16EC30021 Palak Harwani

Experiment 02

Part 1: Window Functions

Code:

```
%%%Exp 01 1 to 8(before a,b,c,d)
clc
clear all
close all

N = 512; %Number of samples
n = 1:1:N;
wc = 0.3*pi;
k = (N-1)/2;
hd = zeros(1,N);

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

subplot(3,1,1);
plot(n,hd);
title("LPF: hd[n]");

%%%%%%%%%%%%rectangular window%%%%%%%%%%%%%%%%%%%%%%%
w_rect = ones(1,N);
subplot(3,1,2);
plot(n, w_rect);
title("rectangular window");

h1 = hd.*w_rect;
w = (-1)*pi : 0.01 : pi;

hf1 = freqz(h1,1,w);
hf_abs1 = 20*log10(abs(hf1));
subplot(3,1,3);
plot(w/pi,hf_abs1);
title("windowed LPF");
```

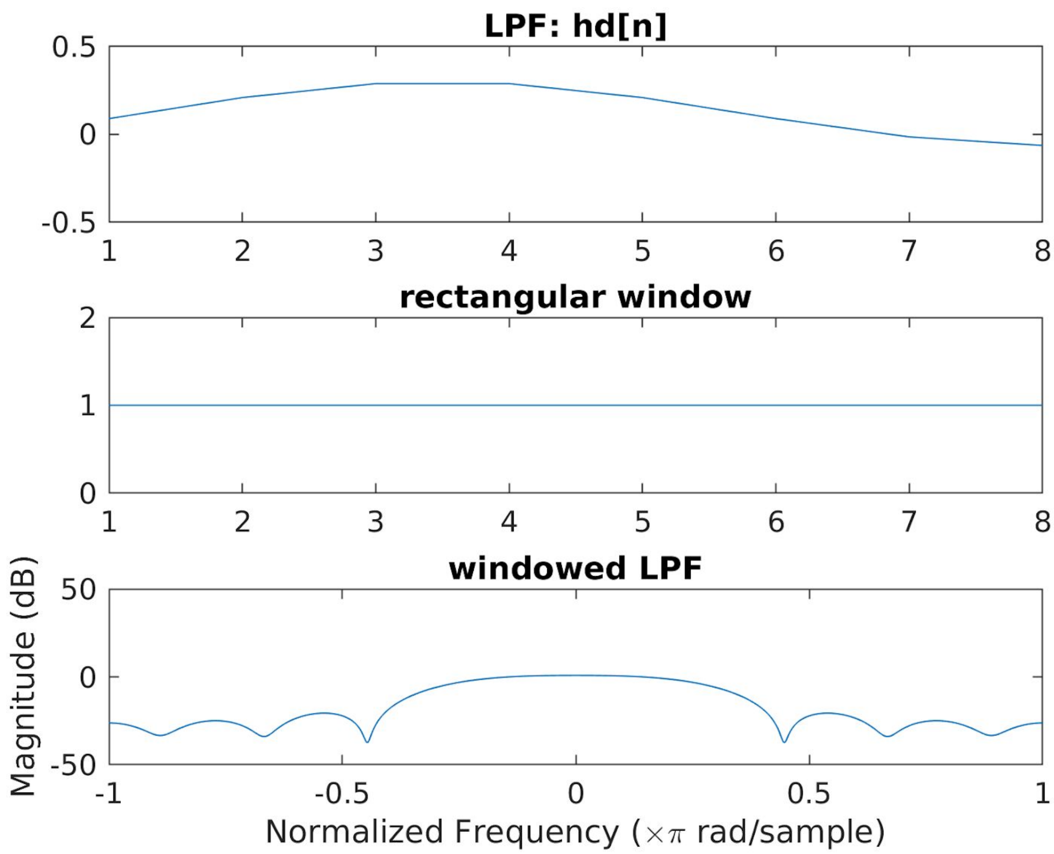
```
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude (dB)')

%%%%%%%%%%%triangular window%%%%%%%%%%%%%%%%%%%%%
figure;
subplot(3,1,1);
plot(n,hd);
title("LPF: hd[n]");

w_tri = zeros(1,N);
for i = 1:N
    w_tri(1,i) = 1 - 2*(i - (N-1)/2)/(N-1);
end
subplot(3,1,2);
plot(n, w_tri);
title("triangular window");

h2 = hd.*w_tri;
w = (-1)*pi : 0.01 : pi;

hf2 = freqz(h2,1,w);
hf_abs2 = 20*log10(abs(hf2));
subplot(3,1,3);
plot(w/pi,hf_abs2);
title("windowed LPF");
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude (dB)')

%%%%%%%%%%%hanning window%%%%%%%%%%%%%%%%%%%%%
figure;
subplot(3,1,1);
plot(n,hd);
title("LPF: hd[n]");

w_hanning = zeros(1,N);
for i = 1:N
    w_hanning(1,i) = 0.5 - 0.5*(cos(2*pi*i/(N-1)));
end
subplot(3,1,2);
plot(n, w_hanning);
title("hanning window");

h3 = hd.*w_hanning;
w = (-1)*pi : 0.01 : pi;
```

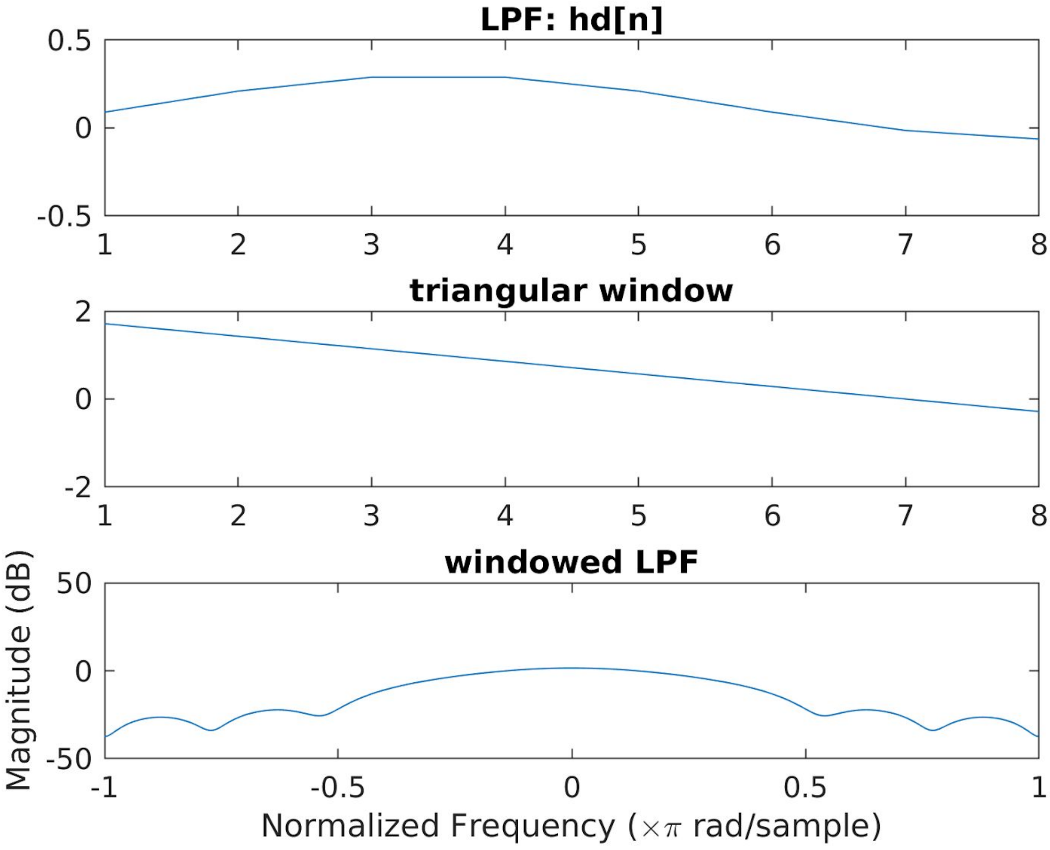
```
hf3 = freqz(h3,1,w);
hf_abs3 = 20*log10(abs(hf3));
subplot(3,1,3);
plot(w/pi,hf_abs3);
title("windowed LPF");
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude (dB)')

%%%%%%%%%%%hamming window%%%%%%%%%%%%%%%%%%%%%%

figure;
subplot(3,1,1);
plot(n,hd);
title("LPF: hd[n]");

w_hamming = zeros(1,N);
for i = 1:N
    w_hamming(1,i) = 0.54 - 0.46*(cos(2*pi*i/(N-1)));
end
subplot(3,1,2);
plot(n, w_hamming);
title("hamming window");

h4 = hd.*w_hamming;
w = (-1)*pi : 0.01 : pi;

hf4 = freqz(h4,1,w);
hf_abs4 = 20*log10(abs(hf4));
subplot(3,1,3);
plot(w/pi,hf_abs4);
title("windowed LPF");
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude (dB)')

%%%%%%%%%%%blackmann window%%%%%%%%%%%%%%%%%%%%%%

figure;
subplot(3,1,1);
plot(n,hd);
title("LPF: hd[n]");

w_blackmann = zeros(1,N);
for i = 1:N
```

