Department of Electronic & Telecommunication Engineering

University of Moratuwa

EN2510 – Digital Signal Processing

REPORT

on

FIR BANDSTOP FILTER DESIGN

Name: H. M. J. Abeysekera

Index No.: 150009H

Date of Sub.: November 22, 2017

This report is submitted for the fulfilment of the FIR filter design project of the module EN2510 – Digital Signal Processing.

CONTENTS

1 ABSTRACT

This report investigates and describes the design procedure of a Finite Impulse Response (FIR) Bandstop filter using the Kaiser Window function. Basic theory of a non-recursive filter is discussed along with the results obtained. MATLAB 2013a software package is used as the programming environment to implement the filter. Final filter has designed to the specifications given in the project description.

2 INTRODUCTION

There are two classical methods that can used for the design of non-recursive filters.

- **Window method/ Fourier Series method**

  This is done by using the Fourier series in conjunction with a class of functions known as *window functions*.

- **Weighted-Chebyshev method**

  This is a multivariable optimization method.

3 BASIC THEORY

It's a fundamental theory is that all digital filters have a periodic frequency response with period equal to the sampling frequency (ω_s).

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

Therefore, we can consider a Fourier series as,

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

where

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

If we take $e^{-j\omega T} = z$, we get the transfer function of non-recursive filter with impulse response $h(nT)$ as follows.

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

We can see that the Fourier series coefficients are defined over the region of $-\infty < n < \infty$. We will encounter two problems with this Fourier series method.

- The filter that we obtain is in infinite length.
- The filter is non-causal since there exist non-zero values of impulse response for negative time.

To overcome from these problems, we define a truncated impulse response of finite length.

Truncated Impulse Response

We define the truncated impulse response as follows.

$$h_t(nT) = \begin{cases} h(nT), & |n| \leq M \\ 0, & |n| > M \end{cases}$$

where $M = \frac{N-1}{2}$.

Here, we have limited the infinite length filter to a length of N $(= 2M + 1)$.

A causal filter can be obtained by delaying the impulse response by MT seconds or by M sampling periods.

By z transform, delaying the impulse response by M sampling periods results to multiplying the transfer function by z^{-M}. So that, the transfer function of the causal filter assumes the form,

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

We can obtain the frequency response of the transfer function by the aforementioned substitution, $z = e^{-j\omega T}$.

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

Since $|e^{-j\omega MT}| = 1$, delaying the impulse response by M sampling periods does not change the amplitude response of the filter.

However, when we truncate the filter, we can observe that the amplitude response of the filter consists oscillations in both passband and stopband. These oscillations are called *Gibbs' oscillations*. Gibbs' oscillations will reduce as the length of the filter increases, but unfortunately it is not able to reduce the passband and stopband errors below a certain limit by increasing the filter length.

As a solution for this problem, we use a *window function* to truncate the infinite duration filter response $h(nT)$ as follows.

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

A modified transfer function can be obtained using the complex-convolution theorem.

$$H_w(z) = Z\{w(nT)h(nT)\}$$

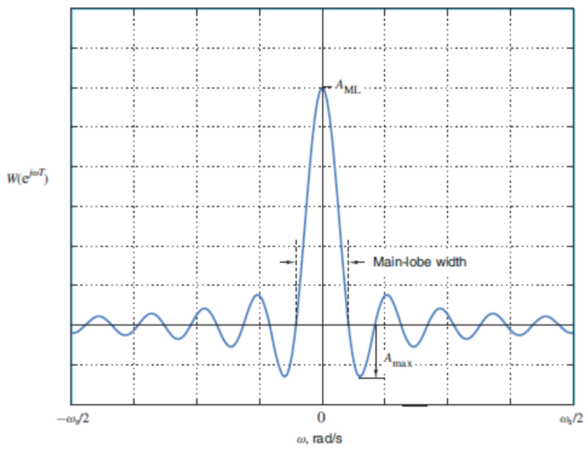

Figure 3.1: Frequency spectrum of a typical window

Windows are characterized by two factors;

Main-lobe width (B_{ML}): Bandwidth between the first positive and first negative zero crossings.

Ripple ratio (r *or* R): $r = \frac{A_{max}}{A_{ML}} \times 100\%$ or $R = 20 \log\left(\frac{A_{max}}{A_{ML}}\right)$ dB

where A_{max} and A_{ML} are the maximum side-lobe and main-lobe amplitudes, respectively.

