Department of Electronic & Telecommunication Engineering

University of Moratuwa

EN 2570 – Digital Signal Processing

DESIGNING A FINITE IMPULSE RESPONSE (FIR) BANDPASS FILTER

Name	Index number
Premathilaka H. D. M.	180497C

This report is submitted in partial fulfillment of the requirements for the module EN 2570 – Digital Signal Processing.

Date of submission: 04/03/2021

Abstract

This is a detailed report of the procedure employed to design a finite-duration impulse response (FIR) band-pass filter for a prescribed set of specifications using the windowing method in conjunction with the Kaiser window. The report includes a review of the core characteristics of the designed digital filter, such as the magnitude response and the phase response. A thorough analysis of the response of the designed filter to an input is presented and compared with expected outputs. The software implementation of designing the digital filter including plots and comparisons, is done on MATLAB 2018a. All the diagrams shown in the report are drawn using 'draw.io' application.

Table of Contents

List of Figures

List of Tables

Introduction

Digital filters are systems that are generally used to enhance or suppress desired aspects of sampled, discrete-time signals. Depending on the portion of the spectrum that the filter is used to suppress/ enhance, filters can be trivially categorized as low-pass, high-pass, band-pass and band-stop. A digital filter can be completely characterized by its transfer function and/or its difference equation. Based on certain characteristics of the filter or the implementation of the filter, digital filters can be categorized as FIR/ IIR, recursive/ non-recursive, direct/ indirect, etc. Although digital filters are comprehensible and are convenient to calculate, the practical implementation of them has continued to be a subject of much advanced research, over the last few decades.

This report is based on the evaluation of a closed form approach of designing a band-pass FIR filter that is implemented through the use of Fourier series in conjunction with the windowing method. The procedure followed to apply a typical Kaiser window function in designing a non-recursive filter for the prescribed specifications is detailed in the subsequent sections.

The filter as well as its behaviour for standard inputs are analysed in the time domain and the frequency domain. Frequency domain analysis is conducted by applying the 'Fast Fourier Transform' algorithm to obtain the 'Discrete Fourier Transform' of the system and signals used.

Method

I. Filter Implementation

1. Prescribed Filter Specifications

The following parameters were determined using the index number (180497C). Thus, the values used for A, B and C are 4, 9 and 7 respectively.

Table 1 - Prescribed Filter Specifications

Specification	Symbol	Derivation	Value	Units
Maximum passband ripple	$\tilde{A}_p$	$0.03 + (0.01 \times A)$	0.07	dB
Minimum stopband attenuation	$\tilde{A}_a$	$45 + B$	54	dB
Lower passband edge	Ω_{p1}	$(C \times 100) + 300$	1000	rad/s
Upper passband edge	Ω_{p2}	$(C \times 100) + 700$	1400	rad/s
Lower stopband edge	Ω_{a1}	$(C \times 100) + 150$	850	rad/s
Upper stopband edge	Ω_{a2}	$(C \times 100) + 800$	1500	rad/s
Sampling frequency	Ω_s	$2[(C \times 100) + 1200]$	3800	rad/s

2. Derived Filter Specifications

Once the prescribed specifications are determined, parameters such as the critical transition width, lower and upper cut-off frequencies can be calculated. The derivations of these parameters are summarized in the following table.

Table 2 - Derived Filter Specifications

Specification	Symbol	Derivation	Value	Units
Lower transition width	B_{t1}	$\Omega_{p1} - \Omega_{a1}$	150	rad/s
Upper transition width	B_{t2}	$\Omega_{a2} - \Omega_{p2}$	100	rad/s
Critical transition width	B_t	$\min(B_{t1}, B_{t2})$	100	rad/s
Lower cut-off frequency	Ω_{c1}	$\Omega_{p1} - \frac{B_t}{2}$	950	rad/s
Upper cut-off frequency	Ω_{c2}	$\Omega_{p2} + \frac{B_t}{2}$	1450	rad/s
Sampling period	T	$\frac{2\pi}{\Omega_s}$	0.0017	s

3. Derivation of the Kaiser Window Parameters

The following equations and derivations are based on the formulae given in Chapter 9 of the text book. (Antoniou, 2005)

- The Kaiser window function is defined as,

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

where $I_o(\beta)$ and $I_o(\alpha)$ are Bessel functions of the parameters α and β.

- The Bessel function is given by,

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

α is an independent parameter where β is a parameter dependent on α. The relation between α and β is given by,

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

To apply the above equations, we need to first find the values of α and N.

But, α and N are indirectly related to another parameter δ. Thus, we should first find an appropriate δ value.

Conditions to be satisfied by δ: $A_p \leq \tilde{A}_p$ and $A_a \geq \tilde{A}_a$

These two conditions can be fulfilled by picking $\delta = \min(\widetilde{\delta_p}, \widetilde{\delta_a})$ where,

$$\widetilde{\delta_p} = \frac{10^{0.05\tilde{A}_p} - 1}{10^{0.05\tilde{A}_p} + 1} \ and \ \widetilde{\delta_a} = 10^{-0.05\tilde{A}_a}$$

Now, using this δ value, we can define the actual passband ripple (A_p) and the actual stopband attenuation (A_a).

$$A_p = 20log\left(\frac{1+\ \delta}{1-\delta}\right)$$

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

Let's choose α to be,

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

Then consider,

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

Finally, we can find the length (N) of the window using the inequality given below.

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

N is chosen such that it's the smallest odd integer satisfying the above inequality.

The parameter values obtained using above equations are summarized below.

Table 3 - Kaiser Window Parameters

Parameter	Value	Units
δ	0.002	-
A_a	54	dB
A_p	0.0347	dB
α	4.9921	-
$I_o(\alpha)$	27.0474	-
D	3.2068	-
N	123	-

4. Derivation of the Idealized Impulse Response

The impulse response (h[nT]) is derived assuming an idealized frequency response. Therefore, if the cut-off frequencies are Ω_{c1} and Ω_{c2}, then we can express the frequency response of the band-pass filter as,

$$H(e^{j\Omega T}) = \begin{cases} 1 & for - \Omega_{c2} \le \Omega \le -\Omega_{c1} \\ 1 & for\ \Omega_{c1} \le \Omega \le \Omega_{c2} \\ 0 & Otherwise \end{cases}$$

Considering the inverse Fourier transform of the above impulse response,

$$h[nT] = \frac{1}{\Omega_s}\left[\int_{-\frac{\Omega_s}{2}}^{\frac{\Omega_s}{2}} H(e^{j\Omega T})\, e^{j\Omega nT} d\Omega\right] = \frac{1}{\Omega_s}\left[\int_{-\Omega_{c2}}^{-\Omega_{c1}} e^{j\Omega nT}\, d\Omega \ + \ \int_{\Omega_{c1}}^{\Omega_{c2}} e^{j\Omega nT}\, d\Omega\right]$$

- When n = 0;

$$h[0] = \frac{1}{\Omega_s}\left[\int_{-\Omega_{c2}}^{-\Omega_{c1}} 1\, d\Omega \ + \ \int_{\Omega_{c1}}^{\Omega_{c2}} 1\, d\Omega\right] = \frac{2}{\Omega_s}[\ \Omega_{c2} - \Omega_{c1}\ ]$$