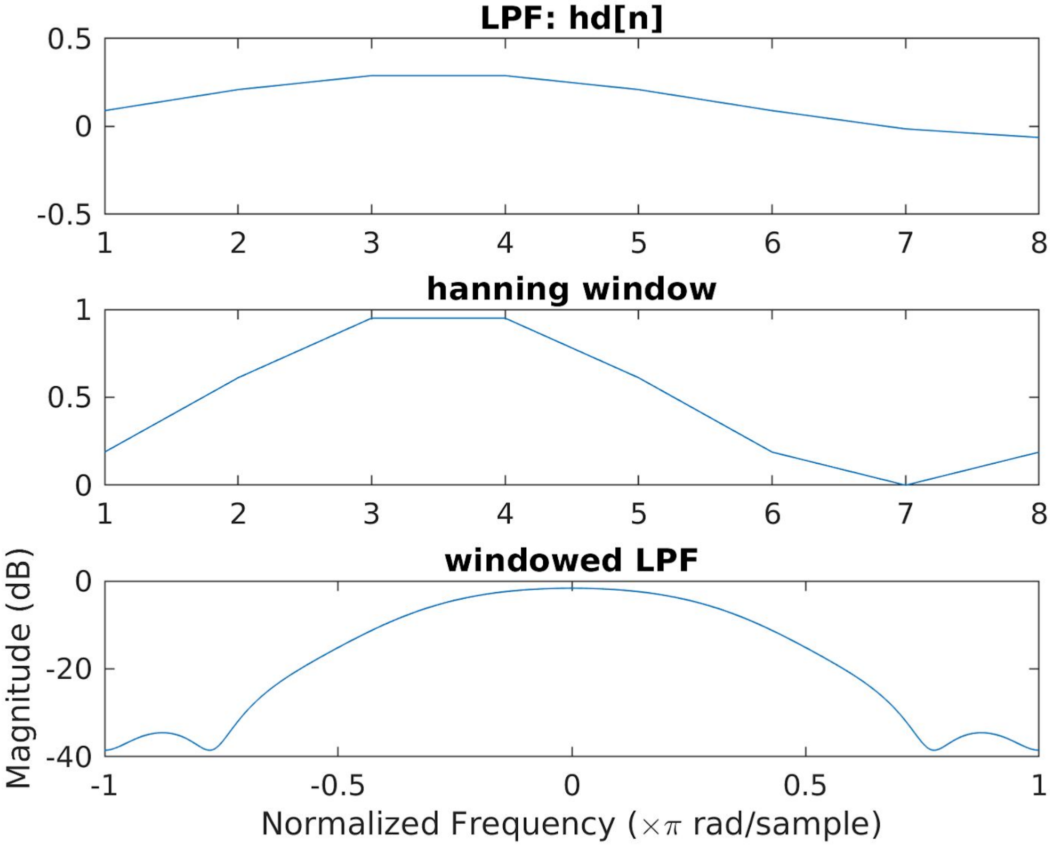
```
    w_blackmann(1,i) = 0.42 - 0.5*(cos(2*pi*i/(N-1))) +
0.08*(cos(4*pi*i/(N-1)));
end
subplot(3,1,2);
plot(n, w_blackmann);
title("blackmann window");

h5 = hd.*w_blackmann;
w = (-1)*pi : 0.01 : pi;

hf5 = freqz(h5,1,w);
hf_abs5 = 20*log10(abs(hf5));
subplot(3,1,3);
plot(w/pi,hf_abs5);
title("windowed LPF");
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude (dB)')
```

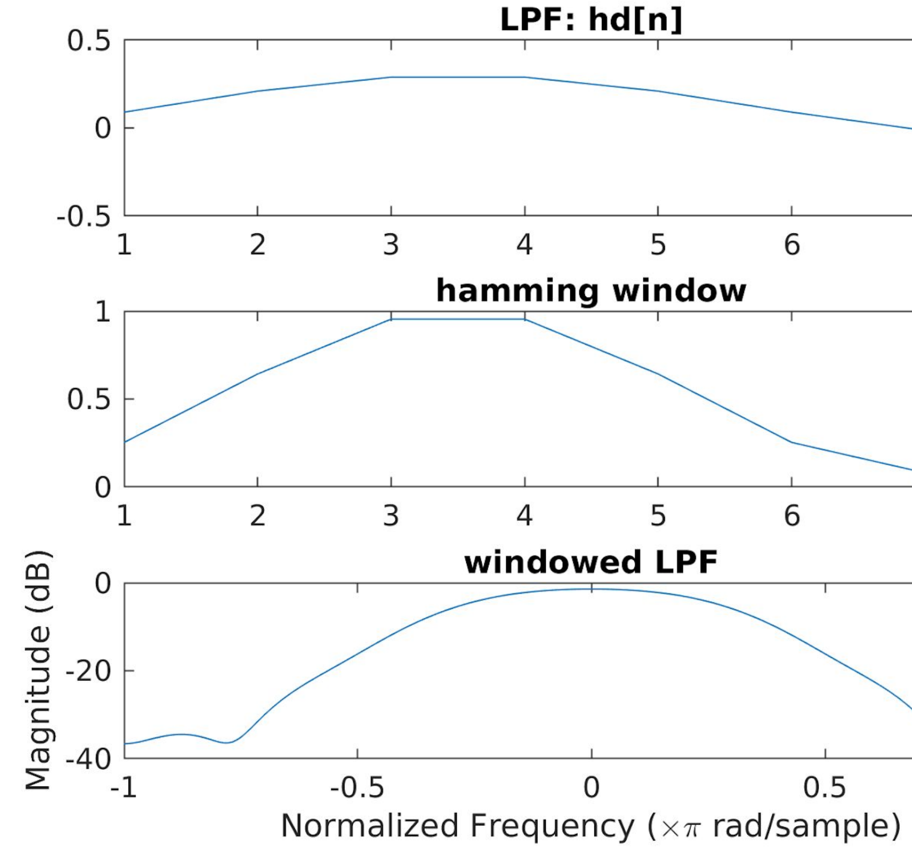
Windows and their frequency response

N = 8

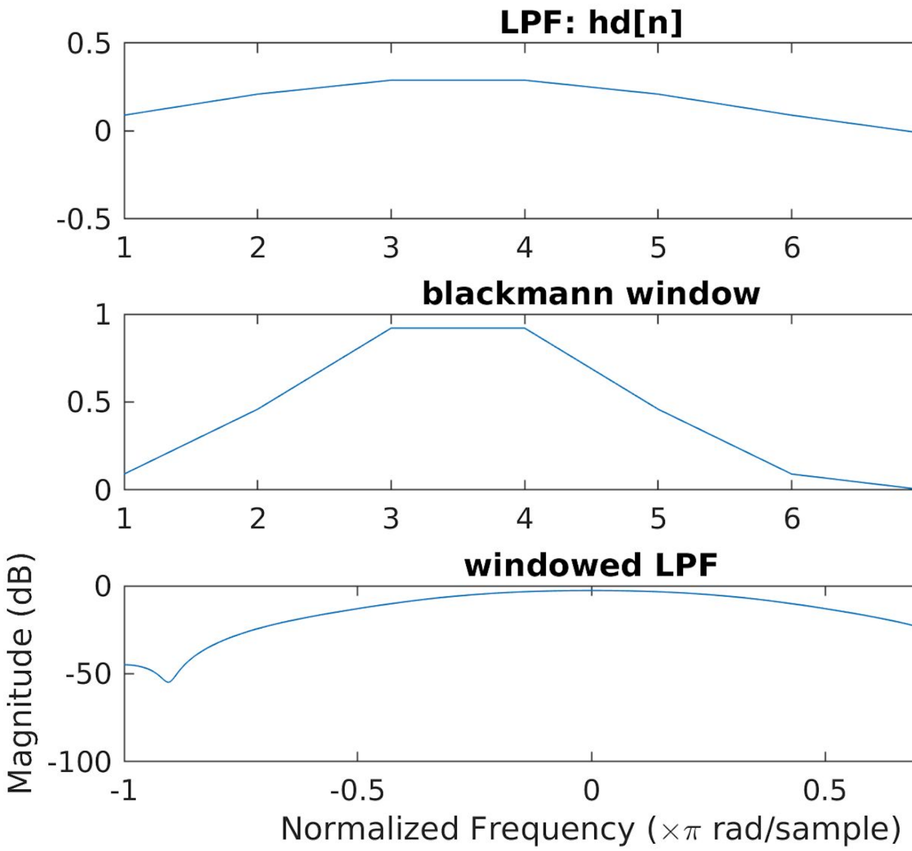



LPF: hd[n]
0.5
0
-0.5
1
2
3
4
5
6
7
8
triangular window
2
0
-2
1
2
3
4
5
6
7
8
windowed LPF
Magnitude (dB)
50
0
-50
-1
-0.5
0
0.5
1
Normalized Frequency ($\times\pi$ rad/sample)


LPF: hd[n]
0.5
0
-0.5
1
2
3
4
5
6
7
8
hanning window
1
0.5
0
1
2
3
4
5
6
7
8
windowed LPF
Magnitude (dB)
0
-20
-40
-1
-0.5
0
0.5
1
Normalized Frequency ($\times\pi$ rad/sample)

