UNIVERSITY OF MORATUWA, SRI LANKA

Faculty of Engineering

Department of Electronic and Telecommunication Engineering

Semester 3 (Intake 2020)

EN2063—SIGNALS AND SYSTEMS

Project

Design of FIR and IIR Digital Filters

W. A. S. S. WANIGATHUNGA

200693D

January 19, 2023

ABSTRACT

This report outlines the design process of FIR and IIR digital bandpass filters using MATLAB 2020a as the programming environment for the project. For FIR filters, the windowing method (in conjunction with the Kaiser window) is used whereas, for IIR filters, the bilinear transformation with elliptic approximation method is used.

INTRODUCTION

There are two classical methods for the design of nonrecursive filters:

1. By using the Fourier series in conjunction with a class of functions known as window functions. The method is referred to as the Fourier series method or as the window method.
2. By using a multivariable optimization method known as the weighted-Chebyshev method.

In this report the Fourier series method is used in conjunction with the window known as the Kaiser Window.

In this report the Bilinear transformation method is used to design the digital IIR filter. The Elliptic Approximation method is a technique used to design IIR digital filters using the Bilinear Transformation. It is based on the elliptic function, which is a mathematical function that can be used to approximate the frequency response of an ideal filter. The method involves using an elliptic function to design an analog prototype filter, and then using the Bilinear Transformation to map the frequency response of the analog filter to a digital filter.

DESIGN PROCEDURE OF A FIR BANDPASS DIGITAL FILTER USING WINDOWING METHOD IN CONJUNCTION WITH THE KAISER WINDOW

The following table gives the design specifications of the desired bandpass filter.

Parameter	Value
Maximum passband ripple, $\overline{A}_p$	0.1 + (0.01 × A) dB = 0.16dB
Minimum stopband attenuation, $\overline{A}_a$	50 + B dB=59dB
Lower passband edge, Ω_{p1}	(C × 100) + 400 rad/s=700 rad/s
Upper passband edge, Ω_{p2}	(C × 100) + 900 rad/s=1200 rad/s
Lower stopband edge, Ω_{s1}	(C × 100) + 100 rad/s=400 rad/s
Upper stopband edge, Ω_{s2}	(C × 100) + 1100 rad/s=1400 rad/s
Sampling frequency, Ω_{sm}	2((C × 100) + 1500) rad/s=3600 rad/s

The more critical transition width is

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

and hence the idealized frequency response for a BP filter is deduced as

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

where $\omega_{c1} = \omega_{p1} - \frac{B_t}{2}$, $\omega_{c2} = \omega_{p2} + \frac{B_t}{2}$, $T = \frac{2\pi}{\omega_{sm}}$

Applying the Fourier series to the idealized frequency response, we get

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

Design Procedure

- Choose a suitable value for δ using the below equations.

$$\tilde{\delta_p} = \frac{10^{0.05\tilde{A}_p} - 1}{10^{0.05\tilde{A}_p} + 1}$$

$$\tilde{\delta_a} = 10^{-0.05\tilde{A}_a}$$

$$\delta = min(\tilde{\delta_p}, \tilde{\delta_a})$$

- Calculate the actual stop band attenuation Aa by

$$A_a = -20log(\delta)$$

- Choose parameter α as,

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

- Choose parameter D as,

$$D = \begin{cases} 0.9222, & \text{for } A_a \leqslant 21 \\ \frac{Aa - 7.95}{14.26}, & \text{for } A_a > 21 \end{cases}$$

- Then select the lowest odd value of N that would satisfy the inequality,

$$N \geqslant \frac{\omega_s D}{B_t} + 1 \text{ where } B_t = min[(\omega_{a1} - \omega_{p1}), (\omega_{p2} - \omega_{a2})]$$

- Then using the definition of the Kaiser Window, find wK (nT) :

$$w_K(nT) = \begin{cases} \frac{I_o(\beta)}{I_o(\alpha)}, & \text{for } |n| < (N-1)/2 \\ 0, & \text{otherwise} \end{cases}$$

here

$$\beta = \alpha\sqrt{1 - \left(\frac{2n}{N-1}\right)^2}$$

$$I_o(x) = 1 + \sum_{k=1}^{\infty} \left[\frac{1}{k!}\left(\frac{x}{2}\right)^k\right]^2$$

- Finally form the modified transfer function of the filter.

