Designing Low Pass Filter by Windowing Method

AIM:

- To design FIR filters for various orders and cut-off Frequencies for five different window functions.

- To determine how FIR filters are responding to the input signals contaminated by noise.

MATLAB FUNCTION USED

fftshift, fft, cos, subplot, abs, freqz, filtfilt, rand

THEORY

FIR filters are digital filters with finite impulse response. They are also known as non-recursive digital filters as they do not have the feedback. The 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.

When designing digital FIR filters using window functions it is necessary to specify:

1. A window function to be used
2. The filter order according to the required specifications (selectivity and stop band attenuation).

These two requirements are interrelated. Each function is a kind of compromise between the two following requirements:

1. The higher the selectivity, i.e. the narrower the transition region
2. The higher suppression of undesirable spectrum, i.e. the higher the stop band attenuation.

PROCEDURE

First we define a low pass filter in time domain as $h_d(n) = \frac{sin(\omega_c(n-k))}{(\pi(n-k))}$ for $n \neq k$

$\frac{\omega_c}{\pi}$ for n = k

Here N is the number of samples and k = (N-1)/2

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

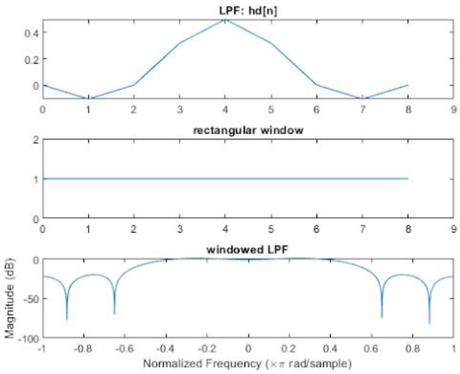
1. The window and low pass filter are multiplied in time domain. ie. h(n)=hd *w(n).
2. Filter characteristics are plotted using the MATLAB function freqz(B, A, w), where B and A are respectively the numerator and denominator of the transfer function. Here the transfer function is FIR low pass filter.
3. A function realized in time domain such that one frequency lies in pass band and other frequency lies in stop band of the filter and all the performance of the filters was observed.
4. In the second part, white noise was added to the signal and SNR was calculated.

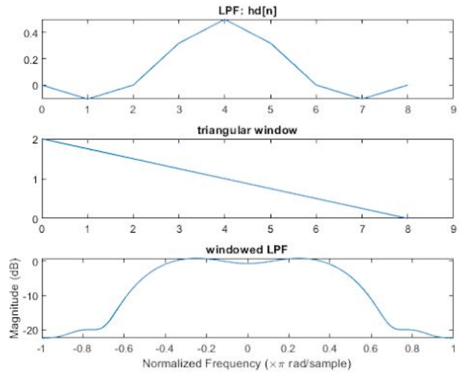
SOURCE CODE AND RESULTS

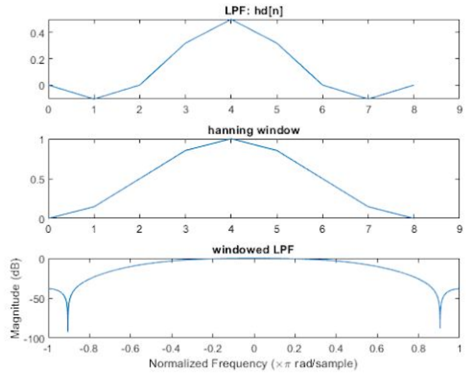
```
N = 513;
k = (N-1)/2;
x = linspace(-k,k,N);

omega = pi/2;

hd = sin(omega*(x))./(pi*(x));
hd(k+1) = omega/pi;

rectwind = 0:N-1;
rectwind = rectwind*0+1;
trangwind = 2-2*(0:N-1)/(N-1);
x = 0:N-1;
hanningwind1 = 0.5 -0.5*cos(2*pi*x/(N-1));
hanningwind2 = 0.54 -0.46*cos(2*pi*x/(N-1));
blackmanwind = 0.42 - 0.5*cos(2*pi*x/(N-1)) + 0.08*cos(4*pi*x/(N-1));

hn_rect = hd.*rectwind;
hn_triang = hd.*trangwind;
hn_hanning1 = hd.*hanningwind1;
hn_hanning2 = hd.*hanningwind2;
hn_blackmanwind = hd.*blackmanwind;

n = x;
w = -pi:.001:pi;

% rectangular window
figure;
subplot(3,1,1);
plot(n,hd);
xlim([0 N]);
title("LPF: hd[n]");
subplot(3,1,2);
plot(n, rectwind);
xlim([0 N]);
title("rectangular window");
hf1 = freqz(hn_rect,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);
xlim([0 N]);
title("LPF: hd[n]");
subplot(3,1,2);
plot(n, trangwind);
xlim([0 N]);
title("triangular window");
hf1 = freqz(hn_triang,1,w);
hf_abs1 = 20*log10(abs(hf1));
subplot(3,1,3);
```

