Department of Electronic & Telecommunication Engineering
University of Moratuwa

PROJECT REPORT

DESIGNING A FINITE IMPULSE RESPONSE BANDPASS FILTER

Name	**Index Number**
K.H. Hansanganie	180220A

This report is submitted as a partial fulfillment for the module
EN2570: Digital Signal Processing

05th of March 2021

Abstract **- This is a detailed report on the design of a Finite Impulse Response (F.I.R) Bandpass filter using the Fourier Series method and its implementation on Matlab. The filter is designed under given specifications by using the Fourier Series method along with a Kaiser Window. The report analyses the magnitude response of the filter to confirm that it corresponds to the starting parameters and also compares the output of the designed filter for an input signal consisting of a summation of sinusoids with the output of an ideal filter.**

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

The purpose of this project is to implement a digital bandpass filter under given specifications. A digital bandpass filter is a signal processing tool which attenuates all frequency components outside a certain frequency range in a sampled, discrete time signal. Digital filters are an essential element of Telecommunication Engineering and are commonplace in everyday electronics.

MATLAB (Version R2019b, MathWorks Inc.) is used to do software implementation of the filter. There are many methods of designing a digital FIR Filter with varying levels of complexity and accuracy. Fourier series method was used to implement the filter and a Kaiser window was used for windowing in accordance with direct closed form approach. The parameters of the Kaiser Window are used to tune the filter to the given specifications. The frequency response of the filter was obtained primarily through the Fast Fourier Transform (FFT) implementation of Discrete Fourier Transform (DFT). A combined signal of sinusoids of three frequencies out of which one lies in the pass band is input to the filter, to test the resulting filter implementation. Then the output is compared with the original sinusoidal component of the input in the pass band. Problems and improvements that can be made to address them, can identify by analyzing the results.

II. METHOD

Implementing a band pass filter and evaluating its performance are the two basic tasks of this project. The basis of these two tasks will be discussed separately. Several stages want to follow to design the filter. First stage is computation of design parameter according to the filter requirements that are provided. The next stage is to obtain the ideal frequency response of the pass band filter. A Kaiser Window function is then obtained by truncating the infinite impulse response on an ideal filter using a Kaiser Window of given parameters. Finally, the time domain and frequency domain representation of the designed filter are obtained. The task of evaluation involves generation of the input signals following which the output is obtained by convolution of the input with the designed filter. Comparison between the expected output and the filter response is done by frequency and time domain plots of the inputs and the outputs.

A. *Implementation of the Filter*

The parameters used for this implementation are given in Table 1.
Index number : 180220A A=2, B=2, C=0

TABLE I

GIVEN SPECIFICATIONS

Specification	Symbol	Value	Units
Maximum pass band ripple	$\tilde{A}_p$	0.03 + (0.01 × A)=0.05	dB
Minimum stop band attenuation	$\tilde{A}_a$	45+B=47	dB
Lower pass band edge	ω_{p1}	(C × 100) + 300=300	rad/s
Upper pass band edge	ω_{p2}	(C × 100) + 700=700	rad/s
Lower stop band edge	ω_{a1}	(C × 100) + 150=150	rad/s
Upper stop band edge	ω_{a2}	(C × 100) + 800=800	rad/s
Sampling frequency	ω_s	2[(C × 100) + 1200]=2400	rad/s

- *Filter Parameter Derivation:*

Using the above parameters, the following parameters shown in Table II were derived in order to design the filter and the Kaiser Window.

TABLE II

DERIVED SPECIFICATIONS

Derived Parameter	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 cutoff frequency	ω_{c1}	$\omega_{p1} - \frac{Bt}{2}$	250	rad/s
Upper cutoff frequency	ω_{c2}	$\omega_{p2} + \frac{Bt}{2}$	750	rad/s
Sampling period	T	$\frac{2\pi}{\omega s}$	0.0026	s

- *Derivation of the Kaiser Window:*

Using the derived parameters, the Kaiser window is obtained in the following form,

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

Where,

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

The parameters α and N are calculated as follows.

δ is chosen such that the actual passband ripple, Ap is equal to or less than specified passband ripple $\tilde{A}p$, and the actual minimum stopband attenuation, Aa is equal or greater than the specified minimum stopband attenuation, $\tilde{A}a$.

This is achieved when,

$$\delta = \min(\delta_p, \ \delta_a)$$

Where $\delta_p = \frac{10^{0.05\,\tilde{A}p} - 1}{10^{0.05\,\tilde{A}p} + 1}$ and $\delta_a = 10^{-0.05\tilde{A}_a}$

With the δ obtained thus,

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

Now α is chosen as

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

In order to obtain N we define a parameter D as

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

The minimum value is selected for N that would satisfy the inequality ,

$$N \ge \frac{\omega_s D}{B_t} + 1$$

A summary of Kaiser Window parameters obtained is shown in Table III.