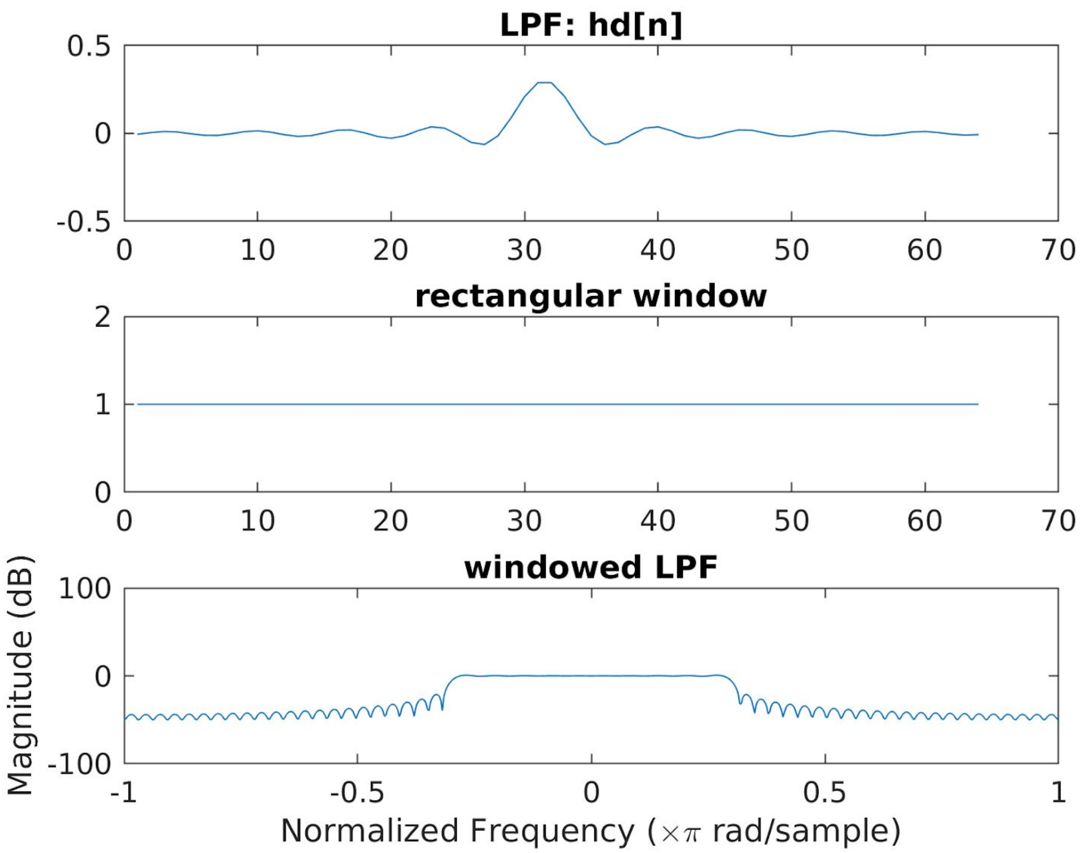


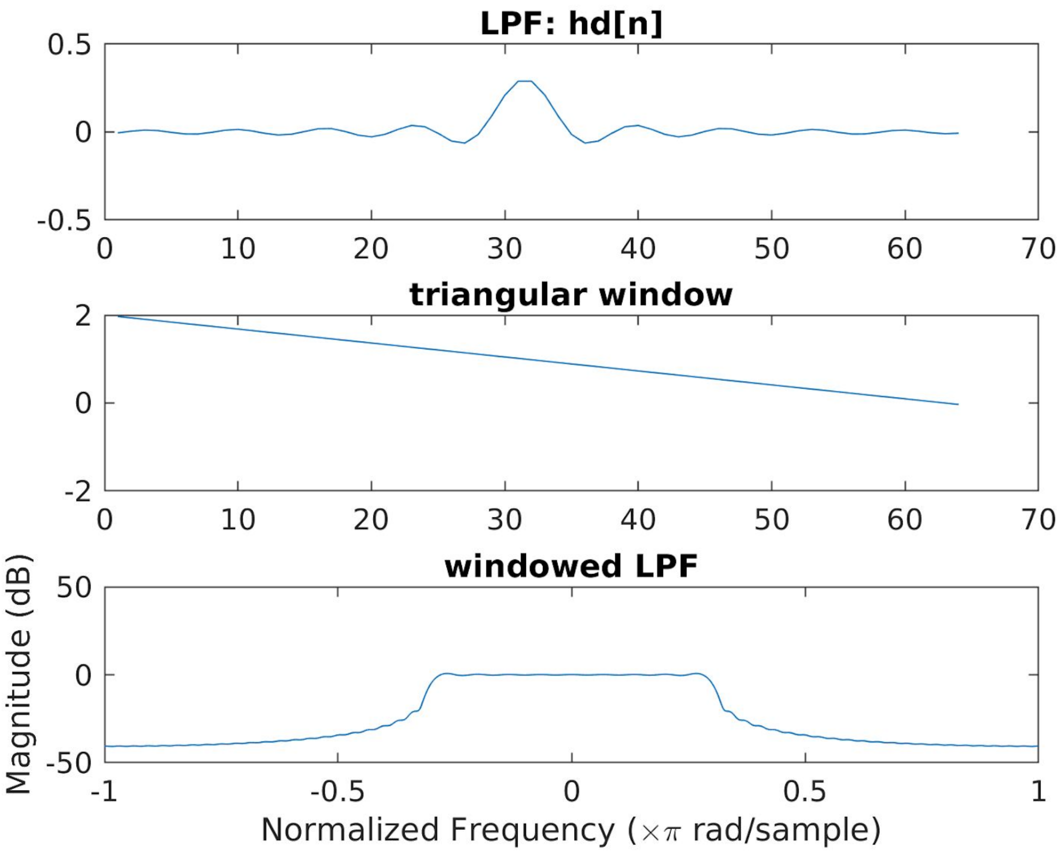
LPF: hd[n]
hamming window
windowed LPF
Magnitude (dB)
Normalized Frequency (×π rad/sample)

