DEPARTMENT OF ELECTRONIC & TELECOMMUNICATION ENGINEERING

UNIVERSITY OF MORATUWA

EN 2510 - DIGITAL SIGNAL PROCESSING PROJECT REPORT

DESIGN OF A FINITE-DURATION IMPULSE RESPONSE (FIR) BANDSTOP FILTER

NAME – D.M.D.P. Dissanayake

INDEX NUMBER – 170147V

THIS REPORT IS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE MODULE EN 2510 - DIGITAL SIGNAL PROCESSING.

30th of December 2019

Abstract

This report includes the details of designing a Finite-Duration Impulse Response Filter (Band stop) for a given set of specifications. The method used here is the Kaiser window method and all the details of the window is stated in the report. Both the lower and upper passbands of the filter and the other figures stated in the report will prove that the filter is up to the given specifications. Finally, the performance of the designed filter is compared with the output of an ideal filter in both time and frequency domains. The implementation was done using MATLAB. The code is attached to the appendix.

Table of Contents

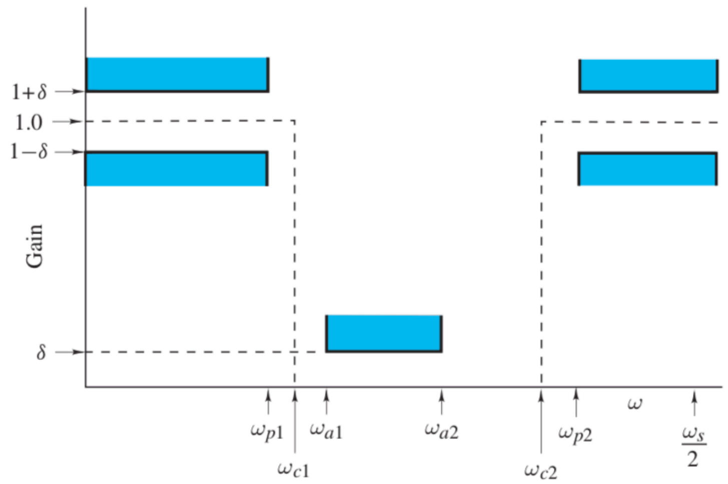
Introduction

This report describes the procedure of designing a band stop FIR digital filter for a given set of specifications. A band stop filter will allow all the frequencies to pass except for a given range. The range starts from the lower cut off frequency to the upper cut off frequency.

The windowing method, also named as the Fourier series method is used in designing the filter. This is a closed form direct approach. Out of the windowing functions available, this filter is designed using the Kaiser window. MATLAB version R2017b is used for the implementation of the band stop filter.

The discrete Fourier transforms of the input and output signals were obtained using the Fast Fourier transform functions available in MATLAB. The input signal containing of three different frequencies (one in the lower passband one in the stopband and one in the upper passband) is then sent through the filter to compare its performance with an ideal band stop filter i.e. one that has a gain of 1 in passbands and 0 in stopband.

The following figure depicts the given specifications for the filter.



Parameter	Value
Maximum passband ripple, $\tilde{A}_p$	0.04 dB
Minimum stopband attenuation, $\tilde{A}_a$	49 dB
Lower passband edge, Ω_{p1}	1100 rad/s
Upper passband edge, Ω_{p2}	1650 rad/s
Lower stopband edge, Ω_{a1}	1200 rad/s
Upper stopband edge, Ω_{a2}	1500 rad/s
Sampling frequency, Ω_s	4000 rad/s

Basic Theory

Introduction to windowing

There are two methods that can be used to design a non-recursive filter.

- By using the Fourier series with a class of functions known as the window functions
- By using multivariable optimization method also known as the weighted-Chebyshev method

Digital filters have a periodic frequency response with period equal to the sampling frequency.

$$H(e^{j(\omega+k\omega_s)T}) = H(e^{j\omega T})$$

This can be represented in Fourier series as;

$$H(e^{j\omega T}) = \sum_{n=-\infty}^{\infty} h(nT)e^{-j\omega nT}$$

$$h(nT) = \frac{1}{\omega_s}\int_{-\omega_s/2}^{\omega_s/2} H(e^{j\omega T})e^{j\omega nT}\, d\omega$$

So, the transfer function can be written for the impulse response of a non-recursive filter as;

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

It can be seen that the Fourier series coefficients are being defined over the range minus infinity to plus infinity, so the filter will be of infinite length and the filter is noncausal as there will be a non-negative response for the negative time. These problems are mitigated by having a finite filter length that is achieved by truncating the impulse response such that;

$$h(nT) = 0 \quad \text{for} \quad |n| > M$$

where $M = (N-1)/2$.

