DEPARTMENT OF ELECTRONIC AND TELECOMMUNICATION ENGINEERING UNIVERSITY OF MORATUWA

Ganeshaaraj Gnanavel

170180N

EN2570-DIGITAL SIGNAL PROCESSING

DESIGN OF FINITE IMPULSE RESPONSE BAND STOP FILTER

This is submitted as a partial fulfillment for the module
EN 2570-Digital signal processing

—

December 21, 2019

Abstract

This is a detailed report discussing the complete procedure used for designing a Finite-Duration Impulse Response (FIR) bandstop filter for a set of prescribed specifications and its implementation on MatLAB 2017a. This digital filter is designed for the prescribed specifications using the Fourier Series method. The truncations of the impulse response are achieved using the Kaiser Window function. The report analyses the magnitude and impulse responses of the filter to confirm its characteristics with initial parameter. Further, the filter is evaluated for its performance by analysing the output signal obtained for a given input consisting of a combination of sinusoids

CONTENTS

1. Introduction

This projects explains how I created a FIR band stop filter using Kaiser window method. Steps are structured in a sequential manner for better understanding. I used Matlab 2017a to do all the designing part. Further, the filter is evaluated for its performance by analysing the output signal obtained for a given input consisting of a combination of sinusoids. I did this evaluation to the designed filter and ideal filter,then compared both the results.

2. Method

Question 1

I. Required Specifications

SPECIFICATION	VALUE	UNITS
Maximum Pass band Ripple	**0.0400**	dB
Maximum Stop band attenuation	**53**	dB
Lower pass band edge	**400**	rads^{-1}
Upper pass band edge	**950**	rads^{-1}
Lower stop band edge	**500**	rads^{-1}
Upper stop band edge	**800**	rads^{-1}
Sampling Frequency	**2600**	rads^{-1}

Table 1

Derived specifications

These specifications were derived from above mentioned required specifications

SPECIFICATION	VALUE	UNITS
Lower transition width	**100**	rads^{-1}
Upper transition width	**150**	rads^{-1}
Critical transition width	**100**	rads^{-1}
Lower cut-off frequency	**450**	rads^{-1}
Upper cut-off frequency	**900**	rads^{-1}
Sampling frequency	**0.0024**	s
Sampling Frequency	**2600**	rads^{-1}

Table 2

II. Derivation of Kaiser Window Parameters

The Kaiser Window function is given by

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

where α is an independent parameter and

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

By using the parameters in Table I and the derived parameters in Table II above, we calculate α and N as

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

where

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

Now, with the defined δ, we calculate the actual stop band loss

$$A_a = -20log|x|$$

and the actual pass band ripple

$$A_p = 20log\frac{|1+\delta|}{|1-\delta|}$$

We can chose α as

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

A parameter D is chosen in order to obtain N, as

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

N is chosen such that it is the smallest odd integer value satisfying the inequality

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

Parameter	Value	Units
δ	0.0022	-
A_a	53	dB
A_p	0.0389	dB
α	4.8899	-
$I_0(\alpha)$	24.51	-
D	3.1370	-
N	83	-

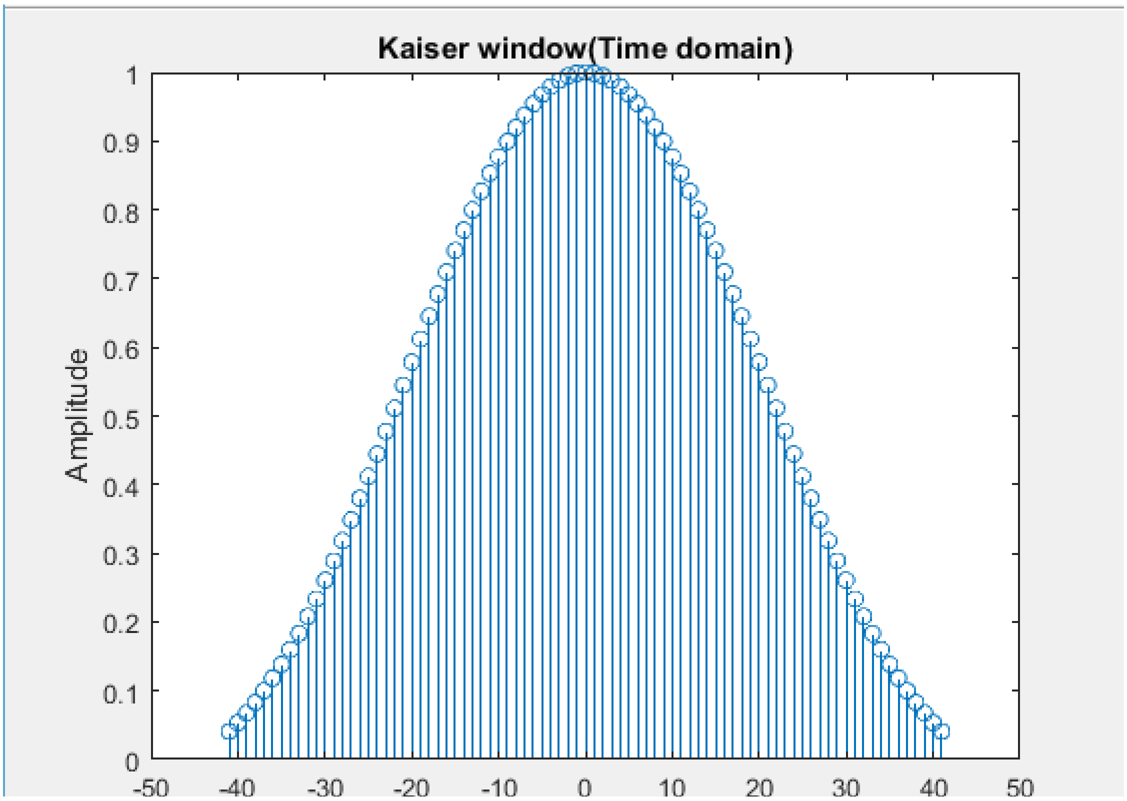


Figure 1-Kaiser Window

III. Derivation of Ideal Impulse Response

The frequency response of an ideal bandstop filter with cutoff frequencies Ω_{c1} and $\Omega c2$ is given by

$$H(e^{j\Omega T}) = \begin{cases} 1 & for\ 0 \leq |\Omega| \leq \Omega_{c1} \\ 0 & for\ \Omega_{c1} \leq |\Omega| \leq \Omega_{c2} \\ 1 & for\ \Omega_{c2} \leq |\Omega| \leq \frac{\Omega_s}{2} \end{cases}$$

