Department of Electronic and Telecommunication Engineering

University of Moratuwa

EN2570 - Digital Signal Processing

FIR Filter Design Report

Submitted by

SIRITHUNGA M.R.A. 180609B

Submitted on

05/03/2021

Contents

Abstract

This project deals with the designing of the finite impulse response (FIR) bandpass filter. The FIR filter is designed using windowing method in conjunction with the kaiser window. The corresponding theories are going to be explaind according to the prescription that is provided. Matlab R2018a is used for demonstrations and, the figures shown below are the corresponding outputs given by the Matlab R2018a. The codes are included in the appendix.

Introduction

A filter is a component wchich eliminates the undesired parts of the input signal with respect to the users requirements. The digital filters can be seperated by considering the impulse response as[1],

- finite impulse response (FIR) filters and
- infinite impulse response (IIR) filters.

In this report it is going to deal with a FIR bandpass filter in conjunction with the kaiser window. Kiser window method is one of the most fascinating techniques.

Basic Theories

This chapter is cited: A.Antoniou's book [1].The design process of the finite impulse response filters can be categorise into two classical methods.

1. Fourier series method or Window method.
2. Weighted-Chebyshev method.

The fourier series is used in the first method and it will be discussed further with the basic theories later. The weighted-chebyshev method is multivariable optimization method. In order to that window functions are conjuncted with the fourier series method.

3.1 Design Using the Fourier Series

The digital filters have a periodic frequency response. The period is equal to the sampling frequecy hence, the frequency response can be denoted as a fourier series as below.

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

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

By sustituting z=e*jwT to the impulse response of the filter, transfer function can be obtained.

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

There are two major consequences with this fourier seriers method that can be shown because, the fourier series coefficients have beed defined over the range -INF to +INF.

- The nonrecursive filter obtained is of infinite length.
- The filter is noncausal because the impulse response is nonzero for negative time.

To overcome these problems, the impulse response has truncated. And a causal filter can be obtained by delaying the impulse response or by sampling periods appropriately.

$$h(nT) = 0 \qquad \text{for } |n| > \frac{N-1}{2}$$

$$H(z) = h(0) + \sum_{n=1}^{(N-1)/2} [h(-nT)z^{n} + h(nT)z^{-n}]$$

3.2 Use of Window Functions

The oscilations can be found in the passband and the stopband. It is because of the slow convergance of the fourier series. These are called Gibb's oscillations and the effect of Gibb's oscillations[2] can be reduced by the complex-convolution with a discrete-time window function.

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

$$H_w(z) = \mathcal{Z}[w(nT)h(nT)] = \frac{1}{2\pi j} \oint_{\Gamma} H(v)W\left(\frac{z}{v}\right) v^{-1}\, dv$$

The above euation can be interpreted in such a way that illustrates the effect of the window spectrum on the frequency response of the filter.

$$H_w(e^{j\omega T}) = \frac{T}{2\pi} \int_{0}^{2\pi/T} H(e^{j\varpi T})W\left(e^{j(\omega-\varpi)T}\right) d\varpi$$

A variety of such windows has been observed and used over the years. Some significant and

common windows are listed below.

1. Rectangular.
2. von Hann2.
3. Hamming.
4. Blackman.
5. Dolph-Chebyshev.
6. Kaiser.

3.2.1 Kaiser Window

Kaiser window is an one of the most widely used window in fir filter designing. It can be easily achieved the prescribed specifications by using the kaiser window method.This is cited:[3]. The window function is given by,

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

where alpha is an independent parameter. The zero-th order, first kind of Bessel function is denoted by the I. The value of beta and the bessel function are given in the following manner.

