Department of Electronic and Telecommunication Engineering

Faculty of Engineering

University of Moratuwa, Sri Lanka

EN2570 Digital Signal Processing - Project Report

DESIGN OF A FINITE-DURATION IMPULSE RESPONSE BANDPASS FILTER

P.M.N.S. Bandara - 180066F

This is submitted as a partial fulfillment for the module

EN2570: Digital Signal Processing

March 5, 2021

Abstract

Finite-duration impulse response filters are considered to a well-established type of digital filters which has a wide range of applications developed throughout the last decades. These filters depict the significant foundations of the field of digital signal processing and further, coherently adhere the elementary cognition of the filter designing. These digital filters do provide the sufficient and necessary back-end potential to emerge the field of digital signal processing further into more advanced stages and applications.

In this fundamental study, a non-recursive finite-duration impulse response band-pass filter is designed using the fundamental blocks and theories, such that a specified set of prescribed filter specifications has been satisfied. The filter implementation is developed upon MATLAB 2018a software and Fourier series method and the Kaiser window method have been applied as necessary to derive the filter characteristics. Further, the study is integrated with sequential time and frequency response plots through the filter in order to validate its obvious and in-depth performance. Eventually, the study discusses the comparative results using the utilized input and the acquired output signals through the designed filter.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Finite-duration impulse response (FIR) filter is eccentrically a ubiquitous type of filter which has an impulse response of finite duration since it settles to zero in finite time. There exists several types of FIR filters such as discrete-time or continuous-time and digital or analog since of their inherited variable characteristics. In general, FIR filter do not require feedback and they are inherently stable. Further, FIR filters are preferred in the phase-sensitive applications since these filters could easily be manipulated to have linear phase by defining its coefficients in a symmetrical manner.

There exists numerous ways to design a FIR filter such as window design methods, frequency sampling method, least mean square error method and Parks-McClellan method. In this fundamental study, a non-recursive FIR band-pass filter is designed using the closed form direct approach with Fourier series method. The filter is windowed using the Kaiser window method in order to tune the filter to the specific required filter characteristics.

The design process of the FIR filter is necessarily consisted of certain time domain and frequency domain response curves as extensively discussed in the following chapters. Further, fast Fourier transform (FFT) algorithmic implementation of the discrete Fourier transform (DFT) has been primarily utilized in order to evaluate the frequency response of the designed filter. Eventually, the evaluation procedure of the filter is presented with a cohesive discussion.

Chapter 2

Methodology

The primary objective of the project is to design a non-recursive FIR band-pass filter while evaluating the designed filter with a straight-forward performance evaluation matrix. The overall methodology for this implementation thus consists of two crucial blocks: the design of the FIR band-pass filter with reference to the given specifications and the evaluation block of the designed filter using a known input signal.

The design of the required filter is comprised of certain stages including the parameter computation stage in accordance with the given specifications, the manipulation stage of the ideal frequency response of the band-pass filter, the Kaiser window development stage where the ideal frequency response is truncated preserving the required filter parameters, the visualization stage of the designed filter for the time and frequency domain representations.

The evaluation block necessarily includes an involvement of a defined input signal in order to identify the output of that signal through the developed system. An obvious comparison is executed via the system output signal and an ideal output signal in the frequency and time domains in order to validate the performance of the designed filter.

2.1 Filter Implementation

2.1.1 Prescribed Filter Specifications

Following specifications in Table 2.1 are required to be acquired by the designed FIR band-pass filter as per the requirements of the project. These necessities were referenced and utilized in the filter design process. The exact values of each parameter were calculated using the index number: *180066F*, hence A=0, B=6, C=6 and "." is F.

Table 2.1: FIR band-pass filter specifications

Parameter	Value
Maximum passband ripple, $\overline{A}_p$	0.03dB
Minimum stopband attenuation, $\overline{A}_a$	51dB
Lower passband edge, Ω_{p1}	900rad/s
Upper passband edge, Ω_{p2}	1300rad/s
Lower stopband edge, Ω_{a1}	750rad/s
Upper stopband edge, Ω_{a2}	1400rad/s
Sampling frequency, Ω_s	3600rad/s

2.1.2 Derived Filter Specifications

The values for the specifications shown in the Table 2.2 were derived using the parameter values in Table 2.1 in order to design the intended FIR *band-pass* filter and the Kaiser window.

Table 2.2: Derived filter specifications

Parameter	Derivation	Value
Lower transition width, B_{t1}	$\Omega_{p1} - \Omega_{a1}$	150rad/s
Upper transition width, B_{t2}	$\Omega_{a2} - \Omega_{p2}$	100rad/s
Critical transition width, B_t	$\min(B_{t1}, B_{t1})$	100rad/s
Lower cutoff frequency, ω_{c1}	$\Omega_{p1} - \frac{B_t}{2}$	850rad/s
Upper cutoff frequency, ω_{c2}	$\Omega_{p2} + \frac{B_t}{2}$	1350rad/s
Sampling period, T	$\frac{2\pi}{\Omega_s}$	0.00175s

2.1.3 Derivation of the Kaiser window

Since the Kaiser window function is given by,

$$\omega_K(nT) = \begin{cases} \frac{I_0(\beta)}{I_0(\alpha)} & \text{for } \mid x \mid \leq \frac{N-1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

where α is an independent parameter and

$$\beta = \alpha\sqrt{1 - \left(\frac{2n}{N-1}\right)^2} \tag{2.2}$$

$$I_0(\alpha) = 1 + \sum_{k=1}^{\infty}\left[\frac{1}{k!}\left(\frac{x}{2}\right)^k\right]^2 \tag{2.3}$$