The steepness of the transition characteristic of the filter depends on the main-lobe width of the window while the amplitudes of the passband and stopband ripples depend on the ripple ratio of the window. Therefore, the main-lobe width and ripple ratio should be as low as possible which means the spectral energy of the window should be concentrated as far as possible in the main-lobe and the energy in the side-lobes should be as low as possible.

There are lot of types of window functions that can be used for the filters.

Fixed windows	Adjustable windows
- Rectangular	- Dolph-Chebyshev
- von-Hann	- Kaiser
- Hamming	- Ultraspherical
- Blackman	

In this project, the filter has been designed using the Kaiser window function.

4 KAISER WINDOW

The Kaiser window function is given by,

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

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

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

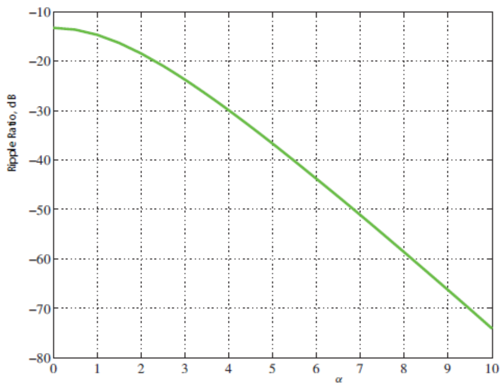

Figure 4.1: Ripple ratio vs α

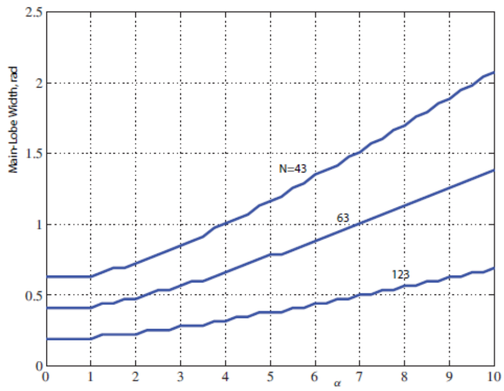

Figure 4.2: Main-lobe width vs α

Kaiser's method can be used to design lowpass (LP), highpass (HP), bandpass (BP) and bandstop (BS) filters.

5 DESIGN OF NON-RECURSIVE BANDSTOP FILTER

Following specifications should be considered in order to design a bandstop filter.

- Passband ripple ≤ $\tilde{A}_p$
- Minimum stopband attenuation ≥ $\tilde{A}_a$
- Lower passband edge ω_{p1}
- Lower stopband edge ω_{a1}
- Upper stopband edge ω_{a2}
- Lower passband edge ω_{p2}
- Sampling frequency ω_s

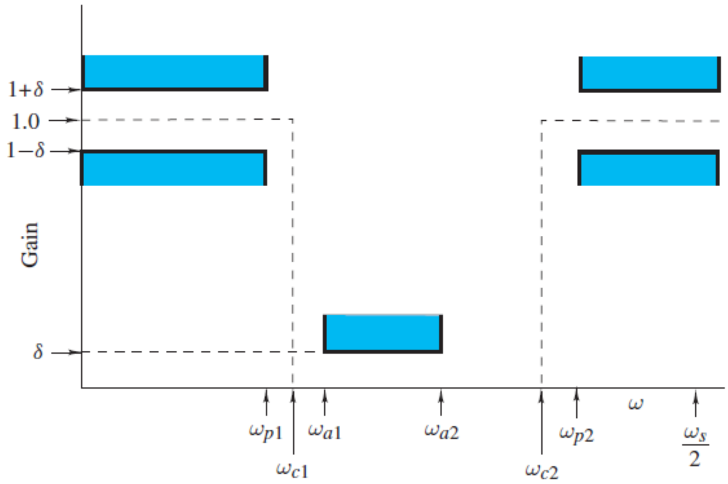

Figure 5.1: BS filter specifications

The design procedure of the BS filter using the Kaiser window function can be stepdown as follows.

➢ Determine the impulse response $h(nT)$ using the Fourier series assuming an idealized frequency response.

$$H(e^{j\omega T}) = \begin{cases} 1 \ \ for\ 0 \le |\omega| \le \omega_{c1} \\ 0 \ \ for\ \omega_{c1} < |\omega| < \omega_{c2} \\ 1\ for\ \omega_{c2} < |\omega| < \dfrac{\omega_s}{2} \end{cases}$$

where
transition width, $B_t = \min\{(\omega_{a1} - \omega_{p1}), (\omega_{p2} - \omega_{a2})\}$ and

cut-off frequencies, $\omega_{c1} = \omega_{p1} + \frac{B_t}{2}$, $\omega_{c2} = \omega_{p2} - \frac{B_t}{2}$.