- When n ≠ 0;

$$h[nT] = \frac{1}{\Omega_s}\left[\left[e^{j\Omega nT}\right]_{-\Omega_{c2}}^{-\Omega_{c1}} + \left[e^{j\Omega nT}\right]_{\Omega_{c1}}^{\Omega_{c2}}\right] = \frac{1}{n\pi}[\, sin(\Omega_{c2} nT) - sin(\Omega_{c1} nT)]$$

In summary, we can write,

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

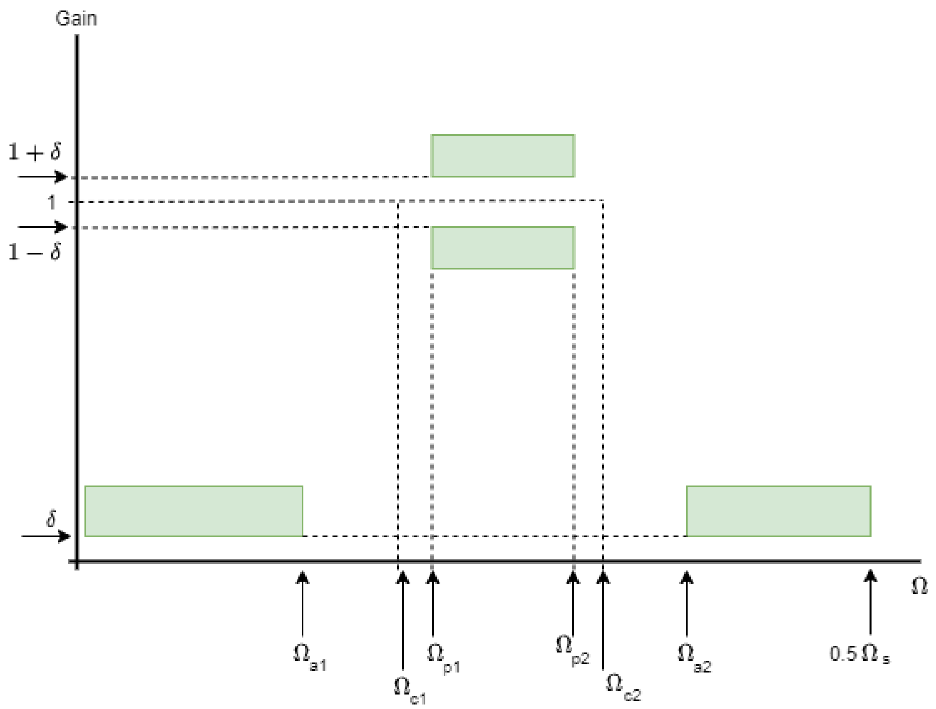

Figure 1 - Idealized Frequency Response of the Bandpass Filter

5. Derivation of the Causal Impulse Response of the Windowed Filter

Now, we will consider the truncated version of the idealized impulse response by multiplying it with the Kaiser window. Thus we get,

$$h_w[nT] = w_k[nT]h[nT]$$

Taking the Z-transform of $h_w[nT]$,

$$H_w(z) = \mathcal{Z}(h_w[nT]) = \mathcal{Z}(w_k[nT]h[nT])$$

To obtain the causal impulse response, we have to shift the system by (N-1)/2 units. When we shift $H_w(z)$, the result would be,

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

For comparison purposes, we will consider another impulse response derived by truncating an ideal impulse response with a rectangular window.

Suppose the rectangular window is given by,

$$w_r(nT) = \begin{cases} 1 & for\ |n| \leq \dfrac{N-1}{2} \\ 0 & otherwise \end{cases}$$

Then we can obtain the corresponding impulse response as,

$$h_r[nT] = w_r[nT]h[nT]$$

The comparisons of these two impulse responses will be performed in the subsequent sections.

II. Filter Evaluation

The designed filter is evaluated using an input signal constructed by summing up three sinusoidal signals. Thus, the input signal can be expressed as,

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

where Ω_1 is the mid frequency of the lower stopband, Ω_2 is the mid frequency of the passband and Ω_3 is the mid frequency of the upper stopband. 500 samples were used to ensure that a steady state response is achieved. Instead of convolving the input signal with the filter, they are multiplied in the frequency domain and then the inverse DFT is calculated to obtain the output signal.

Table 4 - Frequency Components of the Input Signal

Frequency	Derivation	Value	Units
Ω_1	$\Omega_{a1}/2$	425	rad/s
Ω_2	$(\Omega_{p1} + \Omega_{p2})/2$	1200	rad/s
Ω_3	$(\Omega_{a2} + 0.5\Omega_s)/2$	1700	rad/s

Results

In this section, we will go through the plots obtained in the time domain and in the frequency domain. Our main focus will be on the Kaiser window, designed filter and the response from the designed filter to the input signal. In addition to these plots, supplementary plots such as the time domain and frequency domain representation of a filter designed using a rectangular window is included.

I. Time and Frequency Domain Plots for the Filter (Rectangular Window)

The plots in this section depict the impulse response and the frequency response of a filter designed using a rectangular window along with an idealized impulse response. These 2 plots are supplementary.

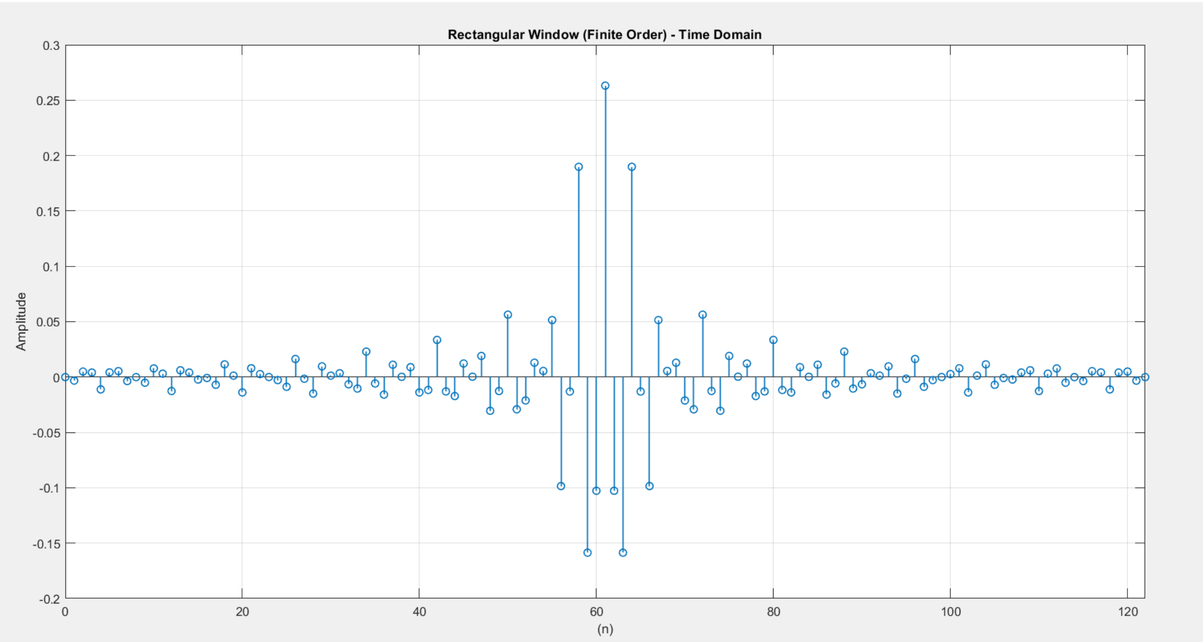

Figure 2 - Time Domain Representation of a Band-pass Filter Designed Using a Rectangular Window