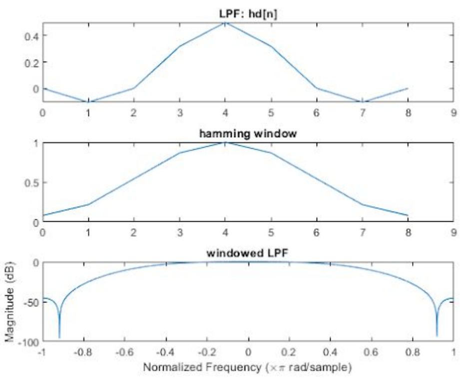
```
% hanning1 window
figure;
subplot(3,1,1);
plot(n,hd);
xlim([0 N]);
title("LPF: hd[n]");
subplot(3,1,2);
plot(n, hanningwind1);
xlim([0 N]);
title("hanning window");
hf1 = freqz(hn_hanning1,1,w);
hf_abs1 = 20*log10(abs(hf1));
subplot(3,1,3);
plot(w/pi,hf_abs1);
title("windowed LPF");
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude (dB)')

% hanning2 window
figure;
subplot(3,1,1);
plot(n,hd);
xlim([0 N]);
title("LPF: hd[n]");
subplot(3,1,2);
plot(n, hanningwind2);
xlim([0 N]);
title("hamming window");
hf1 = freqz(hn_hanning2,1,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)')

% blackman window
figure;
subplot(3,1,1);
plot(n,hd);
xlim([0 N]);
title("LPF: hd[n]");
subplot(3,1,2);
plot(n, blackmanwind);
xlim([0 N]);
title("blackman window");
hf1 = freqz(hn_blackmanwind,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)')
```