Hence, using the inverse fourier transform,

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

$$h(nT) = \frac{1}{\Omega_s} \left[\int_{-\Omega_s/2}^{-\Omega_{c2}} e^{j\Omega nT} d\Omega + \int_{-\Omega_{c1}}^{0} e^{j\Omega nT} d\Omega + \int_{0}^{\Omega_{c1}} e^{j\Omega nT} d\Omega + \int_{\Omega_{c2}}^{\Omega_s/2} e^{j\Omega nT} d\Omega \right]$$

we obtain the ideal impulse response $h(nT)$ of the above filter as

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

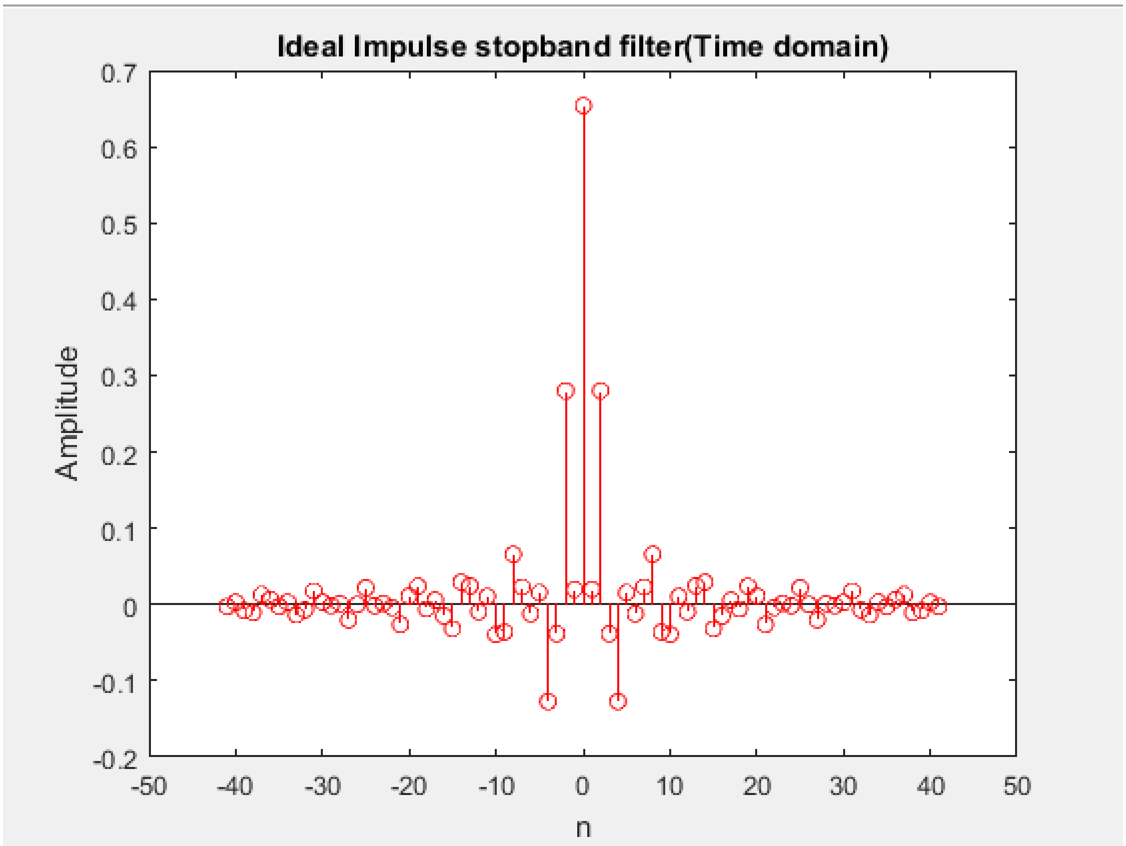


Figure 2: The ideal impulse stopband filter

IV. Derivation of the causal impulse response of windowed filter

QUESTION 2

I.V Derivation of the causal impulse response of windowed filter

We obtain the finite order non-causal impulse response of the windowed filter $h_w(nT)$ by the multiplication of the Ideal impulse response $h(nT)$ by the Kaiser Window function $w_K(nT)$,

$$h_w(nT) = w_K(nT)h(nT)$$

We obtain the $\mathscr{Z}$-tranform of $h_w(nT)$ as

$$H_w(z) = \mathscr{Z}[h_w(nT)] = \mathscr{Z}[w_K(nT)h(nT)]$$

which upon shifting for causality becomes

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

We also present the filter that is obtained from a rectangular windowing, for the purpose of comparing.

$$w_R(nT) = \begin{cases} 1 & for\ |x| \leq \frac{N-1}{2} \\ 0 & otherwise \end{cases}$$