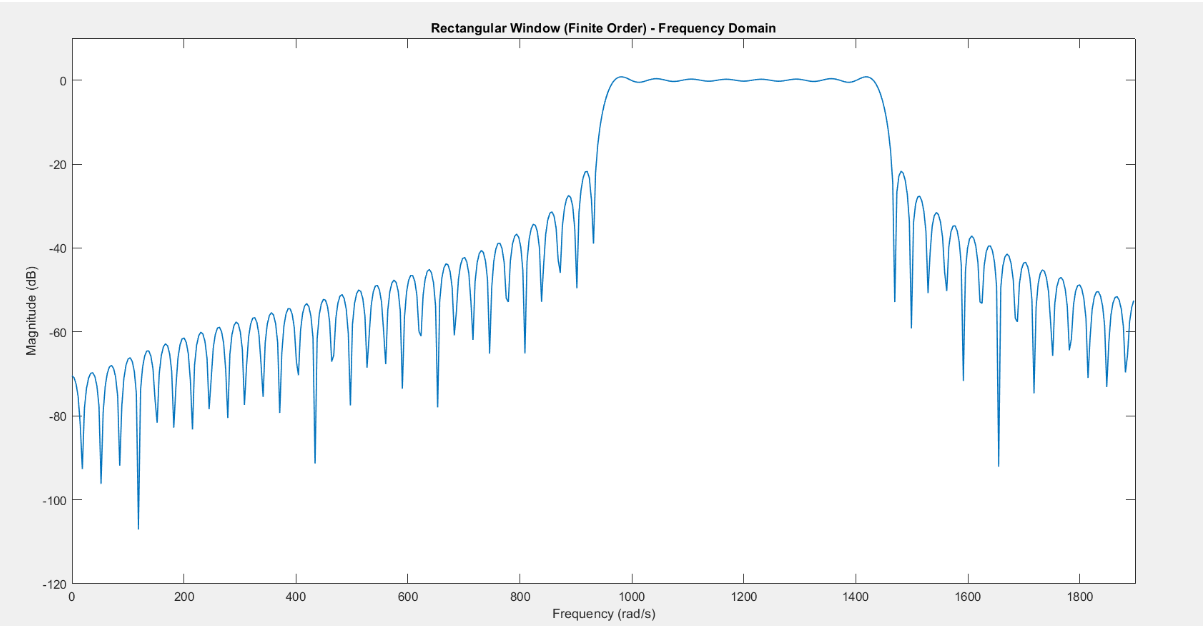

Figure 3 - Frequency Domain Representation of a Band-pass Filter Designed Using a Rectangular Window

II. Time and Frequency Domain Plots for the Filter (Kaiser Window)

The plots in this section depict the impulse response and the magnitude response of a filter designed using the Kaiser window along with an idealized impulse response. Characteristics of the Kaiser window are based on the calculations performed in the previous sections.

