Department of Electronics and Telecommunication Engineering

University of Moratuwa

EN2570- Digital Signal processing

FIR FILTER DESIGN PROJECT

NAME : A.Mithunjha

INDEX NO : 170389M

DATE OF SUBMISSION : 02/01/2020

Abstract

This report discusses about the designing of a non-recursive **Band-stop** filter using **Kaiser window** function. The following sections may give you clear idea about the filter designing, basic theories, result obtained from the MATLAB stimulation also conclusion would be discussed.

Contents

1. INTRODUCTION

There are two methods to design a non-recursive filter. That are,

I. Design using Fourier series
II. Multivariable optimization method

Window Function

An alternative technique for the reduction of Gibbs' oscillations in Fourier series method is to truncate the infinite-duration impulse response $h(nT)$ through the use of a discrete-time window function $w(nT)$. There are variety of windows described in literature in past years. Some of them are,

i. Rectangular
ii. Von hann
iii. Hamming
iv. Blackman
v. Dolph-Chebyshev
vi. Kaiser

In this report Fourier series method is used with Kaiser windowing function for the filter designing.

2. BASIC THEORY

Fourier series

Frequency response of a non-recursive filter can be expressed as fourier series. (period of ω_s and function of ω.

$H\left(e^{j\,\Omega T}\right) = \sum_{n=-\infty}^{\infty} h(nT)e^{-j\,\Omega nT}$ where,

$h(nT) = \frac{1}{\Omega_s}\int_{-\frac{\Omega s}{2}}^{\frac{\Omega_5}{2}} H\left(e^{j\,\Omega T}\right)e^{j\,\Omega nT}\,dw$ where, $\Omega_s = 2\pi/T$

to see the effects in frequency domain, the role of time and frequency are interchanged.

Consider, $z = e^{j\,\Omega T}$.

$H(z) = \sum_{n=-\infty}^{\infty} h(nT)z^{-n}$

this is noncausal and of infinite order since h(nT) is defined over the range $-\infty < n < \infty$.
finite length can be achieved by truncating the impulse response such that,

h(nT)=0 for |n|>M, where M=N-1/2. And causal can be achieved by delaying the impulse response by a period MT seconds. But these two will not change the amplitude response but the phase response. Amplitude response will exhibit gibbs' oscillation. Increasing the filter length will not be useful in this case.

Window function is used as a solution for gibb's oscillation effects, where the window truncate filter response.

$$hw(nT) = h(nT)w(nT)$$

frequency response will be,

Hw(z) = Z [h(nT)w(nT)]

$$H(e^{j\,\Omega' T}) = \frac{T}{2\pi}\int_{0}^{\frac{2\pi}{T}} H(e^{j\,\Omega T})W(e^{j(\Omega-\,\Omega')T})\,d\,\Omega'$$

Where $W(e^{j\,\Omega T})$is a frequency spectrum of window function and the integral is convolution.

Windows are characterized by their **main-lobe width**, BML, which is the bandwidth between the first negative and the first positive zero crossings, and by their **ripple ratio**, R, which is defined as,

R =20log(Amax/AML) **or** r=100* Amax/AML

Where, Amax -maximum side-lobe and AML-main-lobe amplitudes

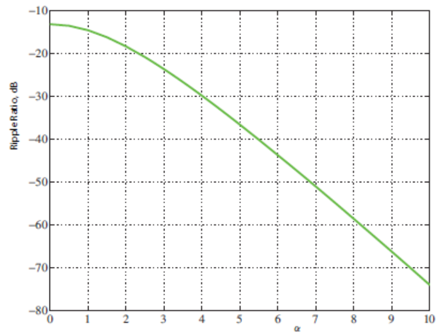
Kaiser window

The window function is given by,

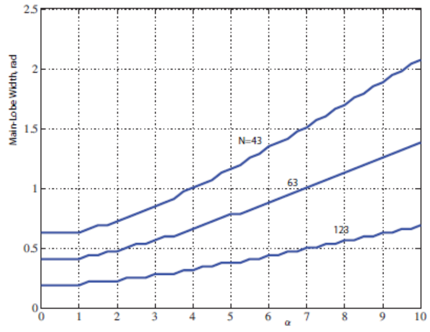
$$w_k[n] = \begin{cases} \dfrac{I_0(\beta)}{I_0(\alpha)}\ ;\ |n| \le \left(\dfrac{N-1}{2}\right) \\ 0\ ;\ otherwise \end{cases}$$

$$I_0(x)\ =\ 1 + \sum_{k=1}^{\infty}\left[\frac{1}{k!}\left(\frac{x}{2}\right)^k\right]^2\ and\ \ \beta = \alpha\sqrt{1-\left(\frac{2n}{N-1}\right)^2}$$

Ripple ratio versus α:



Main-lobe width versus α:



Kaiser’s method can be used to design lowpass, High pass, Band pass, Band stop filters.