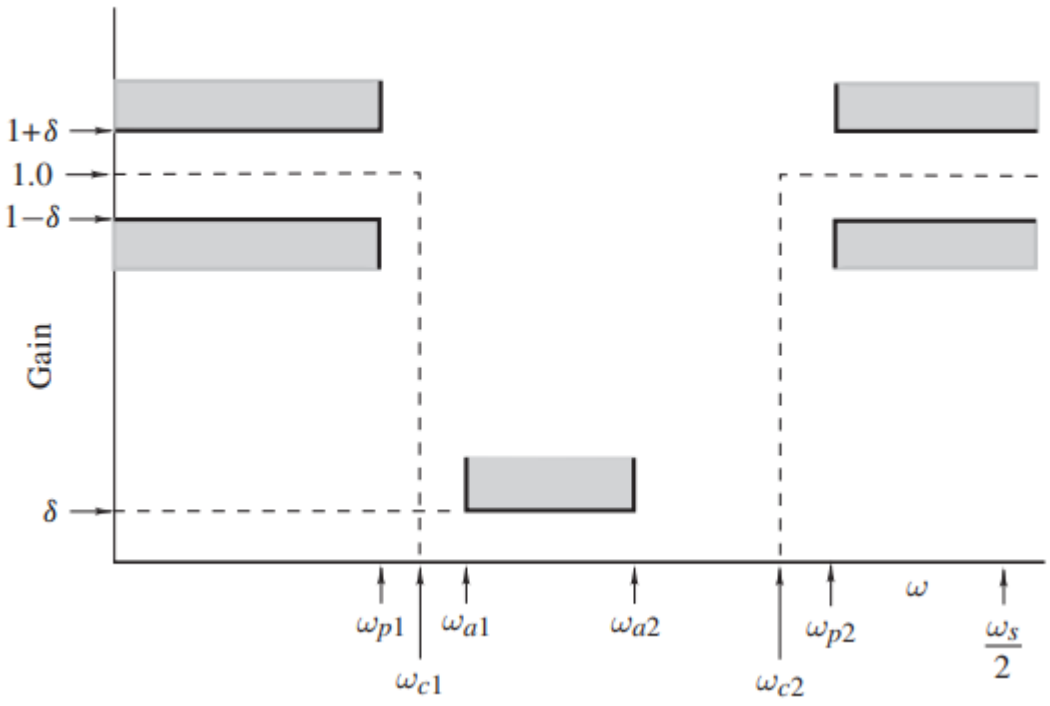
$$\beta = \alpha\sqrt{1-\left(\frac{2n}{N-1}\right)^2} \qquad I_0(x) = 1 + \sum_{k=1}^{\infty}\left[\frac{1}{k!}\left(\frac{x}{2}\right)^k\right]^2$$

3.3 Design of Prescribed Bandpass Filter

The designing procedure of the prescribed filter by using kaiser window will be explained. The required parameters would give as below.

- Passband Ripple $\leqslant \tilde{A}_p$
- Minimum Stop Band Attenuation $\geqslant \tilde{A}_a$
- Lower passband edge ω_{p1}
- Lower stopband edge ω_{a1}
- Upper stopband edge ω_{a2}
- Upper passband edge ω_{p2}
- Sampling frequency ω_s

The above parameters are demonstrated by using a grapgh for further clarifications in the figure 1.



Now, the design of prescribed bandpass filter design can be started.

3.3.1 Determine the Impulse Response.

The minimum transition width Bt is given by the first equation.The frequency response of a band pass filter is given below. The impulse response can be determined by applying the fourier seriers. The lower and upper cutoff frequencies approximate as follow.

$$B_t = \min[(\omega_{p1} - \omega_{a1}),\ (\omega_{a2} - \omega_{p2})]$$

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

$$\omega_{c1} = \omega_{p1} - \frac{B_t}{2} \qquad \omega_{c2} = \omega_{p2} + \frac{B_t}{2}$$

The impulse response for a bandpass filter can be determined as below.

$$h(nT) = \frac{1}{n\pi}(\sin \omega_{c2} nT - \sin \omega_{c1} nT)$$

3.3.2 Choose the Delta

Delta can be calculated using the following equations. Then the actual stopband loss

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

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

of the filter can be determined.

3.3.3 Chose the Alpha

The independent variable alpha shoud determine at this stage. The following conditions must consider when calculating the alpha.

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

3.3.4 Determine the parameter D

It is the last parameter to determine before move in to the kaiser window calcutions. The parameter D can be obtaind after checking these conditions.

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

The N must satisfy the following inequality as well.

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

3.3.5 Apply the Kaiser Window

Finally, compute the kaiser window by using the above determined parameters. It is the last step and the filter has been designed.