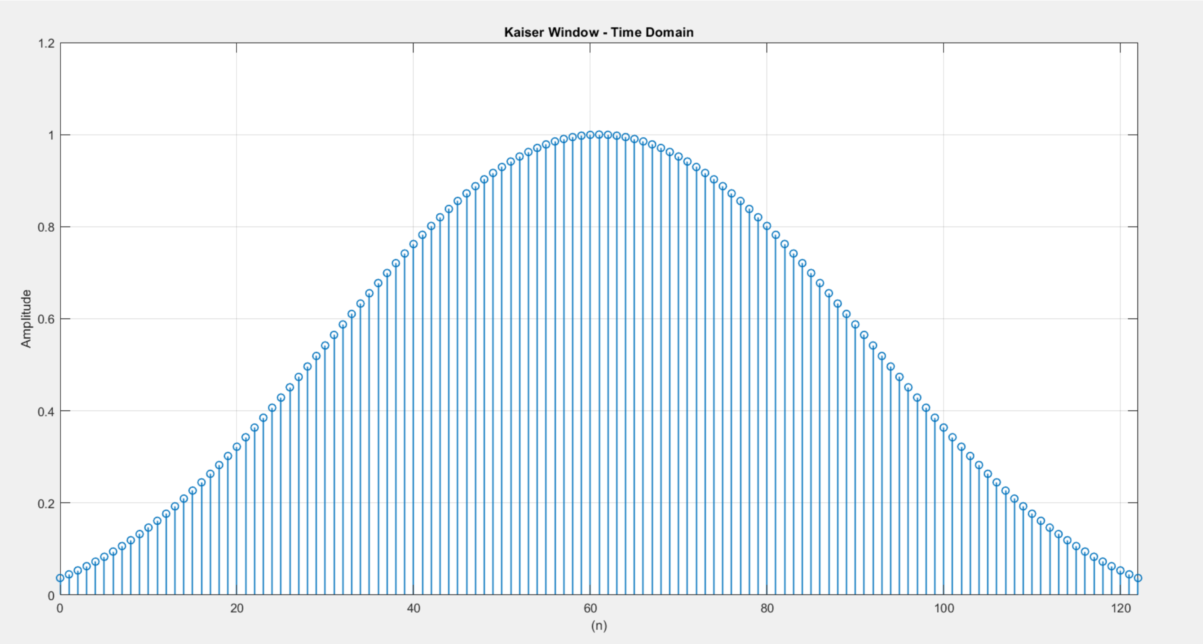

Figure 4 - Kaiser Window Used to Design the Band-pass Filter

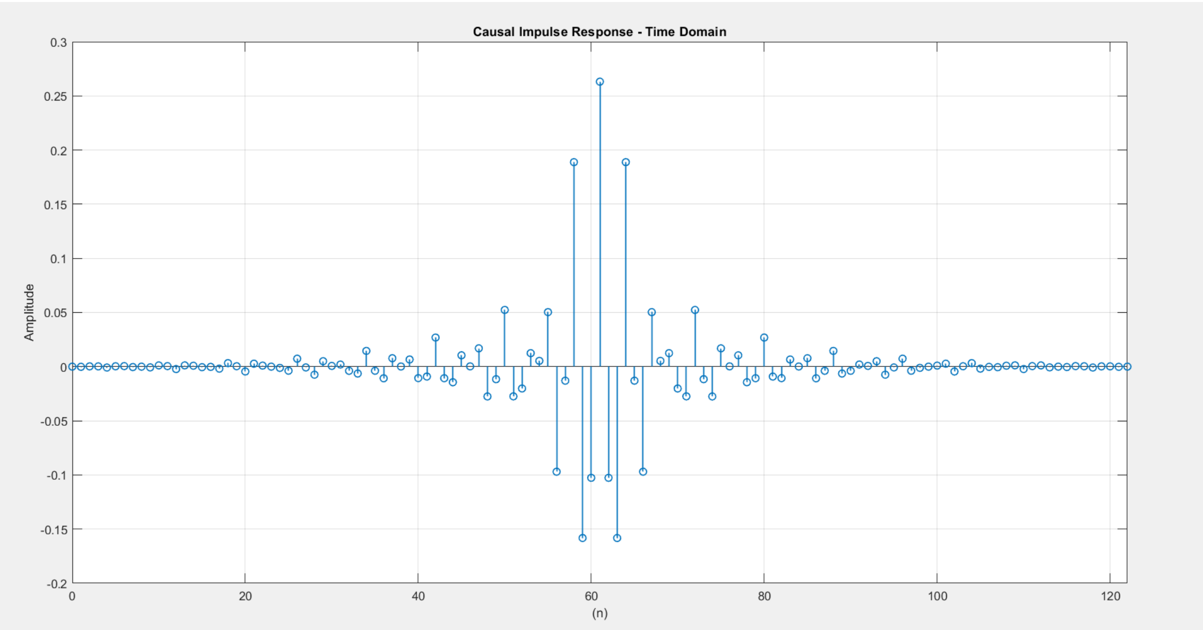

Figure 5 - Causal Impulse Response of the Designed Filter

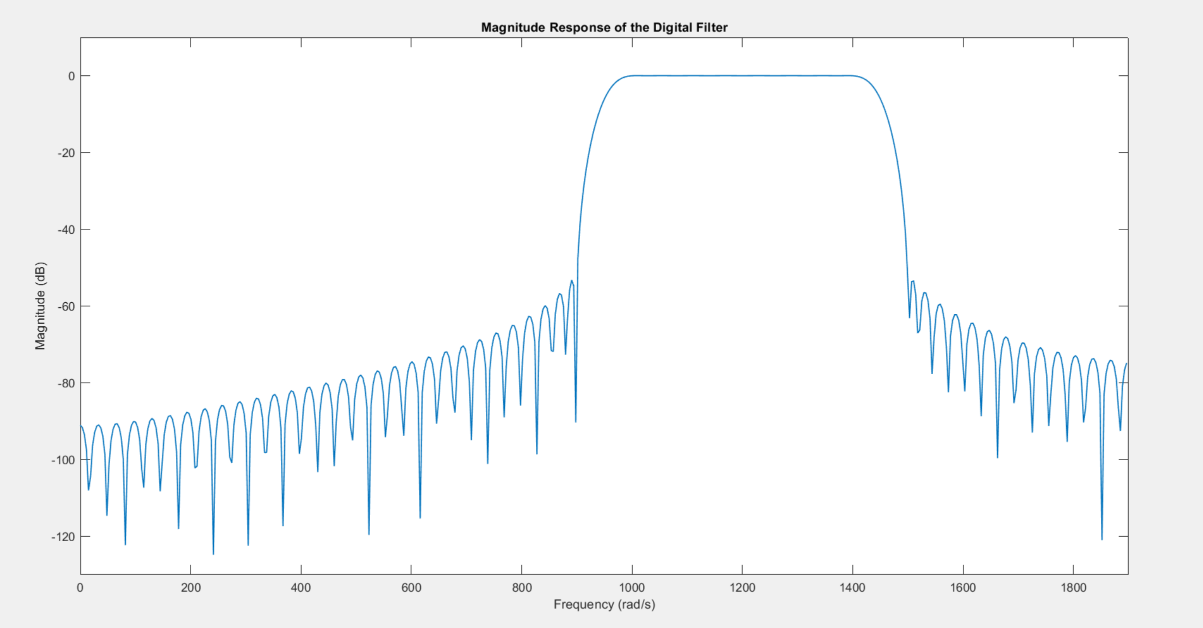

Figure 6 - Magnitude Response of the Filter (from 0 to Omega_s/2 rad/s)

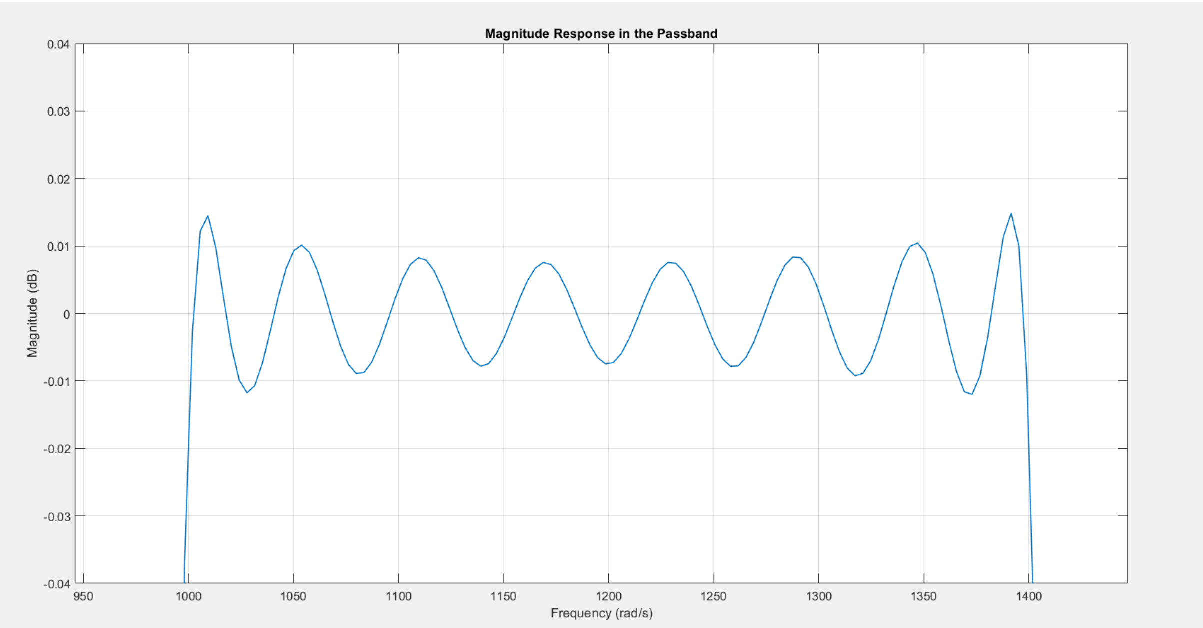

Figure 7 - Magnitude Response of the Filter in the Passband

III. Plots for the Input Signal

The input signal comprises of three sinusoidal signals. The frequencies of these sinusoidal signals lie in the lower stopband, passband and the upper stopband of the digital filter. The following plots depict a portion of the input signal and the single sided spectrum of the input signal.

