PROJECT REPORT

DESIGNING OF FINITE IMPULSE RESPONSE(FIR) BANDPASS FILTER

EN 2570

Name: - Kuruneru C.D

Index No: - 160330R

Abstract

This report includes the technique of designing a Finite Impulse Response(FIR) band pass filter which satisfies some given characteristics. In the implementation I used the Matlab(R2014) software. The design manner used is Fourier Series method where a Kaiser window is used to truncate impulse response. Then the filter is evaluated by analyzing the outputs obtained for given input which is consisted of a combination of sinusoidal signals.

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

This report is consisted of the procedure of designing and evaluating a FIR band pass filter. The software implementation is done by MATLAB (Version R2014, MathWorks Inc.). The filter is comprised given set of characteristics. Fourier series method was used to implement the filter and windowing action was taken placed by Kaiser window. The characteristics of the FIR filter are evaluated by analyzing time domain and frequency domain representation obtained in different stages of the procedure. The frequency response of the filter was obtained primarily through the Fast Fourier Transform(FFT) implementation of Discrete Fourier Transform(DFT). A combination of sinusoids of three frequencies out of which one lies in the pass band is input to the filter to inspect the results of filter implementation. Then the output is compared with the original sinusoidal component of the input in the pass band to compare the results.

METHOD

The two main tasks of the project are implementation of an FIR band pass filter and evaluate its performance. The designation of the filter is done in several stages. At the first stage of the procedure, the design parameters are calculated according to the given filter requirements. The next phase is to obtain the ideal frequency response of the pass band filter. A Kaiser Window function is then obtained such that the ideal frequency response is truncated preserving the required filter parameters. The final stage is to obtain the time domain and frequency domain representation of the designed filter. 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) Filter Implementation

The required parameters for implementing the filter were calculated from the provided filter specifications which are shown in Table 1.

TABLE 1

GIVEN SPECIFICATIONS

Specification	Symbol	Value	Units
Maximum passband ripple	$\tilde{A}_p$	0.08	dB
Minimum stopband attenuation	$\tilde{A}_a$	43	dB
Lower passband edge	ω_{p1}	400	rad/s
Upper passband edge	ω_{p2}	700	rad/s
Lower stopband edge	ω_{a1}	250	rad/s
Upper stopband edge	ω_{a2}	800	rad/s
Sampling frequency	ω_s	2400	rad/s

1) Filter Parameter Derivation:

Based on the provided specifications, the following parameters shown in Table II were derived in order to design the filter and the Kaiser Window.

TABLE 2

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}	$\omega_{p1} - \frac{Bt}{2}$	350	rad/s
Upper cutoff frequency	ω_{c2}	$\omega_{p2} + \frac{Bt}{2}$	750	rad/s
Sampling period	T	$\frac{2\pi}{\omega}$	0.0026	s

1) Derivation of the Kaiser Window:

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

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

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

A parameter δ is defined in a way that the actual pass band ripple $A_p \leq \tilde{A}_p$, and the actual minimum stopband attenuation, $A_a \geq \tilde{A}_a$

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

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

In order to obtain N we define a parameter D as

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

N is chosen so that it is the smallest odd integer satisfying the following inequality

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

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

TABLE 3

KAISER WINDOW PARAMETERS

Parameter	Value	Units
δ	0.0046	-
A_a	46.7351	dB
α	4.1712	-
D	2.7009	-
N	67	-

1) 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,

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

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

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

2) Obtaining the Impulse Response of the Windowed Filter:

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

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

The z – Transform of which takes the form

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

Upon shifting for causality which becomes

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

$filter(nT)$ 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 filter impulse response $filter[nT]$ and $x(nT)$. To avoid the use of convolution operation, we obtain the Discrete Fourier Transform (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 Inverse Discrete Fourier Transform (IDFT). DFT and IDFT are implemented respectively by Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform IFFT methods in MATLAB.