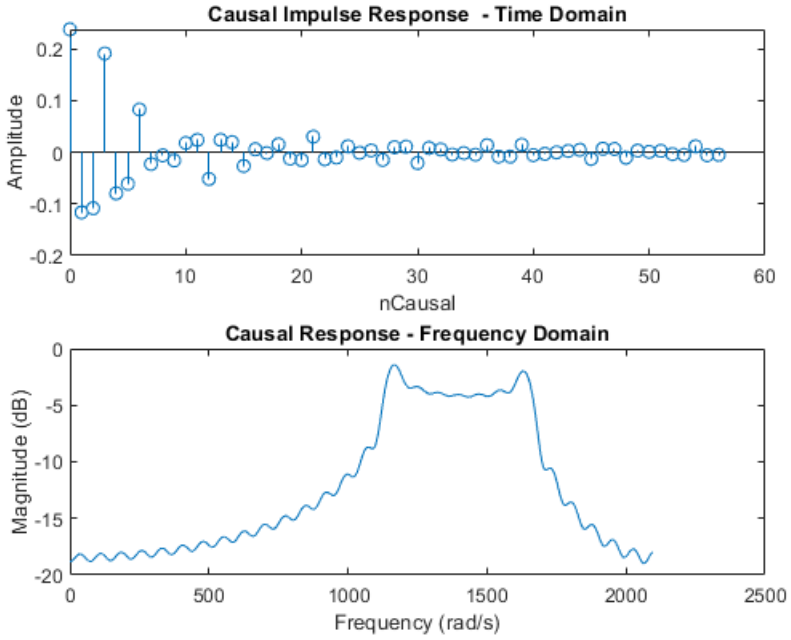
$$H'_w(z) = z^{-(N-1)/2} H_w(z) \qquad \text{where } H_w(z) = \mathcal{Z}[w_K(nT)h(nT)]$$

Results

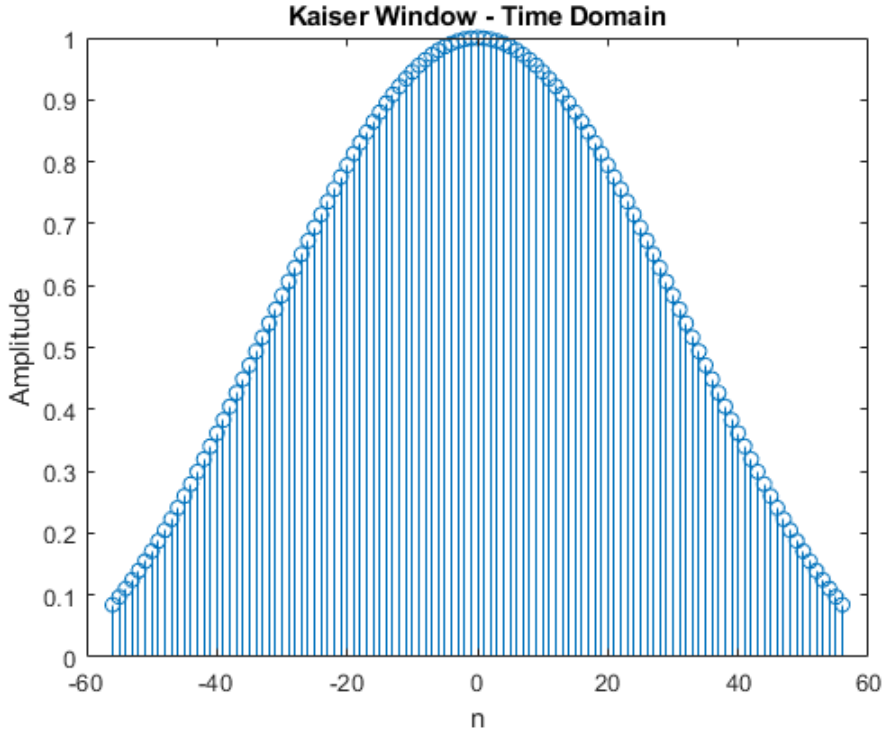
The given parameters and their calculated values are showing in the table. The index is taken as a user input and calculateed the rest by using Matlab R2018a installed version.