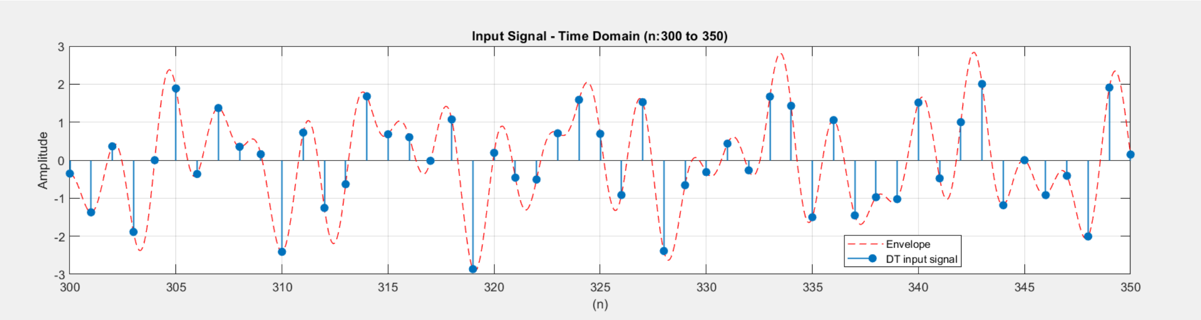

Figure 8 - Input Signal - Time Domain

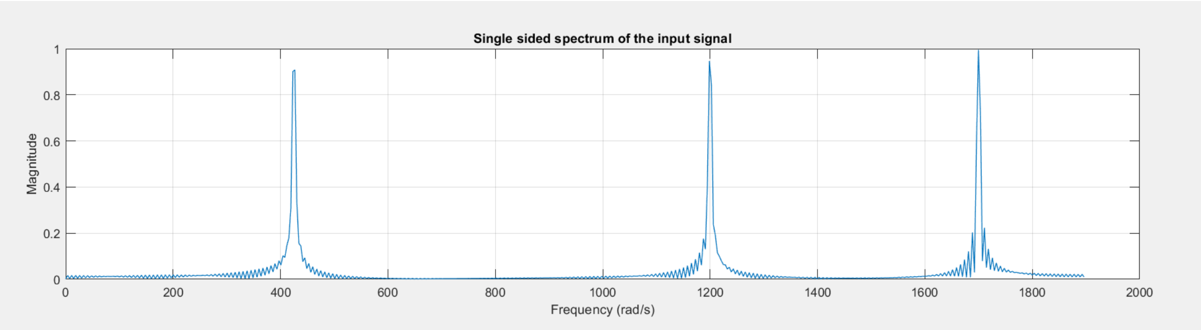

Figure 9 - Input Signal - Frequency Domain

IV. Plots for the Output Signal

We can observe that the digital filter has been able to suppress the frequencies of the input signal in the stopband while preserving the frequency component in the passband.

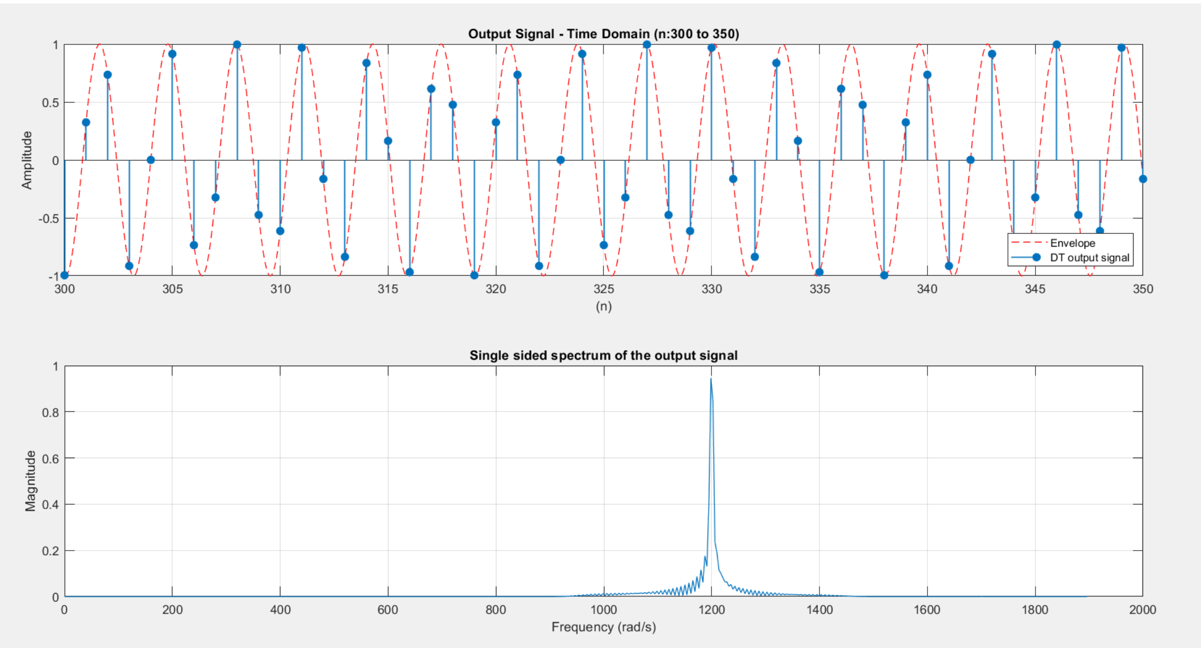

Figure 10 - Output Signal - Time Domain and Frequency Domain

V. Comparison Plots

This section includes plots comparing the output signal with the ideal output. The following plots correspond to the time domain and frequency domain comparison of the two signals.

