Department of Electronic and Telecommunication Engineering

University of Moratuwa

Design of FIR and IIR Digital Filters

This report is submitted as a partial fulfillment of the module

EN2063 – Signals and Systems

19th of January 2023

Name - Wijetunga W.L.N.K

Index Number - 200733D

Introduction

Index number - 200733D

Therefore,

A = 7, B = 3, C = 3

According to the index number, the design parameters of the digital filter are as follows.

Parameter	Value
Maximum passband ripple (A_p)	0.17 dB
Minimum stopband attenuation (A_a)	53 dB
Lower passband edge (Ω_{p1})	700 rad/s
Upper passband edge (Ω_{p2})	1200 rad/s
Lower stopband edge (Ω_{s1})	400 rad/s
Upper stopband edge (Ω_{s2})	1200 rad/s
Sampling frequency (Ω_{sm})	3600 rad/s

Table 1 - Design Parameters

Question 1

Using the windowing method in conjunction with the Kaiser window, a FIR bandpass digital filter was designed which satisfies the specifications given in Table 1.

The plot of the Kaiser window function will be as follows.

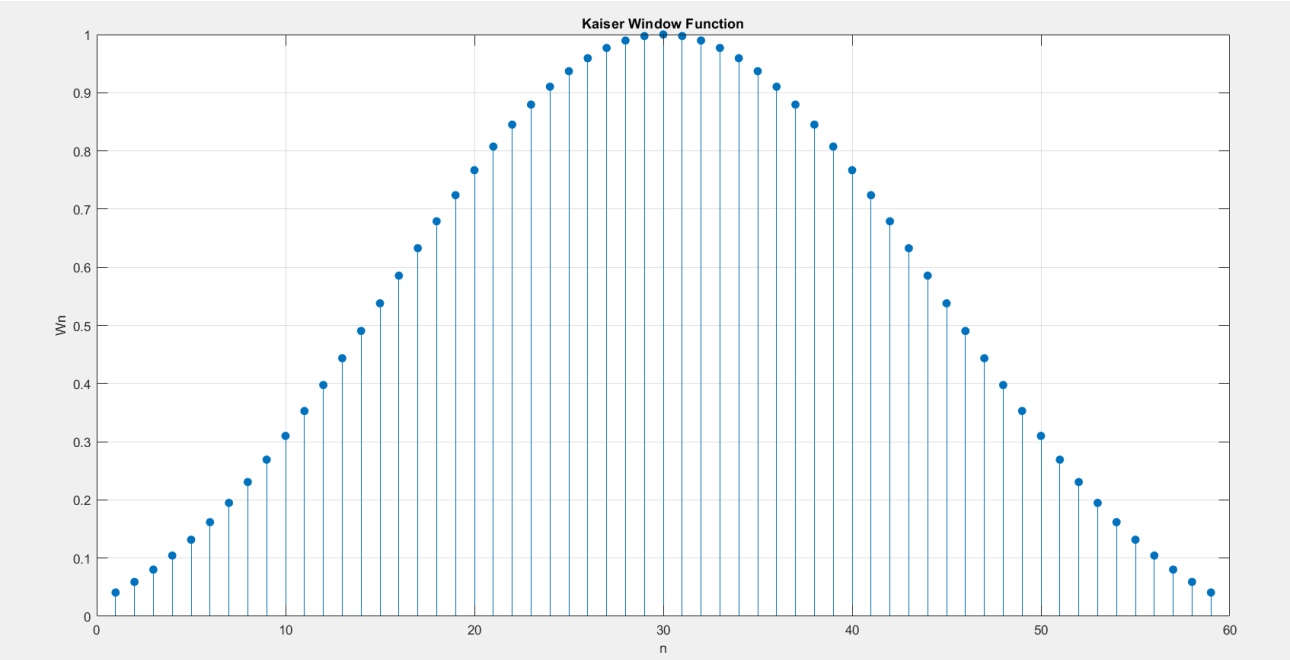

Figure 1

a. The impulse response of the digital filter is as follows

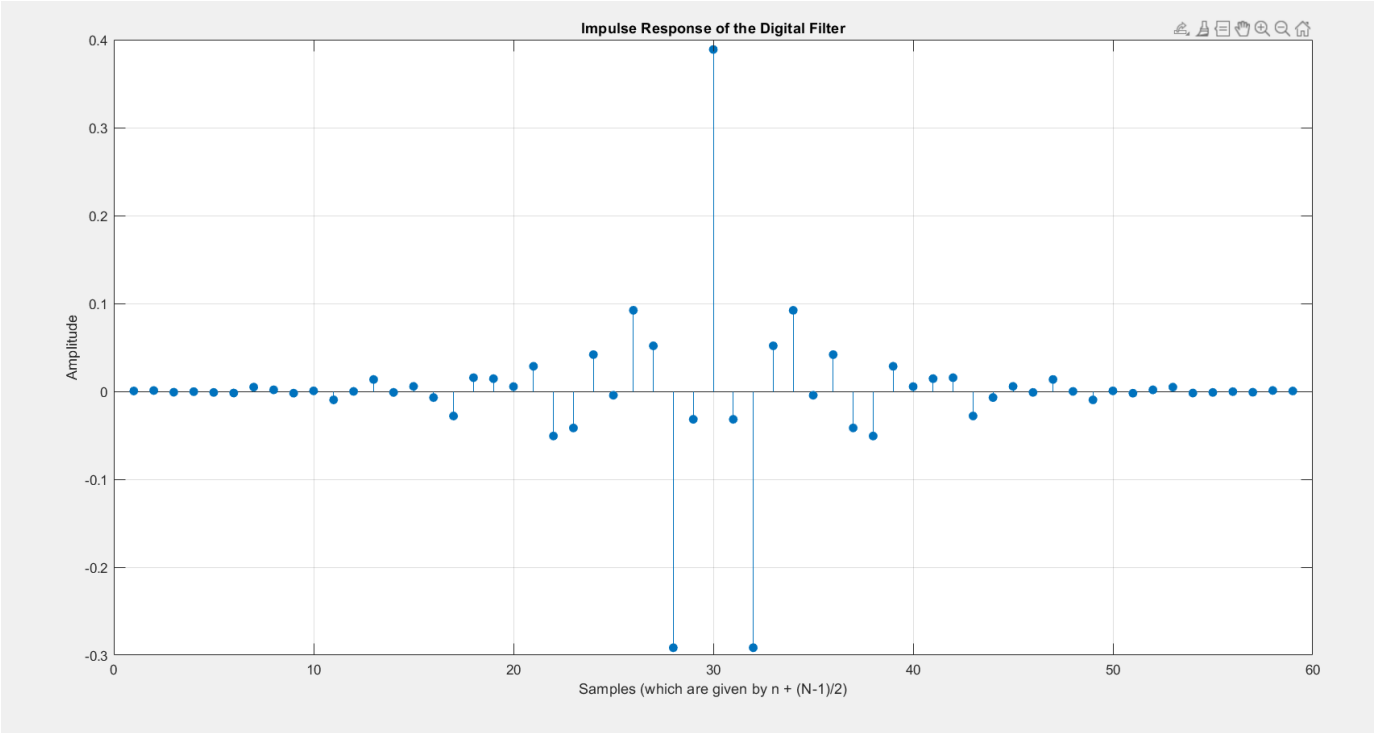

Figure 2

b. The magnitude response of the digital filter for $-\pi \leq \omega < \pi$ rad/sample is as follows