Designing Non-recursive Band Stop Filter

Maximum passband ripple, $\tilde{A}_p$

Minimum stopband attenuation, $\tilde{A}_a$

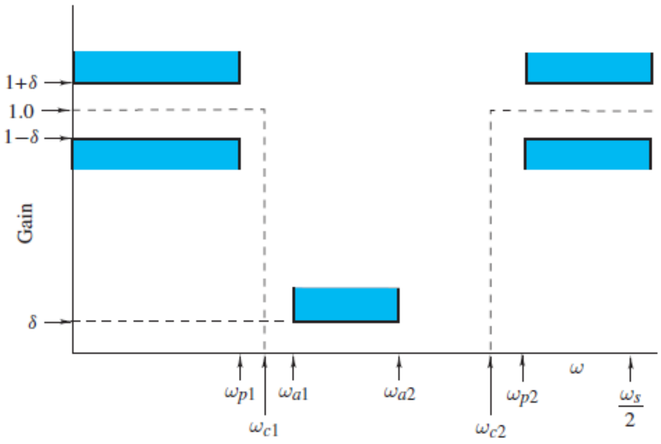
Lower passband edge, Ω_{p1} rad/s

Upper passband edge, Ω_{p2} rad/s

Lower stopband edge, Ω_{a1} rad/s

Upper stopband edge, Ω_{a2} rad/s

Sampling frequency, Ω_s rad/s



From the specification we can define more parameters,

Bt=min[(Ω_p2- Ω_a2),(Ω_a1- Ω_p1)]

Define cut-off frequencies as, $\Omega_{c_1} = \Omega_{p_1} + \frac{B_t}{2}$; $\Omega_{c_2} = \Omega_{p_2} - \frac{B_t}{2}$

Band stop frequency response,

$$H(e^{j\omega}) = \begin{cases} 1\,; & 0 \le |\Omega| \le \Omega_{c_1} \\ 0\,; & \Omega_{c_1} < |\Omega| < \Omega_{c_2} \\ 1\,; & \Omega_{c_2} \le |\omega| \le \pi \end{cases}$$

Impulse response of an ideal band stop filter can be written as,

$$h[n] = \mathrm{h(nT)} = \begin{cases} \dfrac{\sin(\Omega_{c_1} nT) - \sin(\Omega_{c_2} nT)}{n\pi} & ; n \ne 0 \\ \dfrac{2(\Omega_{c_2} - \Omega_{c_1})}{\Omega s} + 1 & ; n = 0 \end{cases}$$

01. choose suitable δ

$$\delta = \min(\tilde{\delta}_p, \tilde{\delta}_a)$$

$$\tilde{\delta}_p = \frac{10^{0.05\widetilde{A_p}} - 1}{10^{0.05\widetilde{A_p}} + 1}$$

$$\tilde{\delta}_a = 10^{-0.05\widetilde{A_a}}$$

02. calculate actual stop band attenuation

Attenuation = -20log(δ)

03. Find α

$$\alpha = \begin{cases} 0 & ; A_a \leq 21 \\ 0.5842(A_a - 21)^{0.4} + 0.07886\,(A_a - 21) & ;\ 21 < A_a \leq 50 \\ 0.1102(A_a - 8.7) & ;\ 50 < A_a \end{cases}$$

04. choose D

$$D = \begin{cases} 0.9222 & ; A_a \leq 21 \\ \frac{A_a - 7.95}{14.36} & ; 21 < A_a \end{cases}$$

05. calculate suitable N

$$N \geq \frac{\Omega_s D}{B_t} + 1$$

06. find window function ($w_k[n]$)

$$w_k[n] = \begin{cases} \frac{I_0(\beta)}{I_0(\alpha)}\ ;\ |n| \leq \left(\frac{N-1}{2}\right) \\ 0\ ;\ otherwise \end{cases}$$

$$I_0(x) \ = \ 1 + \sum_{k=1}^{\infty} \left[\frac{1}{k!}\left(\frac{x}{2}\right)^k\right]^2$$

$$\beta = \alpha\sqrt{1 - \left(\frac{2n}{N-1}\right)^2}$$

07. Find transfer function of the filter

Hw(z) = Z [h(nT).w(nT)]

H'w(z)= z $^{-(N-1)/2}$ Hw(z)