Here, α and N could be calculated through the following equations since the parameter δ is defined such that the actual pass-band ripple in accordance to:

$$A_p \leq \overline{A}_p \tag{2.4}$$

and the minimum stopband attenuation as per:

$$A_a \leq \overline{A}_a \tag{2.5}$$

Hence, δ could be defined as:

$$\delta = \min(\delta_p, \delta_a) \tag{2.6}$$

where,

$$\delta_p = \frac{10^{0.05\overline{A}_p} - 1}{10^{0.05\overline{A}_p} + 1} \tag{2.7}$$

and

$$\delta_a = 10^{-0.05\overline{A}_a} \tag{2.8}$$

The value of δ is utilized to calculated the value of A_a.

$$A_a = -20\log(\delta) \tag{2.9}$$

As per the definitions, α could be chosen as,

$$\alpha = \begin{cases} 0 & \text{for } A_a \leq 21 \\ 0.5842(A_a - 21)^{0.4} + 0.078886(A_a - 21) & \text{for } 21 < A_a \leq 50 \\ 0.078886(A_a - 21) & \text{for } A_a > 50 \end{cases} \tag{2.10}$$

A parameter D is chosen such that N could be obtained through D. Thus,

$$D = \begin{cases} 0.9222 & \text{for } A_a \leq 21 \\ \frac{A_a - 7.95}{14.36} & \text{for } A_a > 21 \end{cases} \tag{2.11}$$

N is thus derived such that it is the smallest odd integer value which satisfies the following inequality,

$$N \geq \frac{\Omega_s D}{B_t} + 1 \tag{2.12}$$

So that, the derived Kaiser window parameters is presented in the Table 2.3

Table 2.3: Derived Kaiser window parameters

Parameter	Value
δ	0.0017
A_a	55.2545dB
A_p	0
α	5.1303
$I_0(\alpha)$	30.6087
D	3.2942
N	121

2.1.4 Derivation of the ideal impulse response

The frequency response of an ideal bandpass filter with the cutoff frequencies of ω_{c1} and ω_{c2} is obtained via,

$$H(e^{j\omega t}) = \begin{cases} 1 & \text{for } -\omega_{c2} \leq \omega \leq -\omega_{c1} \\ 1 & \text{for } \omega_{c2} \leq \omega \leq \omega_{c1} \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

By implementing inverse fourier transform, the ideal impulse response of $H(\exp j\omega t$ could be calculated as follows:

$$h(nT) = \begin{cases} \frac{2}{\Omega_s}(\omega_{c1} - \omega_{c2}) & \text{for } n = 0 \\ \frac{1}{n\pi}(\sin \omega_{c2} nT - \sin \omega_{c1} nT) & \text{otherwise} \end{cases} \tag{2.14}$$

2.1.5 Derivation of the impulse response of the windowed filter

The finite order non-causal impulse response of the windowed filter $h_\omega(nT)$ could be obtained via,

$$h_\omega(nT) = \omega_K(nT)h(nT) \tag{2.15}$$

Afterwards, the Z-transform of the designed filter could be calculated through,

$$H_\omega(z) = Z[h_\omega(nT)] = Z[\omega_K(nT)h(nT)] \tag{2.16}$$

which upon shifting for causality becomes,

$$H'_\omega(z) = z^{-(N-1)/2} H_\omega(z) \tag{2.17}$$

For the primary objective of comparison, the following filter which is in conjunction with rectangular windowing is also considered.

$$h(nT) = \begin{cases} 1 & \text{for } \mid x \mid \leq \frac{N-1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{2.18}$$

2.1.6 Evaluation of the designed filter

In order to evaluate the performance of the designed filter, an input signal x(nT) which is the sum of three sinusoidal signals, each of which has a frequency in the lower stop-band, the pass-band and the upper stop-band, is implemented with the following specific characteristics presented in Table 2.4.

$$x(nT) = \sum_{i=1}^{3} \sin(\omega_i nT) \tag{2.19}$$

Table 2.4: Input signal frequency components

Parameter	Derivation	Value
Ω_1	$\Omega_{c1}/2$	425rad/s
Ω_2	$\Omega_{c1} + \frac{\Omega_{c2}-\Omega_{c1}}{2}$	1100rad/s
Ω_3	$\Omega_{c1} + \frac{\Omega_s/2-\Omega_{c2}}{2}$	1575rad/s

Chapter 3

Results

The results of the filter design is discussed and conveyed in the following manner:

- The frequency domain and time domain impulse response plots which could be utilized to observe the general and specific characteristics of the designed filter in each stage
- The performance evaluation plots of the designed filter using a defined sinusoidal signal

3.1 The frequency and time domain response plots of the filter in several stages of the filter design

The set of MATLAB plots obtained through various stages of the FIR filter design process are presented in Figure 3.1 to Figure 3.11. The impulse response of the designed Kaiser window is conveyed in the Figure 3.1 and Figure 3.2 presents the time domain representation of the filter causal response in conjunction with the Kaiser window.

The frequency domain magnitude response of the designed FIR band-pass filter is presented in Figure 3.3 while it has been further analysed through Figure 3.4 which is obtained through the MATLAB inbuilt *fvtool* and adjoined with the phase response of the designed band-pass filter. Figure 3.5 and Figure 3.6 presents the lower and upper stop-band regions of the designed band-pass filter in the frequency domain, respectively while Figure 3.7 illustrates the pass-band characteristics of the FIR filter in the frequency domain. Further, Figure 3.7 visually represents the pass band ripples in the pass-band of the filter.