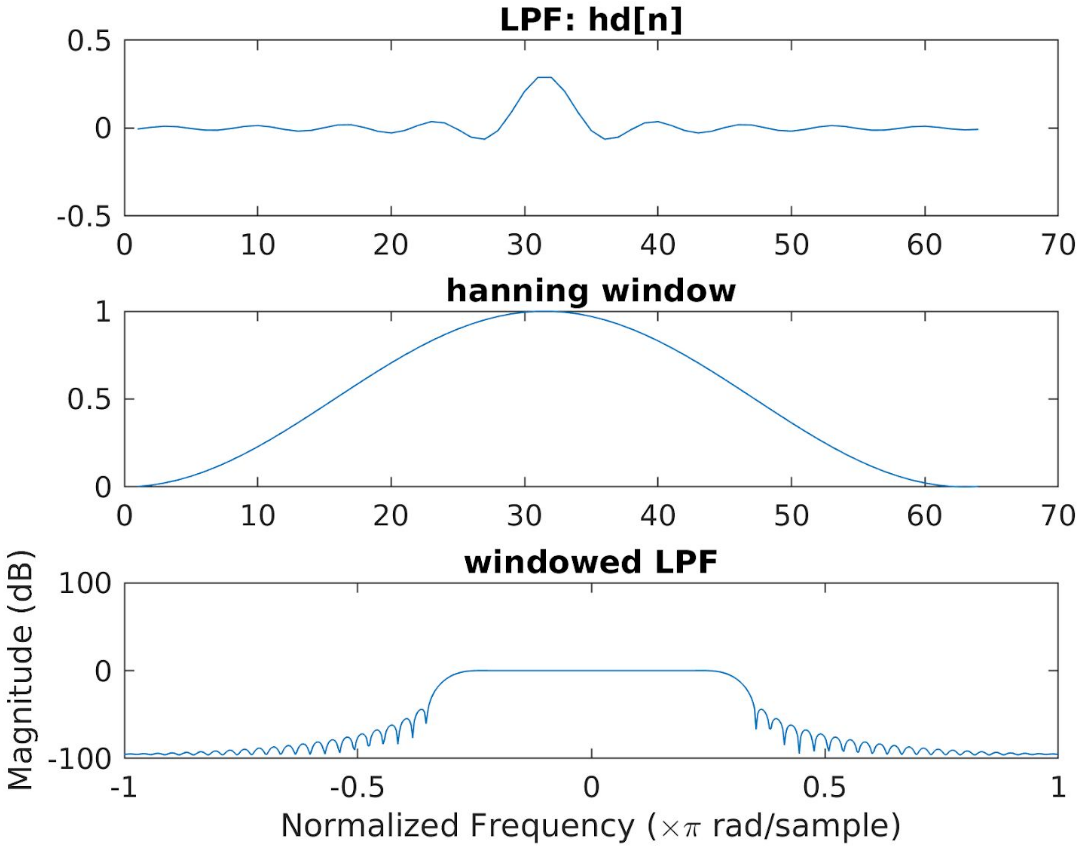


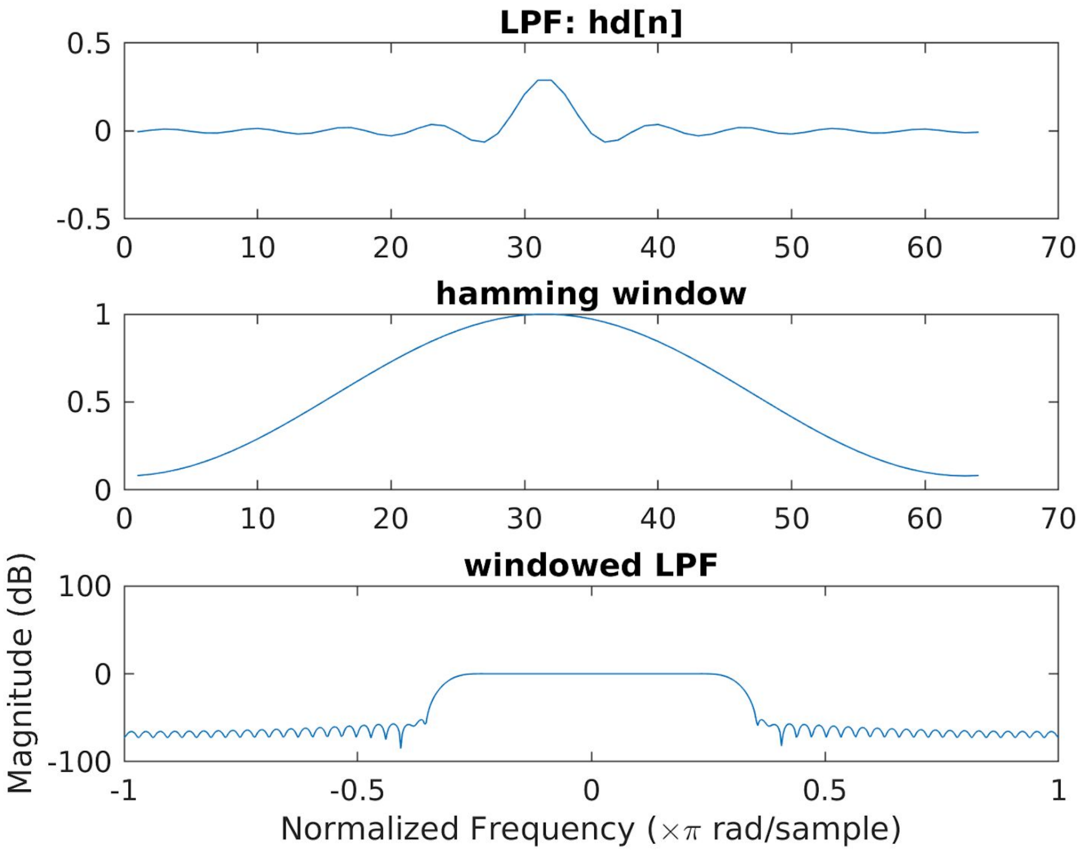
LPF: hd[n]
blackmann window
windowed LPF
Magnitude (dB)
Normalized Frequency (×π rad/sample)

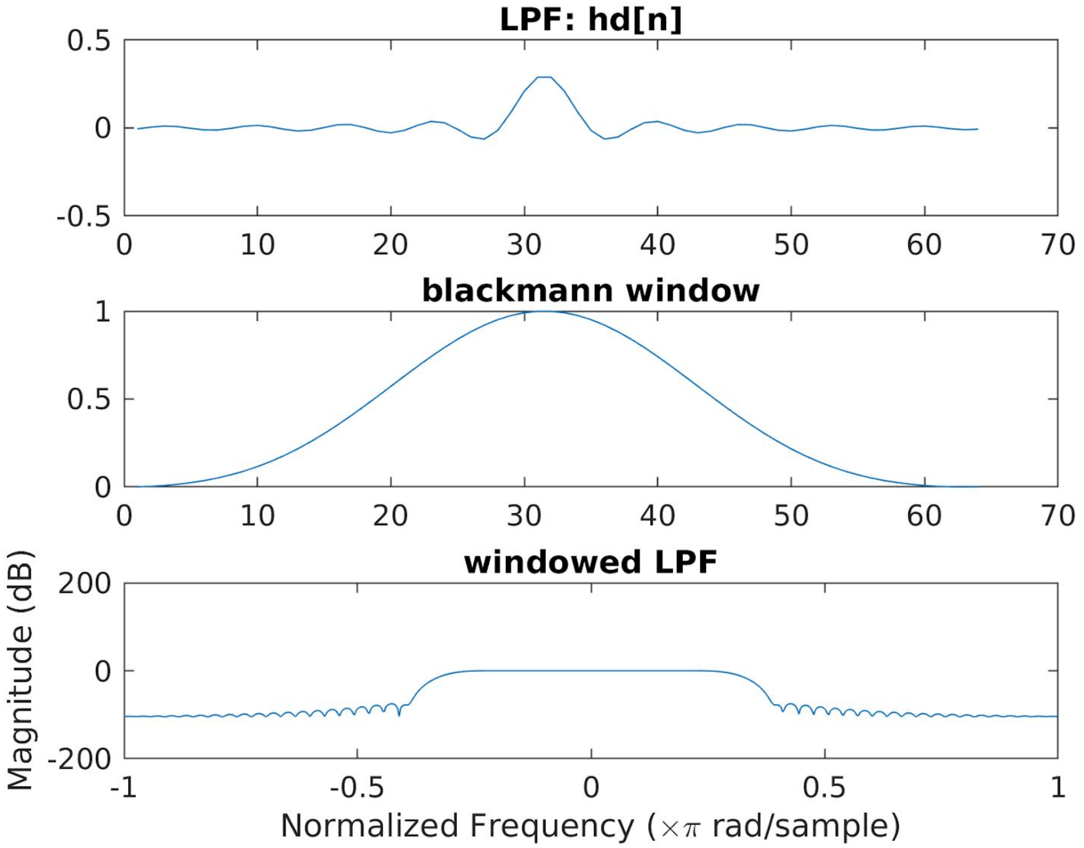
N = 64


LPF: hd[n]
0.5
0
-0.5
0
10
20
30
40
50
60
70
rectangular window
2
1
0
0
10
20
30
40
50
60
70
windowed LPF
Magnitude (dB)
100
0
-100
-1
-0.5
0
0.5
1
Normalized Frequency ($\times\pi$ rad/sample)


LPF: hd[n]
0.5
0
-0.5
0
10
20
30
40
50
60
70
triangular window
2
0
-2
0
10
20
30
40
50
60
70
windowed LPF
Magnitude (dB)
50
0
-50
-1
-0.5
0
0.5
1
Normalized Frequency ($\times\pi$ rad/sample)

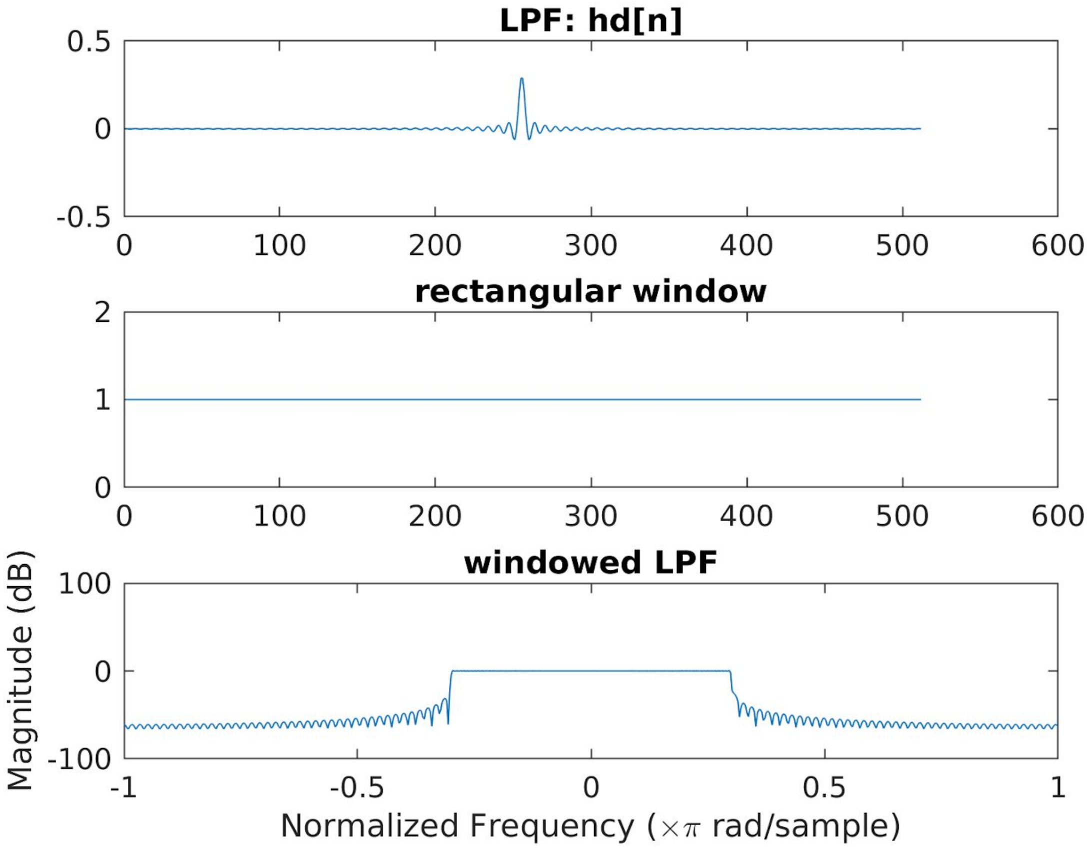

LPF: hd[n]
hanning window
windowed LPF
Magnitude (dB)
Normalized Frequency (×π rad/sample)


LPF: hd[n]
hamming window
windowed LPF
Magnitude (dB)
Normalized Frequency (×π rad/sample)

