DEPARTMENT OF ELECTRONIC AND TELECOMMUNICATION ENGINEERING
UNIVERSITY OF MORATUWA

EN2570 - Digital Signal Processing

FIR Filter Design- Bandstop Filter

H.U.D.B. HAPUTHANTHRI 170208K

This is submitted as a partial fulfillment for the module EN2570

—

January 1, 2020

1 Abstract

This is a detailed report discussing the complete designing procedure of the **Finite Duration Impulse Response (FIR)** bandpass filter for the given specifications. MatLab 2017a is used for the implementation of the design. The filter designing procedure is based on the **Kaiser Windowing Method**. Combination of sinusoidal waves have been used for analyze the performance and the output signal of the filter.

Contents

2 Introduction

This report contains detailed procedure in order to design FIR bandstop filter.
There are mainly 2 types of filters.

- Finite Impulse Response Filter (FIR)
- Infinite Impulse Response Filter (IIR)

When considering the FIR filters, the designing of the FIR method can be done using several techniques/ methods. Windowing method is used for design the filter in this project.
There are number of types of windows can be used to design a filter.

- Rectangular
- Von Hann
- Hamming
- Blackman
- Dolp-Chebyshev
- Kaiser
- Ultraspherical Window

Kaiser Window is used to design the filter in this project due to the flexibility of the Kaiser window. Kaiser window can be optimized in order to obtain the filter with required specifications.
For the frequency representations, Fast Fourier Transform (FTT) and for testing the filter, A sinosoidal signal which has the frequency components from upper passband, lower passband, stopband is used. The output of the filter has been compaired with the ideal output signal.

3 Method

3.1 Filter Specifications

Filter Specifications of the desired Bandstop filter.

Specification	Symbol	Value	Units
Maximum pass band ripple	$\tilde{A}_p$	0.05	dB
Maximum stop band attenuation	$\tilde{A}_a$	45	dB
Lower pass band egde	Ω_{p1}	1300	rads^{-1}
Upper pass band edge	Ω_{p2}	1750	rads^{-1}
Lower stop band egde	Ω_{a1}	1200	rads^{-1}
Upper stop band edge	Ω_{a2}	1600	rads^{-1}
Sampling frequency	Ω_s	4200	rads^{-1}

Table 1: Required Filter Specifications for the Required Bandstop Filter

3.2 Derived Filter Specifications

Filter characteristics those are derived based on the specifications of the filter in order to design the Kaiser window and the filter.

Specification	Symbol	Derivation	Value	Units
Lower transition width	B_{t1}	$\Omega_{a1} - \Omega_{p1}$	100	rads^{-1}
Upper transition width	B_{t2}	$\Omega_{p2} - \Omega_{a2}$	150	rads^{-1}
Critical transition width	B_t	$min[B_{t1}, B_{t1}]$	100	rads^{-1}
Lower cutoff frequency	Ω_{c1}	$\Omega_{p1} + \frac{B_t}{2}$	1250	rads^{-1}
Upper cutoff frequency	Ω_{c2}	$\Omega_{p2} - \frac{B_t}{2}$	1700	rads^{-1}
Sampling period	T	$\frac{2\pi}{\Omega_s}$	0.0015	s

Table 2: Derived Filter Specifications

3.3 Derivation of the Kaiser Window Parameters

The Kaiser Window function is given by

$$w_K(nT) = \begin{cases} I_0(\beta)/I_0(\alpha) & for\ |x| \leq \frac{N-1}{2} \\ 0 & otherwise \end{cases}$$

where α is an independent parameter and

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

By using the parameters in Table 1 and the derived parameters in Table 2 above, we calculate α and N as

$$\delta = min\left(\tilde{\delta_p}, \tilde{\delta_a}\right)$$

where

$$\tilde{\delta_p} = \frac{10^{0.05\tilde{A_p}} - 1}{10^{0.05\tilde{A_p}} + 1} \qquad and \qquad \tilde{\delta_a} = 10^{-0.05\tilde{A_a}}$$

Now, with the defined δ, we calculate the actual stop band loss

$$A_a = -20log|\delta|$$

We can chose α as

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

A parameter D is chosen in order to obtain N, as

$$D = \begin{cases} 0.9222 & for\ A_a \leq 21dB \\ \frac{A_a - 7.95}{14.36} & for\ A_a > 21dB \end{cases}$$

N is chosen such that it is the smallest odd integer value satisfying the inequality

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

The Kaiser Window Parameters thus obtained are given below in Table 3.