TABLE III

KAISER WINDOW PARAMETERS

Parameter	Value	Units
δ	0.0029	-
A_a	51	dB
α	4.6413	-
D	2.9852	-
N	73	-

- *Derivation of The Ideal Impulse Response:*

The frequency response of an ideal band pass filter with cut off frequencies ω_{c1} and ω_{c2} are given by,

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

Using Inverse Fourier transform Impulse response of $H(e^{j\omega T})$ is obtained to be

$$h(nT) = \begin{cases} \dfrac{2}{\omega_s}(\omega_{c2} - \omega_{c1}) & for\ n = 0 \\ \dfrac{1}{n\pi}(\sin\omega_{c2}nT - \sin\omega_{c1}nT) & Otherwise \end{cases}$$

- *Obtaining the Impulse Response of the Windowed Filter:*

The impulse response of the filter (nT) can be obtained by the multiplication of the Ideal Impulse Response $h[nT]$ and the Kaiser Window $w_k(nT)$.

$$\text{filter}(nT) = w_k(nT)\text{h(nT)}$$

The z – Transform of which takes the form

$$H_w(z) = \text{Ƶ}[w_k(nT)\text{h(nT)}]$$

Upon shifting for causality which becomes

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

$\text{filter}(nT)$ in time domain is the final filter. In order for comparison we also obtain a filter that is windowed with a rectangular window.

B. Filter Evaluation

To evaluate the performance of the generated filter, an input signal $x(nT)$ is constructed as the sum of three sinusoid each having frequency below, within and above the pass band as shown in table IV. The output can be obtained by the convolution of the filters impulse response $h_w[nT]$ and $x(nT)$. To avoid the use of convolution operation, we obtain the DFT of these two signals and multiply them in the frequency domain to obtain the frequency domain representation of the output, which is again converted back to time domain by IDFT. DFT and IDFT are implemented respectively by FFT and IFFT methods in MATLAB.

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

TABLE IV

INPUT FREQUENCY COMPONENTS

Frequency Component	Value	Unit
ω_1	75	rad/s
ω_2	500	rad/s
ω_3	1000	rad/s

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.

Ideal output: $\sin(\omega_2 nT)$; Since $\omega 1$ and $\omega 3$ are outside the passband.

III. RESULTS

The results of the filter design can be presented in two forms. The characteristics of the filter in each stage of the filter design process can be seen by the impulse response and frequency response plots. The performance of the filter can be evaluated by comparing the input and output obtained by the filter evaluation stage.

A. Frequency and Time Domain plots of the Filter

The following plots were obtained during the procedure of designing the filter. Impulse response of the Kaiser Window is shown in figure 1. Frequency response of the filter when a Kaiser Window was used is shown in figure 2. Impulse response of the filter when a Kaiser Window was used is shown in figure 3. A zoomed in plot of the pass band of the signal is obtained in order to visually identify the pass band ripples, as shown in figure 4.