3. RESULTS

Design specifications

A=3 B=8 C=9

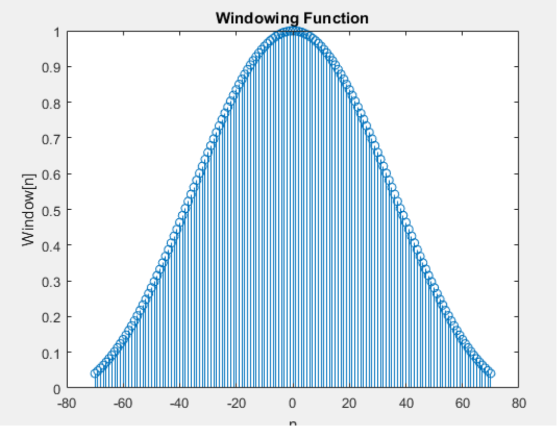
Parameter		values	symbols
Maximum passband ripple, $\tilde{A}_p$	$0.03 + (0.01 \times A)$ dB	0.06dB	Ap
Minimum stopband attenuation, $\tilde{A}_a$	$45+B$ dB	53dB	Aa
Lower passband edge, Ω_p1	$(C \times 100) + 400$ rad/s	1300rad/s	Omega_p1
Upper passband edge, Ω_p2	$(C \times 100) + 950$ rad/s	1850rad/s	Omega_p2
Lower stopband edge, Ω_a1	$(C \times 100) + 500$ rad/s	1400rad/s	Omega_a1
Upper stopband edge, Ω_a2	$(C \times 100) + 800$ rad/s	1700rad/s	Omega_a2
Sampling frequency, Ω_s	$2[(C \times 100) + 1300]$ rad/s	4400rad/s	ts

Parameter derived from equations

α	4.8819
N(length)	141
D	3.1372
δ	0.0022
T	0.0014

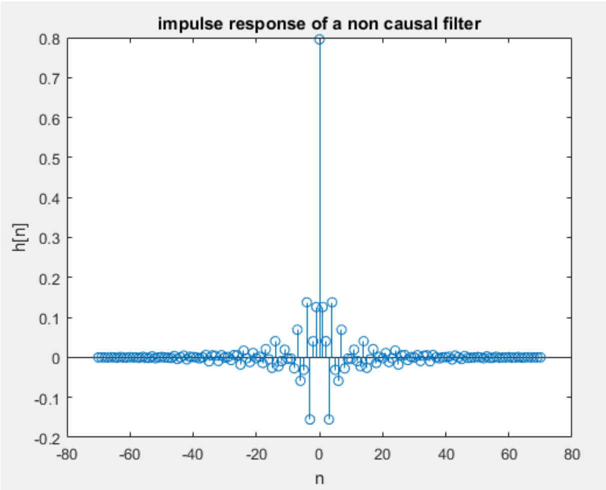
Question 01

A FIR band-stop digital filter with the conjunction of windowing method is achieved, that will satisfy the above specifications.

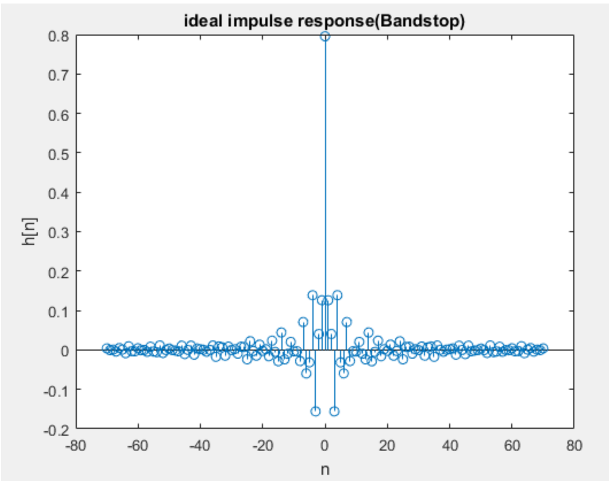


Windowing function

Non-causal impulse response – after windowing

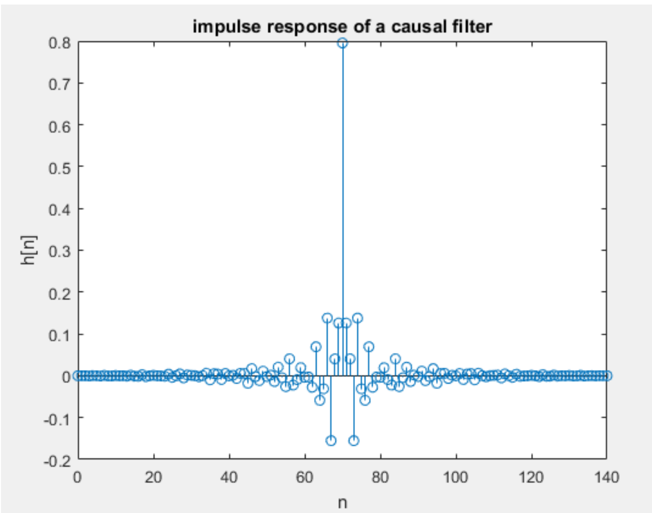


Ideal band-stop filter (there are some minute difference between ideal impulse response and the non-causal impulse response)