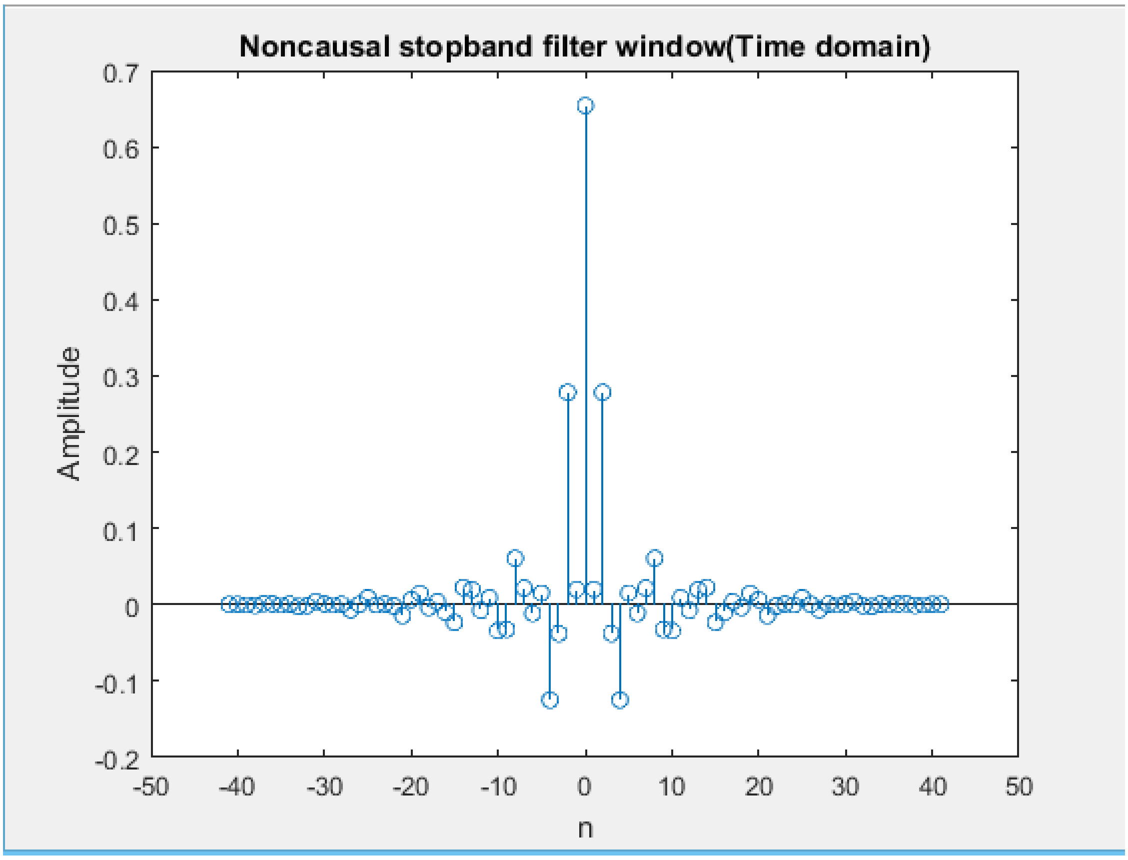


Figure 2: Time Domain representation of the non-causal bandstop filter designed using Kaiser Window

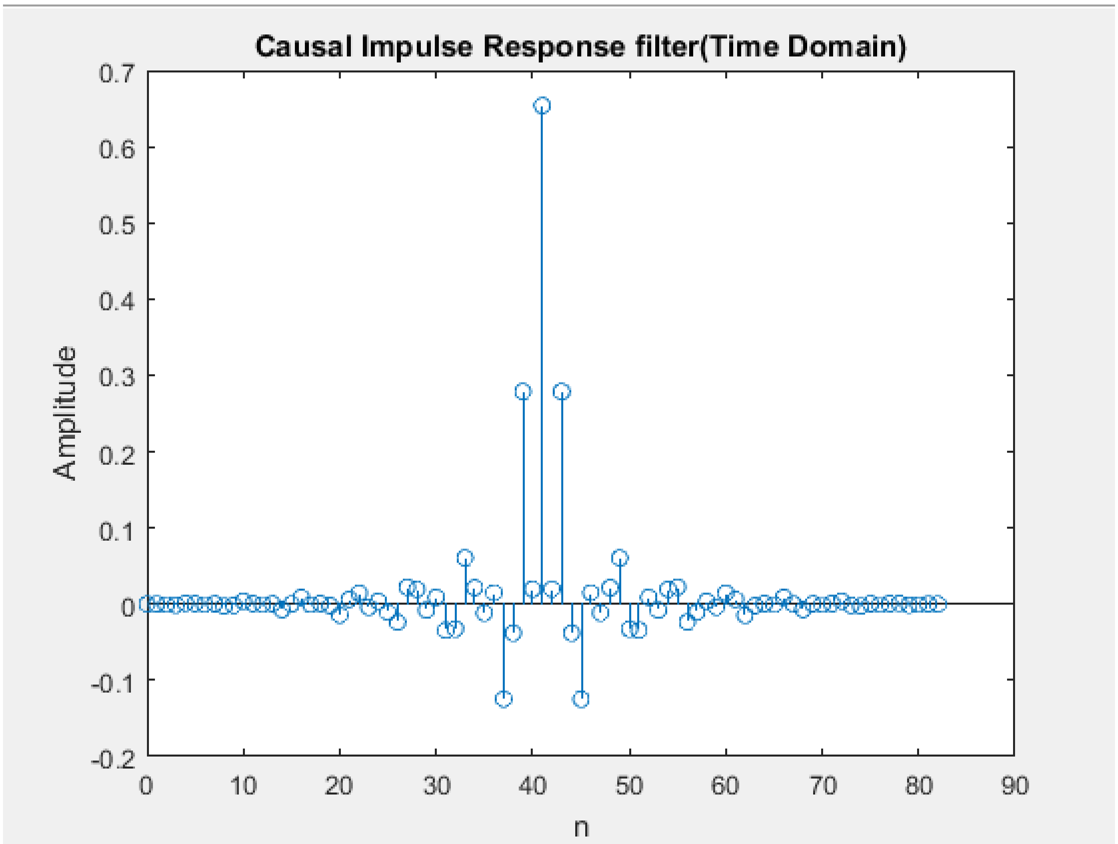


Figure 3: Time Domain representation of the causal bandstop filter designed using Kaiser Window