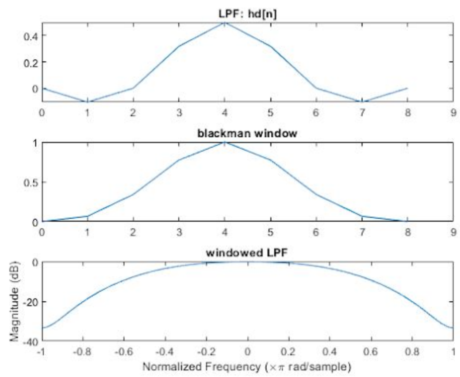
For N=8,



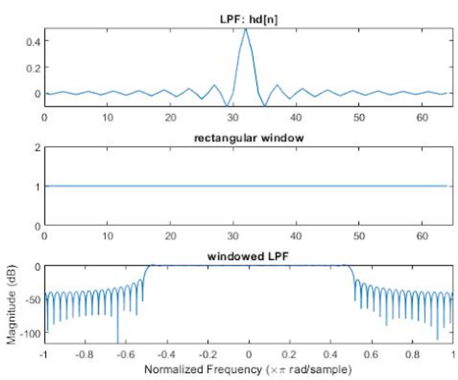


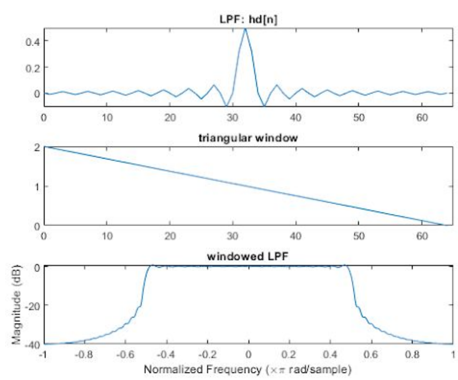


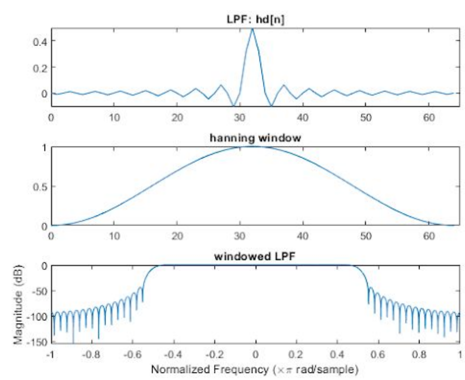


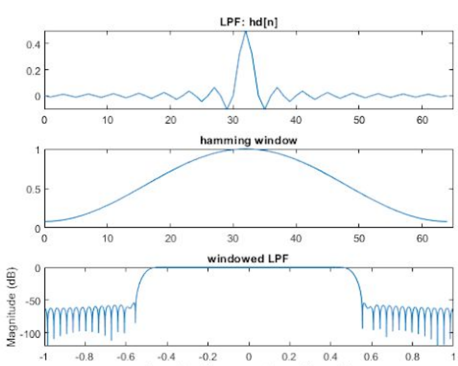


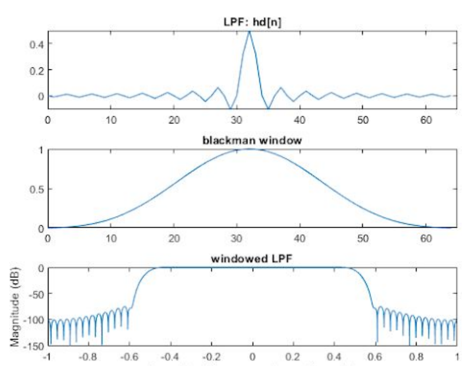
For N=64,



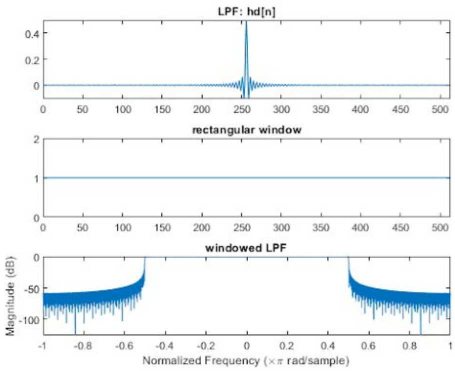


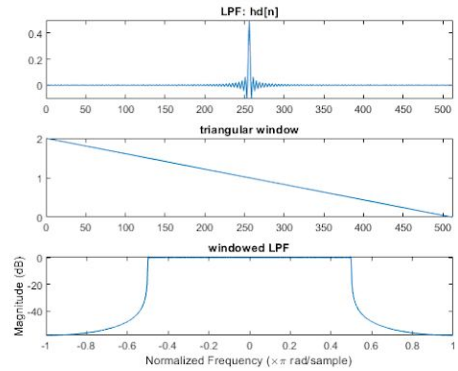


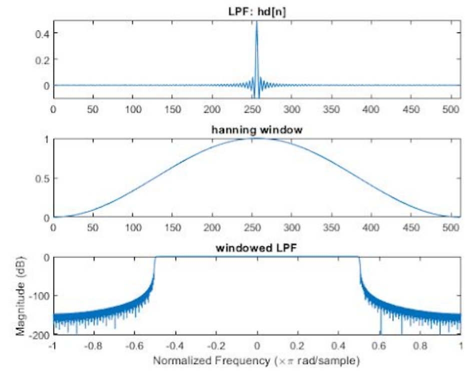


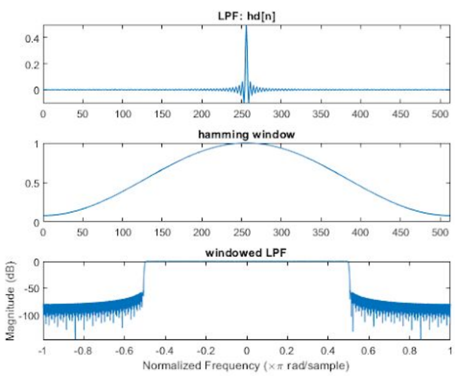


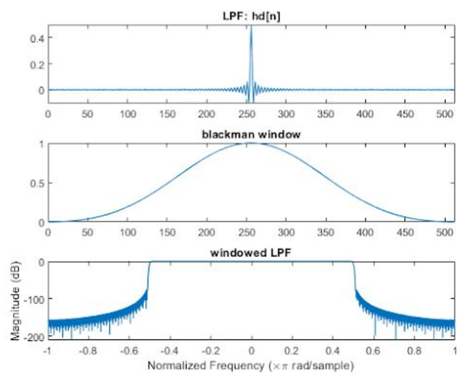
For N=512,











TABLES

For N=8,

Window Type	Transition Width (π rad/sample)	Peak of the first side lobe (dB)	Maximum stop-band attenuation(dB)
Rectangular	0.19	-20.06	37.5
Triangular	0.28	NA	37.37
Hanning	0.52	-38.19	38.51
Hamming	0.67	-45.55	36.58
Blackmann	0.62	NA	33.29

For N=64,

Window Type	Transition Width (π rad/sample)	Peak of the first side lobe (dB)	Maximum stop-band attenuation(dB)
Rectangular	0.03	-21.11	50.84
Triangular	0.03	-22.26	NA
Hanning	0.09	-43.9	77.19
Hamming	0.09	-51.04	93.71
Blackmann	0.15	-75.31	144.01

For N=512,

Window Type	Transition Width (π rad/sample)	Peak of the first side lobe (dB)	Maximum stop-band attenuation(dB)
Rectangular	0.12	-21.00	64.86
Triangular	NA	NA	NA