The causal filter is obtained by delaying the impulse response by a period MT seconds or by M sampling periods. Then the transfer function and the frequency response will be of the form;

$$H'(z) = z^{-M}\sum_{n=-M}^{M} h(nT)z^{-n} \qquad\qquad H'(e^{j\omega T}) = e^{-jM\omega T}\sum_{n=-M}^{M} h(nT)e^{-jn\omega T}$$

The drawback here that occurs is the Gibbs' oscillations due to the truncations of the Fourier series. To reduce this, we truncate the infinite duration impulse response through the use of a discrete time window function.

$$h_w(nT) = w(nT)h(nT)$$

$$H_w(e^{j\omega T}) = \frac{T}{2\pi}\int_0^{2\pi/T} H(e^{j\varpi T})W(e^{j(\omega-\varpi)T})\, d\varpi$$

There are a number of windowing functions available out of that this report will showcase the implementation of the Kaiser window.

Kaiser Window

The Kaiser window function is given by

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

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

α is an independent parameter, and $I_0(x)$ is a *zeroth-order modified Bessel function* of the first kind.

The value of alpha and the filter length, N, that would give us the desired filter will be calculated using some formulas.

Steps used to design a Kaiser window for a Band Stop filter

1. Determine the frequency response- assume an idealized frequency response

$$B_t = \min\left[(\omega_{a1} - \omega_{p1}), (\omega_{p2} - \omega_{a2})\right]$$

$$H(e^{j\omega T}) = \begin{cases} 1 & \text{for } 0 \leq |\omega| \leq \omega_{c1} \\ 0 & \text{for } \omega_{c1} < |\omega| < \omega_{c2} \\ 1 & \text{for } \omega_{c2} \leq |\omega| \leq \omega_s/2 \end{cases} \qquad h(nT) = \begin{cases} 1 + \frac{2(\omega_{c1} - \omega_{c2})}{\omega_s} & \text{for } n = 0 \\ \frac{1}{n\pi}(\sin\omega_{c1}nT - \sin\omega_{c2}nT) & \text{otherwise} \end{cases}$$

where

$$\omega_{c1} = \omega_{p1} + \frac{B_t}{2}, \quad \omega_{c2} = \omega_{p2} - \frac{B_t}{2}$$

2. Choose a delta value such that the actual passband attenuation is equal or less than the needed pass band attenuation and the minimum stop band attenuation is equal or greater than the desired stop band attenuation.

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

$$\text{where} \qquad \tilde{\delta}_p = \frac{10^{0.05\tilde{A}_p} - 1}{10^{0.05\tilde{A}_p} + 1} \quad \text{and} \quad \tilde{\delta}_a = 10^{-0.05\tilde{A}_a}$$

3. With the required delta defined the stop band attenuation is then decided as

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

4. Choose alpha such that

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

5. Choose parameter D as;

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

Then select the lowest odd value of N that would satisfy the inequality

$$N \geq \frac{\omega_s D}{B_t} + 1 \quad \text{where} \quad B_t = \omega_a - \omega_p$$