The rectangular window conjunction with the designed FIR filter is presented in Figure 3.8 as a time domain representation using MATLAB software and the corresponding frequency domain representation is represented via Figure 3.8.

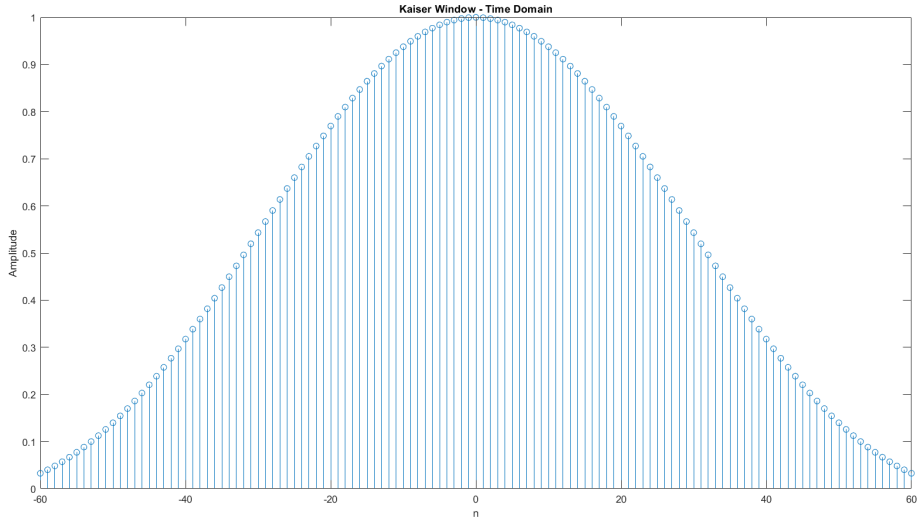

Figure 3.1: Impulse response of the implemented Kaiser window

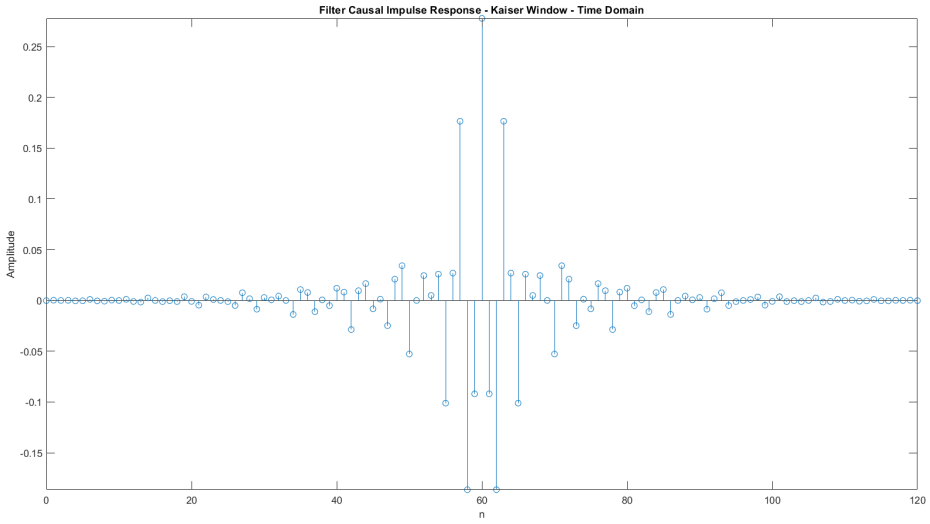

Figure 3.2: Time domain representation of the designed FIR band-pass filter in conjunction with the Kaiser window

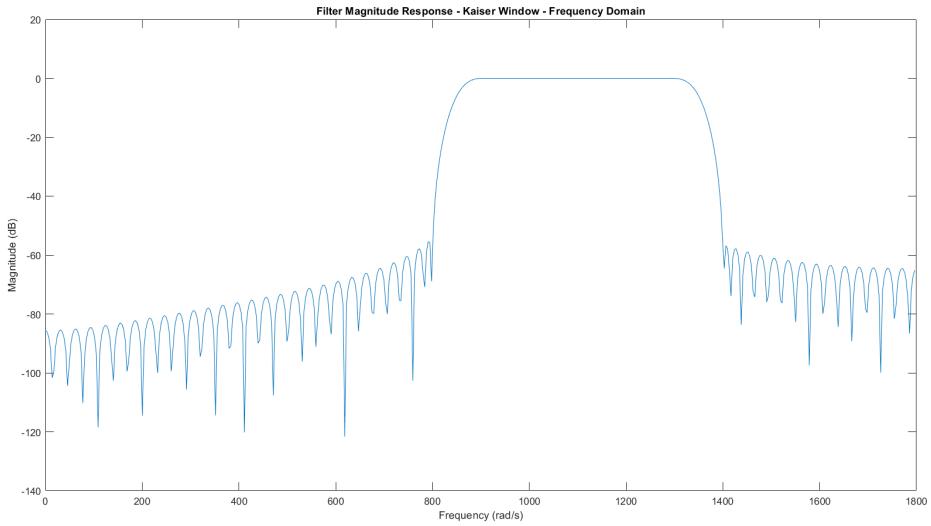

Figure 3.3: Frequency domain representation of the designed FIR band-pass filter in conjunction with the Kaiser window