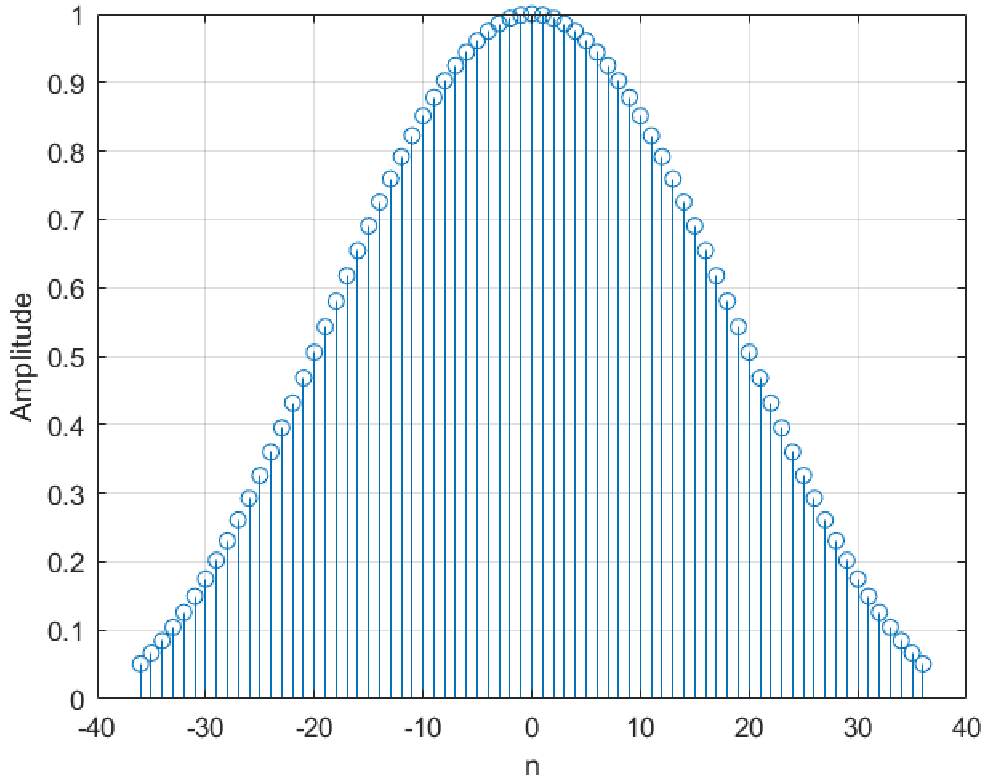

Figure 1. Impulse response of Kaiser Window

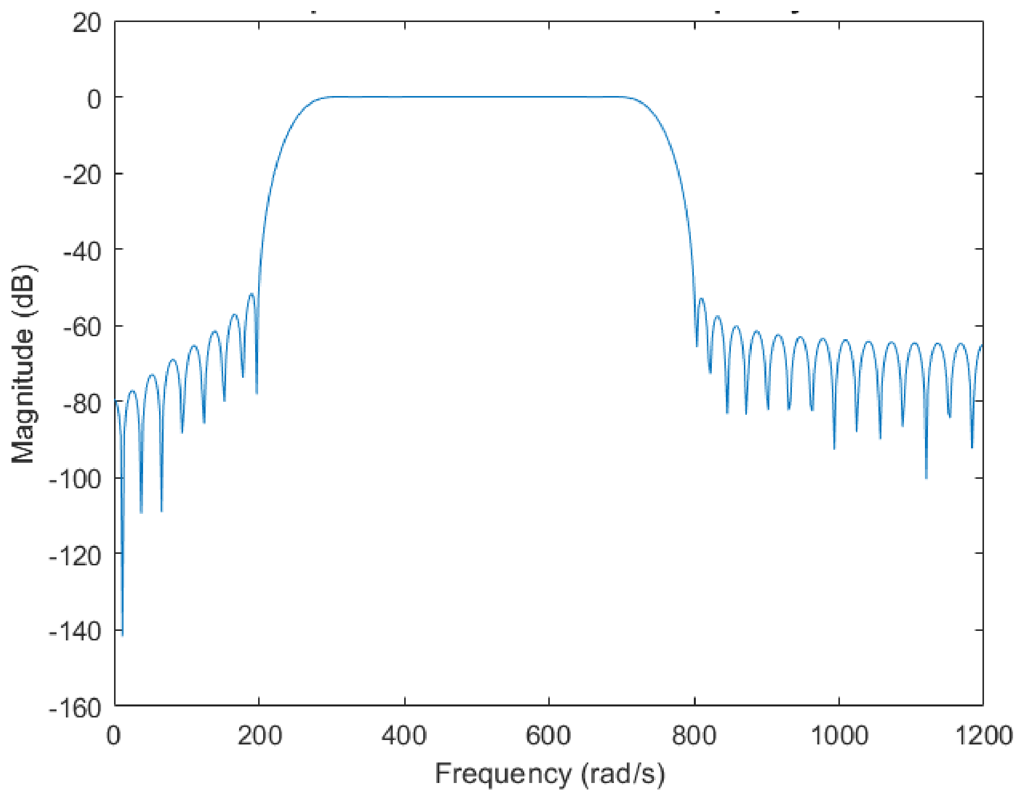

Figure 2. Frequency Domain Representation of the filter designed using a Kaiser Window Function