Using the Fourier series, the impulse response of the ideal bandstop filter can be obtained as,

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

- Choose δ s.t. the actual stopband ripple, A_p, is equal to or less than the specified passband ripple, $\tilde{A}_p$, and the actual minimum stopband attenuation, A_a, is equal to or greater than the specified minimum stopband attenuation, $\tilde{A}_a$.
  A suitable value is

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

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

- With the required δ defined, the actual stopband attenuation A_a can be calculated as

$$A_a = -20log\delta$$

- Choose parameter α as

$$\alpha = \begin{cases} 0 & for A_a \leq 21 \\ 0.5842(A_a - 21)^{0.4} + 07886(A_{a-21}) & for 21 < A_a \leq 50 \\ 0.1102(A + a - 8.7) & for A_a > 50 \end{cases}$$

- Choose parameter D as

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

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

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

- Form $w_k(nT)$ using the following equations:

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

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

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

- Form the modified transfer function of the filter

  $H'_w(z) = z^{-(N-1)/2}H_w(z)$ where $H_w(z) = Z\{w_k(nT)h(nT)\}$.

6 RESULTS

6.1 Filter Specifications

The specifications of the BS filter depend on the final three digits of the index number.

Index No.: 150009H → $A = 0,\ B = 0,\ C = 9$

- Maximum passband ripple ($\tilde{A}_p$) = 0.05dB
- Minimum stopband attenuation ($\tilde{A}_a$) = 40dB
- Lower passband edge (ω_{p1}) = 1200 rad/s
- Lower stopband edge (ω_{a1}) = 1300 rad/s
- Upper stopband edge (ω_{a2}) = 1600 rad/s
- Upper passband edge (ω_{p2}) = 1750 rad/s
- Sampling frequency (ω_s) = 4200 rad/s

- $\delta = \min\{0.0029, 0.01\} = 0.0029$
- $B_t = 100,\ \omega_{c1} = 1250\ rad/s, \omega_{c2} = 1700\ rad/s,$
- $\alpha = 4.6413$
- $D = 2.9852$
- $N = 127$