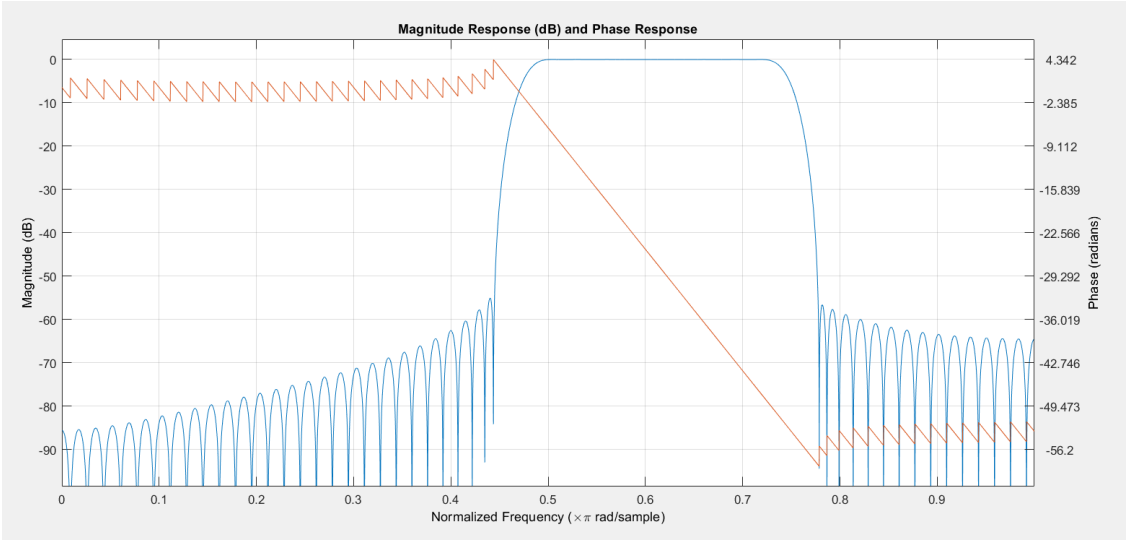

Figure 3.4: Frequency domain representation of the designed FIR band-pass filter in conjunction with the Kaiser window: fvtool

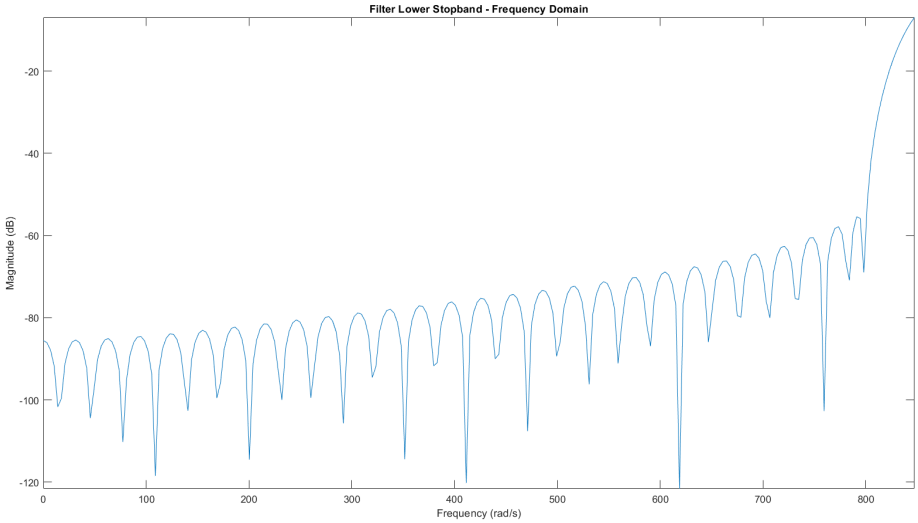

Figure 3.5: Frequency domain representation of the lower stop-band in the designed filter in conjunction with the Kaiser window

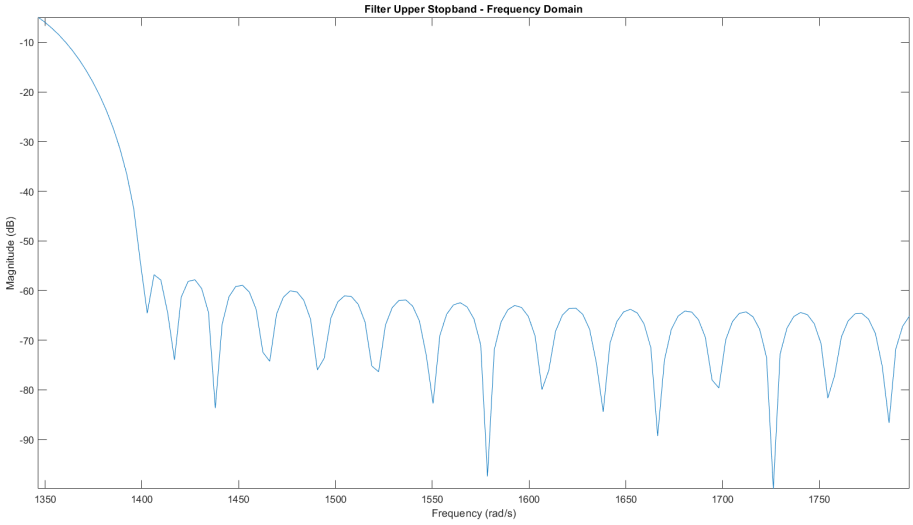

Figure 3.6: Frequency domain representation of the upper stop-band in the designed filter in conjunction with the Kaiser window

