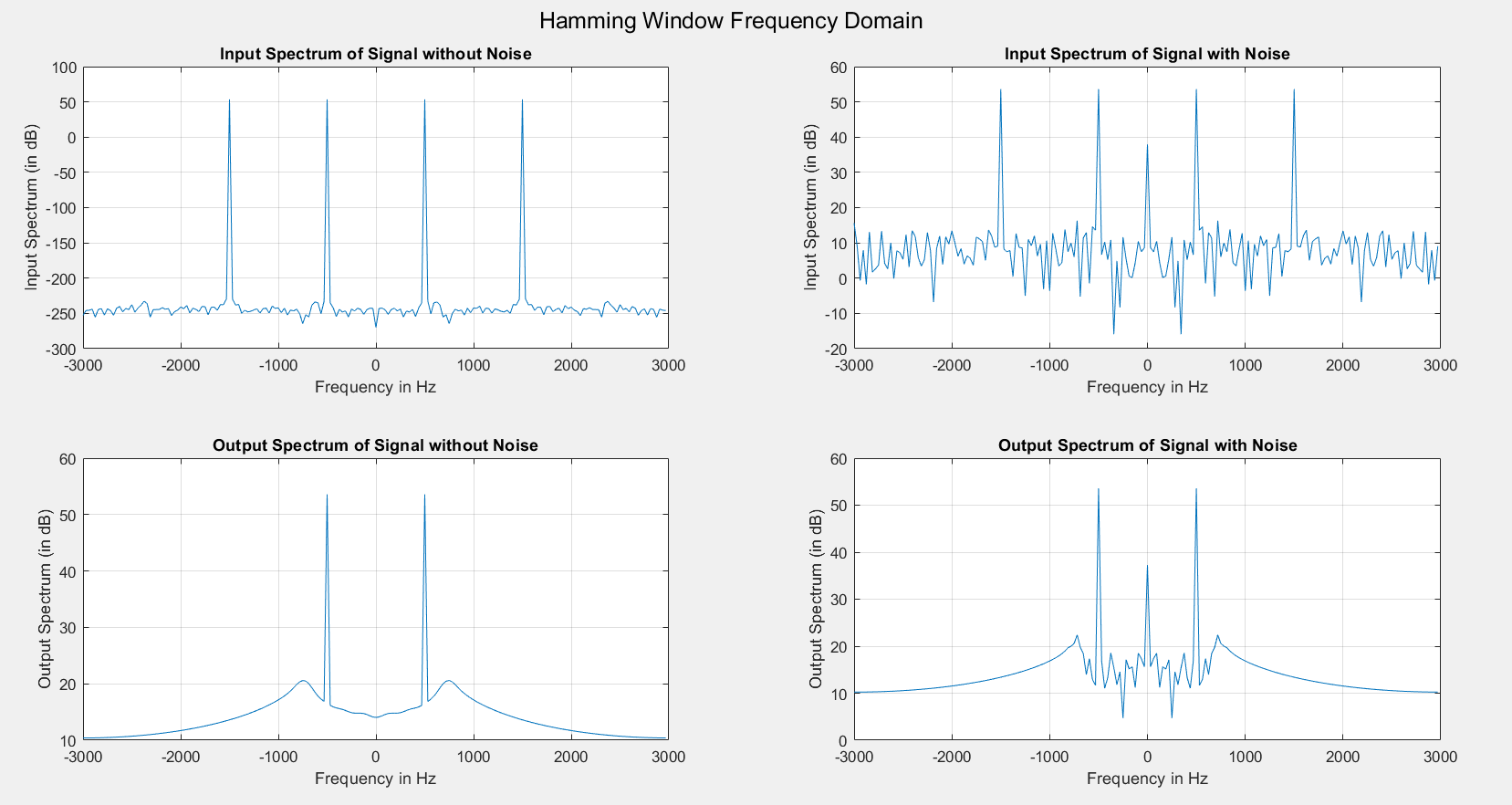
To design various FIR Filters using windowing, and test them on noisy signals