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

TABLE 4

INPUT FREQUENCY COMPONENTS

Frequency Component	Value	Units
ω_1	175	rad/s
ω_2	550	rad/s
ω_3	975	rad/s

RESULTS

A) Time domain and Frequency domain plots of the Filter

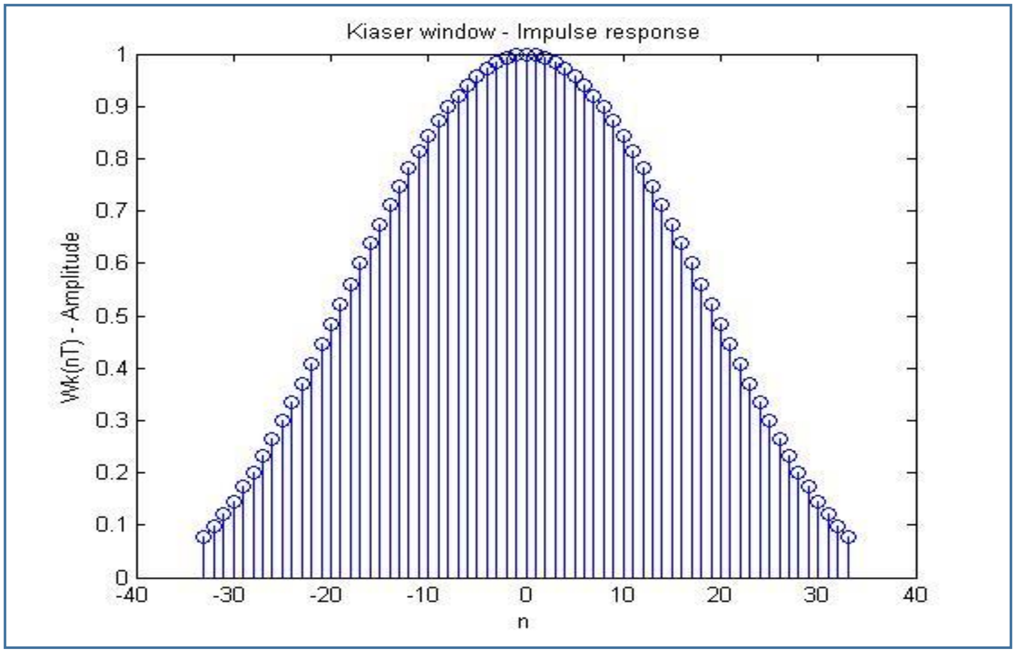

Figure 1: - Impulse response of Kaiser window

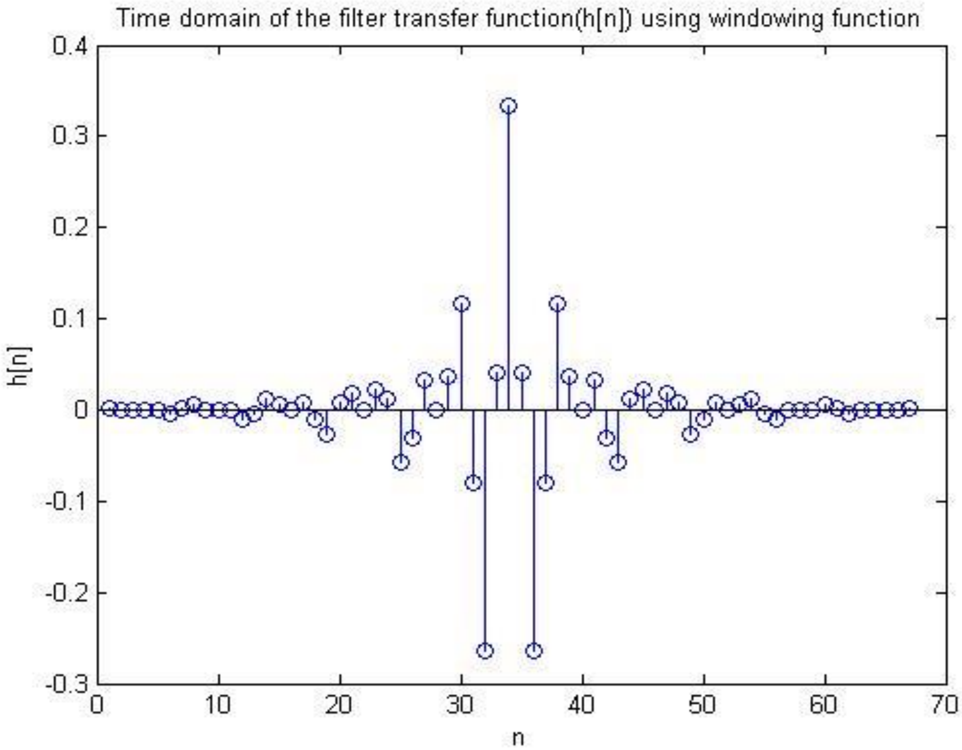

Figure 2: - Time domain representation of the filter designed using the Kaiser window