3. Magnitude Response Analysis

<u>QUESTION 3</u>

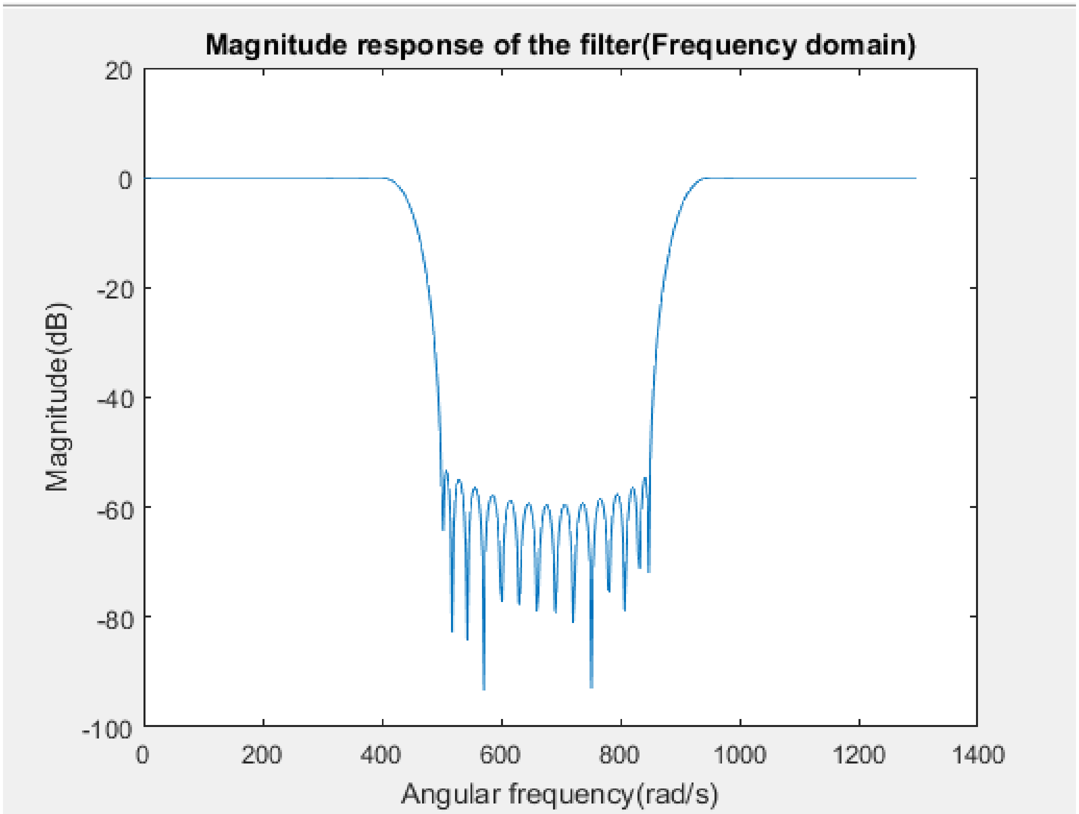


Figure 4: Frequency Domain representation of the bandstop filter designed using Kaiser Window