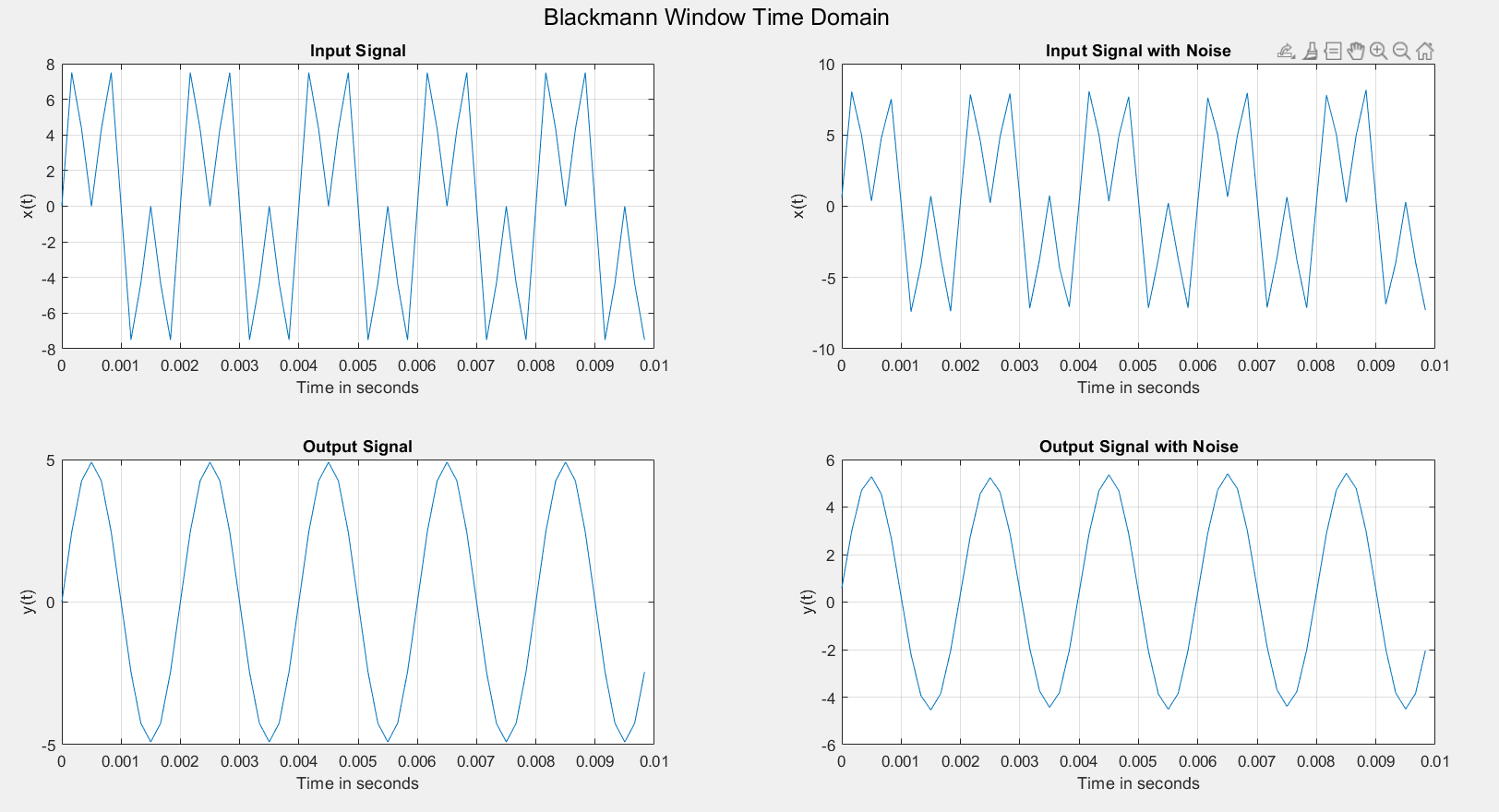
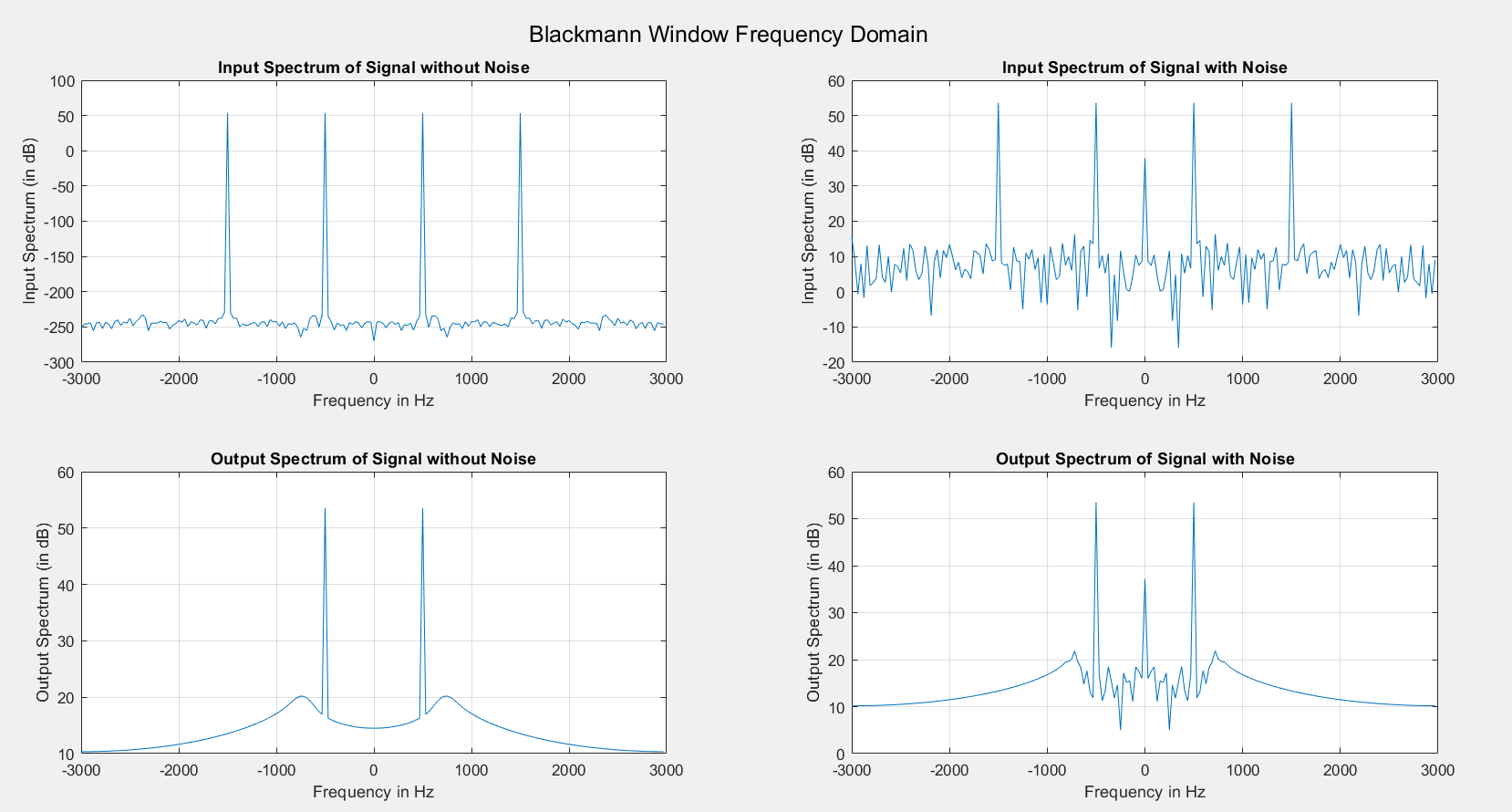
**Plots: (**N = 64 for filter)

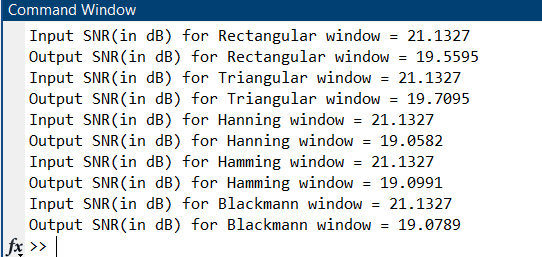












**Code:**

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