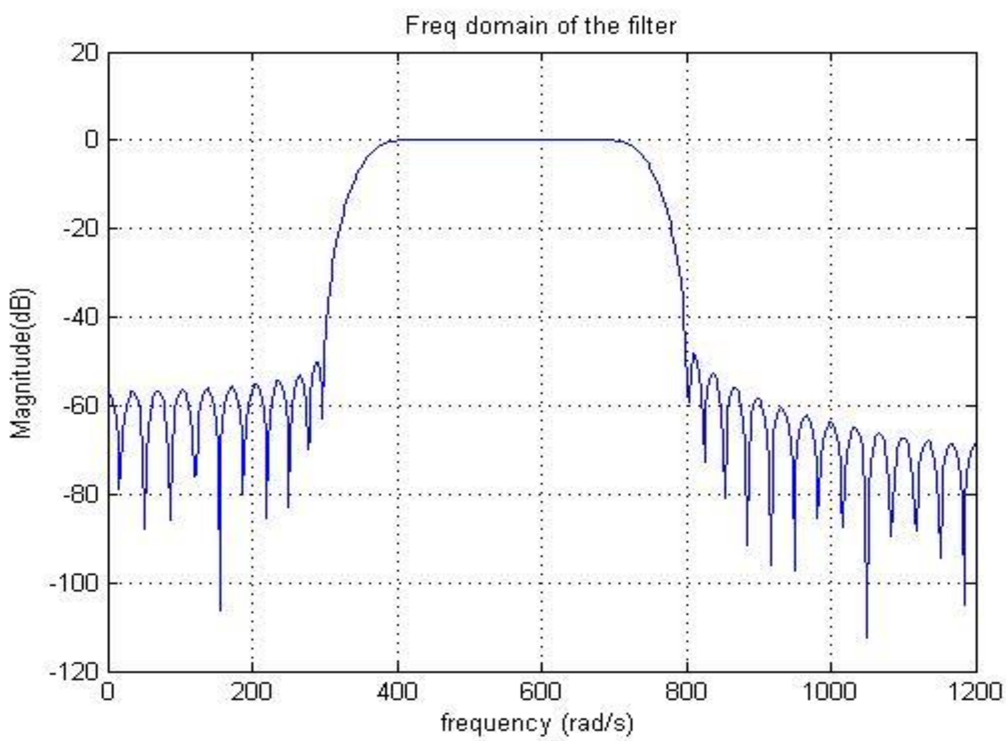

Figure 3: - Time domain representation of the filter designed using the Kaiser window