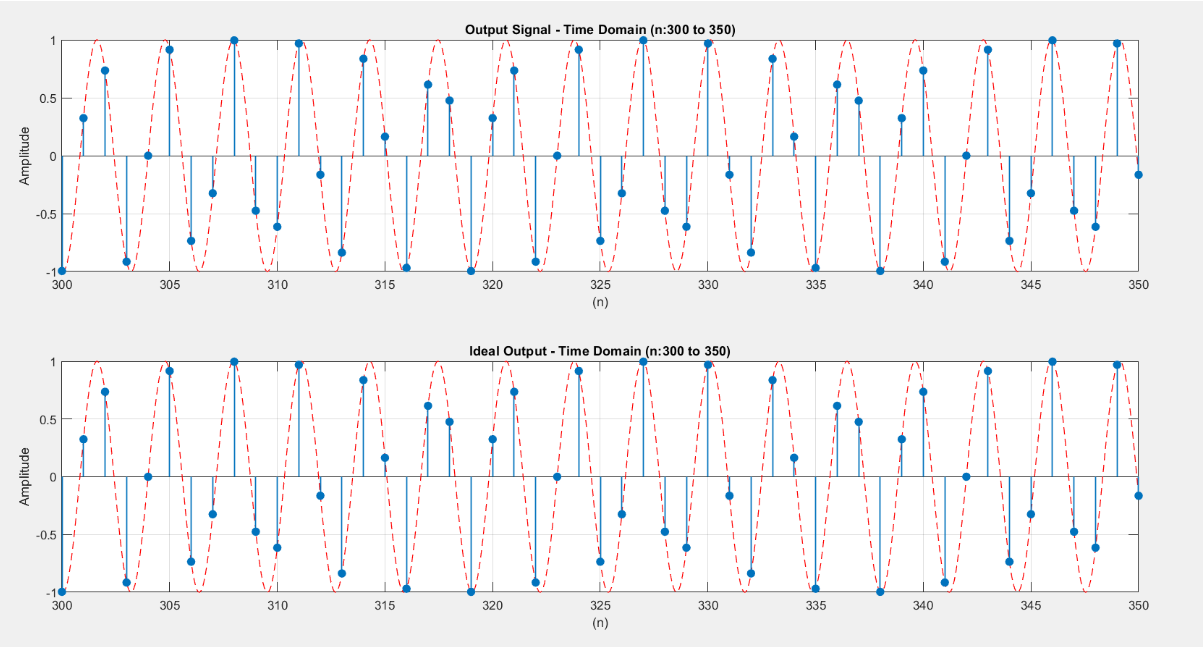

Figure 11 - Comparison in the Time Domain

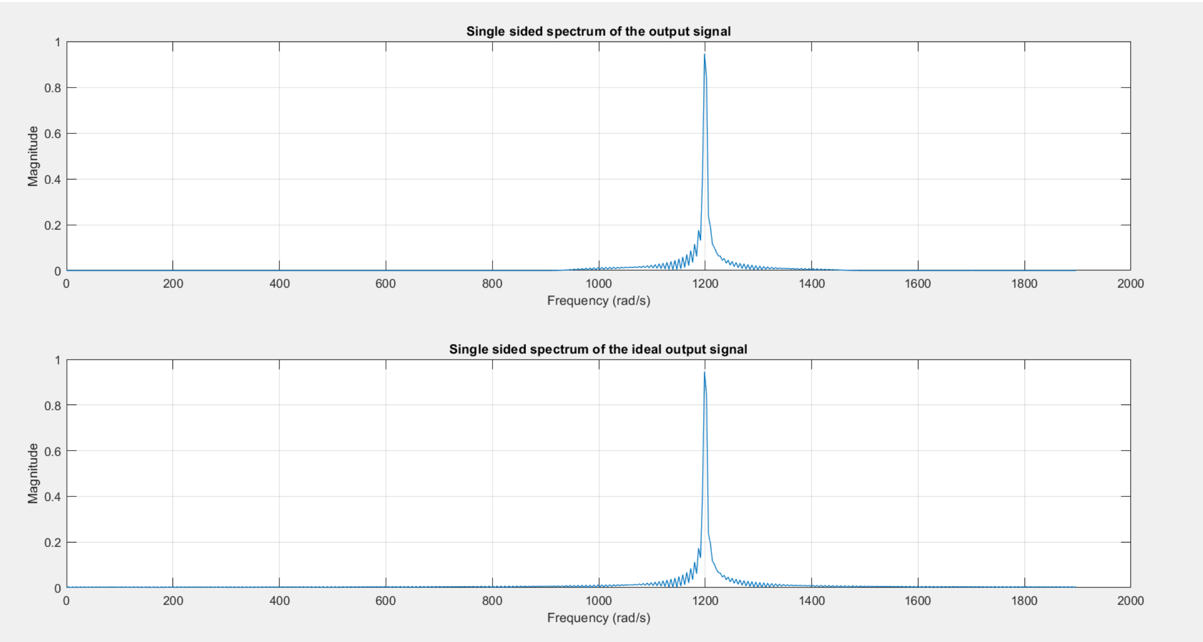

Figure 12 - Comparison in the Frequency Domain

VI. Supplementary Plots for the Filter (Using designfilt and fvtool)

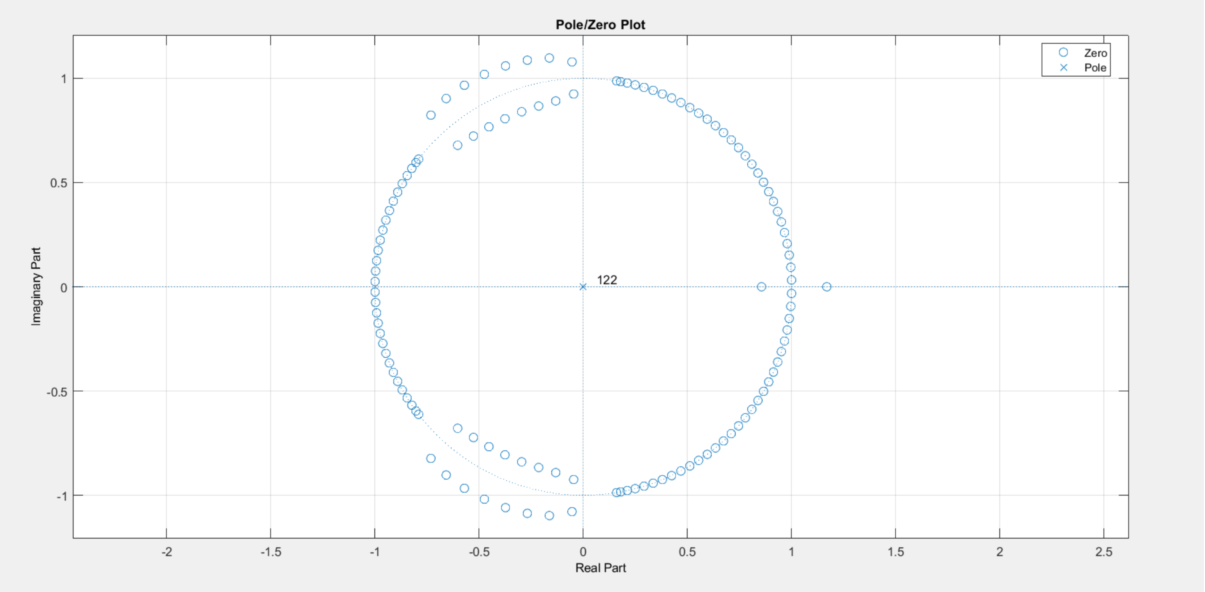

Figure 13 - Pole-Zero Plot of the Filter - Using fvtool

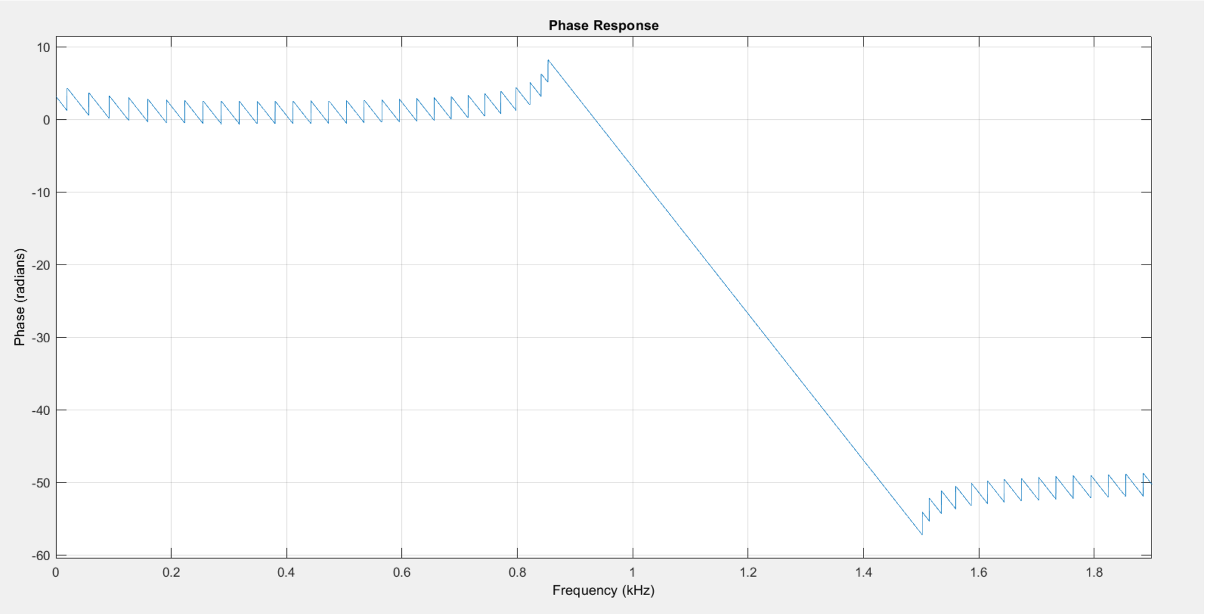

Figure 14 - Phase Response of the Filter - Using fvtool

Discussion

Figure 6 suggests that the filter would suppress frequencies outside the passband, at least with a loss of 54dB. Moreover, figure 7 clearly shows that the maximum fluctuation within the passband is less than the calculated maximum passband ripple (0.0347 dB). Both these facts are supported by the results obtained when an input signal comprised of three sinusoids is passed through the filter. As shown in figure 9, the input signal consists of the frequencies 425 rad/s, 1200 rad/s and 1700 rad/s. But in figure 10 which depicts the frequency spectrum of the output signal, has a spike only at 1200 rad/s. Thus, it is clear that the frequency components of the input within the stopband of the filter are reduced to an insignificant level. On the other hand, when we consider figure 12, we can observe that the amplitudes of the ideal output and the output obtained are approximately equal. Therefore, we can claim that the amplitudes of the frequency components within the passband are preserved in a nearly ideal level. Given these facts, it is safe to assume that the designed filter displays the behaviour of a nearly ideal band-pass filter.

Now, let's compare the Kaiser window approach with the rectangular window approach. From figure 3 and figure 6, we can clearly observe that for a given frequency in the stopband of the filter, the attenuation is higher in the case of Kaiser window approach (i.e. minimum stopband attenuation of the Kaiser window approach is 54dB whereas the minimum stopband attenuation of the rectangular window approach is close to 20dB). Moreover, figure 3 and figure 6 show us that Kaiser window approach leads to a smoother passband which implies that the fluctuations within the passband are minimum in the Kaiser window approach (Smooth passband owes to the fact that the Kaiser window doesn't have any sharp transitions, unlike the rectangular window).

Conclusion

Since it is not feasible to practically implement ideal filters, the best approach we can take is to tweak the parameters that we have control of, to improve the performance of practical filters to a nearly ideal level. As seen in the plots given, we can claim that the Kaiser window approach has led to nearly ideal results, in this scenario. Moreover, Kaiser windowing can be considered as a more flexible option compared to typical windows such as rectangular, von Hann, Hamming, etc. as the Kaiser window has two adjustable parameters (α and N) whereas the rectangular, von Hann, Hamming windows have only one adjustable parameter (N).

However, the major drawback of this technique is that in most cases, Kaiser window approach leads to higher order filters (In our case, the order was 122, which is quite high). Hardware implementation of higher order filters can be complicated and might incur a higher cost. Thus, we can conclude that when designing filters, it is of paramount importance that we choose an optimum approach that would maintain software-hardware balance while delivering nearly ideal performance.