6. Form the Kaiser window as below;

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

7. Then the filter will be of the form;

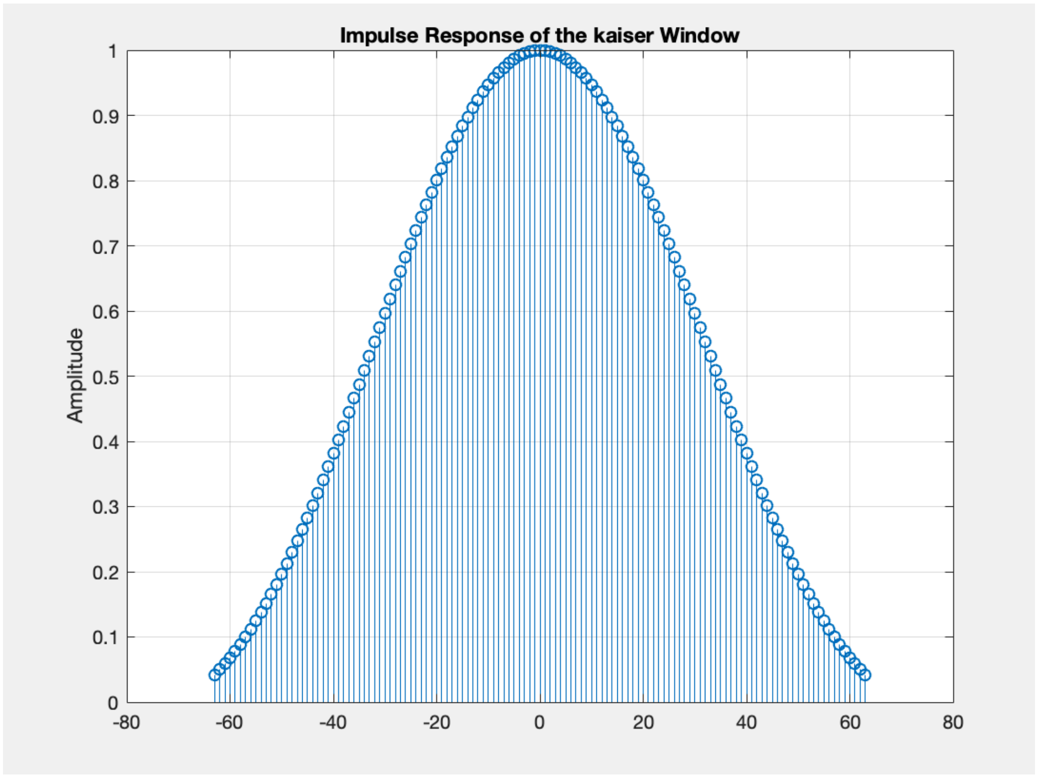
$$H'_w(z) = z^{-(N-1)/2} H_w(z) \quad \text{where } H_w(z) = \mathcal{Z}[w_K(nT)h(nT)]$$

Results

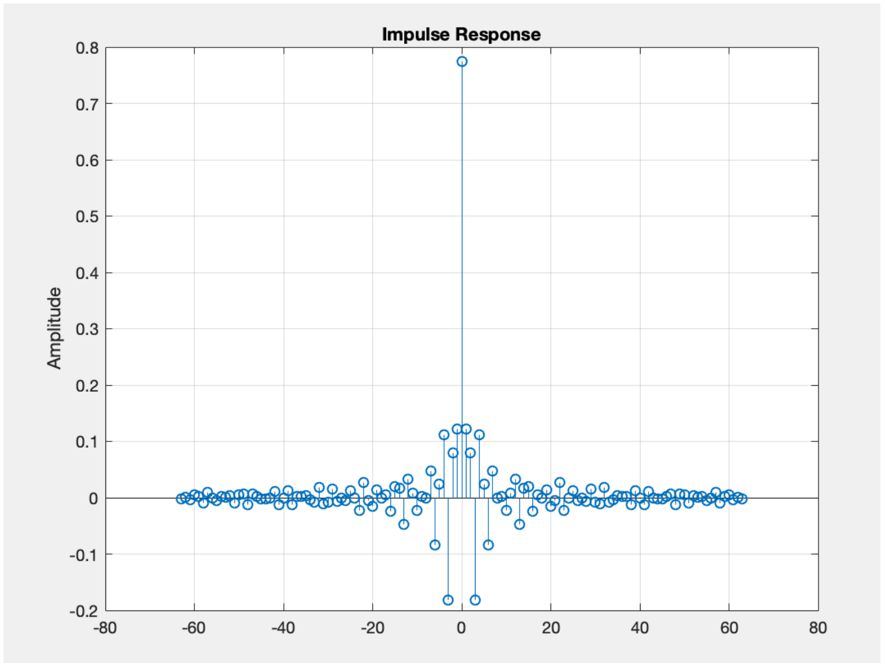
From the specifications available above the following were the computed essential parameters when designing the filter.