6.2 Graphs

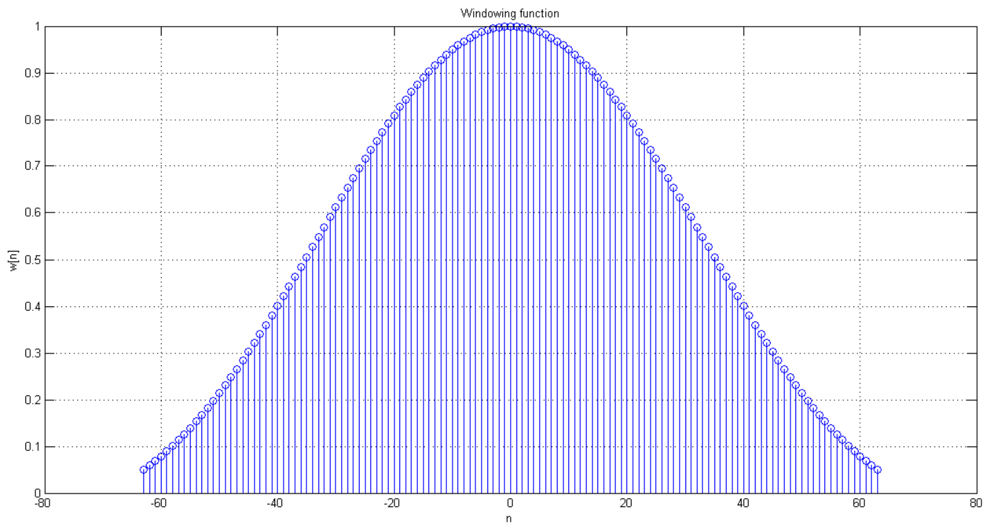

Figure 6.1: Kaiser Window

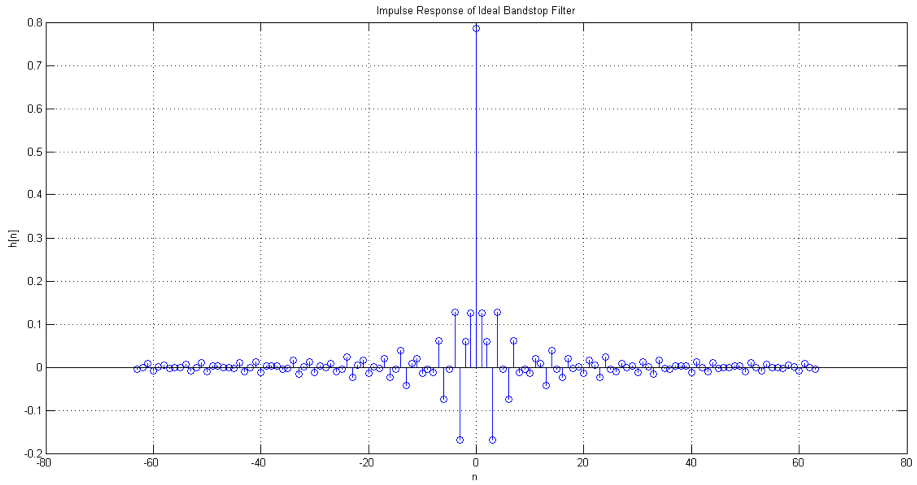

Figure 6.2: Impulse response of Ideal BS filter

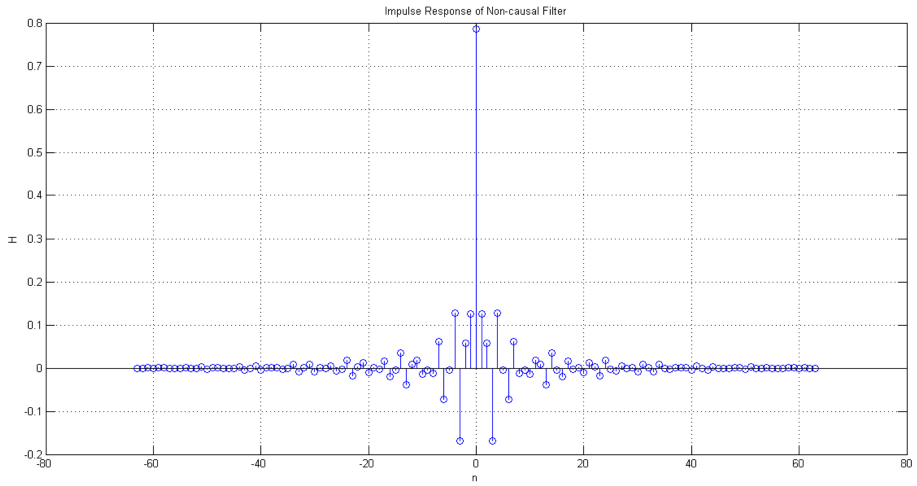

Figure 6.3: Impulse response of Non-causal BS filter

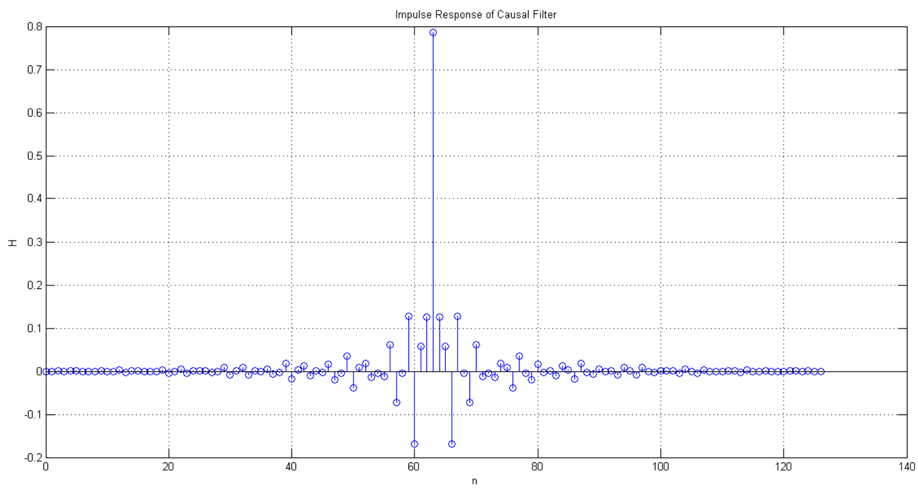

Figure 6.4: Impulse response of Causal BS filter

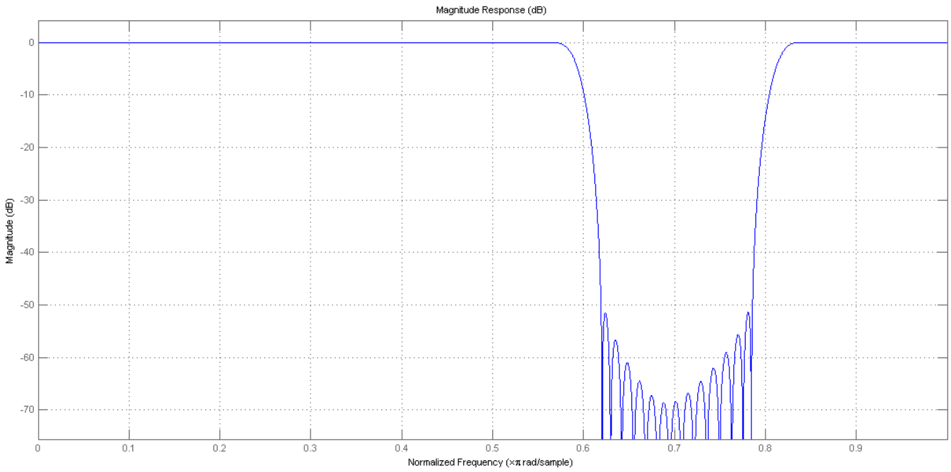

Figure 6.5: Magnitude response

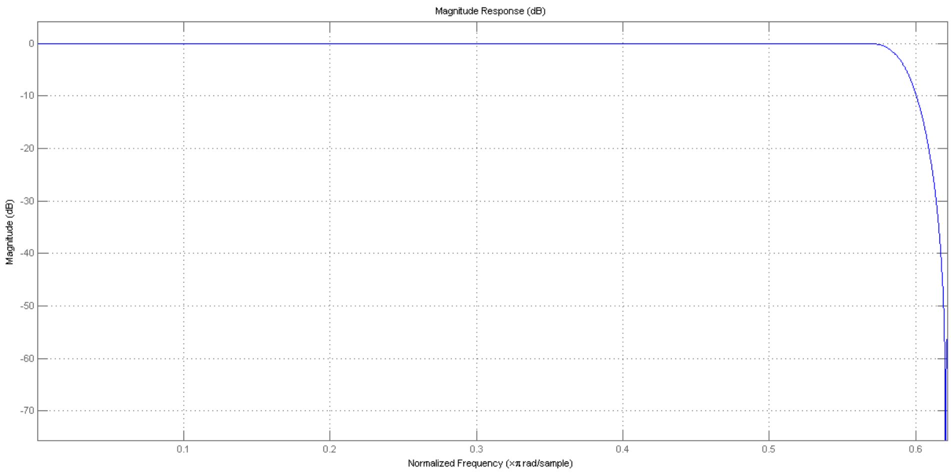