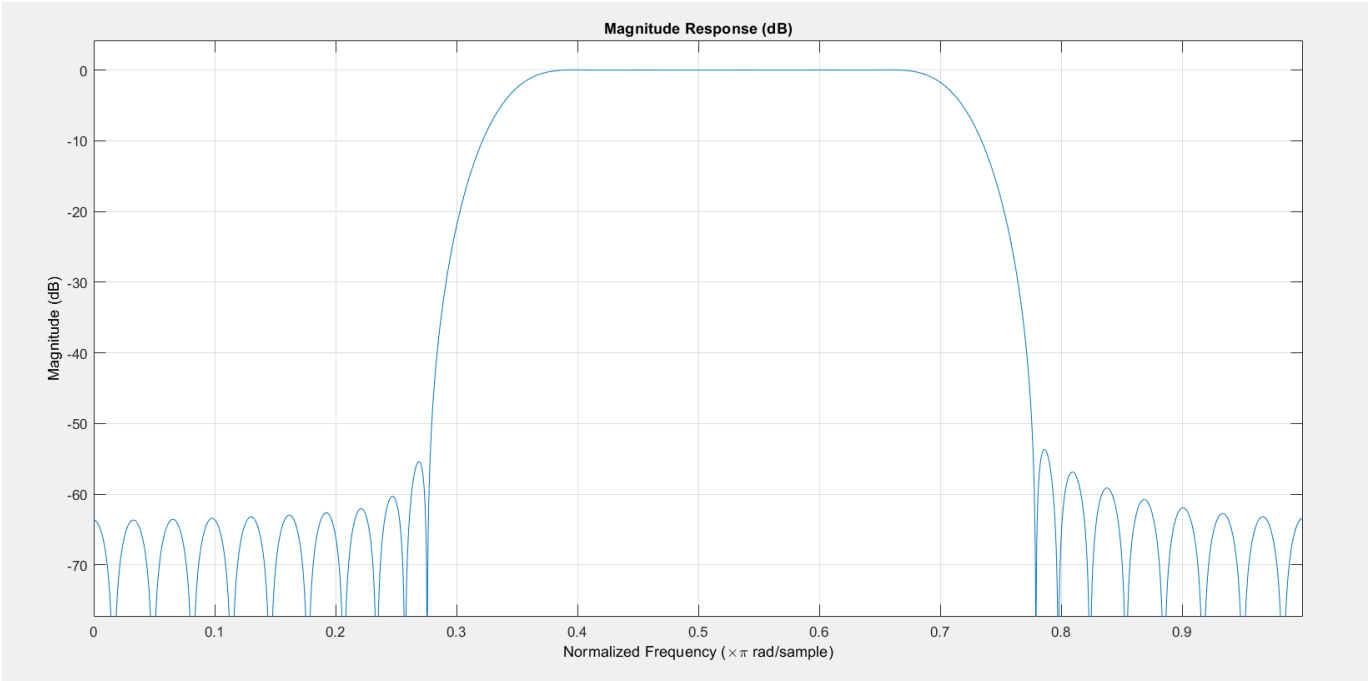

Figure 3

c. The magnitude response for $\omega_{p1} \leq \omega \leq \omega_{p2}$ (in the passband), where ω_{p1} and ω_{p2} are the passband edges in the discrete-time angular frequency domain of the filter is as follows.

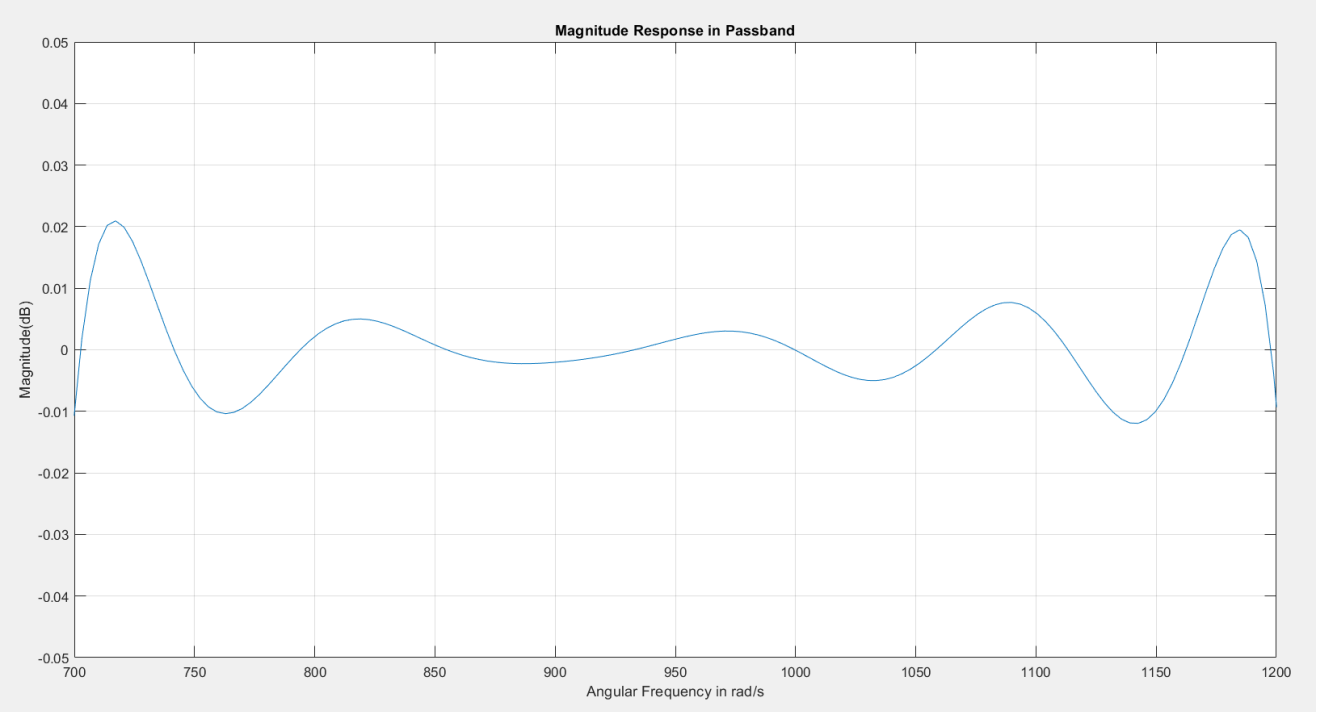

Figure 4

Question 2

Using the bilinear transformation method, an IIR bandpass digital filter which satisfies the specifications given in Table 1 is designed here.

As the first task, an appropriate analog filter was designed, and the required digital filter is obtained by applying the bilinear transform to the transfer function of the designed analog filter.

Pre-warping of frequencies is essential to obtain the required digital filter in this question. Since the approximation method (or the type) of the IIR filter is determined
according to the index number, the approximation method used here is the **elliptic approximation.**

a. The coefficients of the transfer function of the IIR filter.