Hanning	0.09	-43.94	202.67
Hamming	0.09	-59.04	87.54
Blackmann	0.15	-88.85	190

SIGNAL FILTERED WITH DIFFERENT WINDOW FUNCTIONS

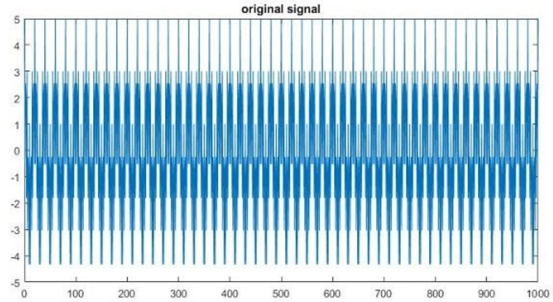
```
t = 0: 1 : 1000;
x = 2*cos(0.1*pi*t) + 3*cos(0.8*pi*t);
noise = 2*randn(1,length(x));
x_noise = x + noise;

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

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

subplot(2,2,3);
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(2,2,4);
plot(f,x1_noise);
title('FFT of original signal with noise');


% plotting outputs
figure;
N = 65;
k = (N-1)/2;
p = linspace(-k,k,N);
omega = pi*0.3;
hd = sin(omega*(p))./(pi*(p));
hd(k+1) = omega/pi;

rectwind = 0:N-1;
rectwind = rectwind*0+1;
trangwind = 2-2*(0:N-1)/(N-1);
p = 0:N-1;
hanningwind1 = 0.5 -0.5*cos(2*pi*p/(N-1));
hanningwind2 = 0.54 -0.46*cos(2*pi*p/(N-1));
blackmanwind = 0.42 - 0.5*cos(2*pi*p/(N-1)) + 0.08*cos(4*pi*p/(N-1));

hn_rect = hd.*rectwind;
hn_triang = hd.*trangwind;
hn_hanning1 = hd.*hanningwind1;
hn_hanning2 = hd.*hanningwind2;
hn_blackmanwind = hd.*blackmanwind;


% rectangular
y = filtfilt(hn_rect,1,x);
y1 = fftshift(abs(fft(y,1024)));
f = -1/2:1/1023:1/2;
subplot(5,2,1);
plot(f,y1);
```