Figure 6.6: Lower passband

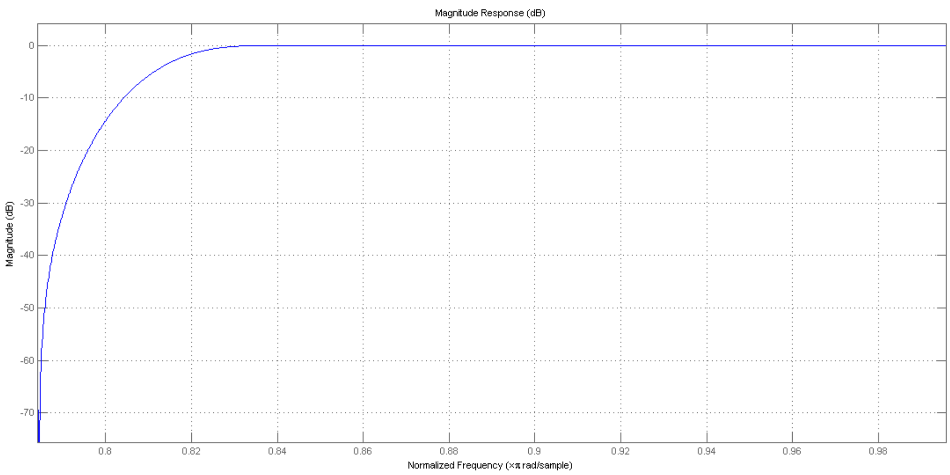

Figure 6.6: Upper passband

6.3 Response of the Filter to an Excitation

Time domain response of the digital filter to an excitation,

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

where

- Middle frequency of the lower passband, Ω_1 = 600 rad/s
- Middle frequency of the stopband, Ω_2 = 1450 rad/s
- Middle frequency of the upper passband, Ω_3 = 1925 rad/s

300 samples were used to achieve a steady state response.