The transfer function of an IIR filter is in the form

$$h(z) = \frac{\sum b_n z^{-n}}{\sum a_n z^{-n}}$$

Therefore, the coefficients of the transfer function of the IIR filter will be as follows.

n in z^{-n}	b_n	a_n
0	0.0140	1.0000
1	0.0052	0.7559
2	-0.0143	3.0957

3	-0.0006	1.9072
4	0.0267	4.4701
5	0.0000	2.1104
6	-0.0267	3.4940
7	0.0006	1.1500
8	0.0143	1.4721
9	-0.0052	0.2586
10	-0.0140	0.2641

Table 2

b. The magnitude response of the digital filter for $-\pi \leq \omega < \pi$ rad/sample is as follows

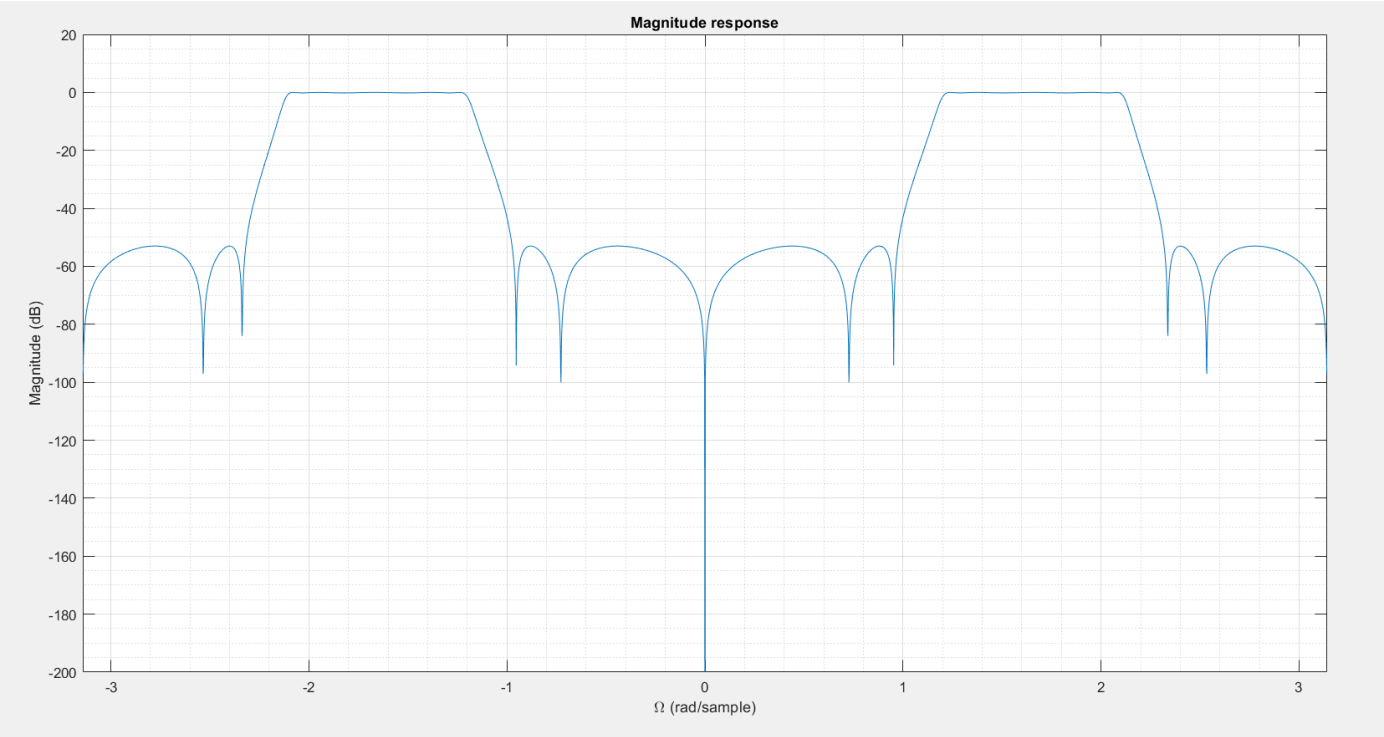

Figure 5

c. The magnitude response for $\omega_{p1} \leq \omega \leq \omega_{p2}$ (in the passband), where ω_{p1} and ω_{p2} are the passband edges in the discrete-time angular frequency domain of the filter is as follows.