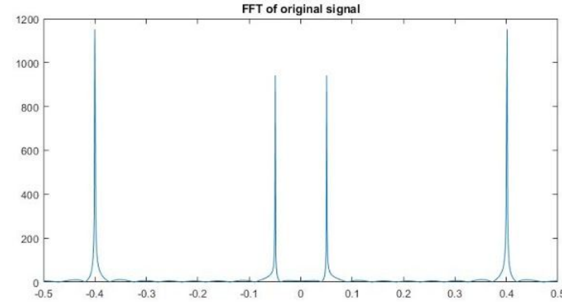
```
% hanning
y = filtfilt(hn_hanning1,1,x);
y1 = fftshift(abs(fft(y,1024)));
f = -1/2:1/1023:1/2;
subplot(5,2,5);
plot(f,y1);
title('filtered signal hanning');

y_noise = filtfilt(hn_hanning1,1,x_noise);
y1_noise = fftshift(abs(fft(y_noise,1024)));
f = -1/2:1/1023:1/2;
subplot(5,2,6);
plot(f,y1_noise);
title('filtered noisy signal hanning');
disp(['SNR hanning is equal to ',num2str(snr(y1_noise, y1_noise-y1)),'.']);


% hamming
y = filtfilt(hn_hanning2,1,x);
y1 = fftshift(abs(fft(y,1024)));
f = -1/2:1/1023:1/2;
subplot(5,2,7);
plot(f,y1);
title('filtered signal hamming');

y_noise = filtfilt(hn_hanning2,1,x_noise);
y1_noise = fftshift(abs(fft(y_noise,1024)));
f = -1/2:1/1023:1/2;
subplot(5,2,8);
plot(f,y1_noise);
title('filtered noisy signal hamming');
disp(['SNR hamming is equal to ',num2str(snr(y1_noise, y1_noise-y1)),'.']);


% blackman
y = filtfilt(hn_hanning2,1,x);
y1 = fftshift(abs(fft(y,1024)));
f = -1/2:1/1023:1/2;
subplot(5,2,9);
plot(f,y1);
title('filtered signal blackman');

y_noise = filtfilt(hn_hanning2,1,x_noise);
y1_noise = fftshift(abs(fft(y_noise,1024)));
f = -1/2:1/1023:1/2;
subplot(5,2,10);
plot(f,y1_noise);
title('filtered noisy signal blackman');
disp(['SNR blackman is equal to ',num2str(snr(y1_noise, y1_noise-y1)),'.']);
```

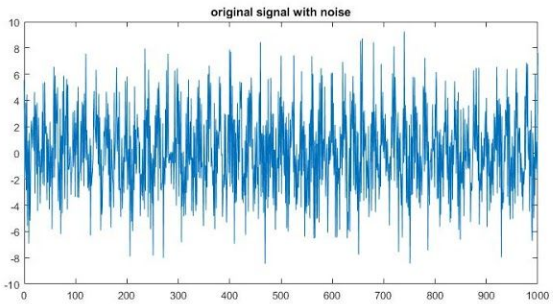
```
SNR rectangular is equal to 5.0621.
SNR triangular is equal to 5.0357.
SNR hanning is equal to 5.201.
SNR hamming is equal to 5.1949.
SNR blackman is equal to 5.1949.
```