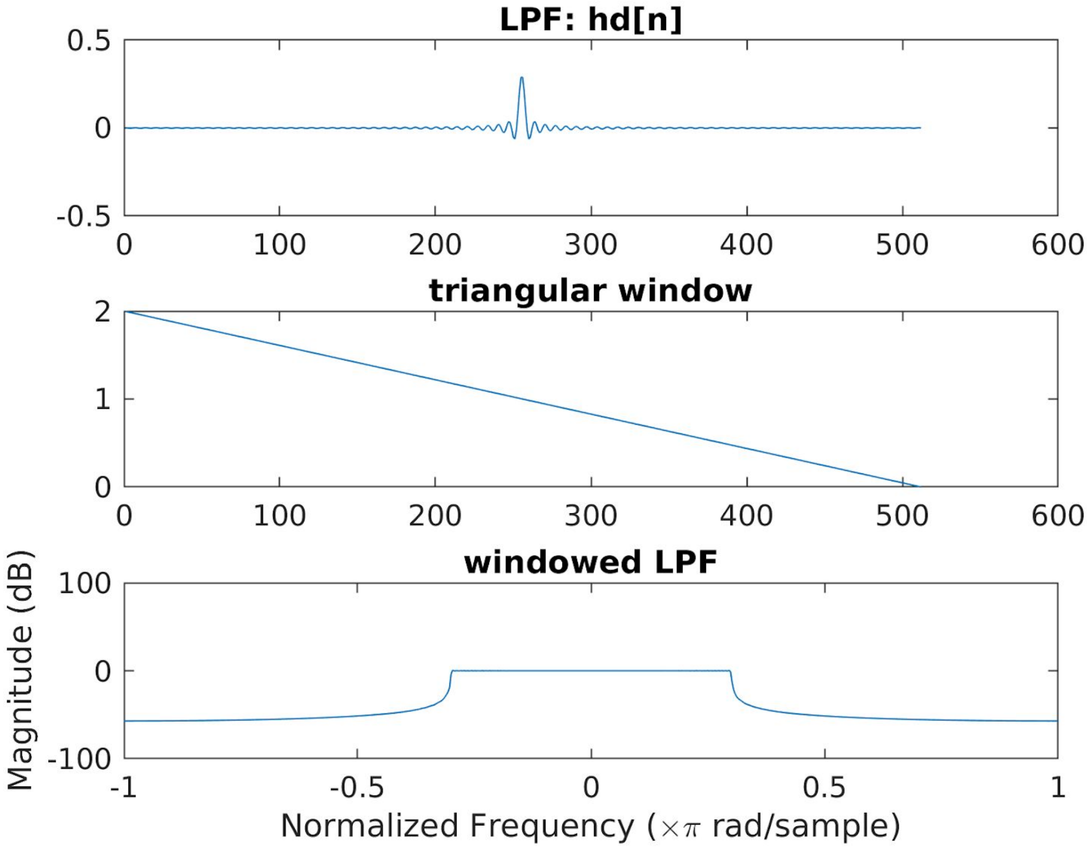


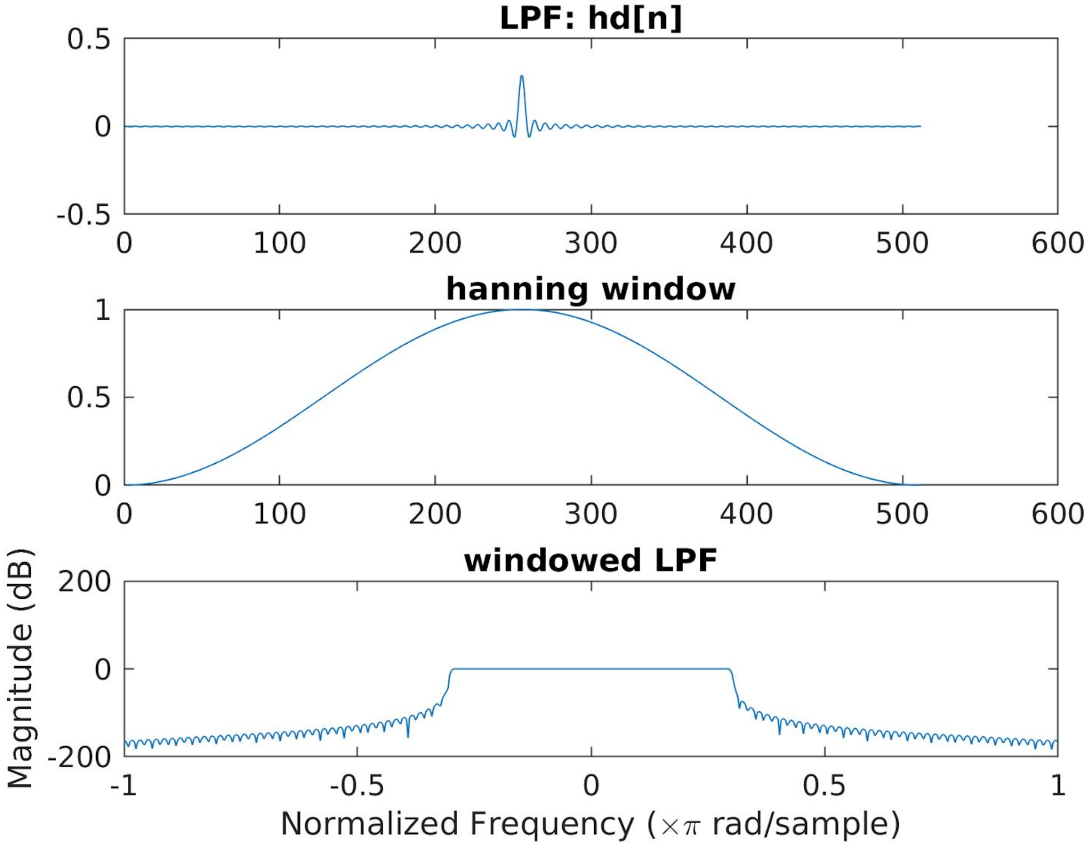
LPF: hd[n]
0.5
0
-0.5
0
10
20
30
40
50
60
70
blackmann window
1
0.5
0
windowed LPF
Magnitude (dB)
200
0
-200
-1
-0.5
0
0.5
1
Normalized Frequency (×π rad/sample)