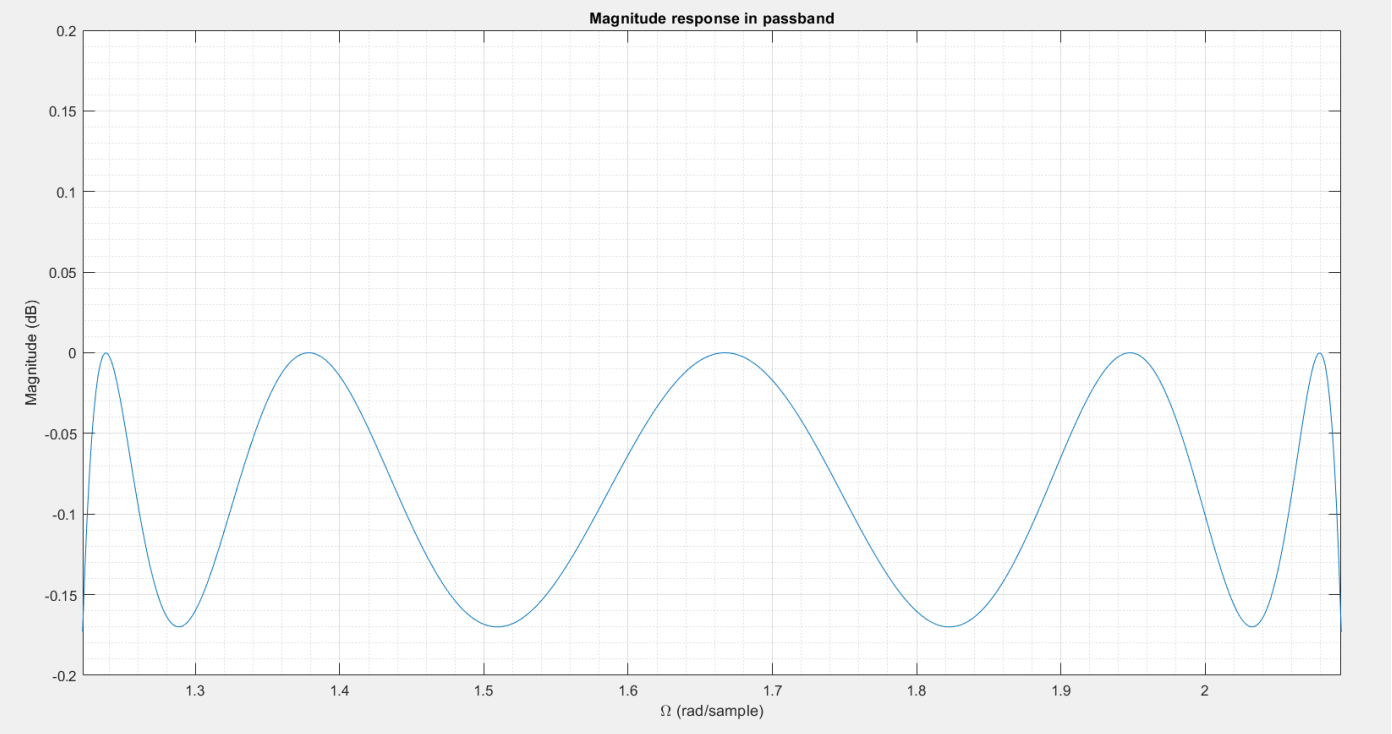

Figure 6

Question 3

It is important to perform a comparison between the designed FIR and IIR filters.

1. Order

The order of the FIR filter is 59 and 5 in the IIR filter designed.

2. Processing of the Samples

Assumption - 2 filters are implemented using the difference equations, and the symmetry of coefficients can be exploited to reduce the number of multiplications.

	FIR	IIR
Number of coefficients in the numerator	60	11
Number of coefficients in the denominator	0	11
Number of multiplications in the numerator	60	11
Number of multiplications in the denominator	0	11
Number of additions in the numerator	29	10
Number of additions in the denominator	0	7
Overall multiplications with symmetry	30 + 0 = 30	6 + 11 = 17
Overall additions with symmetry	29 + 1 = 30	9 + 6 + 1 = 16