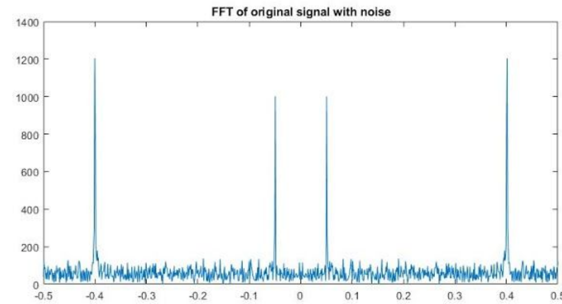
```
>> expt_2_b
SNR input is equal to 4.4386.
SNR rectangular is equal to 6.2075.
SNR triangular is equal to 6.1752.
SNR hanning is equal to 6.3024.
SNR hamming is equal to 6.2984.
SNR blackman is equal to 6.2984.
```

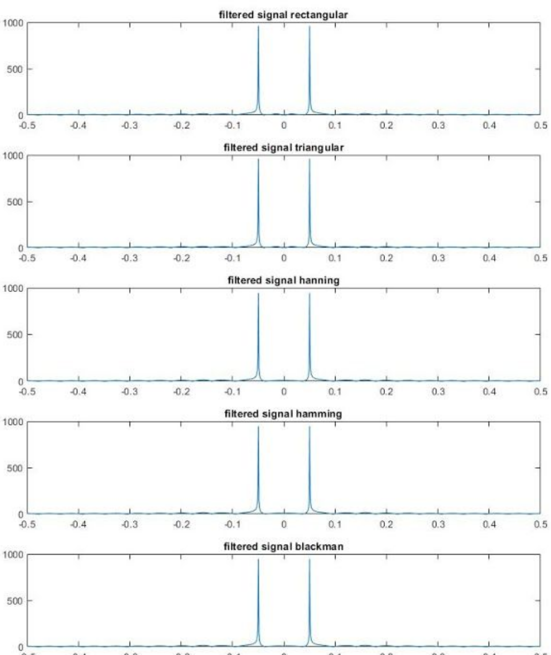
Input SNR = 4.4386 dB
N = 64

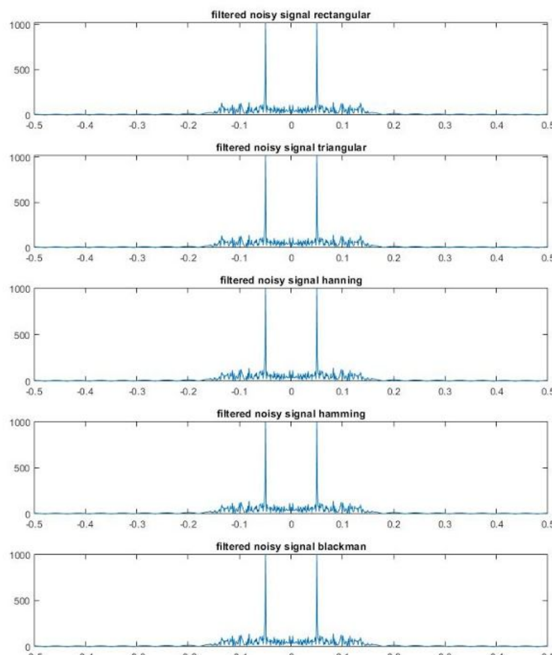












Window Type	Output SNR (dB)
Rectangular	6.2075
Triangular	6.1752
Hanning	6.3024
Hamming	6.2984
Blackmann	6.2984

DISCUSSION

- 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 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.
- 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.

Aadi Swadipto Mondal
17EC10065