Parameter	Value	Units
δ	0.0029	-
A_a	50.8175	dB
A_p	0.05	dB
α	4.6231	-
$I_0(\alpha)$	19.4864	-
D	2.9852	-
N	127	-

Table 3: Derived Kaiser Window Parameters

3.4 Derivation of the ideal impulse response

The frequency response of an ideal bandstop filter with cutoff frequencies Ω_{c1} and Ω_{c2} is given by

$$H(e^{j\Omega T}) = \begin{cases} 1 & for\ 0 \leq |\Omega| \leq \Omega_{c1} \\ 0 & for\ \Omega_{c1} \leq |\Omega| \leq \Omega_{c2} \\ 1 & for\ \Omega_{c2} \leq |\Omega| \leq \frac{\Omega_s}{2} \end{cases}$$

Hence, using the inverse fourier transform,

$$h(nT) = \begin{cases} \frac{2}{\Omega_s}(\Omega_{c2} - \Omega_{c1}) & for\ n = 0 \\ \frac{1}{n\pi}(sin(\Omega_{c2}nT) - sin(\Omega_{c1}nT) & otherwise \end{cases}$$

3.5 Derivation of the causal impulse response of windowed filter

Finite order non-causal impulse response of the windowed filter $h_w(nT)$ can be obtained by the multiplication of the Ideal impulse response $h(nT)$ by the Kaiser Window function $w_K(nT)$,

$$h_w(nT) = w_K(nT)h(nT)$$

$\mathscr{Z}$-tranform of $h_w(nT)$ can be obtained as

$$H_w(z) = \mathscr{Z}[h_w(nT)] = \mathscr{Z}[w_K(nT)h(nT)]$$

In order to derive the frequency response of the causal impulse response,

$$H'_w(z) = z^{-(N-1)/2}H_w(z)$$

3.6 Filter Evaluation

To evaluate the performance of the generated filter, we use an input signal $x(nT)$ which is the sum of three sinusoidal signals, each of which has a frequency in the lower pass band, the stop band and the upper pass band as shown in Table : 4.

$$x(nT) = \sum_{i=1}^{3} Cos(\Omega_i nT)$$

Parameter	Derivation	Value	Units
Ω_1	$\frac{\Omega_{c1}}{2}$	625	rads^{-1}
Ω_2	$\frac{\Omega_{c1}+\Omega_{c2}}{2}$	1475	rads^{-1}
Ω_3	$\frac{\Omega_{c1}+\Omega_s/2}{2}$	1900	rads^{-1}

Table 4: Input Frequency Components

4 Results

4.1 Frequency and Time domain plots of the filter

Frequency responses and the time domain representations of the filter and the Kaiser window which is obtained is shown below. Fig: 4.1.4, 4.1.5 shows the zoomed passband ripples of the lower passband and the upper passband.