Table 3

Appendix 1 - MATLAB Code for FIR Filter Design

```
% Index number - 200733D

% According to the index number,
A=7;
B=3;
C=3;

% Design Specifications
Ap  =  0.1+(0.01* A);       % Maximum passband ripple
Aa  =  50 + B;              % Minimum stopband attenuation
wp1 =  (C * 100) + 400;     % Lower passband edge
wp2 =  (C * 100) + 900;     % Upper passband edge
wa1 =  (C * 100) + 100;     % Lower stopband edge
wa2 =  (C * 100) + 1100;    % Upper stopband edge
ws  =  2*(( C * 100) + 1500) ; % Sampling frequency

%Step 1 - Selecting the transition bandwidth and cutoff points
Bt = min(wp1 - wa1, wa2 - wp2); %Transistion bandwidth is the minumum of lower transistion width and upper
transistion width
wc1 = wp1 - Bt/2 ; %Lower cutoff frequency
wc2 = wp2 + Bt/2 ; %Upper cutoff frequency
T = 2*pi/ws; %Sampling period
disp(Bt);disp(wc1);disp(wc2);disp(T);

%Step 2 - Choosing Delta
delta_p = (10^( Ap/20 ) - 1) / (10^( Ap/20 ) + 1);
delta_a = 10^( -Aa/20 );
delta = min(delta_p,delta_a);
disp(delta)

%Step 3 - Finding the actual stopband attenuation for the defined delta
actual_Aa = -20*log10(delta);

%Step 4 - Choosing the parameter Alpha based on the actual stopband attenuation
if (actual_Aa <= 21)
   alpha = 0;
elseif (actual_Aa <= 50)
   alpha = 0.5842*(actual_Aa - 21)^0.4 + 0.07886*(actual_Aa - 21);
else
   alpha = 0.1102*(actual_Aa - 8.7);
end
disp(alpha)

%Step 5 - Choosing the parameter D based on the actual stop band attenuation
if (actual_Aa <= 21)
   D = 0.9222;
else
   D = (actual_Aa - 7.95) / 14.36;
end
disp(D)

%Selecting the lowest off value N which satifies the inequality N >= ( (ws*D) / Bt )+1
N = ceil(( ws*D / Bt )+1);
if(mod(N,2)==0)
```

```
   N = N + 1;
end
disp("The order of the filter is ")

disp(N)

%Forming Wk(nT), the Kaiser Window function
wk = zeros(N,1);
for n = -(N - 1)/2:(N - 1)/2
  beta = alpha * (1 - (2*n/(N - 1))^2)^0.5;
  numerator = my_Bessel_func(beta);
  denominator = my_Bessel_func(alpha);
  wk(n+(N-1)/2+1) = numerator/denominator;
end

stem(wk,'filled');
title('Kaiser Window Function');
xlabel('n');
ylabel('Wn');
grid on;

%Step 6 - Getting h(nT) using the formed wk(Nt)
h = zeros(N,1);
h(38) = (2/ws)*(wc2 - wc1);
for n = -(N-1)/2:(N-1)/2
  if n==0
    h(n+(N-1)/2+1) = (2/ws)*(wc2 - wc1);
  else
    h(n+(N-1)/2+1) = (1/(n*pi)) * (sin(wc2*n*T) - sin(wc1*n*T));
  end
end
figure;
stem(h,'filled');
title('h(nT)');

disp(wk);
disp(h);

digital_filter = h.*wk;
disp(digital_filter);

%Impulse response of the Filter
figure;
stem(digital_filter,'filled');
title('Impulse Response of the Digital Filter');
xlabel('Samples (which are given by n + (N-1)/2)');
ylabel('Amplitude');
grid on;

% Magnitude Response of the Filter in rad/s
[h,w]=freqz(digital_filter);
magnitude = 20*log10(abs(h));
analogFreq = w*ws/(2*pi);
figure;
plot(analogFreq,magnitude);
grid ON;
title('Magnitude Response of the Digital Filter');
xlabel('Angular Frequency in rad/s');
ylabel('Magnitude(dB)');

% Magnitude Response of the Filter in rad/sample
fvtool(digital_filter);
```

```
%Magnitude Response in Passband
[amp, digiFreq] = freqz(digital_filter);
analogFreq = digiFreq*ws/(2*pi);
ampdb = 20*log10(abs(amp));
figure;
plot(analogFreq, ampdb);
axis([wp1 wp2 -0.05 0.05]);
title('Magnitude Response in Passband');
xlabel('Angular Frequency in rad/s');
ylabel('Magnitude(dB)');
grid on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Testing

%Input Signal Generation
no_of_samples = 300;
w1 = wa1/2;           %Middle freueny of the lower stopband
w2 = (wp1 + wp2)/2;   %Middle frequency of the passband
w3 = (wa2 + ws/2)/2;  %Middle frequency of the upper stopband

Input_signal = zeros(no_of_samples,1);

for i = 1:no_of_samples
  Input_signal(i) = sin(w1*i*T) + sin(w2*i*T) + sin(w3*i*T);
end

%Plotting the Input Signal
figure;
subplot(3,1,1);
stem(Input_signal);
title('Input Signal');
xlabel('Samples');
ylabel('Amplitude');

%Expected Output Signal
expected_output = zeros(no_of_samples,1);

for i = 1:no_of_samples
  expected_output(i) = sin(w2*i*T);
end

subplot(3,1,2);
stem(expected_output);
axis([0 no_of_samples -2 2]);
title('Expected Output Signal');
xlabel('Samples');
ylabel('Amplitude');

%Actual Output Signal
lenfft = length(Input_signal) + length(digital_filter) - 1;

XF = fft(Input_signal,lenfft);  %Taking the FT of the input signal
HF = fft(digital_filter,lenfft);    %Taking the FT of the filter
YF = HF.*XF;

Actual_Output = ifft(YF,lenfft); %Taking the Inverse FT of YF

subplot(3,1,3);
stem(Actual_Output);
```

```
axis([0 no_of_samples -2 2]);
title('Actual Output Signal');
xlabel('Samples');
ylabel('Amplitude');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Bessel function which is required to find the Kaiser window function
function [ result ] = my_Bessel_func( x )
  j = 1;
  result = 0;
  term = 10;
  while (term > 10^(-6))
    term = (((x/2)^j)/(factorial(j)))^2;
    result = result + term;
    j = j + 1;
  end

  result = result + 1;
end
```