Question 02

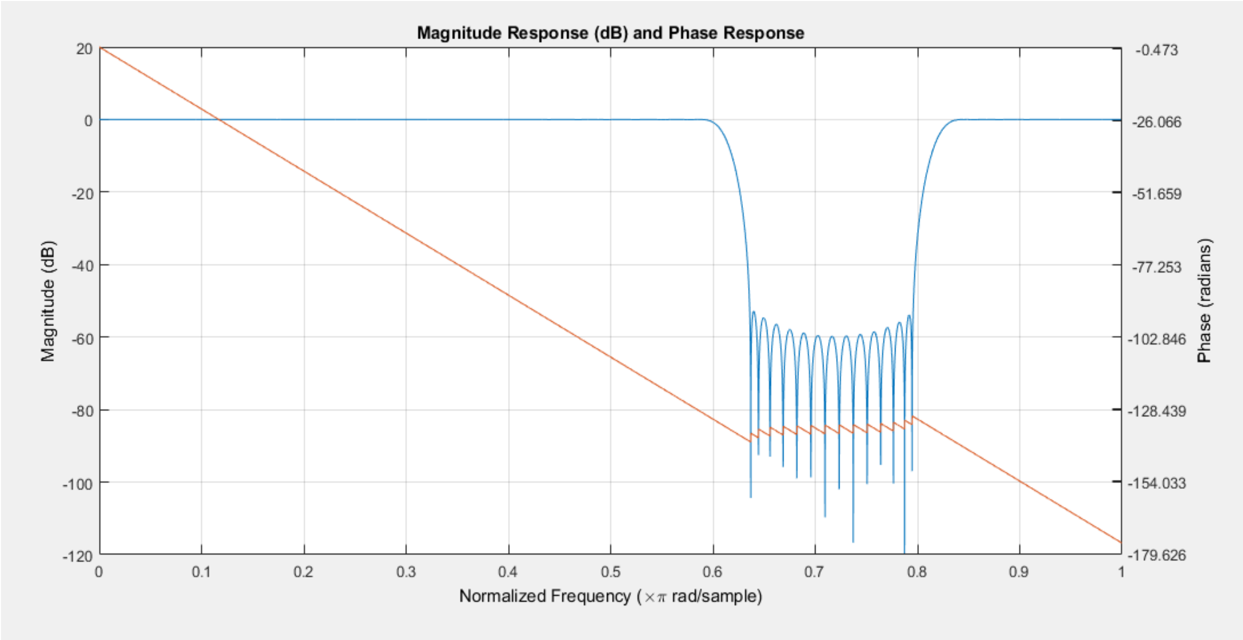
Causal impulse response

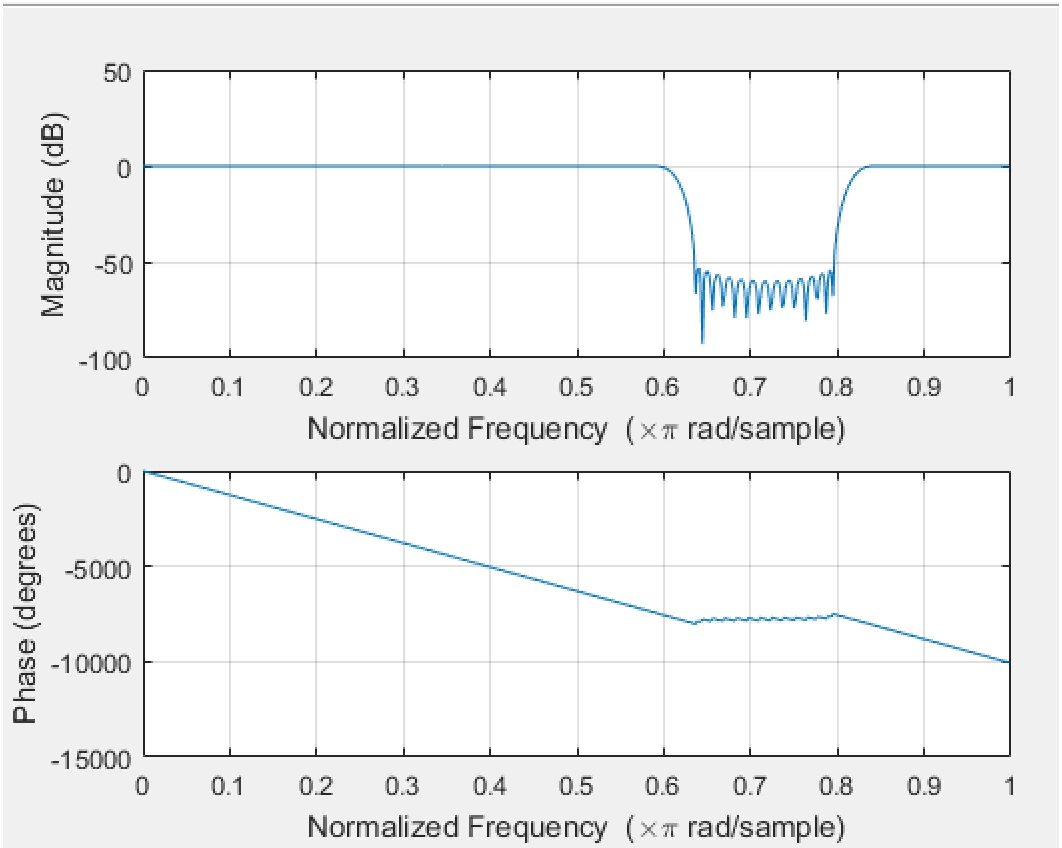


Question 03