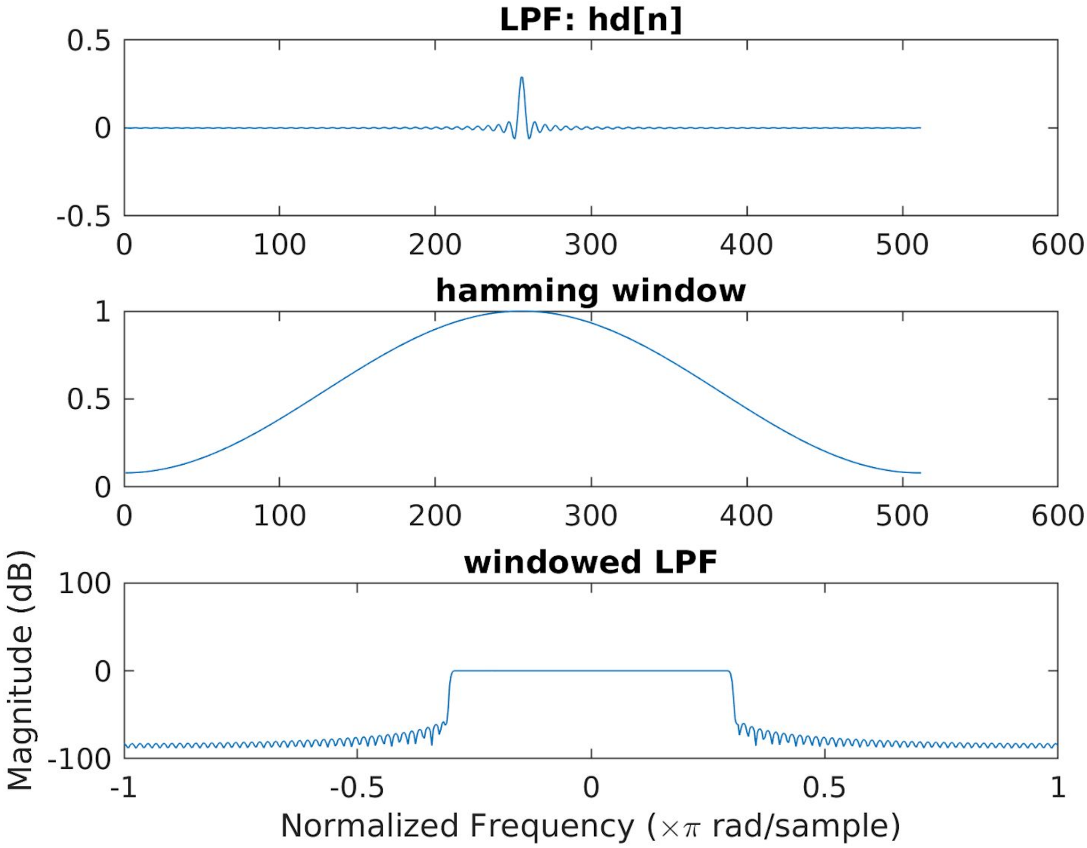
N = 512

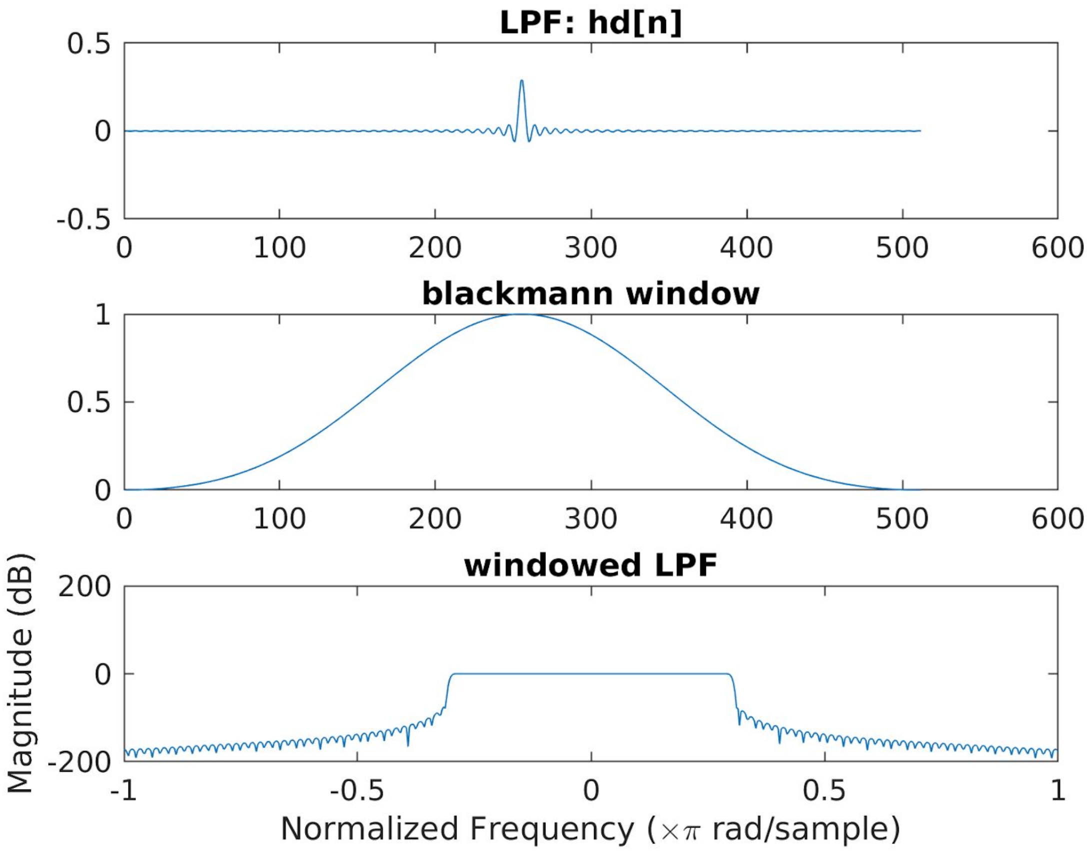


LPF: hd[n]
0.5
0
-0.5
0
100
200
300
400
500
600
rectangular window
2
1
0
windowed LPF
Magnitude (dB)
100
0
-100
-1
-0.5
0
0.5
1
Normalized Frequency (×π rad/sample)


LPF: hd[n]
0.5
0
-0.5
0
100
200
300
400
500
600
triangular window
2
1
0
0
100
200
300
400
500
600
windowed LPF
Magnitude (dB)
100
0
-100
-1
-0.5
0
0.5
1
Normalized Frequency (×π rad/sample)


LPF: hd[n]
0.5
0
-0.5
0
100
200
300
400
500
600
hanning window
1
0.5
0
0
100
200
300
400
500
600
windowed LPF
Magnitude (dB)
200
0
-200
-1
-0.5
0
0.5
1
Normalized Frequency (×π rad/sample)


LPF: hd[n]
hamming window
windowed LPF
Magnitude (dB)
Normalized Frequency ($\times\pi$ rad/sample)
0.5
0
-0.5
0
100
200
300
400
500
600
1
100
-100
-1
1


LPF: hd[n]
blackmann window
windowed LPF
Magnitude (dB)
Normalized Frequency ($\times\pi$ rad/sample)
0.5
0
-0.5
0
100
200
300
400
500
600
1
200
-200
-1
1

Rectangular window

N	Transition Width (π rad/sample)	Peak of first side lobe (dB)	Maximum stop-band Attenuation (dB)
8	0.19	-20.66	37.5
64	0.03	-21.1	50.15
512	0.01	-32	64.86

Triangular window

N	Transition Width (π rad/sample)	Peak of first side lobe (dB)	Maximum stop-band Attenuation (dB)
8	0.28	-22.26	37.37
64	0.03	-20.87	40.62
512	-	-	57.08

Hanning window

N	Transition Width (π rad/sample)	Peak of first side lobe (dB)	Maximum stop-band Attenuation (dB)
8	0.52	-34.53	38.51
64	0.07	-44.4	95.22
512	0.013	-80.28	182.3

Hamming window

N	Transition Width (π rad/sample)	Peak of first side lobe (dB)	Maximum stop-band Attenuation (dB)
8	0.53	-34.47	36.58
64	0.07	-52.09	84.85
512	0.012	-59.79	87.54

Blackmann window

N	Transition Width (π rad/sample)	Peak of first side lobe (dB)	Maximum stop-band Attenuation (dB)
8	0.62	-	54.82
64	0.12	-74.95	104
512	0.03	-89.94	190

- implies not observable from plot

Part 2 : Low Pass filtering using different windows

Code:

```
clc
clear all
close all

t = 0: 1 : 10000;
figure('Name', 'Rectangular window');

x = 2* cos(0.1*pi*t) + 3*cos(0.8*pi*t);
noise = 2*randn(1,length(x));
x_noise = x + noise;
subplot(4,2,1);
plot(t,x);
title('original signal');


x1 = fftshift(abs(fft(x,1024)));
f = -1/2:1/1023:1/2;
subplot(4,2,2);
plot(f,x1);
title('FFT of original signal');

subplot(4,2,3);
plot(t,x_noise);
title('original signal with noise');

x1_noise = fftshift(abs(fft(x_noise,1024)));
f = -1/2:1/1023:1/2;
subplot(4,2,4);
plot(f,x1_noise);
title('FFT of original signal with noise');


N = 512; %Number of samples
n = 1:1:N;
wc = 0.3*pi;
k = (N-1)/2;
hd = zeros(1,N);

for i = 1:N
```

```
    if i ==k
        hd(1,i) = wc/pi;
    else
        hd(1,i) = (sin(wc*(i-k)))/(pi*(i-k));
    end
end



%%%%%%%%%%%rectangular window%%%%%%%%%%%%%%%%%%%%%%
w_rect = ones(1,N);
h1 = hd.*w_rect;

y_rec = filtfilt(h1,1,x);
subplot(4,2,5);
plot(y_rec);
title('filtered signal');
axis([0 1000 -5 5])
y1_rec = fftshift(abs(fft(y_rec,1024)));
f = -1/2:1/1023:1/2;
subplot(4,2,6);
plot(f,y1_rec);
title('FFT of filtered signal');

y_noise_rec = filtfilt(h1,1,x_noise);
subplot(4,2,7);
plot(t,y_noise_rec);
title('filtered signal with noise');
y1_noise_rec = fftshift(abs(fft(y_noise_rec,1024)));
f = -1/2:1/1023:1/2;
subplot(4,2,8);
plot(f,y1_noise_rec);
title('FFT of filtered signal with noise');
axis([-0.3 0.3 0 1500])

%%%%%%%%%%%triangular
window%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('Name', 'Triangular window');

subplot(4,2,1);
plot(t,x);
title('original signal');

subplot(4,2,2);
```

```
plot(f,x1);
title('FFT of original signal');

subplot(4,2,3);
plot(t,x_noise);
title('original signal with noise');

subplot(4,2,4);
plot(f,x1_noise);
title('FFT of original signal with noise');


w_tri = zeros(1,N);
for i = 1:N
    w_tri(1,i) = 1 - 2*(i - (N-1)/2)/(N-1);
end
h2 = hd.*w_tri;

y_tri = filtfilt(h2,1,x);
subplot(4,2,5);
plot(y_tri);
title('filtered signal');
axis([0 1000 -5 5])
y1_tri = fftshift(abs(fft(y_tri,1024)));
f = -1/2:1/1023:1/2;
subplot(4,2,6);
plot(f,y1_tri);
title('FFT of filtered signal');

y_noise_tri = filtfilt(h2,1,x_noise);
subplot(4,2,7);
plot(t,y_noise_tri);
title('filtered signal with noise');
y1_noise_tri = fftshift(abs(fft(y_noise_tri,1024)));
f = -1/2:1/1023:1/2;
subplot(4,2,8);
plot(f,y1_noise_tri);
title('FFT of filtered signal with noise');
axis([-0.3 0.3 0 1500])

%%%%%%%%%%%hanning window%%%%%%%%%%%%%%%%%%%%%%%

figure('Name', 'Hanning');
```

```
subplot(4,2,1);
plot(t,x);
title('original signal');

subplot(4,2,2);
plot(f,x1);
title('FFT of original signal');

subplot(4,2,3);
plot(t,x_noise);
title('original signal with noise');

subplot(4,2,4);
plot(f,x1_noise);
title('FFT of original signal with noise');


w_hanning = zeros(1,N);
for i = 1:N
    w_hanning(1,i) = 0.5 - 0.5*(cos(2*pi*i/(N-1)));
end
h3 = hd.*w_hanning;

y_hann = filtfilt(h3,1,x);
subplot(4,2,5);
plot(y_hann);
title('filtered signal');
axis([0 1000 -5 5])
y1_hann = fftshift(abs(fft(y_hann,1024)));
f = -1/2:1/1023:1/2;
subplot(4,2,6);
plot(f,y1_hann);
title('FFT of filtered signal');

y_noise_hann = filtfilt(h3,1,x_noise);
subplot(4,2,7);
plot(t,y_noise_hann);
title('filtered signal with noise');
y1_noise_hann = fftshift(abs(fft(y_noise_hann,1024)));
f = -1/2:1/1023:1/2;
subplot(4,2,8);
plot(f,y1_noise_hann);
title('FFT of filtered signal with noise');
axis([-0.3 0.3 0 1500])
```