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

DESIGN PROCEDURE OF A IIR BANDPASS DIGITAL FILTER USING ELLIPTIC APPROXIMATION

According to my index number I was given to implement the IIR filter using elliptic approximation method. The design procedure of a IIR bandpass digital filter using elliptic approximation typically involves the following steps:

1. Determine the passband and stopband frequencies of the filter. This includes specifying the lower and upper cutoff frequencies of the passband and the transition bandwidth.
2. Use the Bilinear Transformation to map the frequency response of the analog filter to a digital filter. This involves converting the analog filter coefficients to digital filter coefficients by applying the transformation to each coefficient. (In order to remove warping effect)

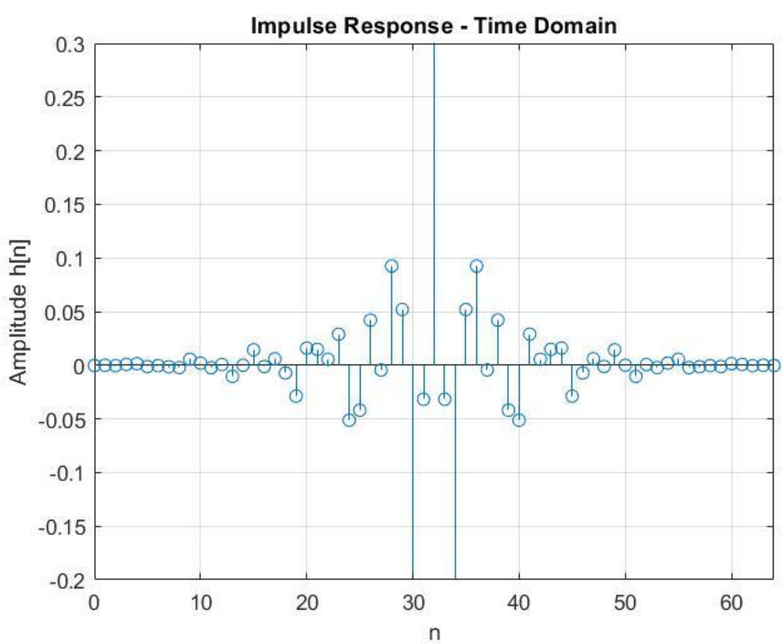
$$\omega = \frac{2}{T} tan \frac{\Omega T}{2}$$

3. Required order of the Elliptic filter can be calculated using the 'ellipord' function in MATLAB.
4. Then generate the analog Elliptic filter using MATLAB.
5. Implement the digital filter using the digital filter coefficients obtained from the Bilinear Transformation. (to convert S-domain transfer function to Z-domain) This is done using MATLABs.