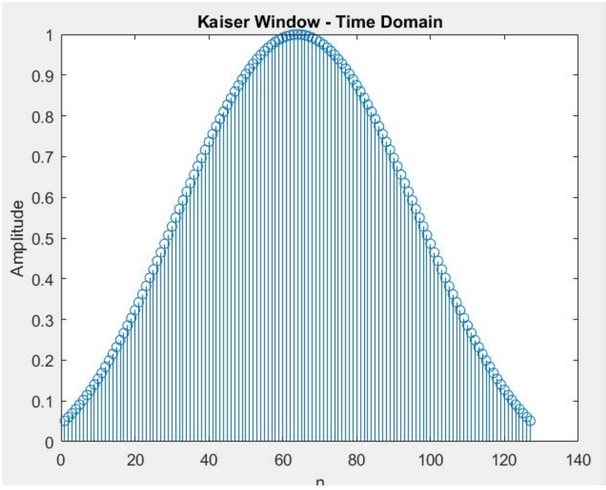

Figure 4.1.1: Kaiser Window

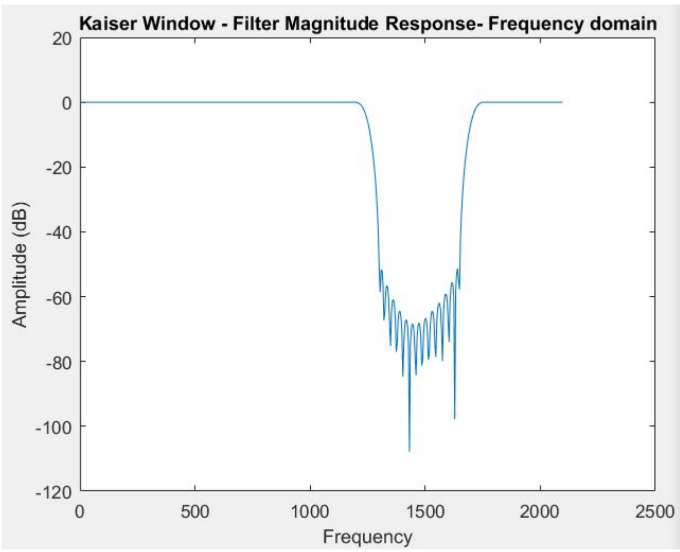

Figure 4.1.2: Frequency Response of the Filter

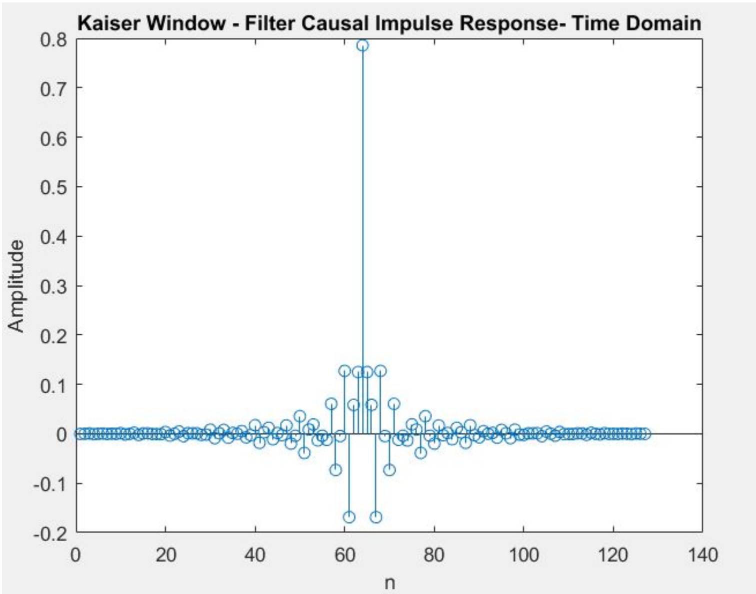

Figure 4.1.3: Impulse Response- Time Domain

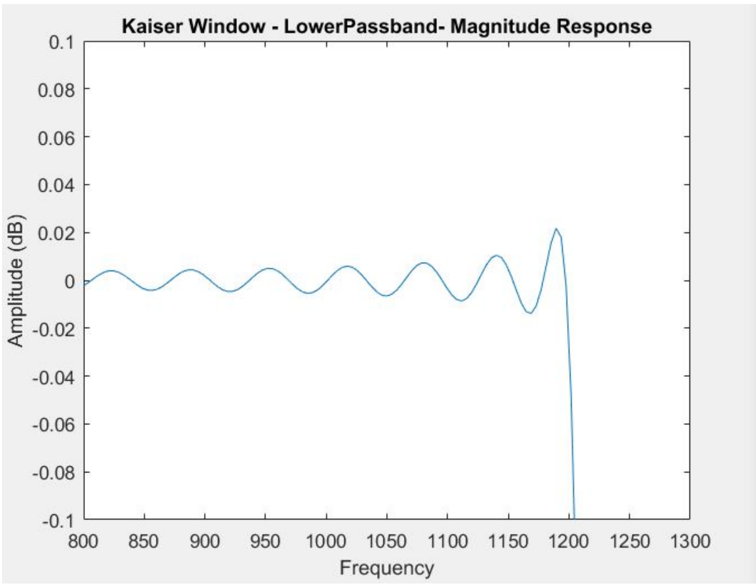

Figure 4.1.4: Magnitude Response of the lower passband (in frequency)