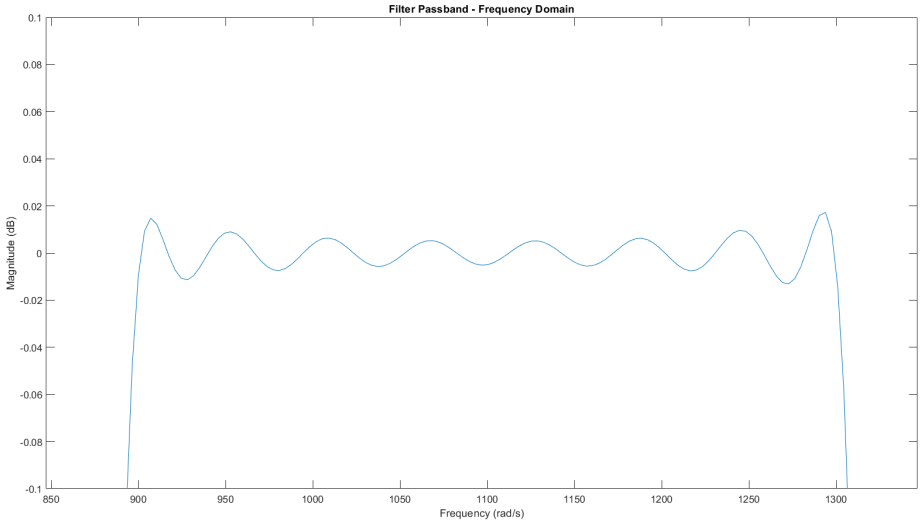

Figure 3.7: Frequency domain representation of the pass-band in the designed FIR band-pass filter in conjunction with the Kaiser window

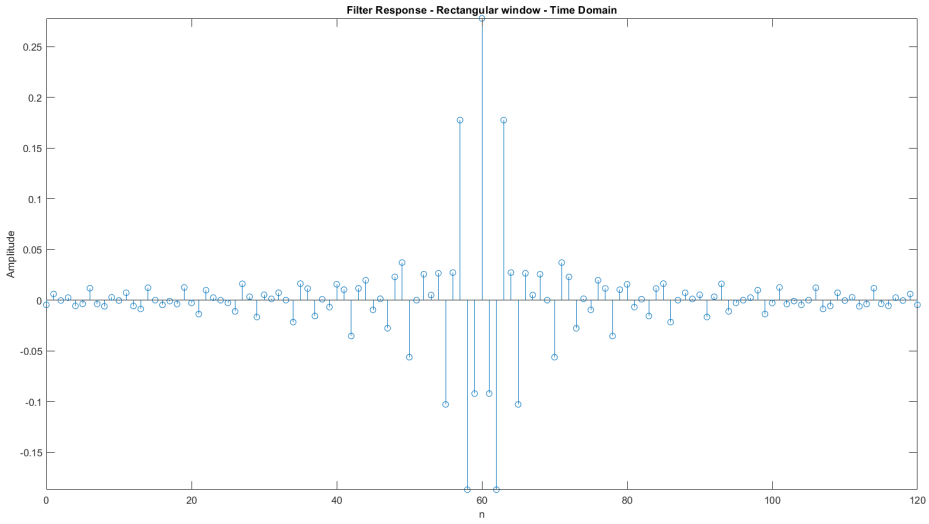

Figure 3.8: Time domain representation of the designed FIR band-pass filter in conjunction with the rectangular window

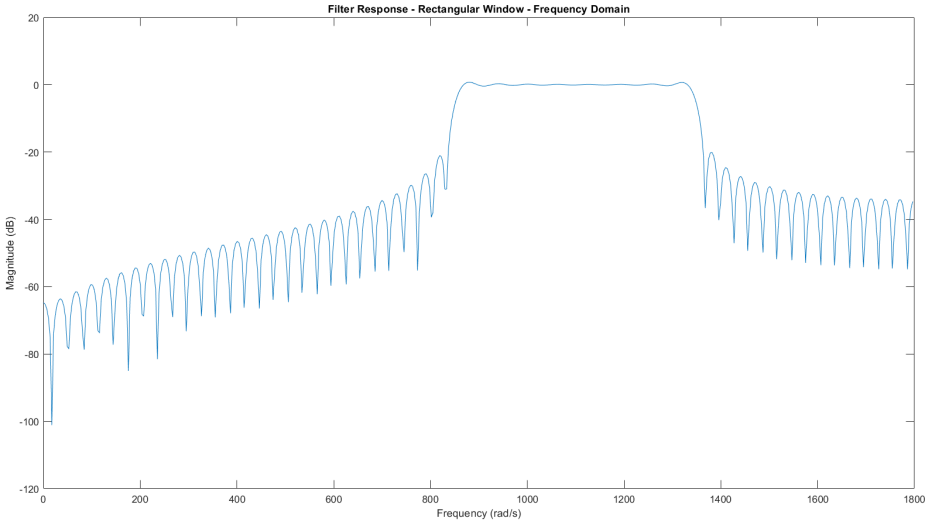

Figure 3.9: Frequency domain representation of the designed FIR band-pass filter in conjunction with the rectangular window

3.2 The performance evaluation plots of the designed filter using a defined sinusoidal signal

The designed FIR filter is fed using a sinusoidal input signal in order to evaluate the obvious performance of the filter. This input signal is consisted of three mid-band frequencies such that where Ω_1 is the middle frequency of the lower stopband, Ω_2 is middle frequency of the passband, and Ω_3 is middle frequency of the upper stopband.