QUESTION 4

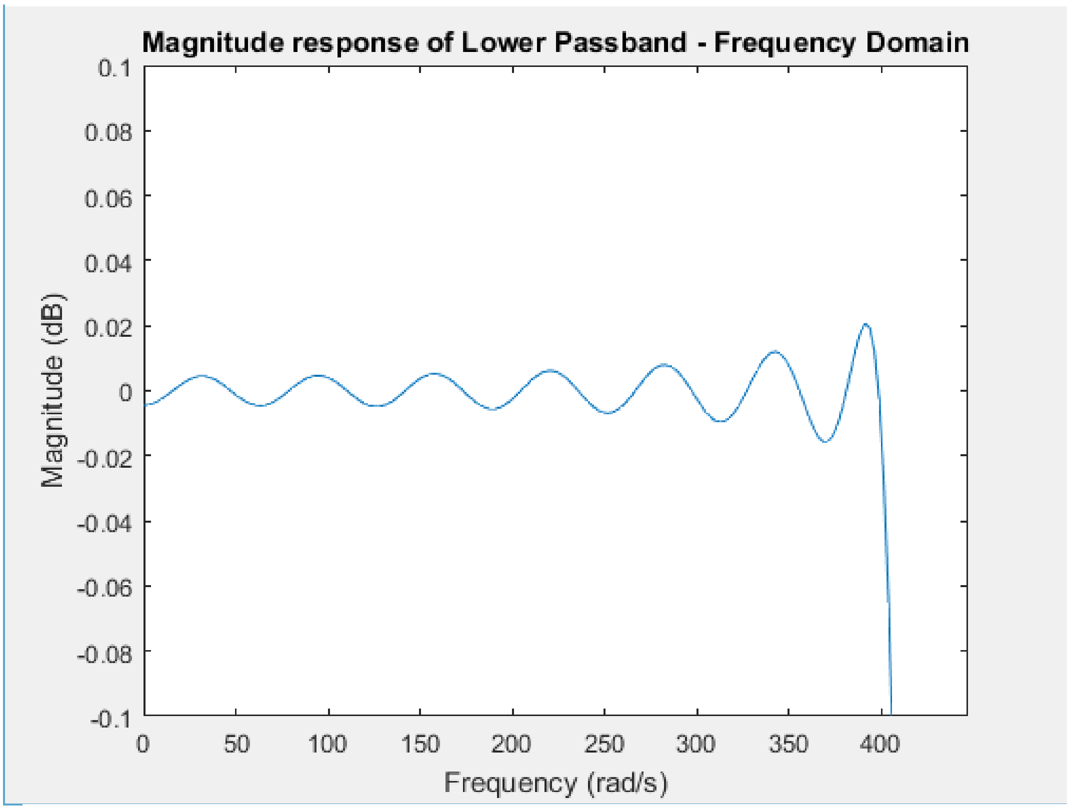


Figure 5: Lower Pass band of the bandstop filter designed using Kaiser Window

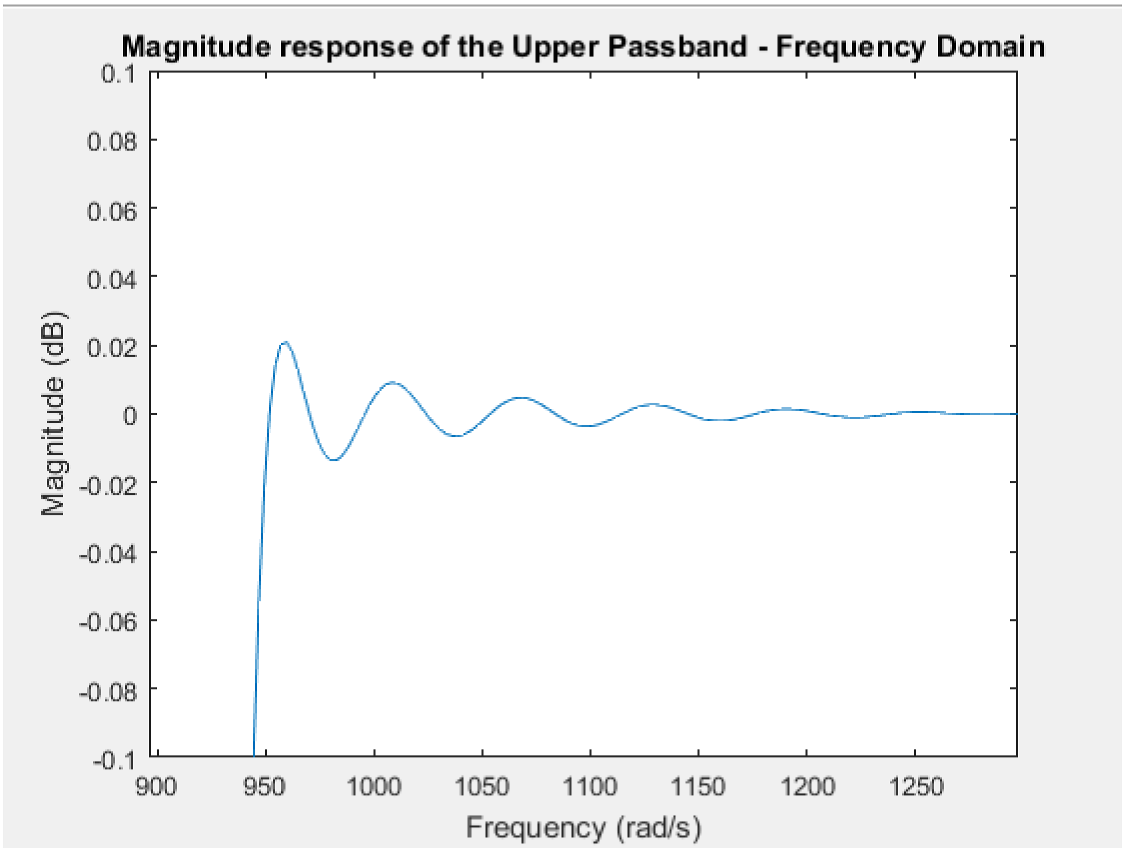


Figure 6: Upper Pass band of the bandstop filter designed using Kaiser Window

4. Testing and Evaluation

Designed filter was tested with an input signal

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

Signal consists 3 sinusoids. The frequencies are

$\Omega_1 = 225$ rads^{-1}