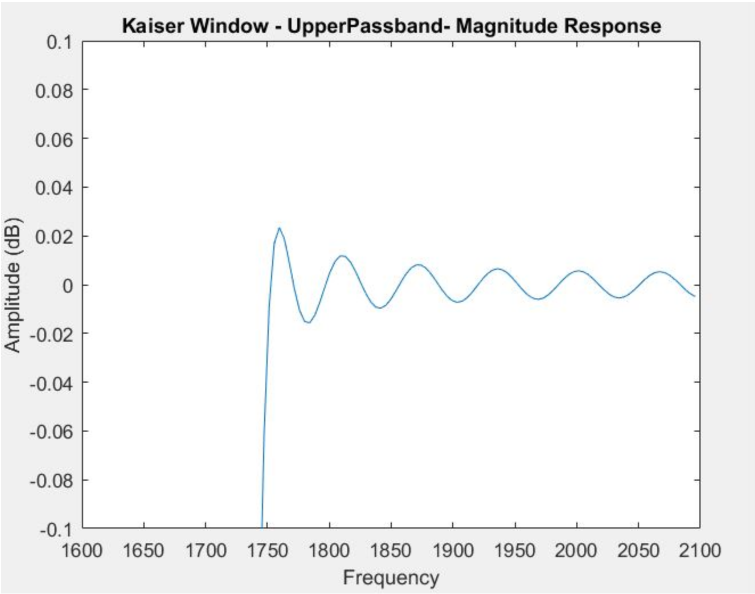

Figure 4.1.5: Magnitude Response of the upper passband (in frequency)

4.2 Inputs and Outputs of the filter

Input x(nT) is given to the filter.

$$x(nT) = \sum_{i=1}^{3} Cos(\Omega_i nT)$$

Parameter	Derivation	Value	Units
Ω_1	$\frac{\Omega_{c1}}{2}$	625	rads^{-1}
Ω_2	$\frac{\Omega_{c1}+\Omega_{c2}}{2}$	1475	rads^{-1}
Ω_3	$\frac{\Omega_{c1}+\Omega_s/2}{2}$	1900	rads^{-1}