Parameter	Value
Passband Ripple	0.0900 dB
Minimum Stop Band Attenuation	45 dB
Lower passband edge	1200 rad/s
Lower stopband edge	1600 rad/s
Upper stopband edge	1050 rad/s
Upper passband edge	1700 rad/s
Sampling frequency	4200 rad/s

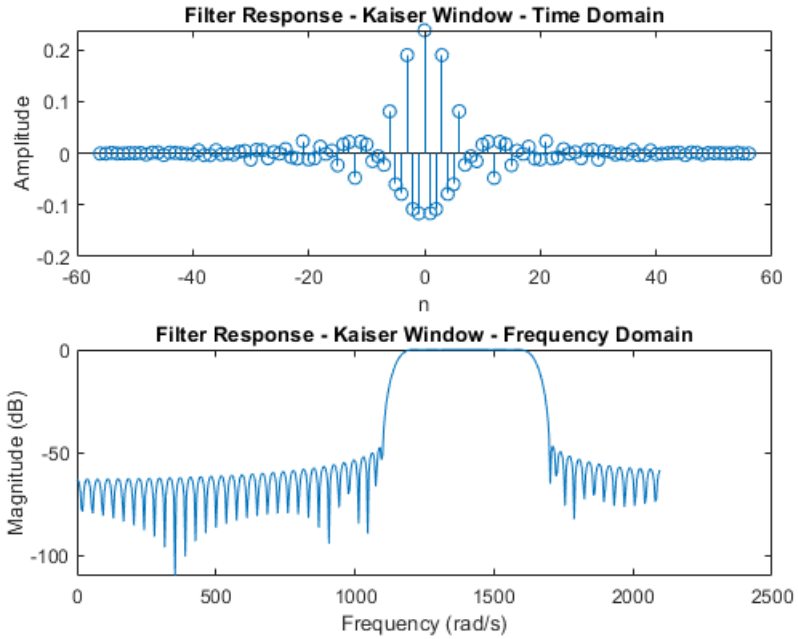
The required impulse response and its frequency domain representation is showing in the figure 2.



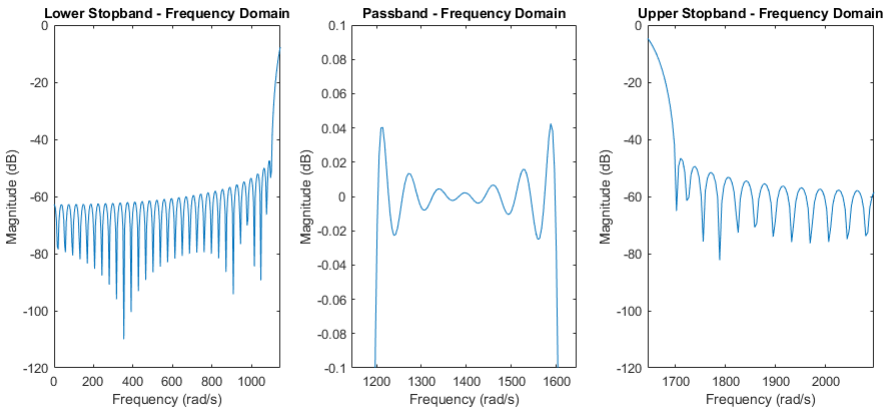
The time domain representation of the bandpass kaiser window is showing in the figure 3.