Magnitude and phase response of the digital filter using fvtool and freqz respectively,

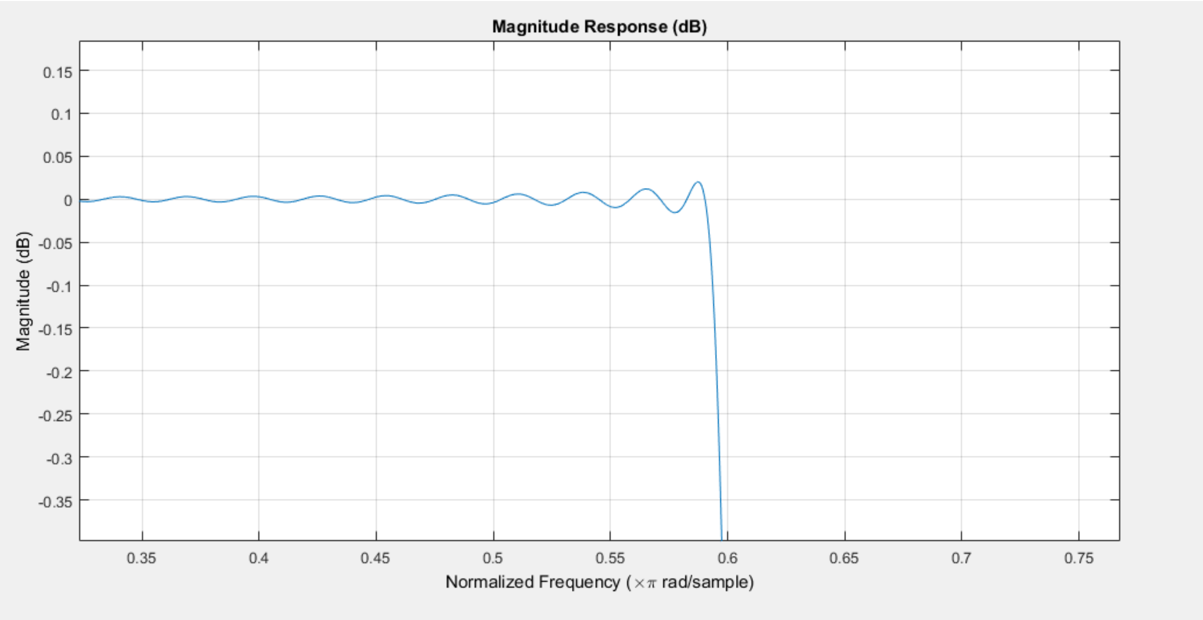
(Blue- magnitude response, red- phase response)