Table 5: Input Frequency Components

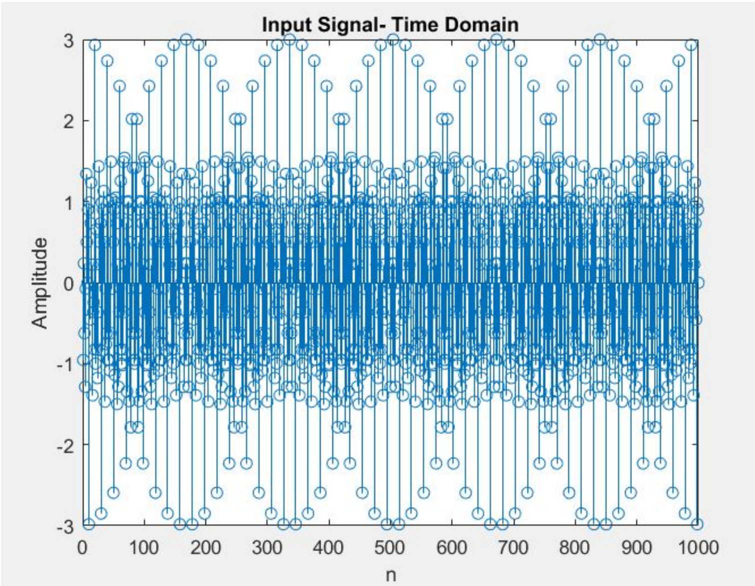

Figure 4.2.1: Input Signal

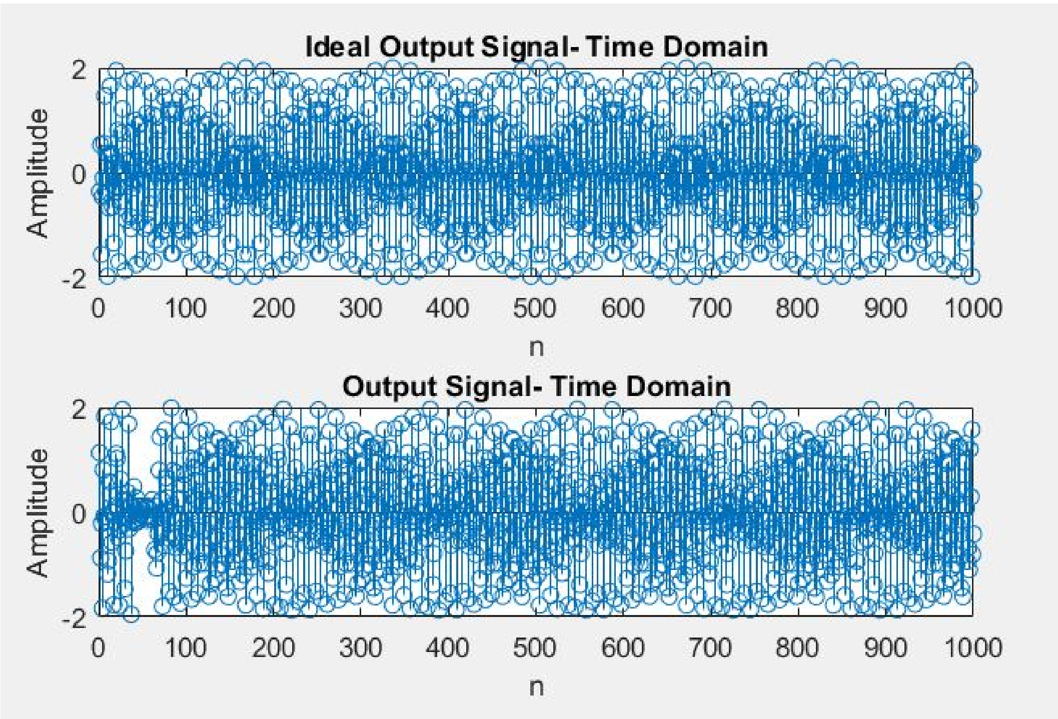

Figure 4.2.2: Output Signal and Expected Output

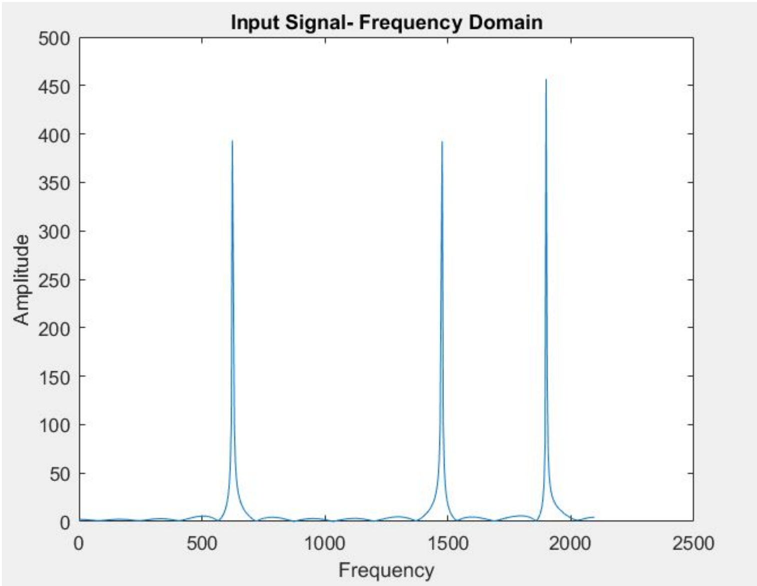

Figure 4.2.3: Frequency Magnitude Response of Input

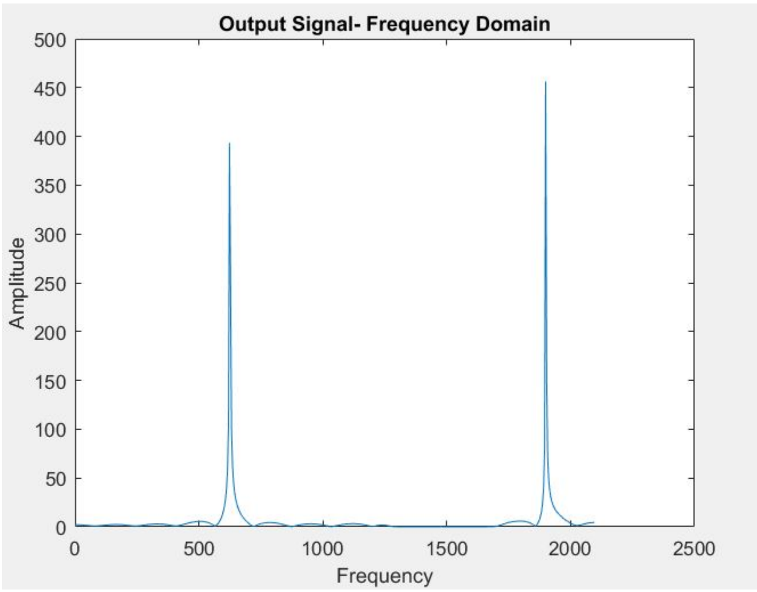

Figure 4.2.4: Frequency Magnitude Response of output

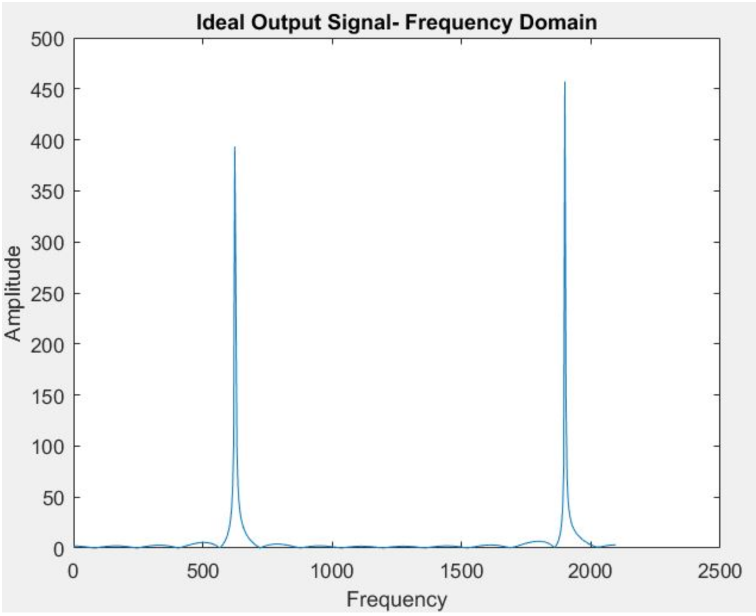