$\Omega_2 = 675$ rads^{-1}

$\Omega_3 = 1050$ rads^{-1}

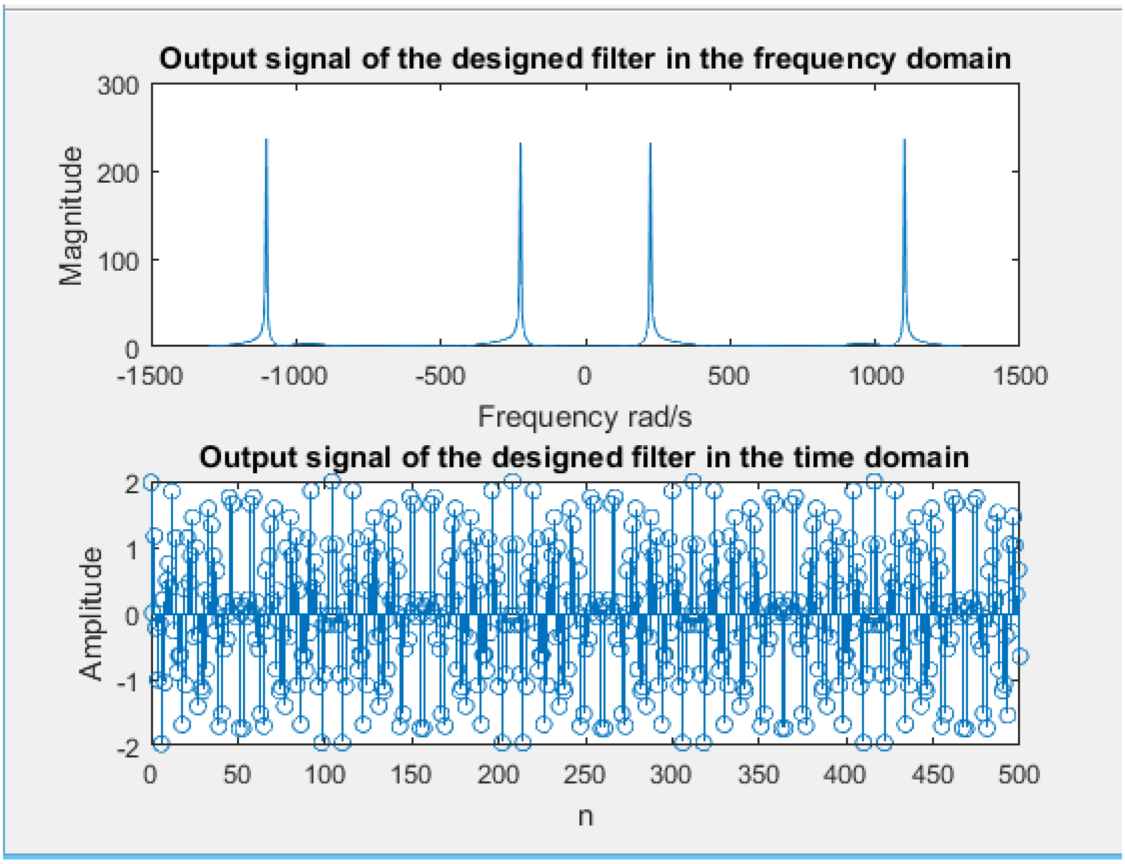
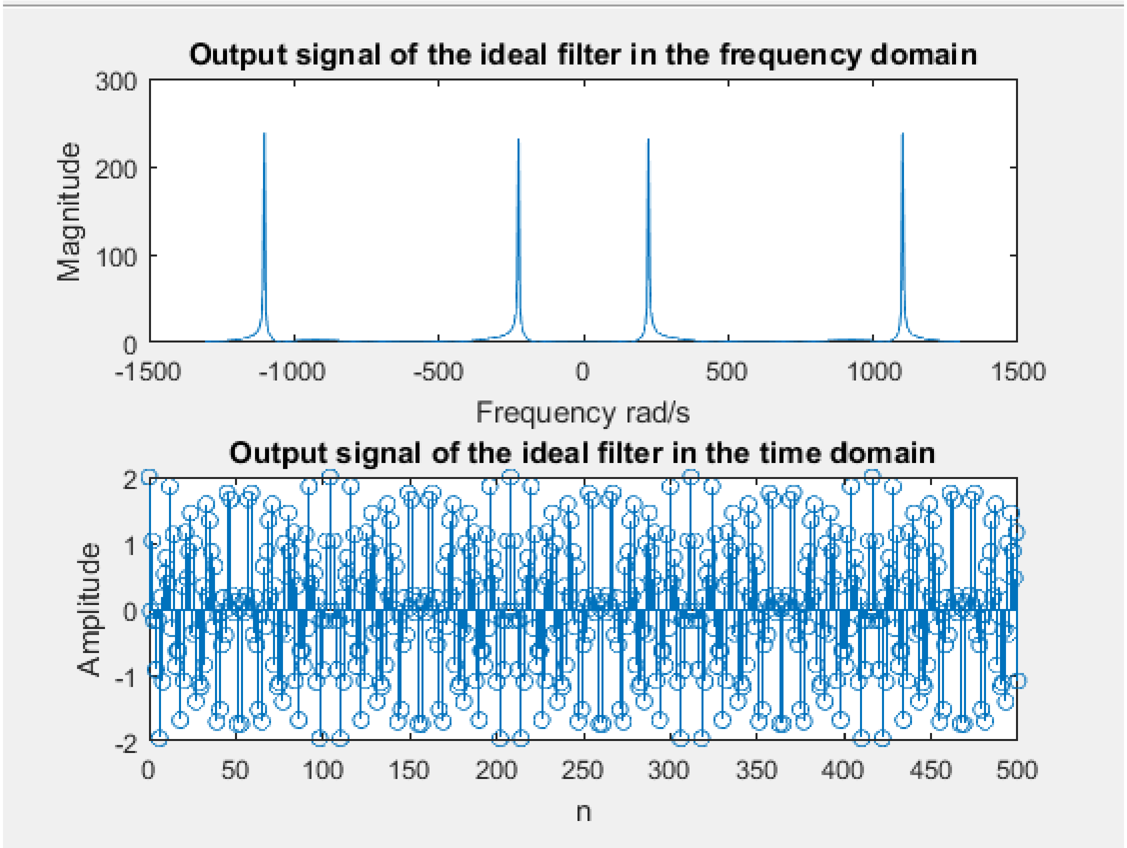


Figure 7: Frequency Domain and Time Domain representation of the output signal



FFigure 9: Frequency Domain and Time Domain representation of the output signal

<u>QUESTION 5</u>

Comparison

- Mean Squared error between designed filter and ideal filter output=0.0012
- Correlation coefficient between above mentioned two filter output