The time domain and the frequency domain representations of the input signal is presented in Figure 3.10 and the sampled input signal in the time domain is conveyed with an envelope in orange. Further, the output signal in the time and frequency domains through the designed filter is presented in Figure 3.11 and it has been obviously conveyed that the FIR filter is capable of filtering the input signal as intended.

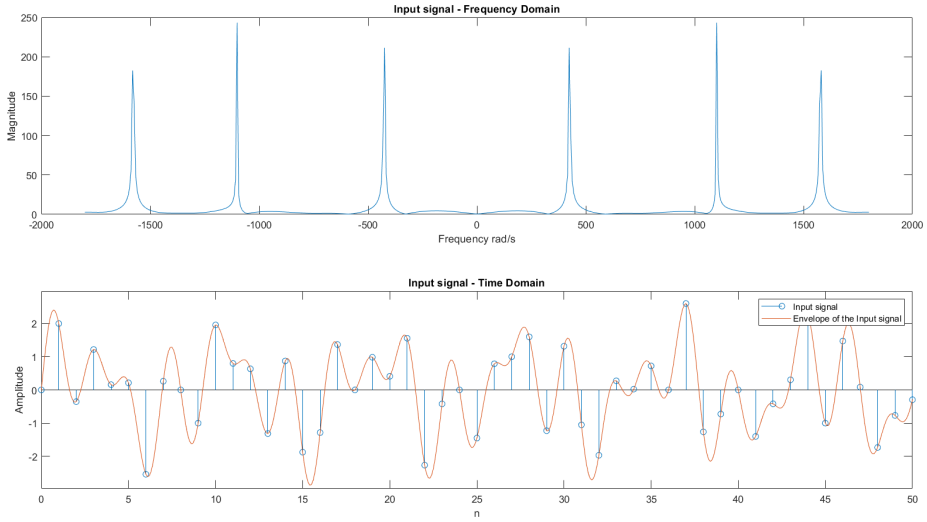

Figure 3.10: Frequency domain and time domain representations of the input signal

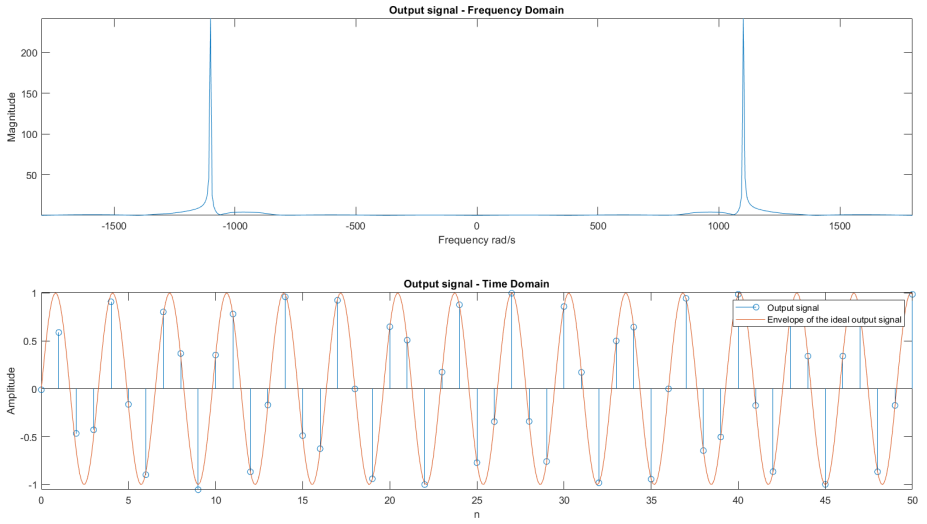

Figure 3.11: Frequency domain and time domain representations of the output signal

Discussion

The results depict that the designed filter is capable of achieving the intended objectives by satisfying the given filter specifications. The magnitude response of the filter in the frequency domain in Figure 3.3 obviously convey that the designed band-pass filter will attenuate the signal components outside the specified band-pass frequency range with a maximum stop-band attenuation of 55.25dB. In addition, the pass-band ripples of the designed filter satisfies the specified filter requirement of being within the given maximum limitation of 0.03dB as per Figure 3.7 and thus, the output signal is not considerably distorted. The comparison of the filter results with the ideal theoretical results conveys that the output signals are almost similar (Figure 3.9) and thus, the filter design is promising to be utilized. Therefore, it is safe to assume that the designed filter is capable of filtering the unwanted frequency components of the input signals to a satisfactory level of performance.

The significant differences in the acquired results through the Kaiser window method and the rectangular window method suggests that the filter requirements are sufficiently satisfied only by the Kaiser window method since the maximum pass-band ripples of the rectangular window method is not within the limit of 0.03dB and further, it was unsuccessful in achieving the maximum stop-band attenuation as required in the filter specifications (Figures 3.8 and 3.9). Further, the time domain representations of the designed FIR filter obviously shows that the Kaiser Window method gradually attenuates samples at the edge of the window (Figure 3.2), while the rectangular window truncates the impulse response abruptly especially at the edges (Figure 3.8). This causes the pass-band of the filter in conjunction with the Kaiser window to be smoother than the rectangular approach. Figure 3.10 and 3.11 clearly show that the filter achieves its intended task since both mid-frequencies in the lower and upper stop bands are evidently removed in the output frequency domain representation of the signal and further, the ideal envelope of the time domain output signal sufficiently correlates with the time domain sampled output signal from the designed FIR band-pass filter.