Figure 4.2.5: Frequency Magnitude Response of Ideal Output

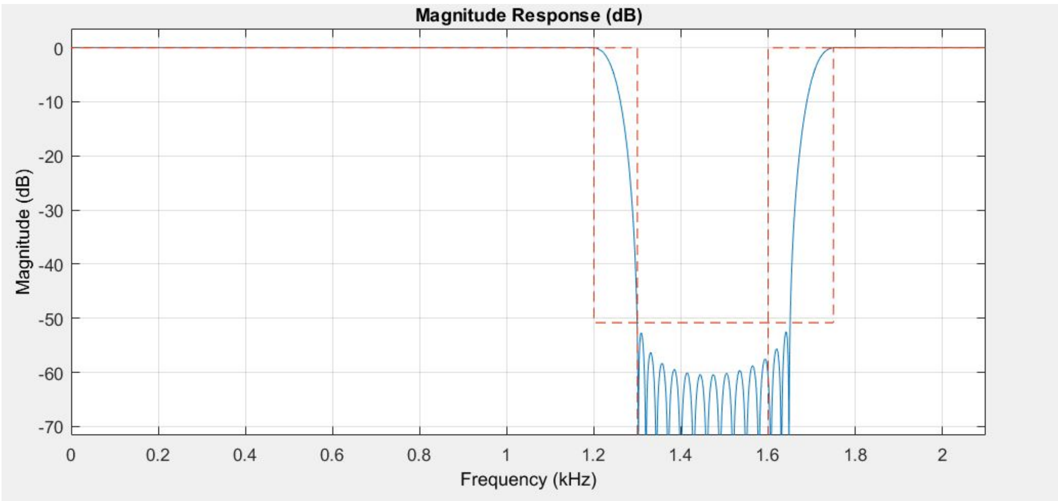

Figure 4.2.6: Filter Response - Using designfilt() on MatLab

5 Discussion

By observing the results which have been obtained, It can be observed that the requirements of the filter has been achieved. When consider the Fig: 4.1.4 and the Fig: 4.1.5 ,it can be observed that the Maximum passband ripple is less than the