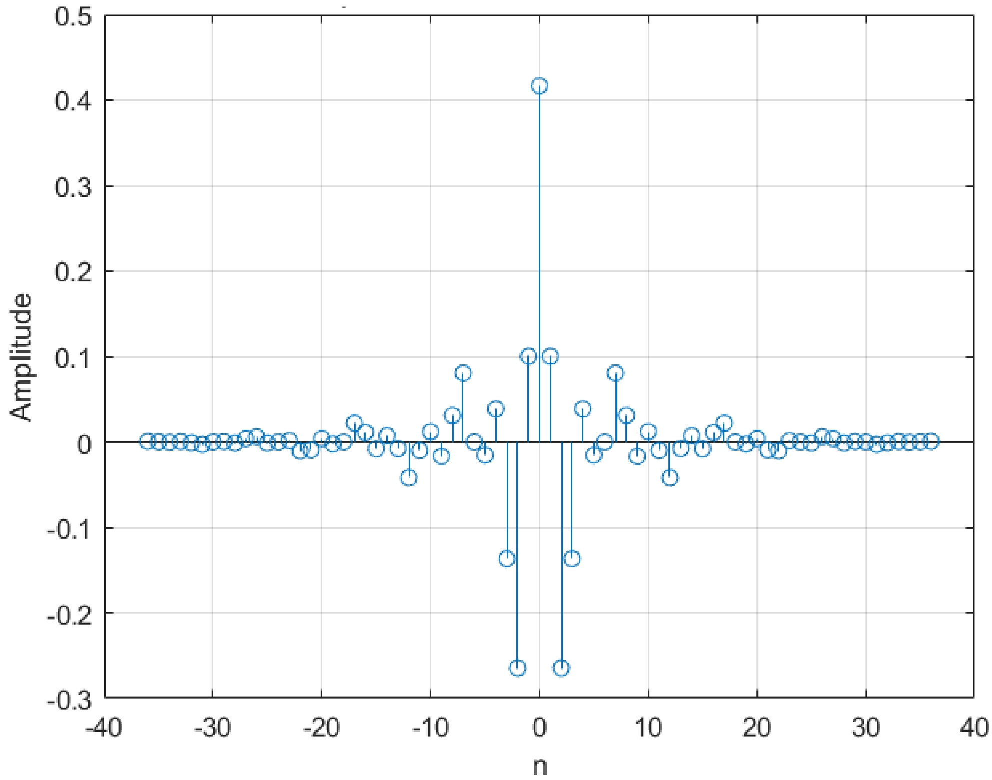

Figure 3. Time Domain Representation of the filter designed using a Kaiser Window Function