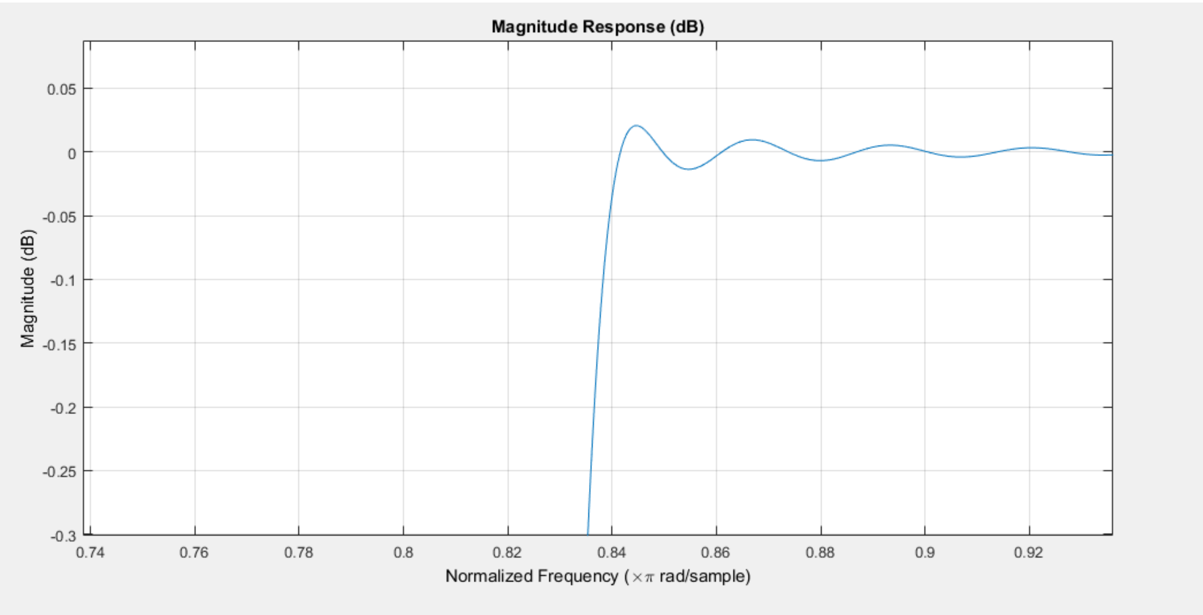


Question 04

Magnitude response of low bandpass



Magnitude response of high bandpass

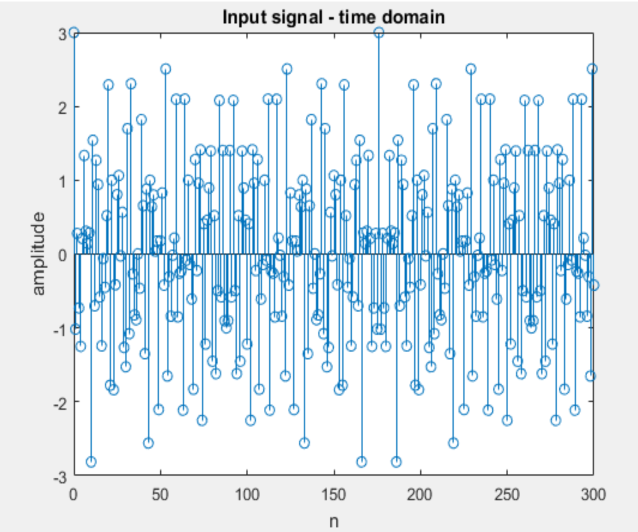


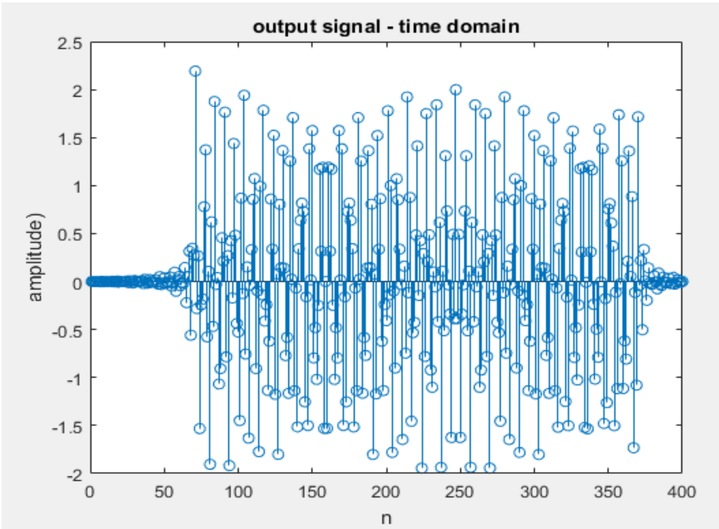
Question 05

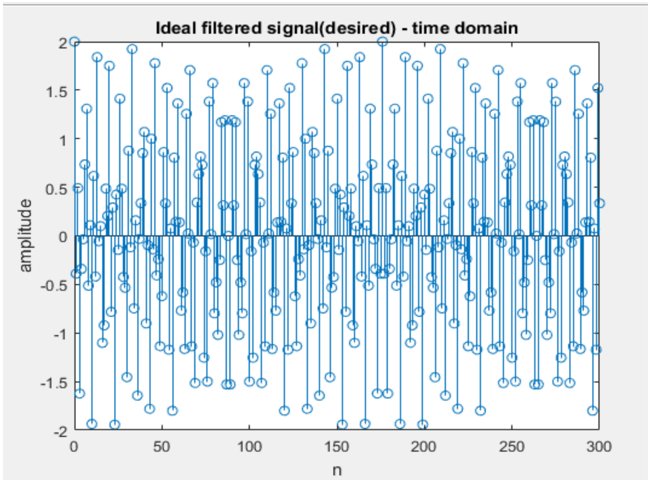
X[nT]= $\sum_{i=1}^{3} \cos(\Omega_i nT)$