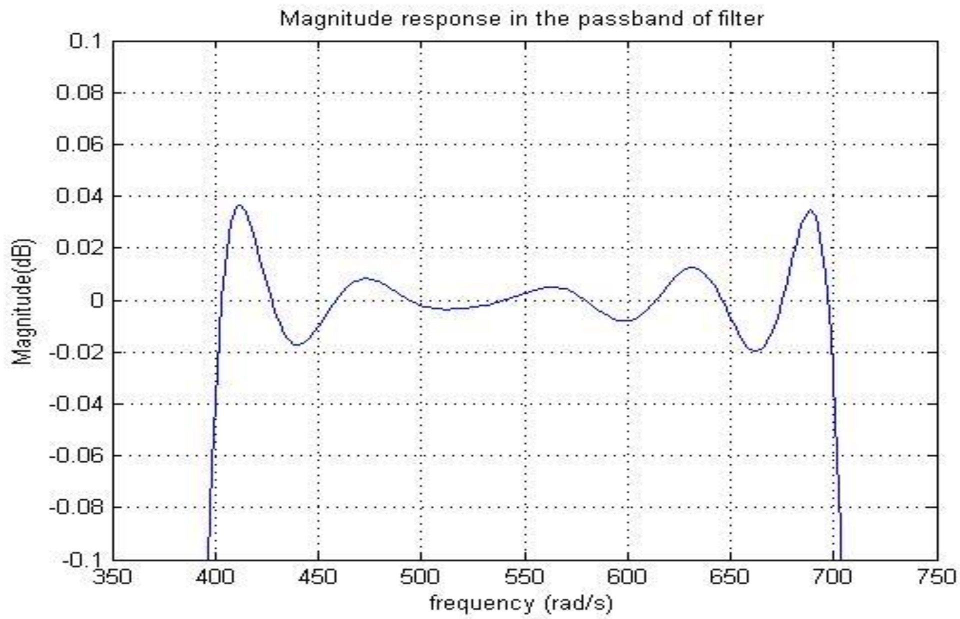

Figure 4: - Pass band of the Filter designed using the Kaiser Window

B) Input and Output waves using the filter

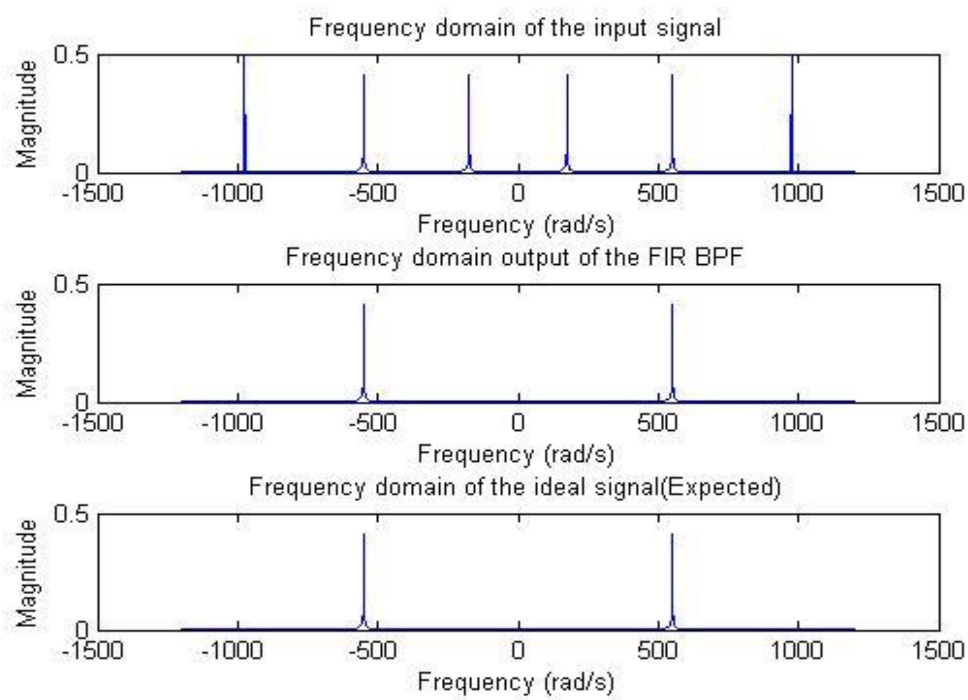

Figure 5: - Frequency domain of input signal, output signal of filter and expected output