Appendix 2 - MATLAB Code for IIR Filter Design

```
% Index number - 200733D
% According to the index number,
clc;
clear all;
close all;
A=7;
B=3;
C=3;

% Design Specifications
Ap  =  0.1+(0.01* A);        % Maximum passband ripple
Aa  =  50 + B;               % Minimum stopband attenuation
wp1 =  (C * 100) + 400;      % Lower passband edge
wp2 =  (C * 100) + 900;      % Upper passband edge
wa1 =  (C * 100) + 100;      % Lower stopband edge
wa2 =  (C * 100) + 1100;     % Upper stopband edge
ws  =   2*(( C * 100) + 1500) ; % Sampling frequency

% Sampling period
Ts = 2*pi/ws;

% Sampling frequency
fs = 1/Ts;

% Passband and stopband frequencies
wp = [wp1 wp2];
ws1 = [wa1 wa2];

% Apply pre - warping transformation
wp_warped = 2*tan(wp*Ts/2)/Ts;
ws_warped = 2*tan(ws1*Ts/2)/Ts;

%Getting the order and the cutoff frequencies
[n,Wp] = ellipord(wp_warped, ws_warped , Ap, Aa,"s");
disp(n);
```

```
disp(Wp);

%Getting the required digital filter coefficients
[num,den] = ellip(n,Ap,Aa,Wp,"bandpass","s");

%Applying Bilinear transform
[dnum,dden]=bilinear(num,den,fs);

%Getting the required digital filter coefficients
disp(dnum);
disp(dden);
printsys (dnum , dden )

[H,w]=freqz(dnum,dden,2001);

Hdb = 20*log10(abs(H));

%Plotting the Magnitude response of the filter
figure;
plot([flip(-w); w], [flip(Hdb); Hdb])
xlabel('\Omega (rad/sample)')
ylabel('Magnitude (dB)')
title('Magnitude response')
ax = gca;
ax.YLim = [-200 20];
ax.XLim = [-pi pi];
grid on;
grid minor;

%Plotting the Magnitude response in the Passband
figure;
plot(w, Hdb);
xlabel('\Omega (rad/sample)')
ylabel('Magnitude (dB)')
title('Magnitude response in passband')
ax = gca;
ax.YLim = [-0.2 0.2];
ax.XLim = [wp1*Ts wp2*Ts];
grid on;
grid minor;
```