300 samples

$\Omega 1$	675 rad/s
$\Omega 2$	1575 rad/s
$\Omega 3$	2000 rad/s

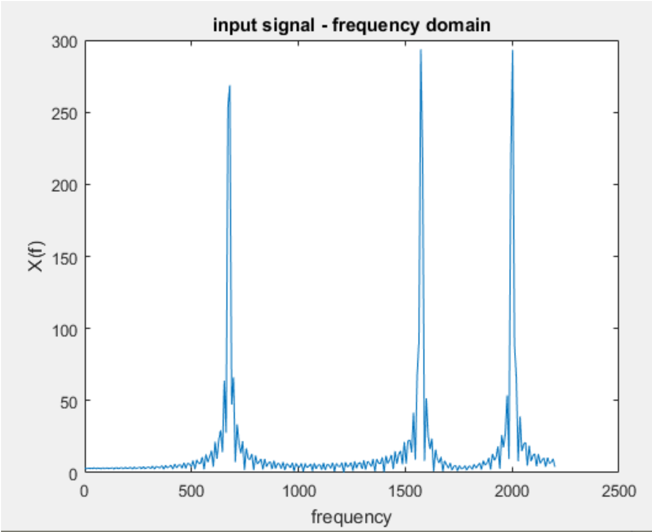


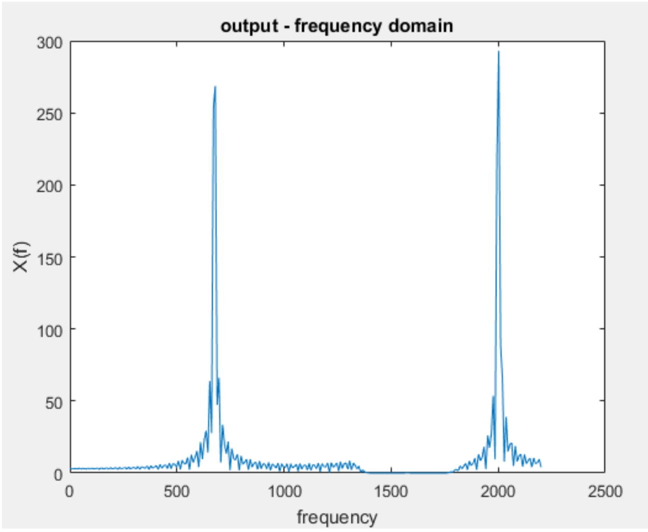


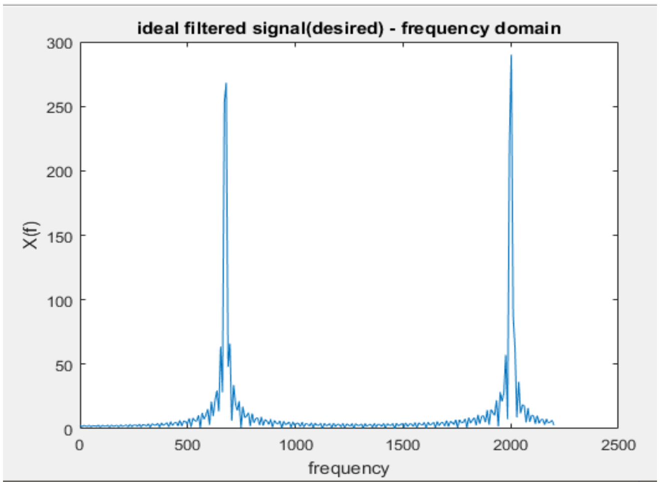


Output through the band-stop signal using Kaiser window is similar to the output of ideal band-stop filter. Except the initial state and end state. Deviations are very less. (except the time taken for the stabilization).

Question 06







Comparing ideal band-stop frequency response and the frequency response of band-stop filter using Kaiser windowing method there are very less deviation. Both are almost similar.

4. CONCLUSION

Designing of FIR digital filters for prescribed specifications using windowing method in conjunction with the Kaiser window is being achieved. We can notice that Band stop filter using Kaiser windowing method produces similar results to ideal band-stop filters. Kaiser window is nearly optimum, it's having most of the energy in main.