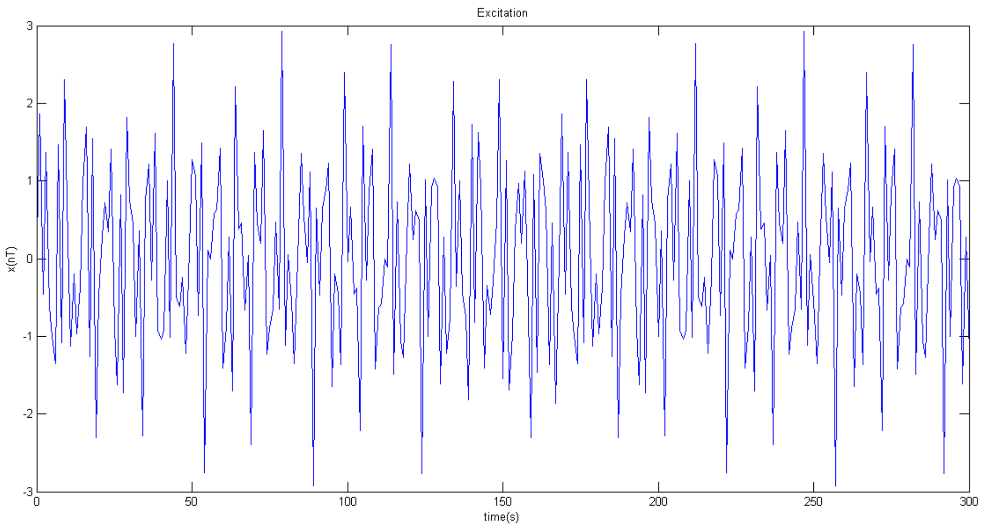

Figure 6.7: Excitation, $x(nT)$

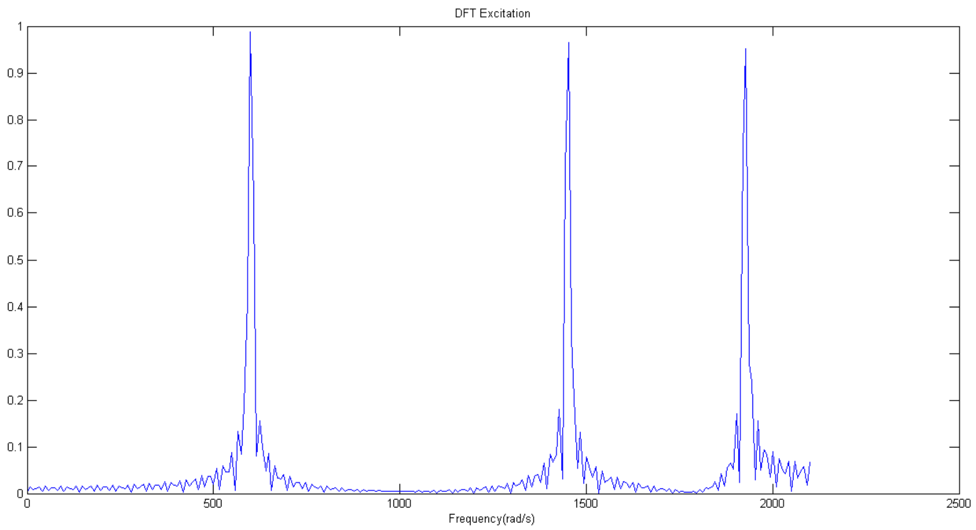

Figure 6.8: DFT Excitation

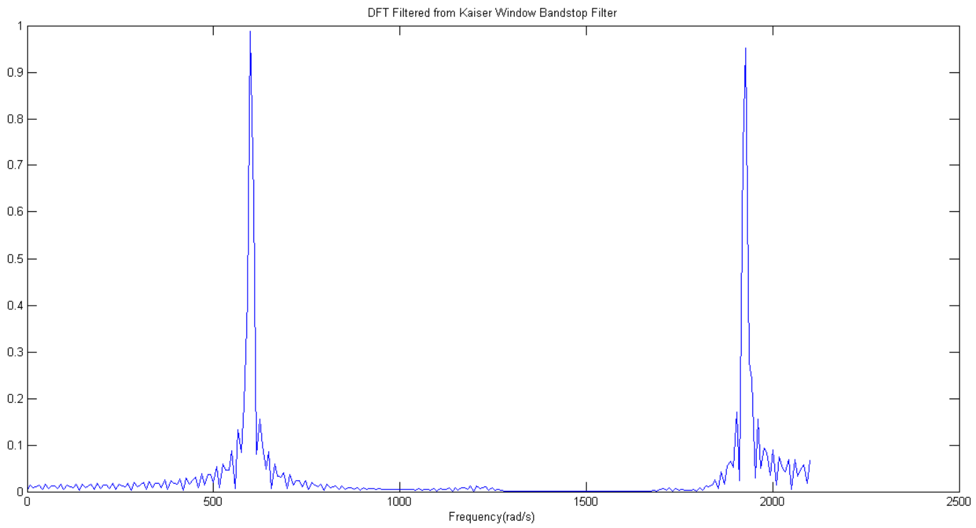

Figure 6.9: DFT filtered from Kaiser window BS filter

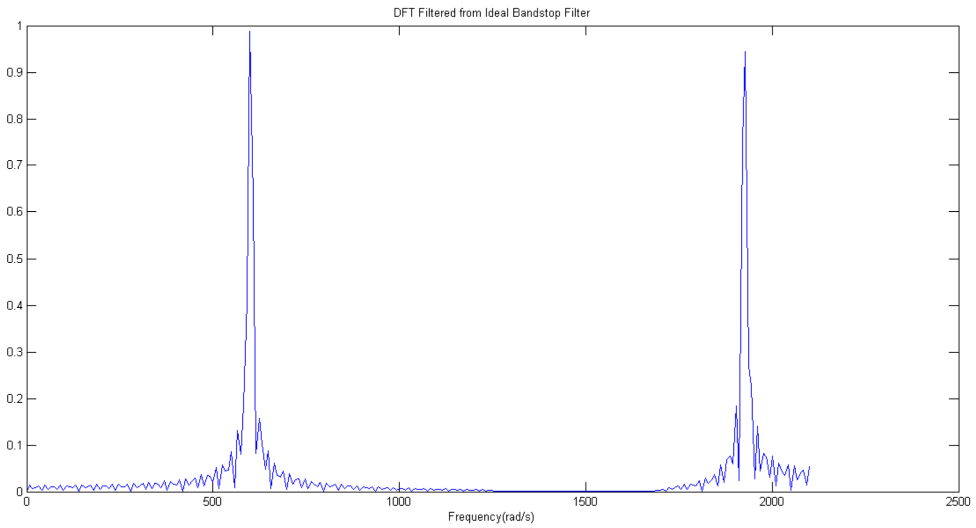

Figure 6.10: DFT filtered from Ideal BS filter

Note: MATLAB 2013a software package does not have the *designfilt()* function that can be used to get the BS filter directly.

7 CONCLUSIONS

A FIR bandstop filter was successfully designed using the Kaiser window method for the aforementioned specifications.

Expected filter response was obtained for the excitation, $x(nT)$.

8 REFERENCES

- “Design of Non-Recursive (FIR) filters” – Andreas Antoniou
- “Discrete-time Signal Processing” – Alan V. Oppenheim
- https://en.wikipedia.org/wiki/Digital_filter
- https://en.wikipedia.org/wiki/Kaiser_window
- https://www.mathworks.com/help/matlab/math/fourier-transforms.html