Conclusion

The efficiency, suitability and flexibility of utilizing the Kaiser windowing method in FIR filter design are cohesively addressed through this fundamental project where the filter specifications and the limitations are defined and successfully controlled. Since there is no ideal filter in practical scenarios, it is utmost important to use flexible filters which have the potential to reach the ideal behaviour with small neglect-able imperfections such as having extremely low pass-band ripples which will not cause observable distortion in the output signal. In these scenarios, the implemented Kaiser windowed filter could be considered as a suitable candidate since of its parameter flexibility which leads it to have indistinguishable performance difference with an ideal filter.

Howsoever, it is needed to note that the higher order of the implemented filter is a significant drawback of the system since the hardware implementation of higher order filters are usually inefficient since of adding more unit delays and multipliers. Further, the software implementation of these kinds of higher order filter demands more computational power per sample. Therefore, more advanced techniques should be implemented to achieve the same acquired performance while having an optimal order number.

Appendix

FIR band-pass filter design project - MATLAB scripts

```
clc;
clear all;
close all;

%indexNo = 180066F; By comparing with 180ABC. -> A=0, B=6, C=6, .=F
A = 0;
B = 6;
C = 6;
```

Figure 3.12: MATLAB implementation of parameter initialization as per the index number

```
A_p = 0.03+(0.01*A); % dB %%max passband ripple
A_a = 45+B; %dB %%min stopband attenuation
Op1 = (C*100)+300; %rad/s %%lower passband edge
Op2 = (C*100)+700; %rad/s %%upper passband edge
Oa1 = (C*100)+150; %rad/s %%lower stopband edge
Oa2 = (C*100)+800; %rad/s %%upper stopband edge
Os = 2*(C*100+1200); %rad/s %%sampling freqency
```

Figure 3.13: MATLAB implementation of required FIR filter specifications

```
Bt1 = Op1-Oa1; %rad/s %%lower transition width
Bt2 = Oa2-Op2; %rad/s %%upper transisiton width
Bt = min(Bt1,Bt2); %rad/s %%critical transition width
Oc1 = Op1-Bt/2; %rad/s %%lower cutoff frequency
Oc2 = Op2+Bt/2; %rad/s %%upper cutoff frequency
T = 2*pi/Os; %s %%sampling period
```

Figure 3.14: MATLAB implementation of derived FIR band-pass filter specifications

```
%Calculating delta
d_P = (10^(0.05*A_p) - 1)/ (10^(0.05*A_p) + 1);
d_A = 10^(-0.05*A_a);
delta = min(d_P,d_A);
Aa = -20*log10(delta); % Actual stopband attenuation
Ap = 20*log10(1+delta/1-delta); % Actual passband ripple

%Calculating 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
```

Figure 3.15: MATLAB implementation of deriving the Kaiser window parameters: δ and α

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

% Calculating order of the filter N
N = ceil(Os*D/Bt +1);
if mod(N,2) == 0
    N = N+1;
end

% Length of the filter
n = -(N-1)/2:1:(N-1)/2;

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

Figure 3.16: MATLAB implementation of deriving the Kaiser window parameters: D, N and β

```
bessellimit = 100;
Io_alpha = 1;
for k = 1:bessellimit
    val_k = (1/factorial(k)*(alpha/2).^k).^2;
    Io_alpha = Io_alpha + val_k;
end

Io_beta = 1;
for m = 1:bessellimit
    val_m = (1/factorial(m)*(beta/2).^m).^2;
    Io_beta = Io_beta + val_m;
end
```

Figure 3.17: MATLAB implementation of deriving $I_0(\alpha)$ and $I_0(\beta)$

```
wk_nT = Io_beta/Io_alpha;
figure
stem(n,wk_nT)
xlabel('n')
ylabel('Amplitude')
title('Kaiser Window - Time Domain');
```

Figure 3.18: MATLAB implementation of obtaining the Kaiser window

```
n_L= -(N-1)/2:1:-1;
hnt_L = 1./(n_L*pi).*(sin(Oc2*n_L*T)-sin(Oc1*n_L*T));
n_R = 1:1:(N-1)/2;
hnt_R = 1./(n_R*pi).*(sin(Oc2*n_R*T)-sin(Oc1*n_R*T));
hnt_0 = (2/Os)*(Oc2-Oc1);
n = [n_L,0,n_R];
h_nT = [hnt_L,hnt_0,hnt_R];
```

Figure 3.19: MATLAB implementation of generating impulse response

```
Hw_nT = h_nT.*wk_nT;
```

Figure 3.20: MATLAB implementation of the application of the Kaiser window to the generated filter

```
n_shifted = [0:1:N-1];
figure
fvtool(Hw_nT);
stem(n_shifted,Hw_nT); axis tight;
xlabel('n')
ylabel('Amplitude')
title(strcat(['Filter Causal Impulse Response - Kaiser Window - Time Domain']));
```

Figure 3.21: MATLAB implementation of plotting the causal impulse response

```
figure
[Hw,f] = freqz(Hw_nT);
w_1 = f*Os/(2*pi);
log_Hw = 20*log10(abs(Hw));
plot(w_1,log_Hw)
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
title(strcat(['Filter Magnitude Response - Kaiser Window - Frequency Domain']));
```