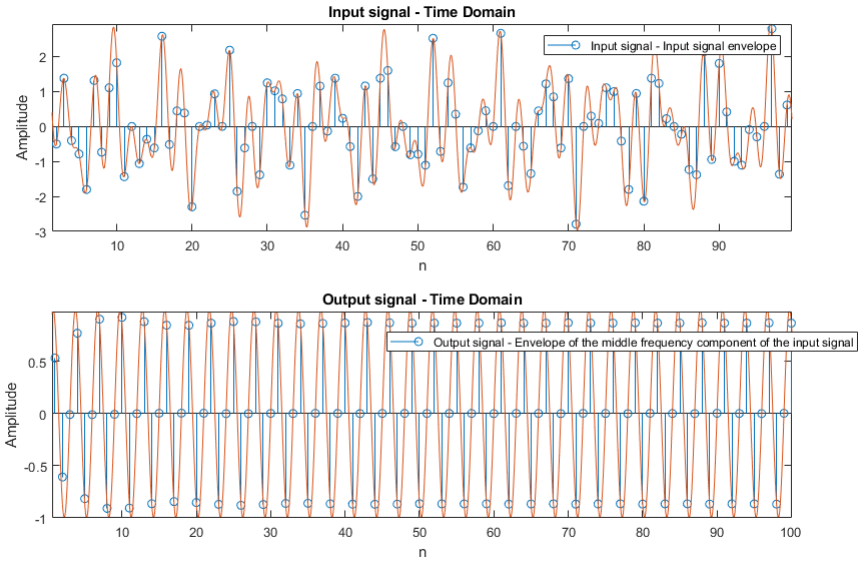
The impulse response of the kaiser window and its frequency domain representation are showing in the figure 4.



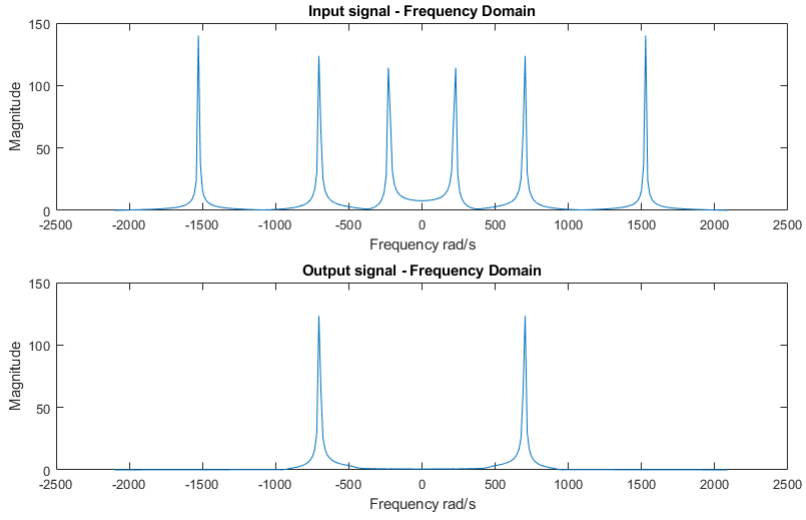
The figure 5 is showing the passband, lower stopband and upper stopband in the frequency domain.



Finally, demonstrate the filter behaviour by using a sample signal and the output is showing in the figure 6.



The filter output in the frequency domain.



Conclusion

Kaiser Window technique is an effective method of planning a FIR filter under the closed form direct methodology, because of its adaptability, the design can oblige many plan prerequisites that are determined on the filter. It is additionally seen that the computational exertion of designing a filter in this strategy is little. Anyway the major downside of this filter configuration is the high order of filter required. There can be ideal filters of lower request that can accomplish similar determinations. At the point when actualized in equipment higher request filters can be wasteful because of the use of more unit delay adders and multipliers. In the product usage, higher order filters require more calculations per test. Further developed plan procedures must be utilized all together to get optimality.

Bibliography

[1] A. Antoniou. *"Digital Signal Processing," 2005 [Online].*. Available: www.ece.uvic.ca/ dsp.

[2] K. Deergha Rao, M.N.S. Swamy *Digital Signal Processing.*. Available: https://link.springer.com/

[3] UKEssays. (November 2018). Filter Design Using Rectangular Window And Kaiser Window Marketing Essay.
`https://www.ukessays.com/essays/marketing/filter-design-using-rectangular-window-and-kaiser-window-marketing-essay.php?vref=1l`