Appendix 3 - MATLAB Code for Comparison of 2 Filters

```
% Index number - 200733D

% According to the index number,
A=7;
B=3;
C=3;

% Design Specifications
Ap =  0.1+(0.01* A);       % Maximum passband ripple
Aa =  50 + B;              % Minimum stopband attenuation
wp1 =  (C * 100) + 400;    % Lower passband edge
wp2 =  (C * 100) + 900;    % Upper passband edge
wa1 =  (C * 100) + 100;    % Lower stopband edge
wa2 =  (C * 100) + 1100;   % Upper stopband edge
ws =  2*(( C * 100) + 1500) ; % Sampling frequency
```

```
%Step 1 - Selecting the transistion bandwidth and cutoff points
Bt = min(wp1 - wa1, wa2 - wp2); %Transistion bandwidth is the minumum of lower transistion width and upper
transistion width
wc1 = wp1 - Bt/2 ; %Lower cutoff frequency
wc2 = wp2 + Bt/2 ; %Upper cutoff frequency
T = 2*pi/ws; %Sampling period
disp(Bt);disp(wc1);disp(wc2);disp(T);

%Step 2 - Choosing Delta
delta_p = (10^( Ap/20 ) - 1) / (10^( Ap/20 ) + 1);
delta_a = 10^( -Aa/20 );
delta = min(delta_p,delta_a);
disp(delta)

%Step 3 - Finding the actual stopband attenuation for the defined delta
actual_Aa = -20*log10(delta);

%Step 4 - Choosing the parameter Alpha based on the actual stopband attenuation
if (actual_Aa <= 21)
   alpha = 0;
elseif (actual_Aa <= 50)
   alpha = 0.5842*(actual_Aa - 21)^0.4 + 0.07886*(actual_Aa - 21);
else
   alpha = 0.1102*(actual_Aa - 8.7);
end
disp(alpha)

%Step 5 - Choosing the parameter D based on the actual stop band attenuation
if (actual_Aa <= 21)
   D = 0.9222;
else
   D = (actual_Aa - 7.95) / 14.36;
end
disp(D)

%Selecting the lowest off value N which satifies the inequality N >= ( (ws*D) / Bt )+1
N = ceil(( ws*D / Bt )+1);
if(mod(N,2)==0)
   N = N + 1;
end

disp("The order of the filter is ")
disp(N)

%Forming Wk(nT), the Kaiser Window function
wk = zeros(N,1);
for n = -(N - 1)/2:(N - 1)/2
  beta = alpha * (1 - (2*n/(N - 1))^2)^0.5;
  numerator = my_Bessel_func(beta);
  denominator = my_Bessel_func(alpha);
  wk(n+(N-1)/2+1) = numerator/denominator;
end

stem(wk,'filled');
title('Kaiser Window Function');
xlabel('n');
ylabel('Wn');
grid on;

%Step 6 - Getting h(nT) using the formed wk(Nt)
h = zeros(N,1);
h(38) = (2/ws)*(wc2 - wc1);
```

```
for n = -(N-1)/2:(N-1)/2
  if n==0
    h(n+(N-1)/2+1) = (2/ws)*(wc2 - wc1);
  else
    h(n+(N-1)/2+1) = (1/(n*pi)) * (sin(wc2*n*T) - sin(wc1*n*T));
  end
end
figure;
stem(h,'filled');
title('h(nT)');

digital_filter = h.*wk;

%Impulse response of the Filter
figure;
stem(digital_filter,'filled');
title('Impulse Response of the Digital Filter');
xlabel('Samples (which are given by n + (N-1)/2)');
ylabel('Amplitude');
grid on;

% Magnitude Response of the Filter in rad/s
[h,w]=freqz(digital_filter);
magnitude = 20*log10(abs(h));
analogFreq = w*ws/(2*pi);
figure;
plot(analogFreq,magnitude);
grid ON;
title('Magnitude Response of the Digital Filter');
xlabel('Angular Frequency in rad/s');
ylabel('Magnitude(dB)');

% Magnitude Response of the Filter in rad/sample
fvtool(digital_filter);

%Magnitude Response in Passband
[amp, digiFreq] = freqz(digital_filter);
analogFreq = digiFreq*ws/(2*pi);
ampdb = 20*log10(abs(amp));
figure;
plot(analogFreq, ampdb);
axis([wp1 wp2 -0.05 0.05]);
title('Magnitude Response in Passband');
xlabel('Angular Frequency in rad/s');
ylabel('Magnitude(dB)');
grid on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sampling period
Ts = 2*pi/ws;

% Sampling frequency
fs = 1/Ts;

% Passband and stopband frequencies
wp = [wp1 wp2];
ws1 = [wa1 wa2];

% Apply pre - warping transformation
wp_warped = 2*tan(wp*Ts/2)/Ts;
ws_warped = 2*tan(ws1*Ts/2)/Ts;
```

```
%Getting the order and the cutoff frequencies
[n,Wp] = ellipord(wp_warped, ws_warped , Ap, Aa,"s");
disp(n);
disp(Wp);


[num,den] = ellip(n,Ap,Aa,Wp,"bandpass","s");

%Applying Bilinear transform
[dnum,dden]=bilinear(num,den,fs);

%Getting the required digital filter coefficients
disp(dnum);
disp(dden);
printsys (dnum , dden )

[H,w2]=freqz(dnum,dden,2001);

Hdb = 20*log10(abs(H));

%Plotting the Magnitude response of the filter
figure;
plot([flip(-w2); w2], [flip(Hdb); Hdb])
xlabel('\Omega (rad/sample)')
ylabel('Magnitude (dB)')
title('Magnitude response')
ax = gca;
ax.YLim = [-200 20];
ax.XLim = [-pi pi];
grid on;
grid minor;

%Plotting the Magnitude response in the Passband
figure;
plot(w2, Hdb);
xlabel('\Omega (rad/sample)')
ylabel('Magnitude (dB)')
title('Magnitude response in passband')
ax = gca;
ax.YLim = [-0.2 0.2];
ax.XLim = [wp1*Ts wp2*Ts];
grid on;
grid minor;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Comparison of filters
figure('name','Comparison of FIR and IIR filters in between [0,pi]');
plot(w/pi,20*log10(abs(h)));
xlabel("\omega (rad/sample)")
ylabel("Magnitude (db)");

hold on;

plot(w2/pi,20*log10(abs(H)))
xlabel("\omega (rad/sample)");
ylabel("Magnitude (db)");
grid on;

legend('FIR','IIR');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Bessel function which is required to find the Kaiser window function
function [ result ] = my_Bessel_func( x )
  j = 1;
  result = 0;
  term = 10;
  while (term > 10^(-6))
    term = (((x/2)^j)/(factorial(j)))^2;
    result = result + term;
    j = j + 1;
  end

  result = result + 1;
end
```