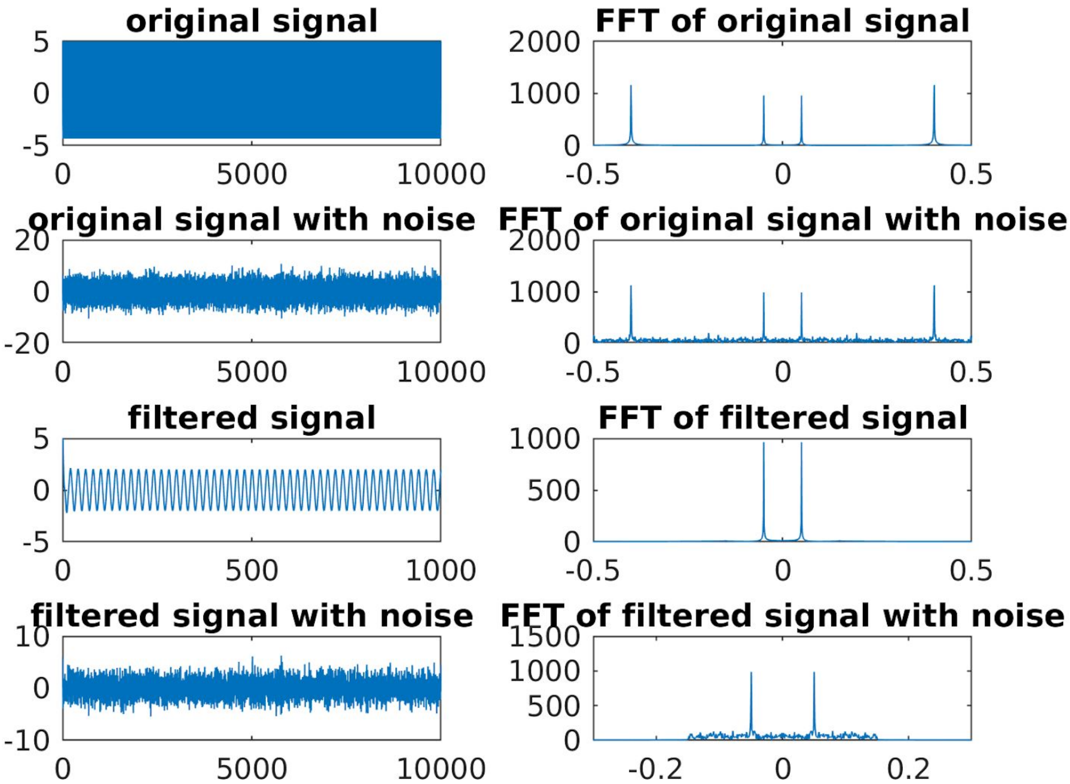
```
%%%%%%%%%%%hamming window%%%%%%%%%%%%%%%%%%%%%%%%
figure('Name', 'Hamming');
subplot(4,2,1);
plot(t,x);
title('original signal');

subplot(4,2,2);
plot(f,x1);
title('FFT of original signal');

subplot(4,2,3);
plot(t,x_noise);
title('original signal with noise');

subplot(4,2,4);
plot(f,x1_noise);
title('FFT of original signal with noise');


w_hamming = zeros(1,N);
for i = 1:N
    w_hamming(1,i) = 0.54 - 0.46*(cos(2*pi*i/(N-1)));
end
h4 = hd.*w_hamming;

y_hamm = filtfilt(h4,1,x);
subplot(4,2,5);
plot(y_hamm);
title('filtered signal');
axis([0 1000 -5 5])
y1_hamm = fftshift(abs(fft(y_hamm,1024)));
f = -1/2:1/1023:1/2;
subplot(4,2,6);
plot(f,y1_hamm);
title('FFT of filtered signal');

y_noise_hamm = filtfilt(h4,1,x_noise);
subplot(4,2,7);
plot(t,y_noise_hamm);
title('filtered signal with noise');
y1_noise_hamm = fftshift(abs(fft(y_noise_hamm,1024)));
f = -1/2:1/1023:1/2;
subplot(4,2,8);
```

```
plot(f,y1_noise_hamm);
title('FFT of filtered signal with noise');
axis([-0.3 0.3 0 1500])

%%%%%%%%%%%blackmann window%%%%%%%%%%%%%%%%
figure('Name', 'Blackmann');
subplot(4,2,1);
plot(t,x);
title('original signal');

subplot(4,2,2);
plot(f,x1);
title('FFT of original signal');

subplot(4,2,3);
plot(t,x_noise);
title('original signal with noise');

subplot(4,2,4);
plot(f,x1_noise);
title('FFT of original signal with noise');


w_tri = zeros(1,N);
for i = 1:N
    w_tri(1,i) = 1 - 2*(i - (N-1)/2)/(N-1);
end
h5 = hd.*w_tri;

y_bm = filtfilt(h5,1,x);
subplot(4,2,5);
plot(y_bm);
title('filtered signal');
axis([0 1000 -5 5])
y1_bm = fftshift(abs(fft(y_bm,1024)));
f = -1/2:1/1023:1/2;
subplot(4,2,6);
plot(f,y1_bm);
title('FFT of filtered signal');

y_noise_bm = filtfilt(h5,1,x_noise);
subplot(4,2,7);
plot(t,y_noise_bm);
title('filtered signal with noise');
```

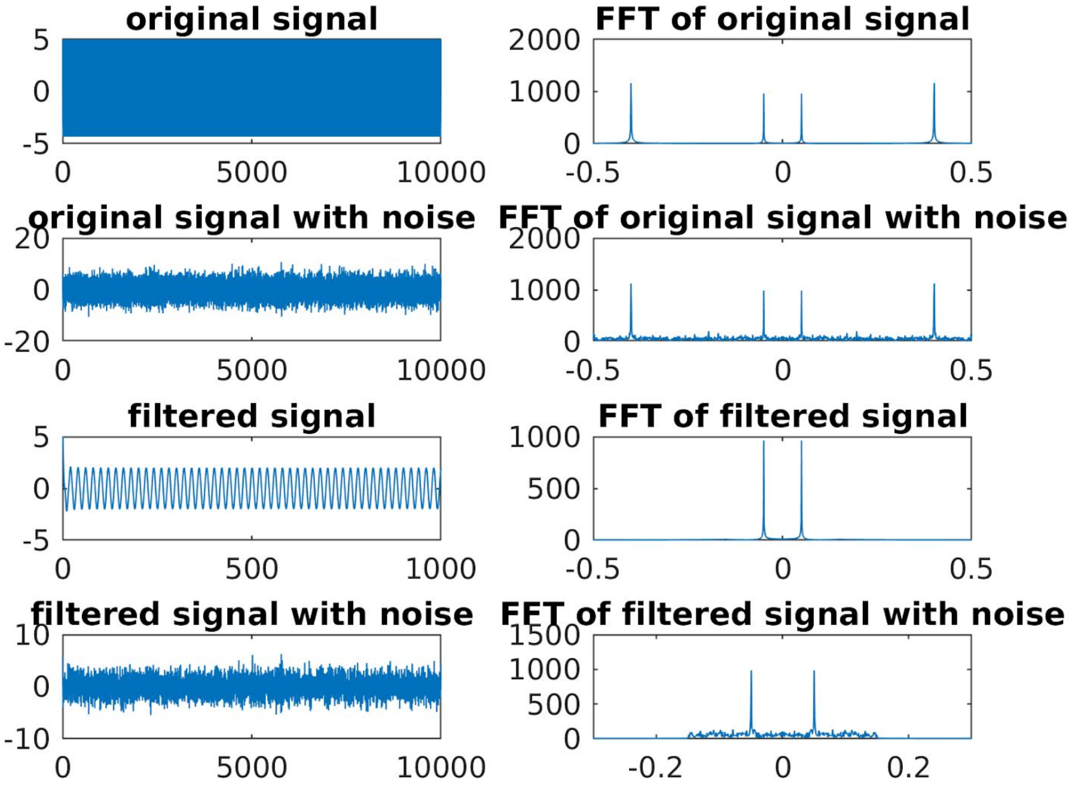
```
y1_noise_bm = fftshift(abs(fft(y_noise_bm,1024)));
f = -1/2:1/1023:1/2;
subplot(4,2,8);
plot(f,y1_noise_bm);
title('FFT of filtered signal with noise');
axis([-0.3 0.3 0 1500])
```