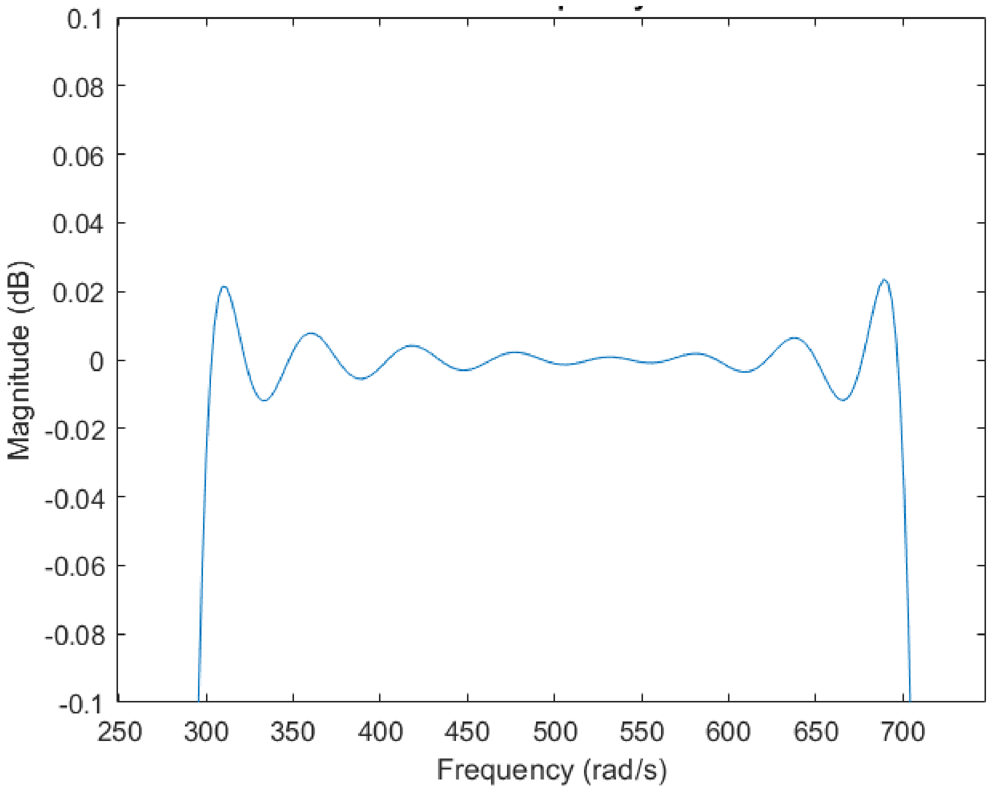

Figure 4. Pass band of the Filter designed using the Kaiser Window

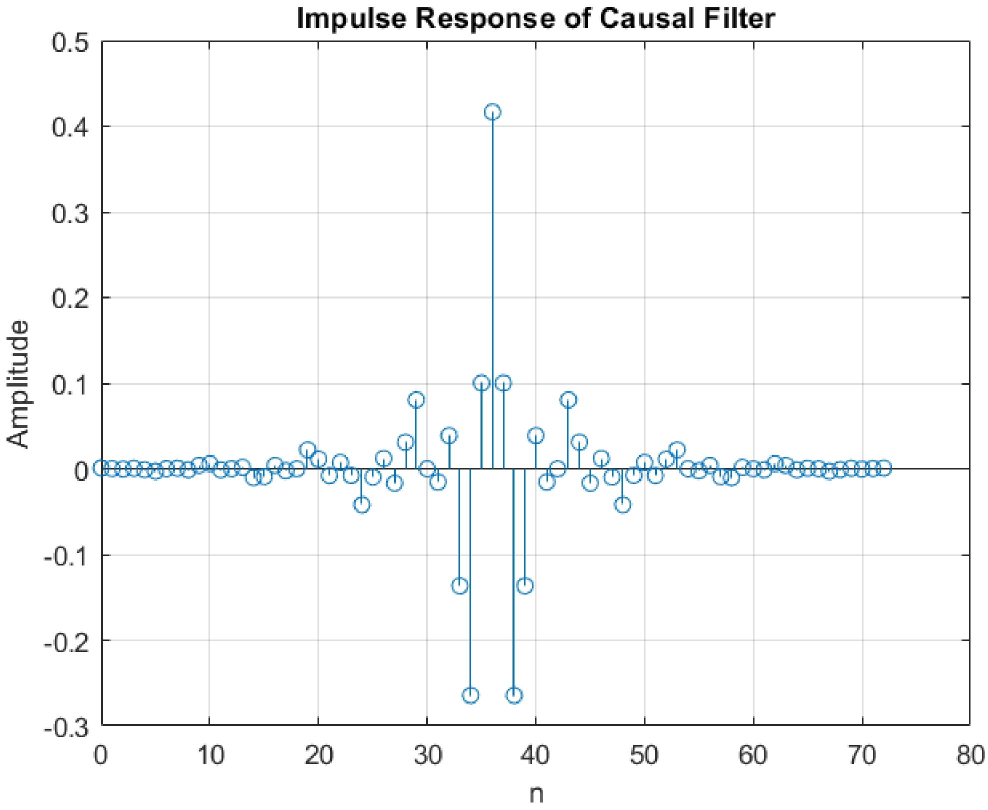

Figure 5. Impulse Response of Causal Filter

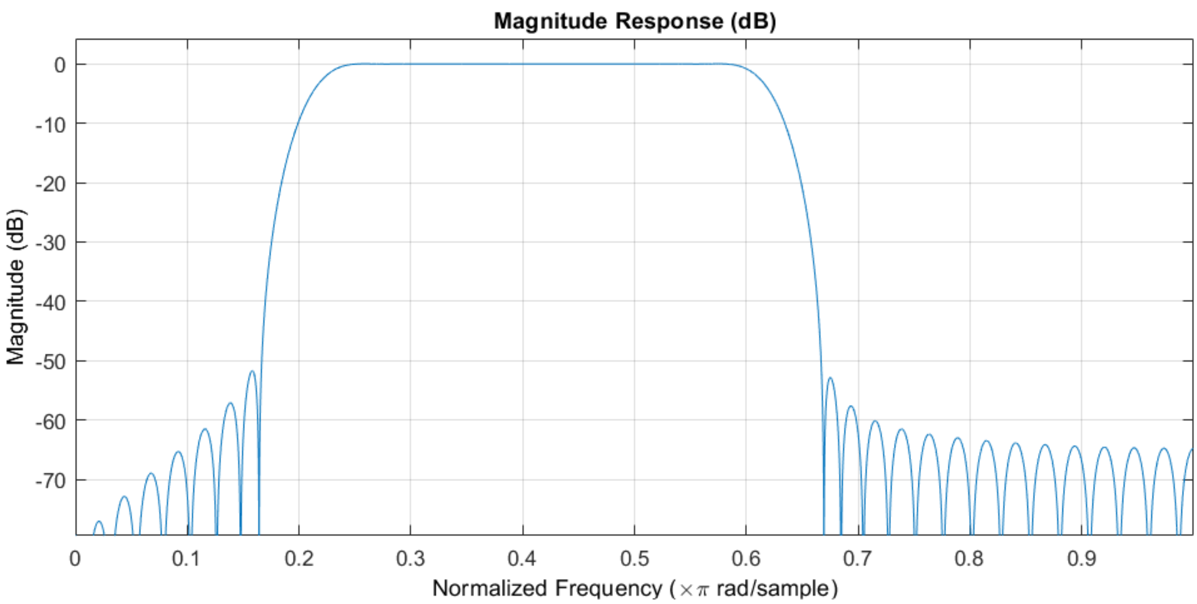

Figure 6. Magnitude Response of the FIR Filter

B. Input and Output of the Filter