Actual stopband Attenuation (Aa)	52.7557 dB
Bt	100
Lower cut off frequency	1150
Upper cut off frequency	1600
Alpha	4.8549
D	3.1202
N	127

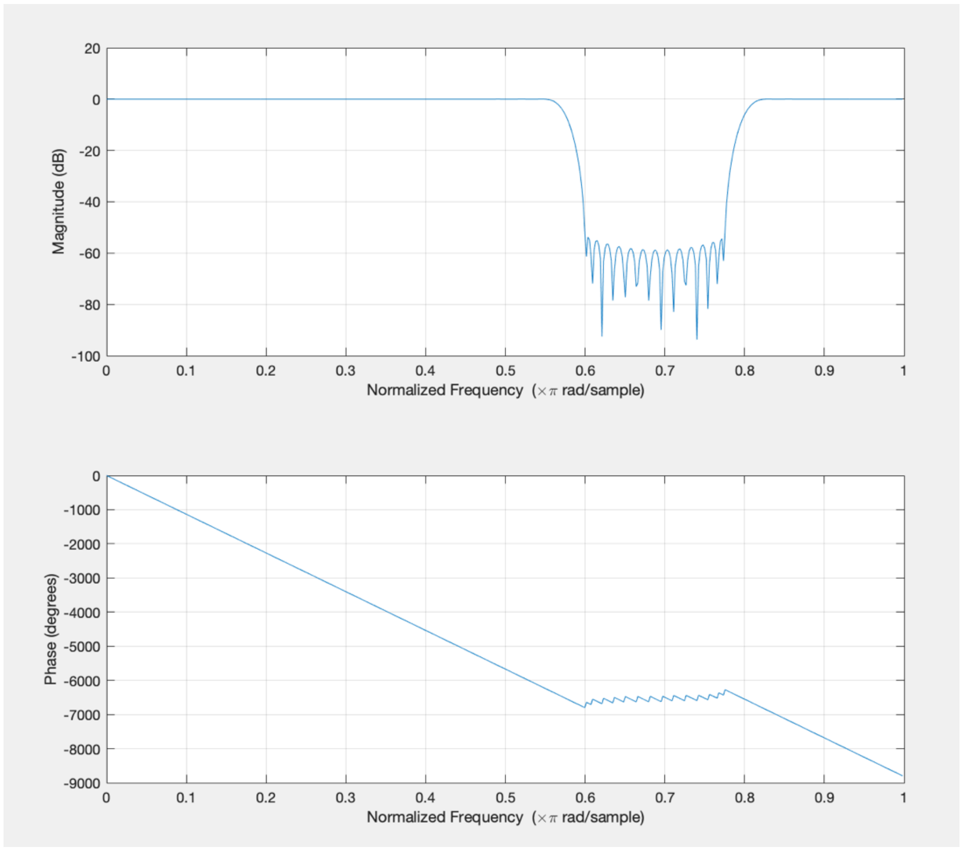
1. Kaiser Window

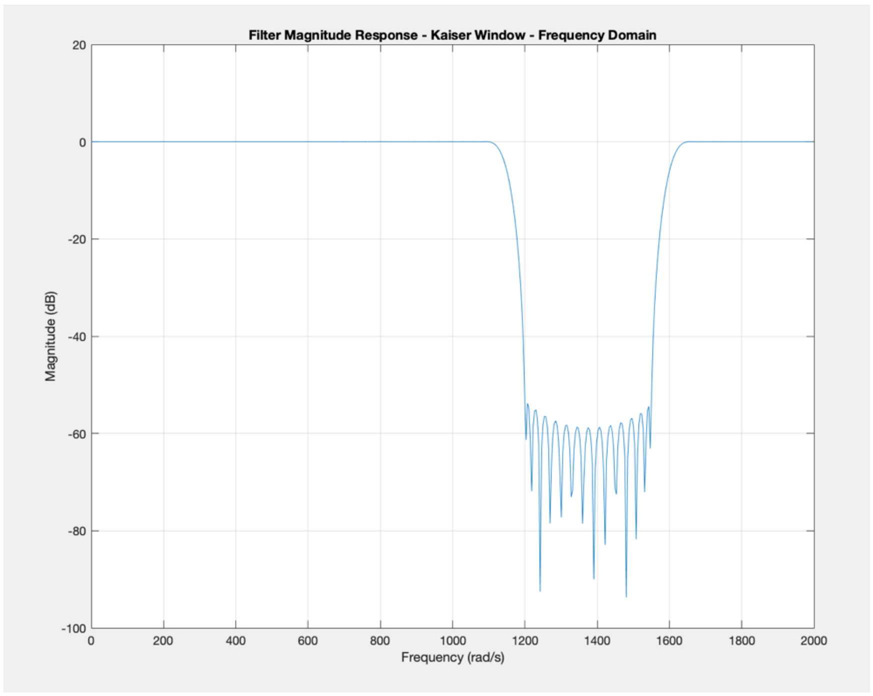


2. Causal Impulse Response

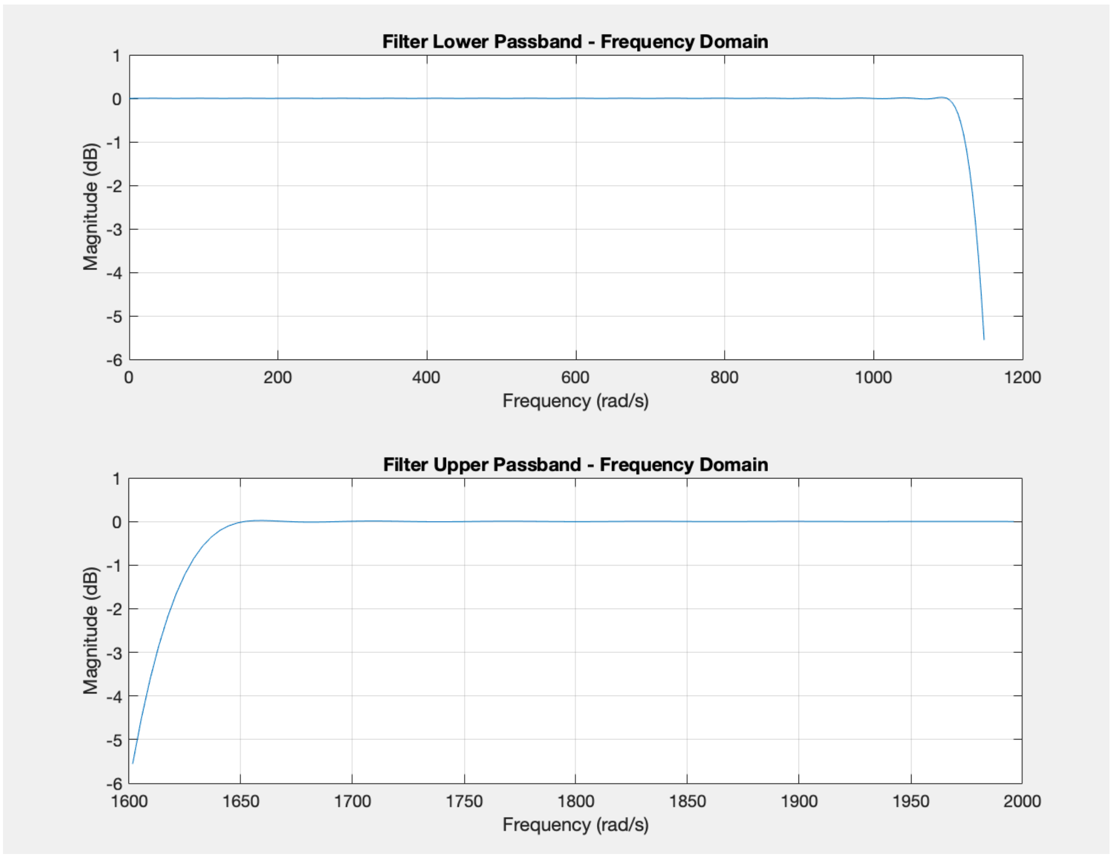


3. Magnitude response of the filter

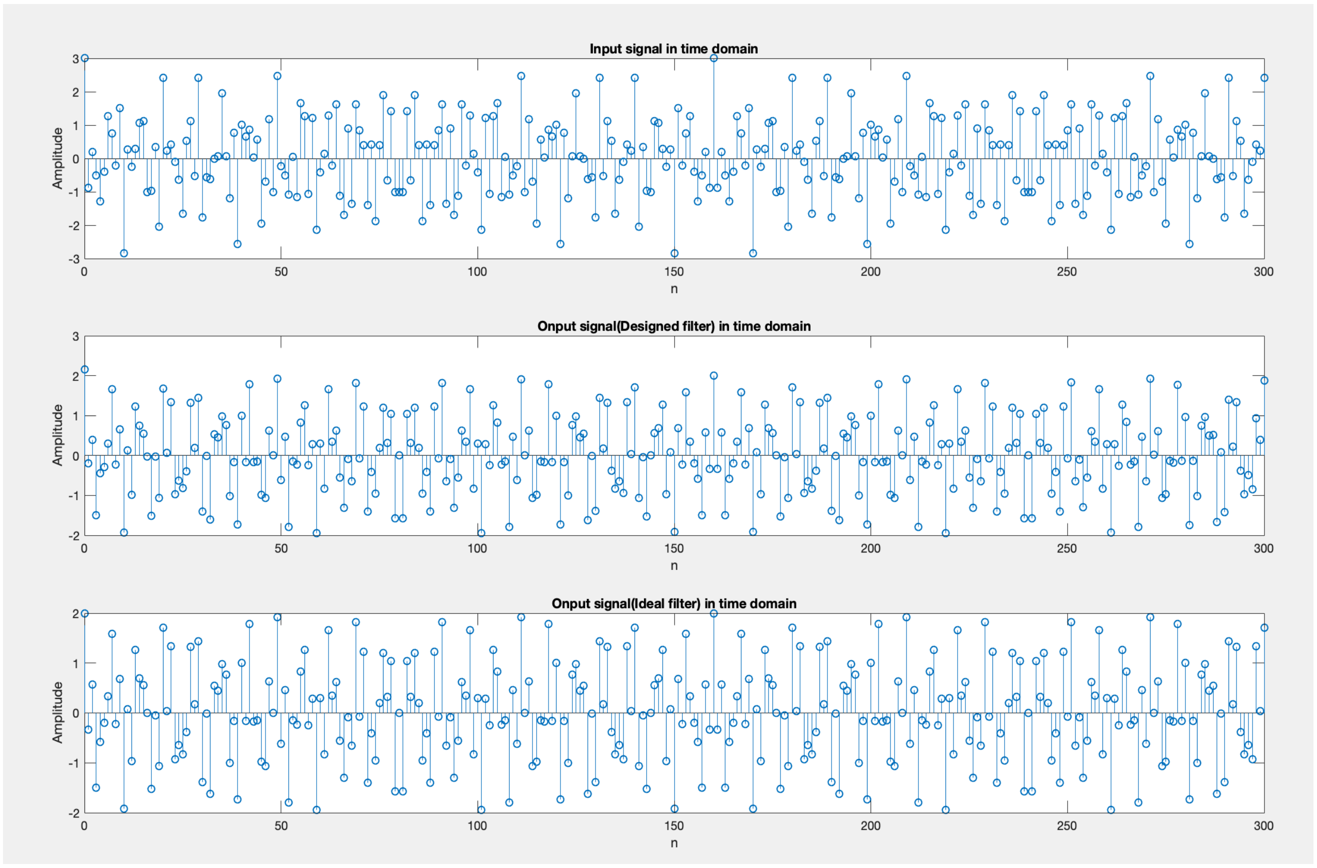




4. Upper and Lower Passbands

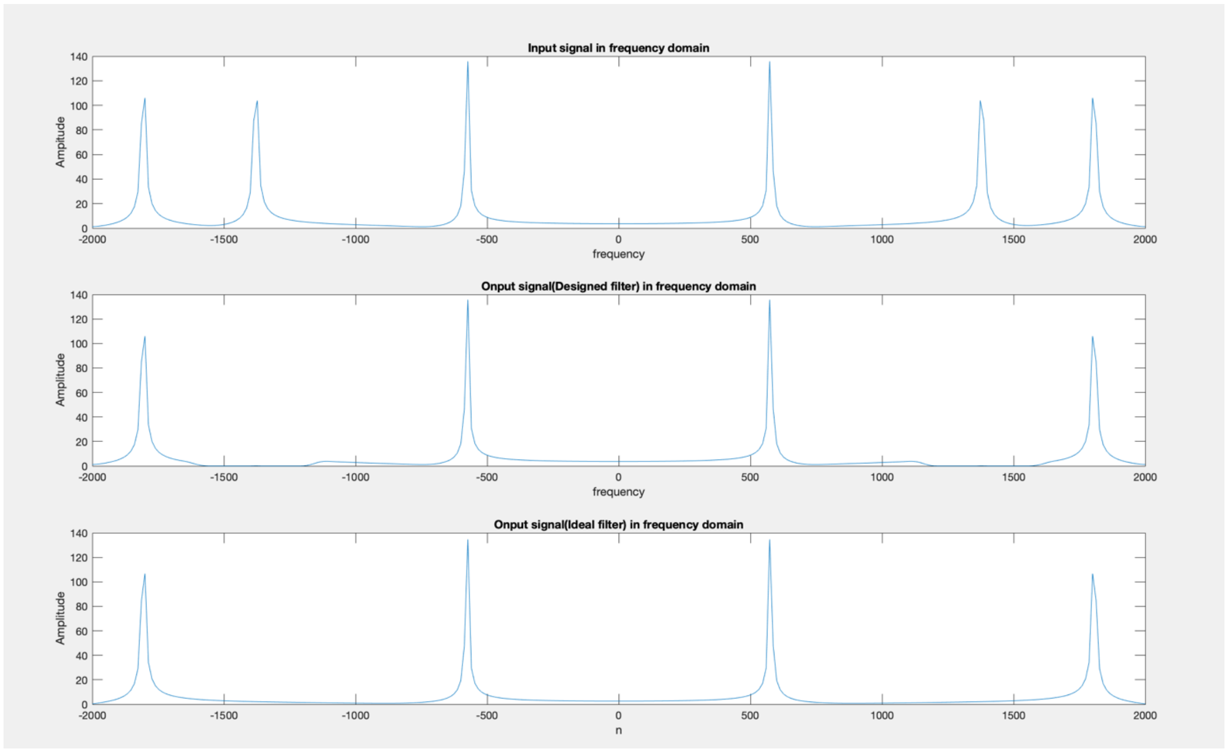


5. Verifying the operation of the filter- Time domain



The first graph depicts the input and the second and third will have the output from the designed filter and the output from an ideal filter. By observing the graphs, we can state that the time domain representation of the outputs from the designed and the ideal filters is somewhat the same. It proves that the implementation of the filter is correct to a certain extent.

6. Using DTFT and checking the frequency response



The above graphs further prove the accuracy of the designed filter. By comparing the last two graphs we can see that there are no major differences between the frequencies belonging to the stopband is not seen in the output signal as desired by the specifications.

Conclusion

It is observed that the ideal filter and the designed filter both behave in the same way. So, the implementation is correct.
'Kaiser Window' as discussed above is a method that is an efficient way of designing a FIR filter under the closed form direct approach. The design can accommodate many design requirements that are specified on the filter. That is due to the flexibility of the design. The computational effort of designing a filter in this method is also small.
The major drawback of this filter design is the high order of filter required when there are more than enough lower order ways that can be used to achieve those specifications. The use of high order filters is a disadvantage as they can be inefficient due to the usage of more unit delays adders and multipliers. In the software implementation more computations per sample are needed by higher order filters.

References

- https://www.mathworks.com/help/matlab/

- A. Antoniou, Digital Signal Processing, United States of America: McGraw-Hill, 2006.