$$\begin{bmatrix} 1.0 & 0.9994 \\ 1.0 & 0.994 \end{bmatrix}$$

- Two outputs are highly correlated

QUESTION 6

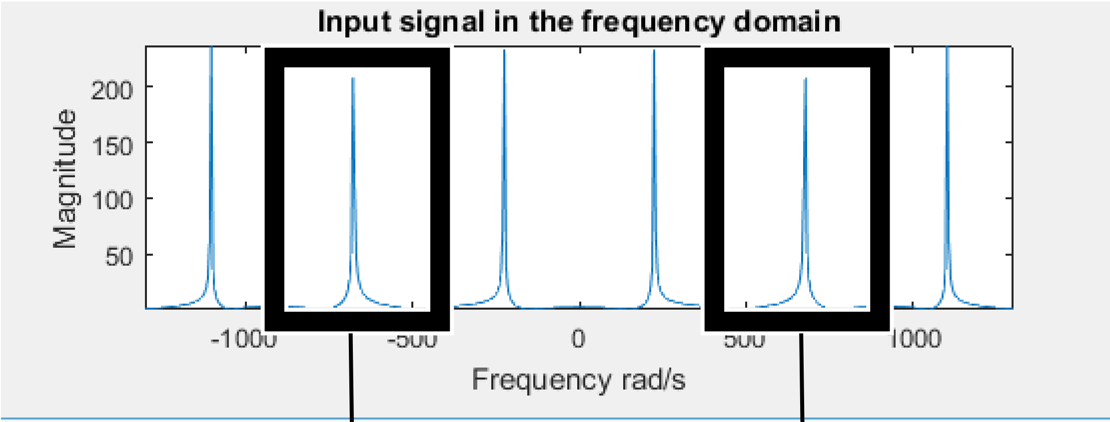


Figure 9: Frequency Domain and Time Domain representation of the input signal

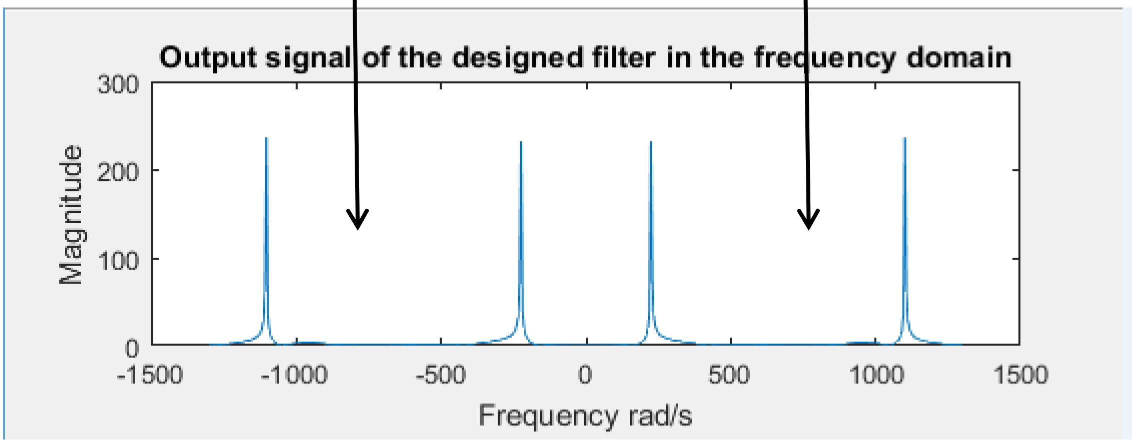


Figure 10: Frequency Domain and Time Domain representation of the output signal

- When comparing figure 9 and figure 10 we can observe that the amplitude waveform occurred during 600-900 Hz frequency range has been suppressed
  But other amplitude waveforms from different frequency ranges have passed through the filter.
- Suppressed waveform occurred in the stop band region and others were in the pass band region, so the designed filter has worked properly by passing only the pass band frequencies and suppressing the stop band frequency range amplitudes.

5. Discussion

By observing the results, it is seen that the intended objectives of the project have been achieved. When the results of the designed filter are compared with the ideal results, Figures 8 and 9 show that they and similar in both the time and frequency domains. Therefore, it is safe to assume that the unwanted frequency component of the input signal has been completely filtered out by using this filter.

6. Conclusion

The flexibility of the Kaiser Window is evident through the above results. 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 pass band 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

7. Acknowledgement

I would like to express my gratitude to Dr. Chameera U.S. Edussooriya, the project supervisor, for the constant guidance he provided throughout this project. I would also like to thank my colleagues for sharing their knowledge and experience on this project

8. Appendix (Matlab scripts)

- Generation of the required specification by the index number

```
I_A = 1; I_B = 8; I_C = 0;
%rad/s(below every parameter)
A_p = 0.03+(0.01*I_A);%max passband ripple
A_a = 45+I_B;%min stopband attenuation
lp = (I_C*100)+400;%lower passband edge
up = (I_C*100)+950;%upper passband edge
ls = (I_C*100)+500;%lower stopband edge
us = (I_C*100)+800;%upper stopband edge
sf = 2*((I_C*100)+1300);%sampling freqency
% A_p
% A_a
% O_p1
% O_p2
% O_a1
% O_a2
% O_s
```

- Obtaining the specifications of the filter

```
ltw = ls-lp; %lower transition width
utw = up-us; %upper transisiton width
ctw = min(ltw,utw); %critical transition width
low_cf = lp+ctw/2; %lower cutoff frequency
up_cf = up-ctw/2;%upper cutoff frequency
st = 2*pi/sf; %sampling period
% B_t1
% B_t2
% B_t
% O_c1
% O_c2
% T
```

- Obtaining the Kaiser Window

```
dp = ((10^(0.05*A_p))-1)/(1+(10^(0.05*A_p)));
da = 10^(-0.05*A_a);
d = min(dp,da);
I_A = -20*log10(d);
Ap = 20*log10((1+d)/(1-d));
Aa = -20*log10(d);
if I_A<=21
    alpha = 0;
elseif I_A>21 && I_A<=50
    alpha = 0.5842.*(I_A-21).^0.4+0.07886.*(I_A-21);
```

```
else
    alpha = 0.1102.*(I_A-8.7);
end
%Calculating D
if I_A<=21
    D = 0.9222;
else
    D = (I_A-7.95)/14.36;
end
%Finding the order of the filter
N = ceil((sf*D/ctw)+1);
%Order of the filter should be odd
if mod(N,2) == 0
    N = N+1;
else
    N =N;
end
n = -(N-1)/2:1:(N-1)/2;
beta = alpha*sqrt(1-(2*n/(N-1)).^2);
%Generating Io_alpha
b_limit = 125;
Io_alpha = 1;
for k = 1:b_limit
    value_k = ((1/factorial(k))*(alpha/2).^k).^2;
    Io_alpha = Io_alpha + value_k;
end
%Generating Io_beta
Io_beta = 1;
for m = 1:b_limit
    value_m = ((1/factorial(m))*(beta/2).^m).^2;
    Io_beta = Io_beta +value_m;
end
wk_nT = Io_beta/Io_alpha;
Io_alpha
Aa
Ap
d
D
N
alpha
figure;
stem(n,wk_nT);
xlabel('n');
ylabel('Amplitude');
title('Kaiser window(Time domain)');
```