Figure 7 show the input, output and expected relationship of the filter observed for a sinusoidal signal (all are in frequency domain plots) and figure 8 shows the time domain plots.

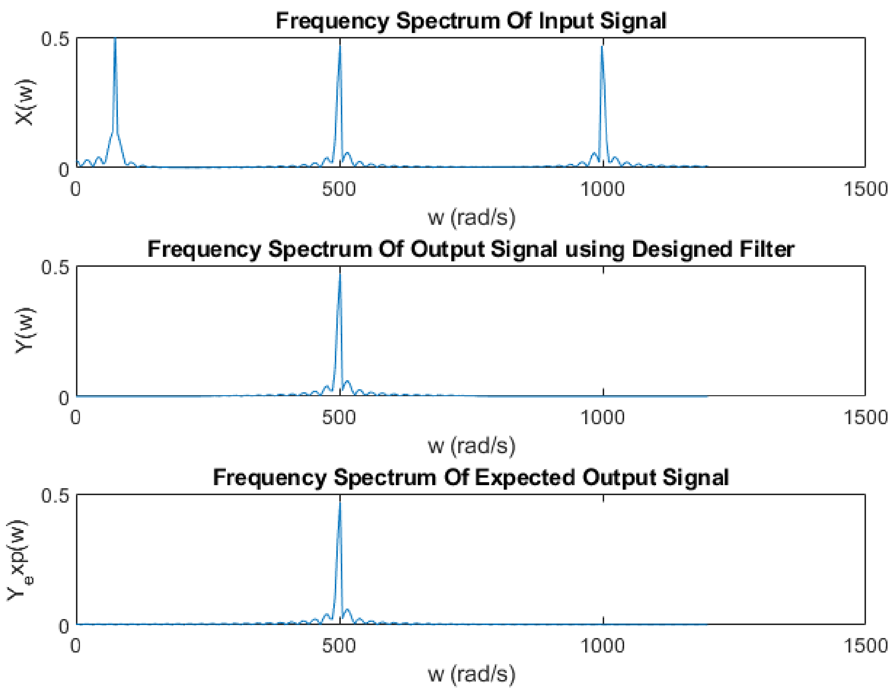

Figure 7. Input, Output and Expected Output Signals Comparison – Frequency Domain