Figure 3.22: MATLAB implementation of plotting the magnitude response of the filter in the range of $(0,\Omega_s/2)$

```
figure
finish = round((length(w_1)/(Os/2)*Oc1));
wpass_l = w_1(1:finish);
hpass_l = log_Hw(1:finish);
plot(wpass_l,hpass_l)
axis([-inf, inf, -inf, inf]);
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
title('Filter Lower Stopband - Frequency Domain');
```

Figure 3.23: MATLAB implementation of plotting the magnitude response of the filter in the lower stop-band

```
figure
start = round(length(w_1)/(Os/2)*Oc2);
wpass_h = w_1(start:length(w_1));
hpass_h = log_Hw(start:length(w_1));
plot(wpass_h,hpass_h)
axis([-inf, inf, -inf, inf]);
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
title('Filter Upper Stopband - Frequency Domain');
```

Figure 3.24: MATLAB implementation of plotting the magnitude response of the filter in the upper stop-band

```
figure
start = round(length(w_1)/(Os/2)*Oc1);
finish = round(length(w_1)/(Os/2)*Oc2);
wpass_h = w_1(start:finish);
hpass_h = log_Hw(start:finish);
plot(wpass_h,hpass_h)
axis([-inf, inf, -0.1, 0.1]);
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
title('Filter Passband - Frequency Domain');
```

Figure 3.25: MATLAB implementation of plotting the magnitude response of the filter in the pass-band

```
figure
stem(n_shifted,h_nT); axis tight;
xlabel('n')
ylabel('Amplitude')
title(strcat(['Filter Response - Rectangular window - Time Domain']));

figure
[hw,f] = freqz(h_nT);
w_2 = f*Os/(2*pi);
log_H = 20*log10(hw);
plot(w_2,log_H)
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
title(strcat(['Filter Response - Rectangular Window - Frequency Domain']));
```

Figure 3.26: MATLAB implementation of plotting the rectangular filter response

```
%Component frequencies of the input
O1 = Oc1/2;
O2 = Oc1 + (Oc2-Oc1)/2;
O3 = Oc2 + (Os/2-Oc2)/2;

%Generate discrete signal and evelope
samples = 500;
n1 = 0:1:samples;
n2 = 0:0.1:samples;
X_nT = sin(O1.*n1.*T)+sin(O2.*n1.*T)+sin(O3.*n1.*T);
X_env = sin(O1.*n2.*T)+sin(O2.*n2.*T)+sin(O3.*n2.*T);
```

Figure 3.27: MATLAB implementation of input signal generation

```
%Filtering using frequency domain multiplication
len_fft = length(X_nT)+length(Hw_nT)-1; %length for fft in x dimension
x_fft = fft(X_nT,len_fft);
Hw_nT_fft = fft(Hw_nT,len_fft);
out_fft = Hw_nT_fft.*x_fft; %a shift in time is added here
out = ifft(out_fft,len_fft);
rec_out = out(floor(N/2)+1:length(out)-floor(N/2)); %account for shifting delay
```

Figure 3.28: MATLAB implementation of using discrete fourier transform for checking the performance of the filter

```
%Ideal Output Signal
ideal_out = sin(O2.*n2.*T);
```

Figure 3.29: MATLAB implementation of the ideal output signal from the filter

```
figure
subplot(2,1,1)
len_fft = 2^nextpow2(numel(n1))-1;
x_fft = fft(X_nT,len_fft);
x_fft_plot = [abs([x_fft(len_fft/2+1:len_fft)]),abs(x_fft(1)),abs(x_fft(2:len_fft/2+1))];
f = Os*linspace(0,1,len_fft)-Os/2;
plot(f,x_fft_plot);
xlabel('Frequency rad/s')
ylabel('Magnitude')
title(strcat(['Input signal',' ','- Frequency Domain']));
```

Figure 3.30: MATLAB implementation of plooting the input signal in the frequency domain

```
%Time domain representation of input signal before filtering
subplot(2,1,2)
stem(n1,X_nT)
axis([0, 50, -inf, inf]);
xlabel('n')
ylabel('Amplitude')
title(strcat(['Input signal',' ','- Time Domain']));
hold on
plot(n2,X_env)
legend('Input signal','Envelope of the Input signal');
```

Figure 3.31: MATLAB implementation of plotting the input signal in the time domain with envelope

```
%Frequency domain representation of output signal after filtering
figure
subplot(2,1,1)
len_fft = 2^nextpow2(numel(n1))-1;
xfft_out = fft(rec_out,len_fft);
x_fft_out_plot = [abs([xfft_out(len_fft/2+1:len_fft)]),abs(xfft_out(1)),abs(xfft_out(2:len_fft/2+1))];
f = Os*linspace(0,1,len_fft)-Os/2;
plot(f,x_fft_out_plot); axis tight;
xlabel('Frequency rad/s')
ylabel('Magnitude')
title(strcat(['Output signal',' ','- Frequency Domain']));
```

Figure 3.32: MATLAB implementation of plotting the output signal in the frequency domain

```
% Time domain representation of output signal after filtering
subplot(2,1,2)
stem(n1,rec_out)
axis([0, 50, -inf, inf]);
xlabel('n')
ylabel('Amplitude')
title(strcat(['Output signal',' ','- Time Domain']));
hold on
plot(n2,ideal_out)
legend('Output signal','Envelope of the ideal output signal');
```

Figure 3.33: MATLAB implementation of plotting the output signal in the time domain with envelope