Acknowledgement

I would like to express my gratitude to Dr. Chameera Edussooriya, the project supervisor, for the persistent guidance he provided via online lectures during this pandemic, while encouraging us to explore more about the subject.

Bibliography

Antoniou, A. (2005). *Digital Signal Processing - Signals, Systems and Filters.* McGraw-Hill.

R., O. A. (2009). *Discrete Time Signal Processing.* PEARSON.

Appendix

Filter Specifications I

```
clc; clear; close all;

%Index number  = 180497C
A = 4;
B = 9;
C = 7;

%Maximum passband ripple in dB
Ap = 0.03 + (0.01*A);
%Minimum stopband attenuation in dB
Aa = 45 + B;
%Lower passband edge in rad/s
Omega_p1 = (C*100) + 300;
%Upper passband edge in rad/s
Omega_p2 = (C*100) + 700;
%Lower stopband edge in rad/s
Omega_a1 = (C*100) + 150;
%Upper stopband edge in rad/s
Omega_a2 = (C*100) + 800;
%Sampling frequency in rad/s
Omega_s = 2*((C*100) + 1200);
%Sampling period
Ts = 2*pi/Omega_s;
```

Filter Specifications II

```
%Transition width in rad/s
Bt = min((Omega_p1-Omega_a1),(Omega_a2-Omega_p2));
%Lower cutoff frequency in rad/s
Omega_c1 = Omega_p1 - Bt/2;
%Upper cutoff frequency in rad/s
Omega_c2 = Omega_p2 + Bt/2;
```

Determining Kaiser Window Parameters

```
%Calculating delta_p
dp = (10^(0.05*Ap)-1)/((10^(0.05*Ap)+1));
%Calculating delta_a
da = 10^(-0.05*Aa);
%Selecting the minimum delta
delta = min(dp,da);

%Actual passband ripple in dB
Ap = 20*log10((1+delta)/(1-delta));
%Actual stopband attenuation in dB
Aa = -20*log10(delta);

%Choosing alpha
if Aa <= 21
    alpha = 0;
elseif Aa > 21 && Aa <= 50
    alpha = 0.5842*(Aa-21)^0.4 + 0.07886*(Aa-21);
```

```
else
    alpha = 0.1102*(Aa-8.7);
end

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

%Choosing the length of the filter
N = ceil((Omega_s*D/Bt)+1);
if rem(N,2) == 0
    N = N+1;
end

%Filter duration
range = 0:1:N-1;
M = (N-1)/2;
n = 0:1:M;

%Calculating beta
beta = alpha*sqrt(1-((2*n)/(N-1)).^2);
```

Generating the Bessel Functions

```
%No.of iterations
Iter =  100;

%Generating I(alpha)
I_alpha = 1;
for i = 1:Iter
    I_alpha = I_alpha + ((1/factorial(i))*((alpha/2)^i))^2;
end

%Generating I(beta)
I_beta = 1;
for i = 1:Iter
    I_beta = I_beta + ((1/factorial(i))*((beta/2).^i)).^2;
end
```

Kaiser Window

```
%defining the kaiser window
kaiser_win = I_beta/I_alpha;
kaiser_win = [fliplr(kaiser_win(2:end)) kaiser_win];
figure(1)
stem(range,kaiser_win,'Linewidth',1)
xlabel('(n)')
ylabel('Amplitude')
axis([0 N-1 0 1.2])
grid on
title('Kaiser Window - Time Domain')
```

Filter Using a Rectangular Window

This is the idealized impulse response truncated using a rectangular window

```
%assuming an idealized frequency response
hnT = zeros(1,N);
for i = 0:M
    if i == 0
        hnT(i+M+1) = (2/Omega_s)*(Omega_c2 - Omega_c1);
    else
        hnT(i+M+1) = (1/(i*pi))*(sin(Omega_c2*i*Ts) - sin(Omega_c1*i*Ts));
    end
end

hnT(1:M) = fliplr(hnT(M+2:end));

%plotting the idealized impulse response of finite order - time domain
stem(range, hnT,'Linewidth',1)
axis([0 122 -0.2 0.3])
grid on
xlabel('(n)')
ylabel('Amplitude')
title('Rectangular Window (Finite Order) - Time Domain')

%Plotting the idealized impulse response of finite order - frequnecy domain
[h_ideal,w_ideal] = freqz(hnT);
Omega_ideal = w_ideal/Ts;
h_ideal = 20*log10(abs(h_ideal));
plot(Omega_ideal,h_ideal,'Linewidth',1)
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
title('Rectangular Window (Finite Order) - Frequency Domain')
axis([0 Omega_s/2 -120 10])
```