required maximum passband ripple of 0.05dB. When considering the Stopband attenuation, in Fig: 4.1.2, it can be clearly shown that the Attenuation of the stopband (50.8175 dB) is larger than the required attenuation (45.0 dB) and there is no significant phenomenon such as the *Gibbs Phenomenon* in the frequency response of the filter.
When observing the output of the filter for the given signal x(nT) (Fig: 4.2.1), In the frequency response, 1475 $rads^{-1}$ component of the input signal has been filtered out because the cutoff frequencies of the filter are 1250, 1700 $rads-1$.(According to the Fig: 4.2.3, 4.2.4) The output time domain signal is much similar to the expected output time signal. (Fig: 4.2.2) The reason for the insignificant dissimilaries of the expected output signal and the actual output signal, because the input to the filter x(nT) has only contains 1000 samples.
Finally all the specifications required are satisfied by the implemented filter using the Kaiser Window.

6 Conclusion

The Kaiser window can be used for designing the filter with given specifications. Since the ideal filters cannot be implemented due to the length of the time domain response, there should be a way to implement a better filter with the required specifications and using the Kaiser window, the filter has been generated for the given specifications successfully. The parameters of the Kaiser window can be changed in order to obtain the required specifications.
Computational complexity of the Kaiser windowing method is also much less and there is no recursive way in order to obtain the filter function. .Therefore the Kaiser window method is a much flexible method to implementing filters.

7 Acknowledgements

I would like to extend my sincere gratitude to Dr. Chamira U.S. Edussooriya for his valuable support and guidance in making the project success.

8 References

[1] Antoniou.A, "Digital Signal Processing - Signals Systems and Filters"(1st ed.).(2005). McGraw-Hill.