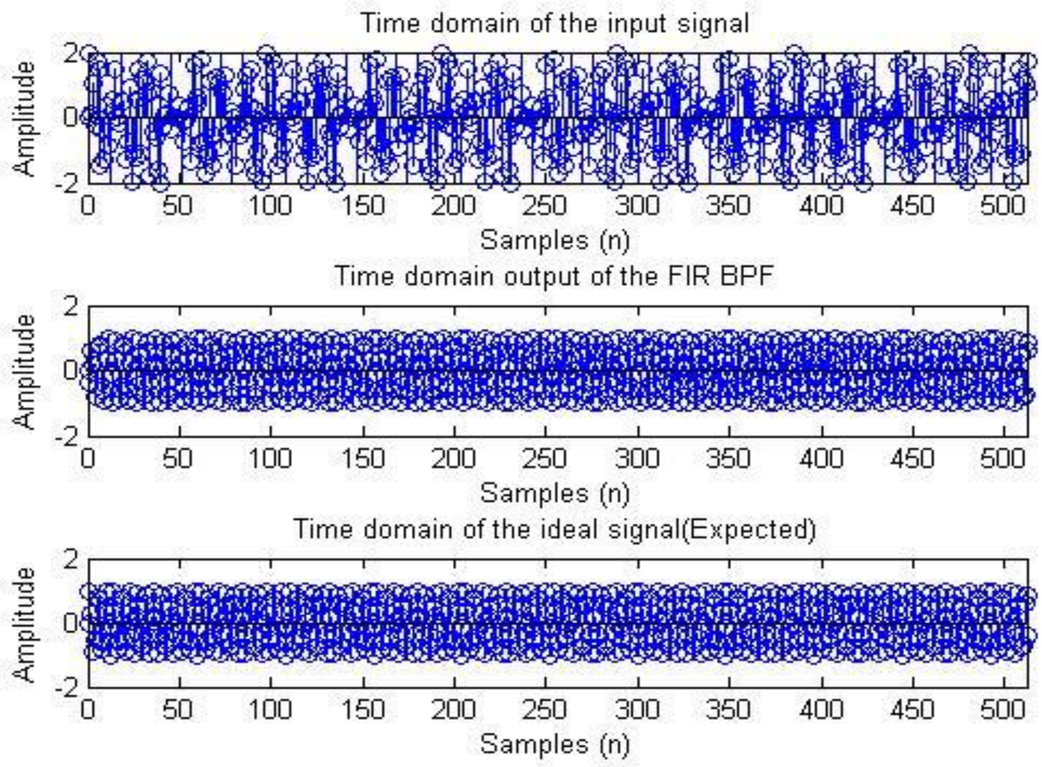

Figure 6: - Frequency domain of input signal, output signal of filter and expected output

CONCLUSION

Kaiser Windowing method is an efficient way of designing an FIR filter under the closed form direct approach. The design can accommodate many design requirements that are specified on the filter. It is also seen that the computational effort of designing is small. However, the major drawback of this filter design is the high order of filter required. There can be optimal filters of lower order that can achieve the same specifications. When implemented in hardware higher order filters can be inefficient due to the usage of more unit delays adders and multipliers. In the software implementation, higher order filters require more computations per sample. More advanced design techniques have to be used in order to obtain optimality.