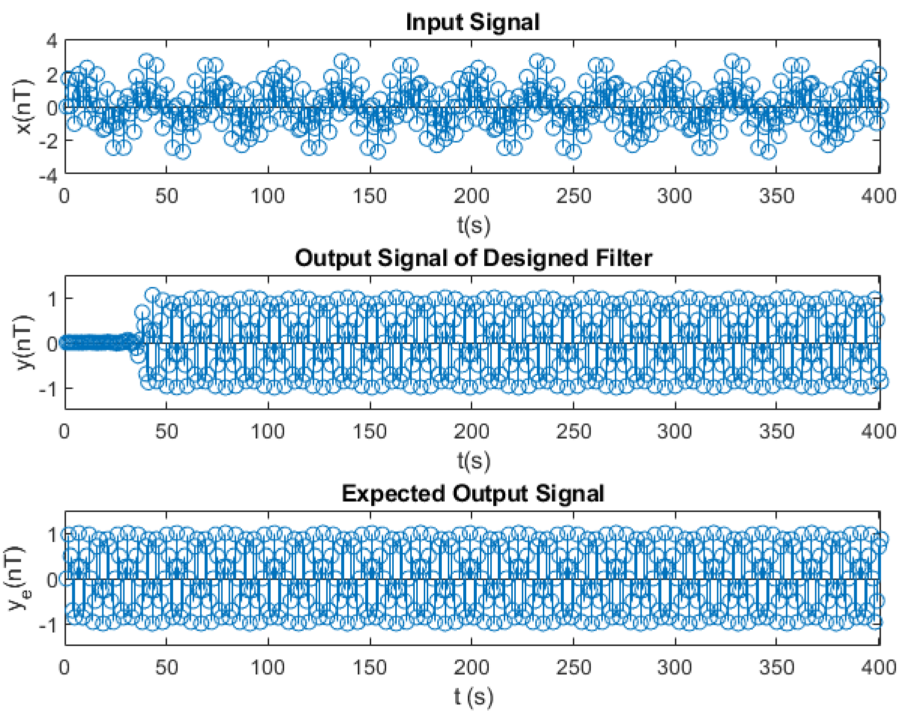

Figure 8. Input, Output and Expected Output Signals Comparison – Time Domain

IV. DISCUSSION

Results show that the expected objectives of the project have been achieved. According to the figure 2, the amplitude response of the filter attenuates signal components outside the specified pass band frequency range with the specified stopband attenuation of -51db. Figure 4 shows that the passband ripple is within the given limitation (0.05 dB) in order to reduce distortions in the output signal. According to the figures 7 and 8, the results of the designed filter and the ideal results are similar in both the time and frequency domains. These results ensure that unwanted frequencies have been completely filtered out by using this filter.

Above results evident the flexibility of the Kaiser window. Since ideal filters cannot be practically implemented, it is advantageous to be able to make a flexible filter of which the limitations can be controlled. This is a practical approach since small imperfections such as passband ripple will not cause an observable difference in the filtered output. This means that the parameters of the filter can be adjusted until the differences between the output and an ideal output become indistinguishable for all practical purposes.

V. CONCLUSION

From the comparison above we see that the results of the designed filter are identical to that of an ideal filter. This means that the Kaiser Window method can give acceptable practical results. Even though it is not an ideal filter (as seen by the plots of the passband and stopband ripples), this method is flexible and can be adjusted using the input parameters. However, this method is suboptimal due to the high order of the resulting filter. This leads to lower efficiency and is therefore addressed by other optimal filter design methods that can generate the same results using a lower order.

VI. ACKNOWLEDGEMENT

I would like to thank the lecturer in charge of this project, Dr. Chamira Edussooriya, for giving me the necessary knowledge and guidance needed to complete this project. I would also like to thank Ms. S. M. Salgado for her advice and training on technical report writing. Finally, I would like to thank my peers who helped me when in doubt.

VII. REFERENSES

[1] A. Antoniou, "Digital Signal Processing," 2005. [Online]. Available: www.ece.uvic.ca/~dsp. [Accessed 29 12 2016].

[2] Discrete-Time Signal Processing by Alan V. Oppenheim, Ronald W. Schafer (z-lib.org)

APPENDIX I

MATLAB SCRIPT FOR THE SOFTWARE IMPLMENTATION

Index Number 180220A

```
clc;
clear all;
close all;
```

Specifications Given

```
A = 2;
B = 2;
C = 0;

A_p = 0.03+0.01*A;  % dB max passband ripple
A_a = 45+B; %dB min stopband attenuation
wp1 = C*100+300; %rad/s lower passband edge
wp2 = C*100+700; %rad/s upper passband edge
wa1 = C*100+150; %rad/s lower stopband edge
wa2 = C*100+800; %rad/s upper stopband edge
ws = 2*(C*100+1200); %sampling frequency
```

Derived Specifications

```
bt1 = wp1-wa1;      %lower transition width
bt2 = wa2-wp2;      % upper transisiton width
bt = min(bt1,bt2); %critical transition width
wc1 = wp1-bt/2;     % lower cutoff frequency
wc2 = wp2+bt/2;     % upper cutoff frequency
T = 2*pi/ws;        % sampling period
```

Kaiser Window Parameters

```
deltaP = (10^(0.05*A_p) - 1)/ (10^(0.05*A_p) + 1); % calculating delta
deltaA = 10^(-0.05*A_a);
delta = min(deltaP,deltaA);

Aa = -20*log10(delta);  % Actual stopband attenuation

if Aa<=21               % Calculating alpha
    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

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

```
if mod(ceil((ws*D)/bt+1),2) == 0 %Choosing the Lowest Odd Value of N
 N = ceil((ws*D)/bt+1) + 1;
else
 N = ceil((ws*D)/bt+1);
end

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

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

Generating I(alpha)and I(beta)

```
Ibeta = 0; Ialpha = 0;
bessellimit = 50;
for k = 1 : 1 : bessellimit
Ibeta = Ibeta + ((1/factorial(k))*(beta/2).^k).^2;
Ialpha = Ialpha + ((1/factorial(k))*(alpha/2)^k)^2;
end
Ibeta = Ibeta + ones(1,numel(Ibeta));
Ialpha = Ialpha + ones(1,numel(Ialpha));
```