Appendix

- FIR.m

```
clc;
clear all;
close all;

A=2;B=0;C=8;

Ap=0.03 + (0.01 * A);
Aa=45+B;
Op1=(C * 100) + 400;
Op2=(C * 100) + 950;
Oa1=(C * 100) + 500;
Oa2=(C * 100) + 800;
Os=2*((C * 100) + 1300);
Ap
Aa
Bt=min([(Oa1-Op1),(Op2-Oa2)]);

Oc1=Op1+ Bt/2;
Oc2=Op2-Bt/2;
T=2*pi/Os;

%compute h[n]
%%%

delta_p= (10^(0.05*Ap)-1)/(10^(0.05*Ap)+1);
delta_a= 10^(-0.05*Aa);
delta=min(delta_p,delta_a);
Aa=-20*log10(delta);
Ap=20*log10((1+delta)/(1-delta));
if Aa<=21
    alpha=0;
    D=0.9222;
elseif 21<Aa<=50
    alpha=0.5842*(Aa-21)^0.4+0.07886*(Aa-21);
    D=(Aa-7.95)/14.36;
else
    alpha=0.1102*(Aa-8.7);
    D=(Aa-7.95)/14.36;
end
```

```
N=ceil(Os*D/Bt+1)+~mod(ceil(Os*D/Bt+1),2);

%%%%%%%%%%%%%%
n=[-(N-1)/2:1:(N-1)/2];
%%%%%%%%%%%%%%

beta=alpha*(1-((2.*n)./(N-1)).^2).^0.5;
h_d=(sin(Oc1*T*n)-sin(Oc2*T*n))./(n*pi);
h_d((N+1)/2)=1+2*(Oc1-Oc2)/Os;


inf_=100;
w= I_note(beta,inf_)./I_note(alpha,inf_);

h=w.*h_d;

figure;
stem(w);
title('Kaiser Window - Time Domain');
xlabel('n');
ylabel('Amplitude');
figure;
stem(h);
title('Kaiser Window - Filter Causal Impulse Response- Time
    Domain');
xlabel('n');
ylabel('Amplitude');
hold;
[H, H_freq_DT]=freqz(h,1);
H_=fft(h);
freq=H_freq_DT*Os/(2*pi);
figure;
H_mag_log=20*log10(abs(H));
plot(freq,H_mag_log);
xlabel('Frequency');
ylabel('Amplitude (dB)');
title('Kaiser Window - Filter Magnitude Response- Frequency
    domain');


%%%%%%%%%%%%%%%%%%%%%%%%%% 2----ANOTHER TWO PLOTS - LOW/HIGH ...
figure;
```

```
plot(freq,H_mag_log);
axis([800,1300,-0.1,0.1])
xlabel('Frequency');
ylabel('Amplitude (dB)');
title('Kaiser Window - LowerPassband- Magnitude Response');
figure;
plot(freq,H_mag_log);
axis([1600,2100,-0.1,0.1])
xlabel('Frequency');
ylabel('Amplitude (dB)');
title('Kaiser Window - UpperPassband- Magnitude Response');


w_rect=[];
%h_rect=w_rect.*h_d;
h_rect=h_d;
figure;
stem(h_rect);
title('Rectangular Window - Filter Causal Impulse Response- Time Domain');
xlabel('n');
ylabel('Amplitude');
hold;
[H_rect, H_freq_DT_rect]=freqz(h_rect,1);
freq_rect=H_freq_DT_rect*Os/(2*pi);
figure;
H_mag_log_rect=20*log10(abs(H_rect));
plot(freq_rect,H_mag_log_rect);
xlabel('Frequency');
ylabel('Amplitude (dB)');
title('Rectangular Window - Filter Magnitude Response- Frequency domain');



%%%%%%%%%%%%%%%%%%%%%%% INPUT - OUTPUT
O1=Oc1/2;
O2=(Oc1+Oc2)/2;
O3=(Oc2+Os/2)/2;
width=[1:1000];
x=cos(O1*T*width)+cos(O2*T*width)+cos(O3*T*width);
x_ideal=cos(O1*T*width)+cos(O3*T*width);

```

```
figure;
[X,freq_DT]=freqz(x,1);
freq=freq_DT*Os/(2*pi);
plot(freq,abs(X));
xlabel('Frequency');
ylabel('Amplitude');
title('Input Signal- Frequency Domain');

figure;
stem(x);
xlabel('n');
ylabel('Amplitude');
title('Input Signal- Time Domain');

figure;
X_filtered=X.*H;
x_filtered=abs(ifft(X_filtered));
plot(freq,abs(X_filtered));
title('Output Signal- Frequency Domain');
xlabel('Frequency');
ylabel('Amplitude');

figure;
[X_ideal,freq_DT_ideal]=freqz(x_ideal,1);
freq_id=freq_DT_ideal*Os/(2*pi);
plot(freq_id,abs(X_ideal));
title('Ideal Output Signal- Frequency Domain');
xlabel('Frequency');
ylabel('Amplitude');



[u_,d_]=invfreqz(X_filtered,freq_DT,length(x)-1,1);
figure;
subplot(311);stem(x_ideal);
title('Ideal Output Signal- Time Domain');
xlabel('n');
ylabel('Amplitude');
subplot(312);stem(u_);
title('Output Signal- Time Domain');
xlabel('n');
ylabel('Amplitude');
axis([0 1000 -2 2]);
```

```
%%%% Impulse Response by fvtool
%%%% Freq represetation by fvtool
fvtool(h);

%%%% Impule Resp-designfilt()
%%%% Freq resp-designfilt()

bandstop = designfilt('bandstopfir', 'PassbandFrequency1', Op1,
    'StopbandFrequency1', Oa1, 'StopbandFrequency2', Oa2,
    'PassbandFrequency2', Op2, 'PassbandRipple1', Ap,
    'StopbandAttenuation', Aa, 'PassbandRipple2', Ap,
    'SampleRate', 4200, 'DesignMethod', 'kaiserwin');
fvtool(bandstop);
```

- $I_{note}.m$

```
function I=I_note(x,inf_)
k=[1:inf_]';
I=1+sum((((x./2).^k)./factorial(k)).^2);
end
```

[Github repo]