Signal filtered with different window functions

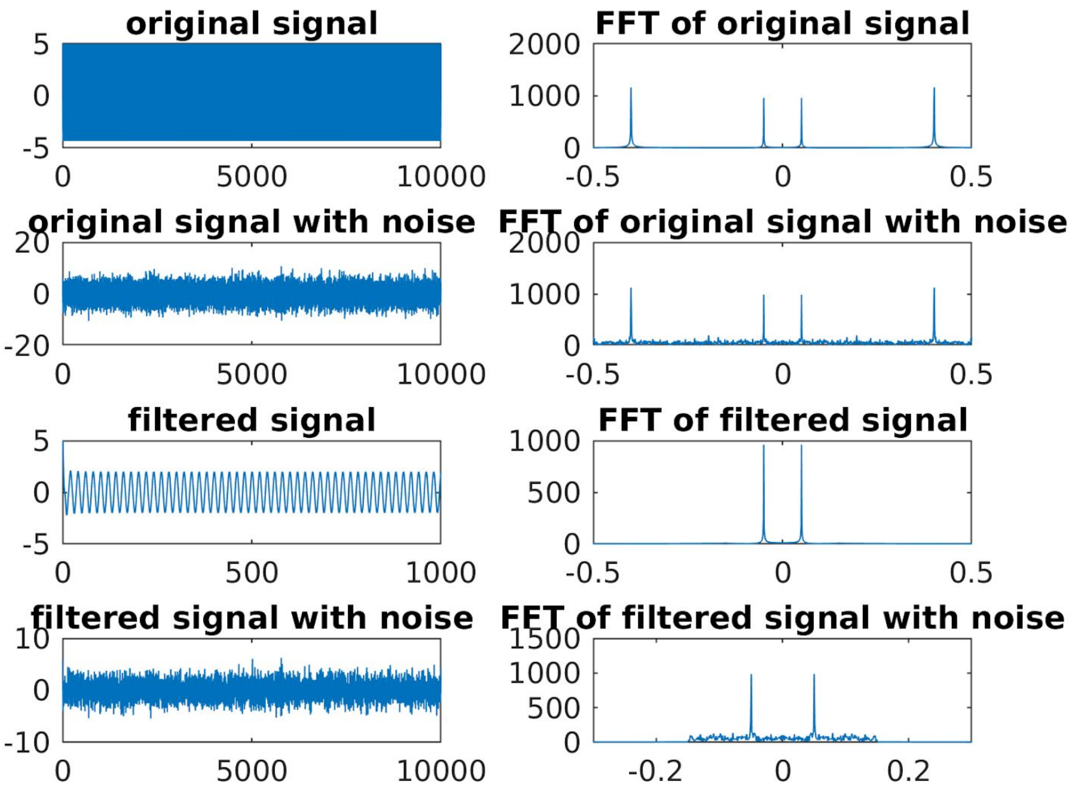
Filtering by Rectangular window



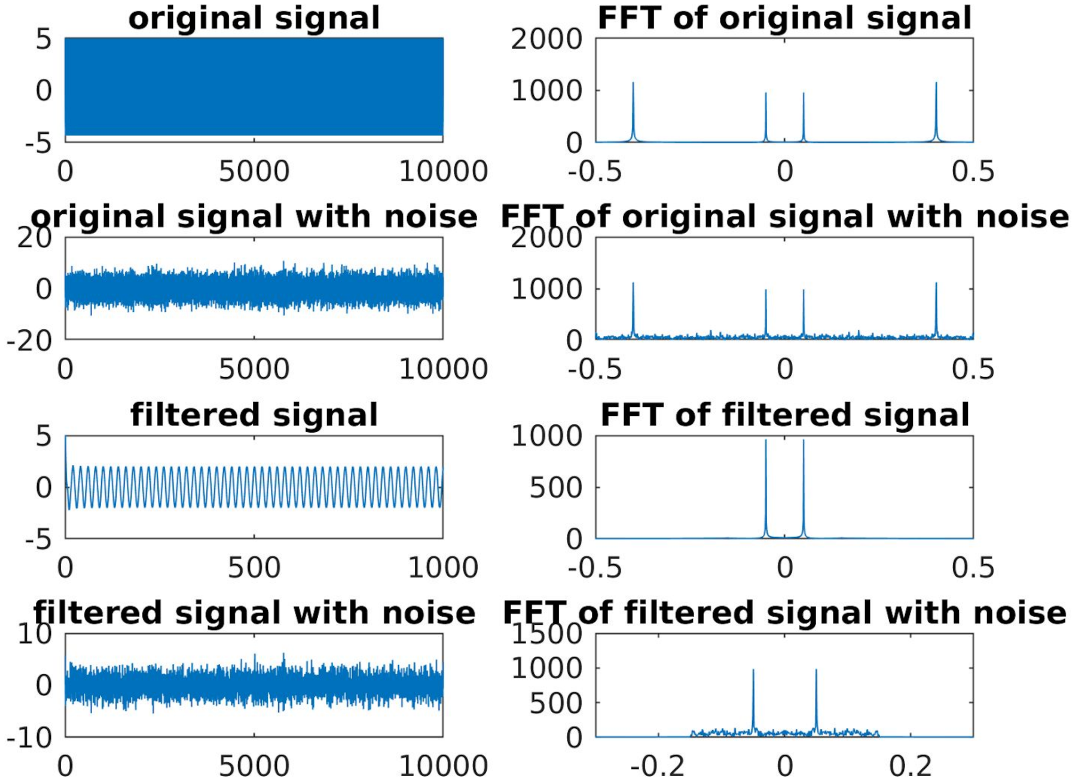
Filtering by Triangular window



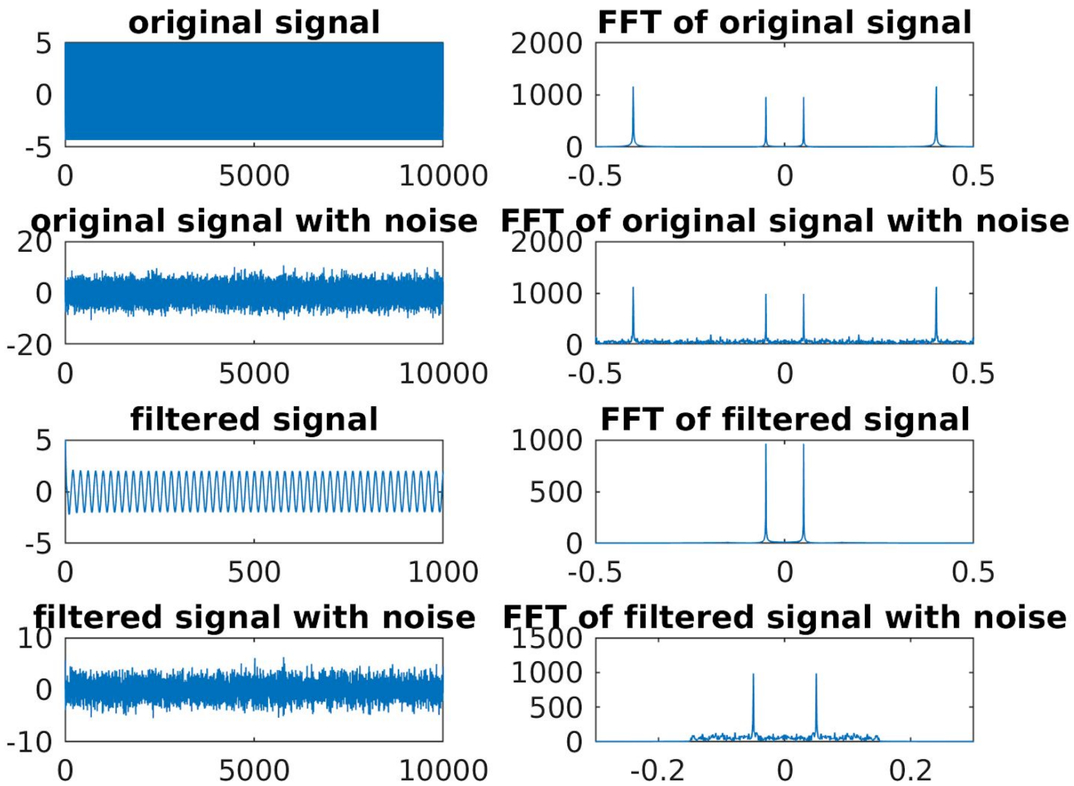
Filtering by Hanning window



Filtering by Hamming window



Filtering by Blackmann window



Rectangular window

N	Input SNR(dB)	Output SNR(dB)
8	1.9677	5.2328
64	2.0937	4.1791
512	2.045	4.2242

Triangular window

N	Input SNR(dB)	Output SNR(dB)
8	1.9677	5.1590
64	2.0937	4.0118
512	2.045	4.2190

Hanning window

N	Input SNR(dB)	Output SNR(dB)
8	1.9677	4.8587
64	2.0937	4.3761
512	2.045	4.2250

Hamming window

N	Input SNR(dB)	Output SNR(dB))
8	1.9677	4.9101
64	2.0937	4.3640
512	2.045	4.2253

Blackmann window

N	Input SNR(dB)	Output SNR(dB)
8	1.9677	5.1590
64	2.0937	4.0118
512	2.045	4.2190

- Ideal LPF is equivalent to a sinc function of infinite length in time domain which can't be realized practically. Therefore we use windowing method to restrict the sinc function using different windowing functions so as to essentially design a FIR filter.
- To obtain the impulse response, hd(n) is multiplied with w(n) in time domain, which is equivalent to convolution in frequency domain. It has the effect of smoothing Hd(w) as a result of which transition width comes into play.
- As the length of the window function(N) increases, the main lobe width of W (w) is reduced which in turn reduces the transition width but we have to deal with more ripples.
- As N increases, peak of first side lobe decreases.
- As N increases, maximum stop band attenuation increases
- For N=8 the component of the signal in the stopband is not rejected out completely.This is because the attenuation in the stop band is not sufficient, due to the fall of the filter not being sharp enough.
- The Blackman Filter have good stop band attenuation relative to other window filters.
- SNR is observed to be increased after passing through the filter as it removes out the noisy components to some extent.
- Change in SNR of the signal is more when it is filtered by higher order filter(Larger N) .