Causal Impulse Response

This section plots the causal impulse response for a BPF designed using a Kaiser window

```
%Defining the filter using the Kaiser window
filter = kaiser_win.*hnT;

%Plotting the causal impulse response of the filter - time domain
stem(range, filter,'Linewidth',1)
axis([0 N-1 -0.2 0.3])
grid on
xlabel('(n)')
ylabel('Amplitude')
title('Causal Impulse Response - Time Domain')
```

Magnitude Response of the Filter (0 to (Ω_s)/2 (rad/s))

In this section, we are trying to plot the magnitude response of the digital filter obtained, for the frequency range 0 to $\frac{\Omega_s}{2}$(rad/s).

```
%Plotting the magnitude of the filter
[h,w] = freqz(filter);
Omega = w/Ts;
h = 20*log10(abs(h));
plot(Omega,h,'Linewidth',1)
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
title('Magnitude Response of the Digital Filter')
axis([0 Omega_s/2 -130 10])
```

Magnitude Response of the Filter in the Passband

```
lower_margin = round((length(Omega)/(Omega_s/2))*Omega_c1);
upper_margin = round((length(Omega)/(Omega_s/2))*Omega_c2);
%Passband frequencies
passband = Omega(lower_margin:upper_margin);
mag_passband = h(lower_margin:upper_margin);
plot(passband,mag_passband,'Linewidth',1)
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
title('Magnitude Response in the Passband')
axis([-Inf Inf -0.04 0.04])
grid on
```

Generating the Input Signal

```
%Finding the component frequencies
Omega_1 = (0 + Omega_a1)/2;
Omega_2 = (Omega_p1 + Omega_p2)/2;
Omega_3 = (Omega_a2 + Omega_s/2)/2;

samples = 500;
%to generate the discrete signal
n1 = 0:1:samples;
%to envelope the the discrete signal using a CT signal
n2 = 0:0.0001:samples;

%input DT signal
xnT = sin(Omega_1.*n1.*Ts) + sin(Omega_2.*n1.*Ts) + sin(Omega_3.*n1.*Ts);
%CT signal (envelope)
envelope = sin(Omega_1.*n2.*Ts) + sin(Omega_2.*n2.*Ts) + sin(Omega_3.*n2.*Ts);
```

Displaying the Input Signal

```
%plotting a sample of the input signal in time domain
subplot(2,1,1)
plot(n2,envelope,'r--','LineWidth',0.75)
hold on
stem(n1, xnT,'filled','LineWidth',1)
xlabel('(n)')
ylabel('Amplitude')
title('Input Signal - Time Domain (n:300 to 350)')
xlim([300,350])
grid on
```

```
legend('Envelope','DT input signal','Location','best')
hold off

%Calculating FFT of the input signal
numpoints = 2^(nextpow2(numel(xnT)+numel(filter)));
x_fft = fft(xnT,numpoints);
magnitudes = abs(x_fft/numel(xnT));
magnitudes = magnitudes(:,1:numpoints/2+1);
magnitudes(:,2:end-1) = 2*magnitudes(:,2:end-1);
freq = 0:(Omega_s/numpoints):(Omega_s/2)-Omega_s/numpoints;

%plotting the input signal in the frequency domain
subplot(2,1,2)
plot(freq,magnitudes(1:numpoints/2),'LineWidth',0.75)
xlabel('Frequency (rad/s)')
ylabel('Magnitude')
title('Single sided spectrum of the input signal')
grid on
```

Filtering the Input Signal

```
filter_fft = fft(filter,numpoints);
y_fft = x_fft.*filter_fft;
ynT = ifft(y_fft,numpoints);
ynT = ynT(floor(N/2)+1:numel(ynT)-floor(N/2));

%constructing an envelope similar to the ideal output
y_ideal = sin(Omega_2.*n2.*Ts);
```

Displaying the Output Signal

```
%plotting a sample of the output signal in time domain
subplot(2,1,1)
plot(n2,y_ideal,'r--','LineWidth',0.75)
hold on
stem(n1, ynT(1:samples+1),'filled','LineWidth',1)
xlabel('(n)')
ylabel('Amplitude')
title('Output Signal - Time Domain (n:300 to 350)')
xlim([300,350])
grid on
legend('Envelope','DT output signal','Location','southeast')
hold off

%FFT parameters of the output signal
magnitudes_out = abs(y_fft/(samples+1));
magnitudes_out = magnitudes_out(:,1:numpoints/2+1);
magnitudes_out(:,2:end-1) = 2*magnitudes_out(:,2:end-1);
freq_out = 0:(Omega_s/numpoints):(Omega_s/2)-Omega_s/numpoints;

%plotting the output signal in the frequency domain
subplot(2,1,2)
plot(freq_out,magnitudes_out(1:numpoints/2),'LineWidth',0.75)
xlabel('Frequency (rad/s)')
ylabel('Magnitude')
```

```
title('Single sided spectrum of the output signal')
grid on
```

Comparing the Output with the Ideal Output

```
%ideal output
yT = sin(Omega_2.*n1.*Ts);
y_ideal_fft = fft(yT,numpoints);
%FFT parameters of the ideal output signal
magnitudes_ideal = abs(y_ideal_fft/(samples+1));
magnitudes_ideal = magnitudes_ideal(:,1:numpoints/2+1);
magnitudes_ideal(:,2:end-1) = 2*magnitudes_ideal(:,2:end-1);
freq_ideal = 0:(Omega_s/numpoints):(Omega_s/2)-Omega_s/numpoints;

figure(1)
%comparing the results with the ideal output - time domain
subplot(2,1,1)
plot(n2,y_ideal,'r--','LineWidth',0.75)
hold on
stem(n1, ynT(1:samples+1),'filled','LineWidth',1)
xlabel('(n)')
ylabel('Amplitude')
title('Output Signal - Time Domain (n:300 to 350)')
xlim([300,350])
grid on
hold off

subplot(2,1,2)
plot(n2,y_ideal,'r--','LineWidth',0.75)
hold on
stem(n1, yT,'filled','LineWidth',1)
xlabel('(n)')
ylabel('Amplitude')
title('Ideal Output - Time Domain (n:300 to 350)')
xlim([300,350])
grid on
hold off

%comparing the results with the ideal output- frequency domain
figure(2)
subplot(2,1,1)
plot(freq_out,magnitudes_out(1:numpoints/2),'LineWidth',0.75)
xlabel('Frequency (rad/s)')
ylabel('Magnitude')
title('Single sided spectrum of the output signal')
grid on

subplot(2,1,2)
plot(freq_ideal,magnitudes_ideal(1:numpoints/2),'LineWidth',0.75)
xlabel('Frequency (rad/s)')
ylabel('Magnitude')
title('Single sided spectrum of the ideal output signal')
grid on
```

Published with MATLAB® R2018a