REFERENCES

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

APPENDIX I

MATLAB SCRIPT FOR THE SOFTWARE IMPLMENTATION

```
Ap = 0.08;  %maximum passband ripple
Aa = 43;    %minimum stopband attenuation
wp1 = 400;  %lower passband edge
wp2 = 700;  %upper passband edge
wa1 = 250;  %lower stopband edge
wa2 = 800;  %upper stopband edge
ws = 2400;  %sampling frequency

Bt = min((wp1-wa1),(wa2-wp2));
wc1 = wp1 - Bt/2;
wc2 = wp2 + Bt/2;
T = 2*pi/ws; %sampling period
deltaph = (10^(0.05*Ap) - 1)/(10^(0.05*Ap) + 1);
deltaah = 10^(-0.05*Aa);
delta = min(deltaph,deltaah);
Aaact = -20*log10(delta); %actual stopband attenuation
```

```
%choosing alpha parameter
if Aaact <= 21
    alpha = 0;
elseif Aaact <= 50
    alpha = 0.5842*((Aaact - 21)^0.4) + 0.07886* (Aaact - 21);
else
    alpha = 0.1102*(Aaact - 8.7);
end
disp(alpha)
%choosing D parameter
if Aaact <= 21
    D = 0.9222;
elseif Aaact > 21
    D = (Aaact - 7.95)/14.36;
end
```

```
% Choosing N value
val = ((ws*D)/Bt) + 1 ;
val = ceil(val);
if mod(val,2) == 0
    N = val + 1;
else
    N = val;
end
%filter n values
n = -(N-1)/2 : 1 : (N-1)/2;
beta = alpha * sqrt(1 - ((2*n)/(N-1)).^2);
%Io(beta) and Io(alpha)
Ioalpha = bessfun(alpha);
Iobeta = bessfun(beta);
%generating window function
wknt = Iobeta/Ioalpha;
```

```
%Impulse response representation of kaizer window
figure;
stem(n,wknt);
xlabel('n')
ylabel('Wk(nT) - Amplitude')
title('Kiaser window - Impulse response')
%generating the ideal frequency response
hnt0 = (2/ws)*(wc2-wc1);
rrange = 1 : (N-1)/2;
hntright = (1./(rrange*pi)).*(sin(wc2.*rrange*T) - sin(wc1.*rrange*T)) ;
lrange = -(N-1)/2 : -1;
hntleft = (1./(lrange*pi)).*(sin(wc2.*lrange*T) - sin(wc1.*lrange*T)) ;
hnt = [hntleft,hnt0,hntright];
figure;
stem(n,hnt);
xlabel('n');
ylabel('hi[n]');
title('Time domain of the ideal band pass filter');
```

```
% combining the ideal bandpass filter with kaiser window to get the h[n]
hn = hnt.*wknt;
figure;
n = 1:N;
stem(n,hn)
xlabel('n')
ylabel('h[n]')
title('Time domain of the filter transfer function(h[n]) using windowing function')
%obtaining the magnitude response of h[n]
[hnMag,f] = freqz(hn);
w = f/T;
loghnMag = 20*log10(abs(hnMag));
figure;
plot(w,loghnMag);
grid on;
xlabel('frequency (rad/s)');ylabel('Magnitude(dB)')
title('Freq domain of the filter')

%obtaining the magnitude response for frequncies in the passband
figure;
plot(w,loghnMag);
grid on;
axis([wc1 wc2 -0.1 0.1]);
xlabel('frequency (rad/s)'); ylabel('Magnitude(dB)');
title('Magnitude response in the passband of filter')
```

```
%Generating the input signal
w1 = wc1/2;
w2 = wc1 + (wc2-wc1)/2;
w3 = wc2 + (ws/2 - wc2)/2;
L = 2048;   %sample size
n = 0:L-1;
f = (-L/2:L/2-1)*ws/L;
x_n=sin(n*w1*T)+sin(n*w2*T)+sin(n*w3*T);   %generated input
t_n=sin(n*w2*T);    %ideal output signal


X_N = fftshift(fft(x_n,L));
T_N = fftshift(fft(t_n,L));
H_N = fftshift(fft(hn,L));

Y_N = X_N.*H_N; % frequency domain multiplication of the signals
```

```
%convolution of x_n and hn
m=length(x_n);
n=length(hn);
X=[x_n,zeros(1,n)];
H=[hn,zeros(1,m)];
Y = zeros(n+m-2);
for i=1:n+m-1
    Y(i)=0;
    for j=1:m
        if(i-j+1>0)
            Y(i)=Y(i)+X(j).*H(i-j+1);
        else
        end
    end
end
y_n = Y;
```

```
figure;
subplot(3,1,1);plot(n,x_n);
ylim([-2 2]);xlim([0 L]);
xlabel('Samples (n)');ylabel('Amplitude');
title('Time domain of the input signal');
subplot(3,1,2);plot(n,y_n((N-1)/2:L+(N-1)/2-1));
ylim([-2 2]);xlim([0 L]);
xlabel('Samples (n)');ylabel('Amplitude');
title('Time domain output of the FIR BPF');
subplot(3,1,3);plot(n,t_n);
ylim([-2 2]);xlim([0 L]);
xlabel('Samples (n)');ylabel('Amplitude');
title('Time domain of the ideal signal(Expected)');
```

```
figure;
subplot(3,1,1);plot(f,abs(X_N/L));
xlabel('Frequency (rad/s)');
ylabel('Magnitude');
title('Frequency domain of the input signal');
subplot(3,1,2);plot(f,abs(Y_N/L));
xlabel('Frequency (rad/s)');
ylabel('Magnitude');
title('Frequency domain output of the FIR BPF');
subplot(3,1,3);plot(f,abs(T_N/L));
xlabel('Frequency (rad/s)');
ylabel('Magnitude');
title('Frequency domain of the ideal signal(Expected)');
```