Appendix

A.1 Matlab Code

```
close all;
clear;
clc

%prescribed parameters A,B, and C are calculated by the user input

indexIn = inputdlg ( ' Enter Index Number' , 'Index Number' , 1 )  ;
indexNo = str2double (indexIn{1}(1:end-1));
C = mod(indexNo , 10 ) ;
B = mod( floor(indexNo/10) ,10 ) ;
A = mod( floor(indexNo/100) ,10 ) ;

%parameters for the Bandpass filter

A_tilda_p=0.03+0.01*A;%Maximum passband ripple-:0.0900
A_tilda_a=45+B;%Minimum stopband attenuation-:45
wp1=C*100+300;%Lower passband edge-:1200
wp2=C*100+700;%Upper passband edge-:1600
wa1=C*100+150;%Lower stopband edge-:1050
wa2=C*100+800;%Upper stopband edge-:1700
ws=2*(C*100+1200);%Sampling frequency-:4200

% 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_tilda_p) - 1)/ (10^(0.05*A_tilda_p) + 1); % calculating
```

```
    delta
deltaA = 10^(-0.05*A_tilda_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

N = ceil(ws*D/bt +1); % order of the filter
if mod(N,2) == 0
    N = N+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)

bessellimit = 50;

Ialpha = 1;
for k = 1:bessellimit
    termk = (1/factorial(k)*(alpha/2).^k).^2;
    Ialpha = Ialpha + termk;
end

% Generating I(beta) %%

Ibeta = 1;
for k = 1:bessellimit
    termk = (1/factorial(k)*(beta/2).^k).^2;
    Ibeta = Ibeta + termk;
end

% Obtaining Kaiser Window

wknt = Ibeta/Ialpha;

```

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

% 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/ws*(wc2-wc1);

hnt = [hntleft,hnt0,hntright];
figure
stem(n,hnt)
xlabel('n')
ylabel('Amplitude')
title(strcat('Filter Response - Rectangular window - Time Domain'));

figure
[hi_f,wi_f] = freqz(hnt);
wi_f = wi_f/T;
hi_f = 20*log10(abs(hi_f));
plot(wi_f,hi_f)
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
title(strcat('Filter Response - Rectangular Window - Frequency Domain'));

% 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'));

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'));

```

```
% 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 signal generation %%

w1 = wc1/2;                     %% component frequencies of the input
w2 = wc1 + (wc2-wc1)/2;
w3 = wc2 + (ws/2-wc2)/2;
wi = [w1,w2,w3];

n1 = 0:1:500;                   %% x axis for discrete signal
n2 = 0:0.1:500;                 %% x axis for envelope

xnt = 0;
xdash =0;

for each = wi
    xnt = xnt+ sin(each.*n1.*T);     %% generate discrete signal
    xdash = xdash+sin(each.*n2.*T);  %% generate envelope
end


% Using DFT to check the filtering %%

% Filtering using frequency domain multiplication - see function getfiltered.m
    %%

Npoint = length(xnt) + length(filter) - 1;  % length for fft in x dimension
xfft = fft(xnt,Npoint);
filterfft = fft(filter,Npoint);
outfft = filterfft .* xfft;
out = ifft(outfft,Npoint);

out1 = out(floor(N/2)+1:length(out)-floor(N/2)); % account for shifting delay


% Frequency domain representation of input signal before filtering

figure
```

```
subplot(2,1,1)

Npoint = length(xnt);
xfft = fft(xnt,Npoint);
x1 = n1/length(n1)*ws-ws/2;
plot(x1,abs(xfft))
xlabel('Frequency rad/s')
ylabel('Magnitude')
title(strcat(['Input signal',' ','- Frequency Domain']));

% Frequency domain representation of input signal after filtering

subplot(2,1,2)
Npoint = length(out1);  % length for fft in x dimension
xffout = fft(out1,Npoint);
x1 = n1/length(n1)*ws-ws/2;
plot(x1,abs(xffout))
xlabel('Frequency rad/s')
ylabel('Magnitude')
title(strcat('Output signal',' ','- Frequency Domain'));



% time domain representation of input signal before filtering

figure
subplot(2,1,1)
stem(n1,xnt)
xlabel('n')
ylabel('Amplitude')
title(strcat(['Input signal',' ','- Time Domain']));
hold on
plot(n2,xdash)
legend('Input signal','Input signal envelope');



% Time domain representation of input signal after filtering

subplot(2,1,2)
stem(n1,out1)
xlabel('n')
ylabel('Amplitude')
title(strcat('Output signal',' ','- Time Domain'));

hold on
plot(n2,sin(w2.*n2.*T))
legend('Output signal','Envelope of the middle frequency component of the input
    signal');
```