APPENDIX

MATLAB Code

```
%FIR Bandstop Filter

close all;
clear all;

indexNo = 150009;
%Get A, B, C values from the index number
A = fix(mod(indexNo,1000)/100);
B = fix(mod(indexNo,100)/10);
C = mod(indexNo,10);

%Define filter specifications
Ap = 0.05 + 0.01*A;      %Passband ripple
Aa = 40 + B;             %Minimum stopband attenuation
Omega_p1 = C*100 + 300; %Lower passband frequency
Omega_p2 = C*100 + 850; %Upper passband frequency
Omega_a1 = C*100 + 400; %Lower stopband frequency
Omega_a2 = C*100 + 700; %Upper stopband frequency
Omega_s = 2*(C*100 + 1200); %Sampling frequency
T = 2*pi/Omega_s;

Bt = min((Omega_a1 - Omega_p1),(Omega_p2 - Omega_a2));  %Transition width
Omega_c1 = Omega_p1 + Bt*0.5;    %Lower cut-off frequency
Omega_c2 = Omega_p2 - Bt*0.5;    %Upper cut-off frequency

%Choose delta value
delta_p = (10^(0.05*Ap)-1)/(10^(0.05*Ap)+1);
delta_a = 10^(-0.05*Aa);
delta = min(delta_p, delta_a);

%Actual Stopband attenuation
Aaa = -20*log10(delta);

%Choose parameter alpha
if Aaa<=21
    alpha = 0;
elseif (21<=Aaa) && (Aaa<=50)
    alpha = 0.5842*(Aaa-21)^0.4 + 0.07886*(Aaa-21);
elseif Aaa>50
    alpha = 0.1102*(Aaa-8.7);
end

%Choose parameter D
if Aaa<=21
    D = 0.9222;
else
    D = (Aaa-7.95)/14.36;
end

%Choose the lowest odd value of N
if mod(ceil((Omega_s*D)/Bt+1),2) == 0;
    N = ceil((Omega_s*D)/Bt+1) + 1;
else
```

```
    N = ceil((Omega_s*D)/Bt+1);
end

%Plot the Window function (wk) from Kaiser window function
nr = -(N-1)/2 : 1 : (N-1)/2;      %define the range where wk is non-zero.
beta = alpha*(1 - ((2*nr)/(N-1)).^2).^0.5;
I_beta = 1; I_alpha = 1;
for k = 1 : 1 : 100
    I_beta = I_beta + ((1/factorial(k))*(beta/2).^k).^2;
    I_alpha = I_alpha + ((1/factorial(k))*(alpha/2).^k).^2;
end
wk = I_beta./I_alpha;
figure;
stem(nr,wk);
title('Windowing function');
xlabel('n'); ylabel('w[n]');
grid on;

%Compute h[n]
n1 = -(N-1)/2 : 1 : -1; %Range for negative values
n2 = 1 : 1 : (N-1)/2;   %Range for positive values
h1 = (((1/pi)./n1).*(sin(Omega_c1*n1*T) - sin(Omega_c2*n1*T)));
h2 = (((1/pi)./n2).*(sin(Omega_c1*n2*T) - sin(Omega_c2*n2*T)));
h0 = 1 + (2*(Omega_c1-Omega_c2))/Omega_s;
n = [n1, 0, n2];
hn = [h1, h0, h2];      %h[n] array
figure;
stem(n,hn);
title('Impulse Response of Ideal Bandstop Filter');
xlabel('n'); ylabel('h[n]');
grid on;

%Compute Digital filter
filt = hn.*wk;

figure;
stem(n, filt);
title('Impulse Response of Non-causal Filter');
xlabel('n'); ylabel('H');
grid on;

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

%Magnitude Response
fvtool(filt)

%compute the Omega values and plot the Excitation in time domain
w1 = Omega_p1/2;
w2 = (Omega_a1+Omega_a2)/2;
w3 = (Omega_p2+Omega_s/2)/2;

ns = 0:1:300;   %No. of samples
xnT = sin(w1*ns*T)+sin(w2*ns*T)+sin(w3*ns*T); %Excitation function
figure;
plot(ns, xnT);
```

```
title('Excitation');
xlabel('time(s)');
ylabel('x(nT)');

y = conv2(xnT,filt);
figure;
plot([1:length(y)]*T*(length(xnT))/(length(y)),y);
title('Filtered Signal');
xlabel('time(s)');

N_FFT = 2^nextpow2(numel(ns));    %Next power of 2 from length of y
xnT_FFT = fft (xnT, N_FFT)/numel(ns);
f = (Omega_s)/2* linspace(0, 1, N_FFT/2+1);
figure;
plot(f, 2*abs(xnT_FFT(1:N_FFT/2+1)));
title('Discrete Fourier Transform Excitation');
xlabel('Frequency(rad/s)');

N_FFT = 2^nextpow2(numel(ns));    %Next power of 2 from length of y
Y_FFT = fft(y, N_FFT)/numel(ns);
f = (Omega_s)/2*linspace(0, 1, N_FFT/2+1);
figure;
plot(f, 2*abs(Y_FFT(1:N_FFT/2+1)));
title('Discrete Fourier Transform Filtered from Kaiser Window Bandstop
Filter');
xlabel('Frequency(rad/s)');

%Ideal Filter
xnT=sin(w1*ns*T)+sin(w3*ns*T);
Y=conv2(xnT,filt);
N_FFT = 2^nextpow2(numel(ns));    %Next power of 2 from length of y
Y_FFT = fft(Y, N_FFT)/numel(ns);
f = (Omega_s)/2*linspace(0, 1, N_FFT/2+1);
figure;
plot(f, 2*abs(Y_FFT(1:N_FFT/2+1)));
title('DFT Filtered from Ideal Bandstop Filter');
xlabel('Frequency(rad/s)');

%%%%%%%%%%%%end%%%%%%%%%%%%
```