Obtaining Kaiser Window

```
wknt = Ibeta/Ialpha;

figure
stem(n,wknt)
xlabel('n')
ylabel('Amplitude')
title('Kaiser Window - Time Domain');
grid on;
```

Generating Impulse Response

```
nleft = -(N-1)/2:-1;
hntleft = 1./(nleft*pi).*(sin(wc2*nleft*T)-sin(wc1*nleft*T));

nright = 1:(N-1)/2;
hntright = 1./(nright*pi).*(sin(wc2*nright*T)-sin(wc1*nright*T));

hnt0 = 2*(wc2-wc1)/ws;

hnt = [hntleft,hnt0,hntright];
```

Applying the window to the filter

```
filter = hnt.*wknt;
figure
stem(n,filter)
xlabel('n')
ylabel('Amplitude')
title(strcat(['Filter Response - Kaiser Window - Time Domain']));
grid on;
```

```
figure
[h,w] = freqz(filter);
w = w/T;
h = 20*log10(abs(h));
plot(w,h)
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
title(strcat(['Filter Response - Kaiser Window - Frequency Domain']));

figure;
n_new = 0:1:(N-1);
stem(n_new, filter);
title('Impulse Response of Causal Filter');
xlabel('n'); ylabel('Amplitude');
grid on;

%Magnitude Response of the Filter
fvtool(filter) %Magnitude response
```

Plotting the Passband

```
figure
start = round(length(w)/(ws/2)*wc1);
finish = round((length(w)/(ws/2)*wc2));
wpass = w(start:finish);
hpass = (h(start:finish));
plot(wpass,hpass)
axis([-inf, inf, -0.1, 0.1]);
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
title('Passband - Frequency Domain');
```

Input, Output and Expected Signal of the Designed Filter

```
%Input Signal Generating
samples = 400;
len_fft = 2^nextpow2(samples);

w1 = wa1/2;                    %% component frequencies of the input
w2 = (wp2 + wp1)/2;
w3 = (wa2 + ws/2)/2;
wi = [w1,w2,w3];

nT = (0:T:samples*T);

xnT = sin(w1*nT)+sin(w2*nT)+sin(w3*nT);
Xw = fft(xnT, len_fft);
X1 = T*abs(Xw(1:len_fft/2+1));

%Output Signal of the Designed Filter
hnT = fft(filter,len_fft);
Yw = hnT.*Xw;
yt = ifft(Yw, len_fft);
Y = T*abs(Yw(1:len_fft/2+1));
```

```
%Expected Output Signal from the Ideal Filter
exp_ynT = sin(w2*nT);
exp_Yw = fft(exp_ynT, len_fft);
Y1 = T*abs(exp_Yw(1:len_fft/2+1));
```

Using DFT to check the filtering

```
% Frequency domain representation of signals

w = ws*(0:1/len_fft:1/2);
figure, subplot(3, 1, 1) ;
plot(w,X1);
title('Frequency Spectrum Of Input Signal');
xlabel('w (rad/s)'); ylabel('X(w)');
axis([0, 1500, 0, 0.5]);

subplot(3, 1, 2);
plot(w, abs(Y'));
title('Frequency Spectrum Of Output Signal using Designed Filter');
xlabel('w (rad/s)'); ylabel('Y(w)');
axis([0, 1500, 0, 0.5]);

subplot(3, 1, 3);
plot(w, abs(Y1'));
title('Frequency Spectrum Of Expected Output Signal');
xlabel('w (rad/s)'); ylabel('Y_exp(w)');
axis([0, 1500, 0, 0.5]);


% Time domain representation of signals

figure,
subplot(3, 1, 1);
stem(1:(samples+1),xnT(1:(samples+1)));
title('Input Signal');
xlabel('t(s)'); ylabel('x(nT)');
axis([0, (samples+1), -4, 4]);

subplot(3, 1, 2);
stem(1:(samples+1),yt(1:(samples+1)));
title('Output Signal of Designed Filter');
xlabel('t(s)'); ylabel('y(nT)');
axis([0,(samples+1), -1.5, 1.5]);

subplot(3, 1, 3);
stem(1:(samples+1),exp_ynT(1:(samples+1)));
title('Expected Output Signal');
xlabel('t (s)'); ylabel('y_e(nT)');
axis([0, (samples+1), -1.5, 1.5]);
```

Published with MATLAB® R2019b