5. REFERENCES

I. Digital Signal Processing: Signals, Systems, and Filters 1st Edition by Andreas Antoniou

II. Matlab documentation https://www.mathworks.com/help/matlab/

6. APPENDIX

```
% index no 170389M
A=3;B=8;C=9;

ap=0.03+(0.01*A); %maximum passband ripple
aa=45+B; % minimum stopband attenuation

omega_p1=(C*100)+400; % lower passband edge
omega_p2=(C*100)+950;  % upper passband edge
omega_a1=(C*100)+500; %lower stopband edge
omega_a2=(C*100)+800;  %upper stopband edge
omega_s=2*((C*100)+1300);  % sampling frequency

%finding cutoff frequencies
bt= min((omega_a1-omega_p1),(omega_p2-omega_a2));
omega_c1=omega_p1+(bt/2);
omega_c2=omega_p2-(bt/2);

%finding delta
delta_p=((10^(0.05*ap))-1)/((10^(0.05*ap)) + 1);
delta_a = 10^(-0.05*aa);
delta=min(delta_p,delta_a);

%finding actual stopband attenuation
attenuation=-20*log10(delta);

%find alpha
if (attenuation<=21)
    alpha=0;
elseif (attenuation>21) && (attenuation<=50)
    alpha = 0.5842*(attenuation-21).^0.4 + 0.07886*(attenuation-21);
else
    alpha = 0.1102*(attenuation-8.7);
end
%alpha=0;

%finding D
if (attenuation<=21)
    D=0.9222;
else
    D=(attenuation-7.95)/14.36;
end

%finding N
if mod(ceil(omega_s*D/bt)+1,2)==1      %ceil function
    N=ceil(omega_s*D/bt)+1;
elseif mod(ceil(omega_s*D/bt)+1,2)==0
    N=ceil(omega_s*D/bt)+2;
end

%find beta
n_t = (N-1)/2;
n = -n_t:1:n_t;
beta = alpha*((1-(2*n/(N-1)).^2).^0.5);

%% kaiser window plotting
i_beta = 1;
i_alpha = 1;
```

```matlab
limit=150;
for k = 1 : limit
    i_beta = i_beta + ((1/factorial(k))*(beta/2).^k) .^2;
    i_alpha = i_alpha + ((1/factorial(k))*(alpha/2).^k).^2;
end


window = i_beta./i_alpha ;
figure;
stem(n,window);
xlabel('n');
ylabel('Window[n]') ;
title ( 'Windowing Function');

%% ideal impulse response filter
T=(2*pi)/omega_s;

n_negative = -n_t:1:-1;
h_negative = (1./(pi*n_negative)).*(sin(omega_c1*T.*n_negative)-
sin(omega_c2*T.*n_negative)) ;

n_positive = 1:1:n_t;
h_positive = (1./(pi*n_positive)).*(sin(omega_c1*T.*n_positive)-
sin(omega_c2*T.*n_positive));

n0=0;
h0 = 1+2*(omega_c1-omega_c2)/omega_s;

h = [h_negative,h0,h_positive];
n = [n_negative,n0,n_positive];

figure;
stem(n,h);
xlabel('n');
title('ideal impulse response(Bandstop)');
ylabel('h[n]');

%% non-causal
h_filtered=h.*window;
figure;
stem(n,h_filtered);
xlabel('n');
ylabel('h[n]');
title('impulse response of a non causal filter');

%% causal
n_causal=[0:1:(n_t*2)];
figure;
stem(n_causal,h_filtered);
title('impulse response of a causal filter');
xlabel('n');
ylabel('h[n]');

%% frequency response
fvtool(h_filtered);
freqz(h_filtered);%obtaining the frequency response and corresponding
frequencies
```

```
%%

%define frequencies of input signals%
omega_1=omega_c1/2;
omega_2=omega_c1+(omega_c2-omega_c1)/2;
omega_3=omega_c2+(omega_s/2-omega_c2)/2;
ns=0:1:300;
excitation= cos(omega_1*T.*ns)+cos(omega_2*T.*ns)+cos(omega_3*T.*ns);

%plot input signal- time domain
figure;
stem(ns,excitation);
title('Input signal - time domain');
xlabel('n');
ylabel('amplitude');
%%
%frequency domain - input signal
n_y_1=2^nextpow2(numel(ns));
y=fft(excitation,n_y_1)
func_x=omega_s/2*linspace(0,1,n_y_1/2+1);
figure;
plot(func_x,2*abs(y(1:n_y_1/2+1)));
title('input signal - frequency domain');
xlabel('frequency');
ylabel('X(f)');

%frequency domain - bandstop filter
func_h=fft(h_filtered,n_y_1)
figure;
plot(func_x,2*abs(func_h(1:n_y_1/2+1)));
title('filter - frequency domain');
xlabel('frequency');
ylabel('X(f)');

%%output through filter
out=y.*func_h;
figure;
plot(func_x,2*abs(out(1:n_y_1/2+1)));
title('output - frequency domain');
xlabel('frequency');
ylabel('X(f)');



out_t=ifft(out);
figure;
stem(out_t(1:400));
title('output signal - time domain');
xlabel('n');
ylabel('amplitude)');
%%

%to check plot ideal response
%don't send the signal which is in bandstop frequency
%time domain
x_out=cos(omega_1*T.*ns)+cos(omega_3*T.*ns);
figure;
stem(ns,x_out);
title('Ideal filtered signal(desired) - time domain');
xlabel('n');
ylabel('amplitude');
```

```
%frequency domain
y=fft(x_out,n_y_1)
func=omega_s/2*linspace(0,1,n_y_1/2+1);
figure;
plot(func,2*abs(y(1:n_y_1/2+1)));
title('ideal filtered signal(desired) - frequency domain');
xlabel('frequency');
ylabel('X(f)');
```