Appendix

1. Code used

```
clc;
clear all;
close all;

%declaring the variables
A=1;
B=4;
C=7;
global Ap N ws wc1 wc2 alpha T
Ap=0.03+(0.01*A);
Aa=45+B;
wp1=(C*100)+400;
wp2=(C*100)+950;
wa1=(C*100)+500;
wa2=(C*100)+800;
ws=2*((C*100)+1300);
Bt=min((wa1-wp1),(wp2-wa2));
wc1=wp1+Bt/2;
wc2=wp2-Bt/2;
T=2*pi/ws;

%choosing delta
deltap=(10^(0.05*Ap)-1)/(10^(0.05*Ap)+1);
deltaa=10^(-0.05*Aa);
delta=min(deltap,deltaa);

%calculating the actual stopband attenuation
Aa=-db(delta);

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

%choosing D
if Aa<=21
    D=0.9222;
else
    D=(Aa-7.95)/14.36;
end
```

```
%selecting N
N=(ws*D/Bt)+1;
n=round(N);
if N>n
    N=n+1;
else
    N=n;
end
modulus=(mod(N,2))
if modulus==0
    N=N+1;
end
disp(N);
%*************************************
%create and plot the impulse response
n=[-(N-1)/2:(N-1)/2];
%hwnt=computehn(n)
hwnT=zeros(1,(N));
n1=-(N-1)/2;
for i=1:(N);
    hwnT(i)=computehn(n1);
    n1=n1+1;
end
disp(hwnT);
%hw_nT=computehn(n)
figure, stem(n, hwnT);
title('Impulse Response');
ylabel('Amplitude');
grid on;

%*************************************
%create the windowing function

n=[-(N-1)/2:(N-1)/2];
wnT=zeros(1,(N));
n2=-(N-1)/2;
for i=1:(N);
    wnT(i)=wknT(n2);
    n2=n2+1;
end
figure;
stem(n,wnT);
title('Impulse Response of the kaiser Window');
ylabel('Amplitude');
grid on;
```

```
%***************************************
%create the magnitude of the filter
h_fil=hwnT.*wnT;
%fvtool ( h_fil ) ;
freqz(h_fil);

figure;
[Hw,f] = freqz(h_fil);
f = f*ws/(2*pi);
Hw = 20*log10(abs(Hw));

plot(f,Hw);
xlabel('Frequency (rad/s)');
ylabel('Magnitude (dB)');
title(strcat(['Filter Magnitude Response - Kaiser Window - Frequency
Domain']));
grid on;

%******************************
%upper and lower passbands
figure;
Hw1=Hw(f<=wc1);
f1=f(f<=wc1);
subplot(2,1,1);
plot(f1,Hw1);
xlabel('Frequency (rad/s)');
ylabel('Magnitude (dB)');
title('Filter Lower Passband - Frequency Domain');
grid on;

Hw2=Hw(f>=wc2);
f2=f(f>=wc2);
subplot(2,1,2);
plot(f2,Hw2);
xlabel('Frequency (rad/s)');
ylabel('Magnitude (dB)');
title('Filter Upper Passband - Frequency Domain');
grid on;

%*************************************
%Creating the excitation of the input signal
omega1=(wc1-0)/2;
omega2=wc1+(wc2-wc1)/2;
omega3=wc2+(2000-wc2)/2;

nn=[0:1:300];
x=cos(omega1*nn*T) + cos(omega2*nn*T)+ cos(omega3*nn*T);

%**************************************
%obtaning the output for the input signal
y=conv(x,h_fil,'same');

%**************************************
%output from the ideal bandstop filter
ideal_y=cos(omega1*nn*T)+cos(omega3*nn*T);
```

```
%**************************************
%time domain representation of the input signal and the output signal
%from the designed filter and ideal filter

figure;
subplot(3,1,1);
stem(nn,x);
xlabel('n');
ylabel('Amplitude');
title('Input signal in time domain');

subplot(3,1,2);
stem(nn,y);
xlabel('n');
ylabel('Amplitude');
title('Onput signal(Designed filter) in time domain');

subplot(3,1,3);
stem(nn,ideal_y);
xlabel('n');
ylabel('Amplitude');
title('Onput signal(Ideal filter) in time domain');

%*********************************
%Use of DTFT and plotting the frequency responses
len=max(length(x),length(h_fil));
x_f=fft(x,len);
ideal_y_f=fft(ideal_y,len);
h_fil_f=fft(h_fil,len);
y_f=x_f.*h_fil_f;

freq = linspace(-ws/2,ws/2,len);
figure;
subplot(3,1,1);
plot(freq,abs(fftshift(x_f)));
xlabel('frequency');
ylabel('Ampitude');
title('Input signal in frequency domain');

subplot(3,1,2);
plot(freq,abs(fftshift(y_f)));
xlabel('frequency');
ylabel('Amplitude');
title('Onput signal(Designed filter) in frequency domain');

subplot(3,1,3);
plot(freq,abs(fftshift(ideal_y_f)));
xlabel('n');
ylabel('Amplitude');
title('Onput signal(Ideal filter) in frequency domain');
```

```
%***********************************
%Functions which were used for the above calculations

%for the kaiser window
function wknTval=wknT(n)
    global alpha N;
    beta=alpha*(1-(2*n/(N-1))^2)^0.5;
    Io_beta=1+Bessel(beta);
    Io_alpha=1+Bessel(alpha);

    if abs(n)<=(N-1)/2
        wknTval=Io_beta/Io_alpha;
    else
        wknTval=0;
    end
end

%bessel function
function Bess = Bessel(x)
    Bess=0;
    fact=1;
    k=1;
    limit=1;
    while (limit>10^-6)
        fact=fact*k;
        limit=((1/fact)*((x/2)^k))^2;
        Bess=Bess+limit;
        k=k+1;
        %disp(k)
    end
    %disp(Bess)
end

%impulse response generating function
function hn= computehn(n)
    global ws wc1 wc2 T
    if n==0
        hn=1+(2/ws)*(wc1-wc2);
    else
        hn=(1/(n*pi))*(sin(wc1*n*T)-sin(wc2*n*T));
    end
end
```