- <u>Obtaining the ideal impulse stopband filter</u>

```
n_L = -(N-1)/2:1:-1;
hn_L = (1./(n_L*pi)).*(sin(low_cf*n_L*st)-sin(up_cf*n_L*st));
n_R = 1:1:(N-1)/2;
hn_R = (1./(n_R*pi)).*(sin(low_cf*n_R*st)-sin(up_cf*n_R*st));
hn_0 = 1+(2/sf).*(low_cf-up_cf);
n = [n_L,0,n_R];
h_nT = [hn_L,hn_0,hn_R];
hw_nT = h_nT.*wk_nT;
% %Plotting the ideal filter
figure;
stem(n,h_nT,'-r');
xlabel('n');
ylabel('Amplitude');
title('Ideal Impulse stopband filter(Time domain)');
```

- <u>Plotting causal stopband filter</u>

```
%Plotting the noncausal stopband filter
figure;
stem(n,hw_nT);
xlabel('n');
ylabel('Amplitude');
title('Noncausal stopband filter window(Time domain)');
%Plotting the causal stopband filter
shift = [0:1:N-1];
figure;
stem(shift,hw_nT);
xlabel('n');
ylabel('Amplitude');
title('Causal Impulse Response filter(Time Domain)');
```

- <u>Magnitude response analysis</u>

```
%Question 3
[Hw,f] = freqz(hw_nT);%obtaining the frequency response and
corresponding frequencies
af = f*sf/(2*pi);%Angular frequency
log_Hw = 20.*log10(abs(Hw));
figure;
plot(af,log_Hw);
xlabel('Angular frequency(rad/s)');
ylabel('Magnitude(dB)');
title('Magnitude response of the filter(Frequency domain)');
% fvtool(af,log_Hw)
%Question 4
%Plotting the magnitude response of the passbands
%considering the lower passband
figure;
```

```
fi = round((length(af)/(sf/2)*low_cf));
hp_l = log_Hw(1:fi);
wp_l = af(1:fi);
plot(wp_l,hp_l);
axis([-inf, inf, -0.1, 0.1]);
xlabel('Frequency (rad/s)');
ylabel('Magnitude (dB)');
title('Magnitude response of Lower Passband - Frequency Domain');
%Considering the upperpassband
figure;
begin = round(length(af)/(sf/2)*up_cf);
wp_h = af(begin:length(af));
hp_h = log_Hw(begin:length(af));
plot(wp_h,hp_h);
axis([-inf, inf, -0.1, 0.1]);
xlabel('Frequency (rad/s)');
ylabel('Magnitude (dB)');
title('Magnitude response of the Upper Passband - Frequency Domain');
```

- <u>Generating the input signal</u>

```
%Component frequencies of the input
f1 = low_cf/2;
f2 = low_cf + (up_cf-low_cf)/2;
f3 = up_cf + (sf/2-up_cf)/2;
samples=500;
%Generating the discrete signal
n1 = 0:1:samples;
X_ax = cos(f1.*n1.*st)+cos(f2.*n1.*st)+cos(f3.*n1.*st);
figure;
subplot(2,1,1);
stem(n1,X_ax);
xlabel('n');
ylabel('Amplitude');
title('Input signal(Time domain)')
subplot(2,1,2);
l_fft = 2^nextpow2(numel(n1))-1;
x_fft = fft(X_ax,l_fft);
x_fft_plot =
[abs([x_fft(l_fft/2+1:l_fft)]),abs(x_fft(1)),abs(x_fft(2:l_fft/2+1))];
f = sf*linspace(0,1,l_fft)-sf/2;
plot(f,x_fft_plot);
xlabel('Frequency rad/s');
ylabel('Magnitude');
title('Input signal in the frequency domain');
axis tight;
```

- <u>Generating the output signal from our designed filter & ideal filter</u>

```
%Question 6
% Filtering using frequency domain multiplication
l_fft = length(X_ax)+length(hw_nT)-1; % length for fft in x dimension
x_fft = fft(X_ax,l_fft);
hw_nT_fft = fft(hw_nT,l_fft);
out_fft = hw_nT_fft.*x_fft;
out = ifft(out_fft,l_fft);
design_out = out(floor(N/2)+1:length(out)-floor(N/2));
% Ideal Output Signal
ideal_out = cos(f1.*n1.*st)+cos(f3.*n1.*st);
%O_2 is left out because it is in the  stopband
%Obtaining the output waveforms
% Frequency domain representation of output signal after filtering
using
% the designed filter
figure;
subplot(2,1,1);
l_fft = 2^nextpow2(numel(n1))-1;
xfft_out = fft(design_out,l_fft);
x_fft_out_plot =
[abs([xfft_out(l_fft/2+1:l_fft)]),abs(xfft_out(1)),abs(xfft_out(2:l_ff
t/2+1))];
f = sf*linspace(0,1,l_fft)-sf/2;
plot(f,x_fft_out_plot);
xlabel('Frequency rad/s');
ylabel('Magnitude');
title('Output signal of the designed filter in the frequency domain');
% Time domain representation of output signal after filtering using
the
% designed filter
subplot(2,1,2);
stem(n1,design_out);
xlabel('n');
ylabel('Amplitude');
title('Output signal of the designed filter in the time domain');
%Obtaining the outputs of the ideal filter
figure;
subplot(2,1,1);
xfft_ideal_out = fft(ideal_out,l_fft);
x_fft_ideal_out_plot =
[abs([xfft_ideal_out(l_fft/2+1:l_fft)]),abs(xfft_ideal_out(1)),abs(xff
t_ideal_out(2:l_fft/2+1))];
plot(f,x_fft_ideal_out_plot);
xlabel('Frequency rad/s');
ylabel('Magnitude');
title('Output signal of the ideal filter in the frequency domain');
% Time domain representation of output signal after filtering using
ideal filter
subplot(2,1,2);
```

```
stem(n1,ideal_out);
xlabel('n');
ylabel('Amplitude');
title('Output signal of the ideal filter in the time domain');
```

- <u>Comparision</u>

```
mean( (design_out(:)-ideal_out(:)).^2)
mean( (design_out(:)-ideal_out(:)).^2)/(mean( (design_out(:).^2 )))
corrcoef(design_out,ideal_out)
```