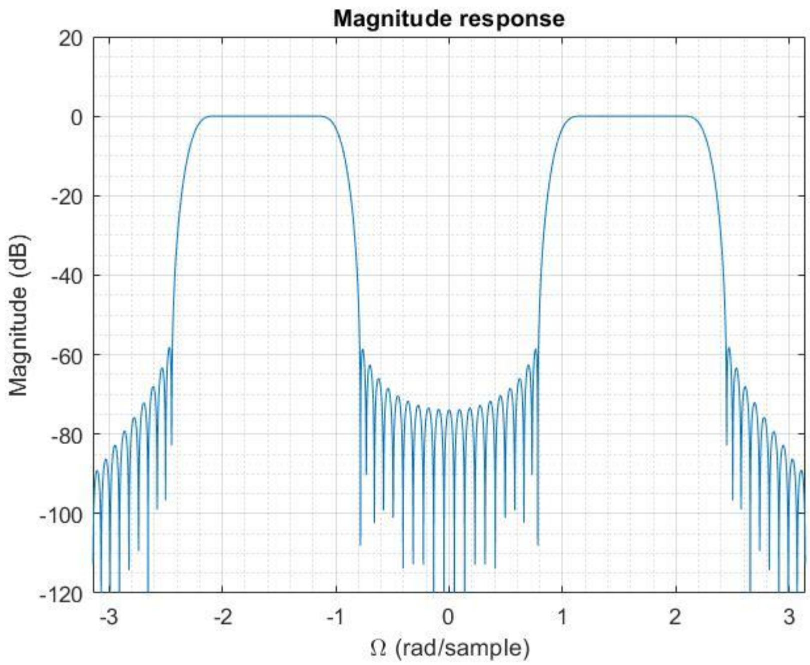
RESULTS

1. FIR BANDPASS DIGITAL FILTER USING WINDOWING METHOD IN CONJUNCTION WITH THE KAISER WINDOW

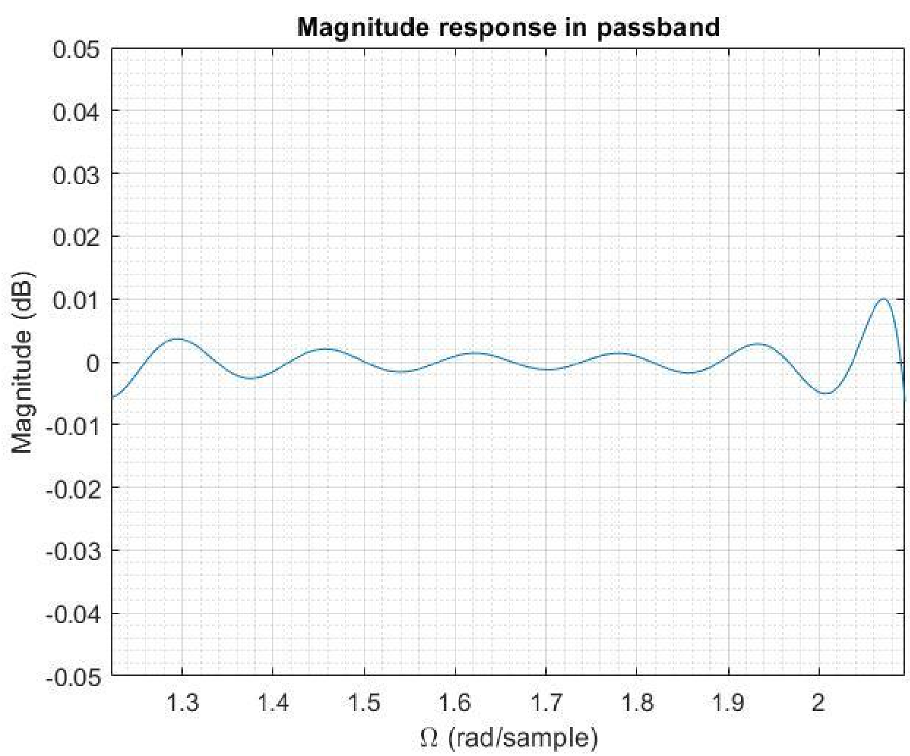
1.1 Impulse response



1.2 Magnitude response of the digital filter for -π ≤ ω < π rad/sample



1.3 Magnitude response in the passband



2. IIR BANDPASS DIGITAL FILTER USING BILINEAR TRANSFORMATION METHOD

2.1 Coefficients of the transfer function of the IIR filter

The transfer function of the IIR filter:

$$H(z) = \frac{a_0 z^{0} + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_N z^{-N}}{b_0 z^{0} + b_1 z^{-1} + b_2 z^{-2} \ldots + b_N z^{-N}}$$

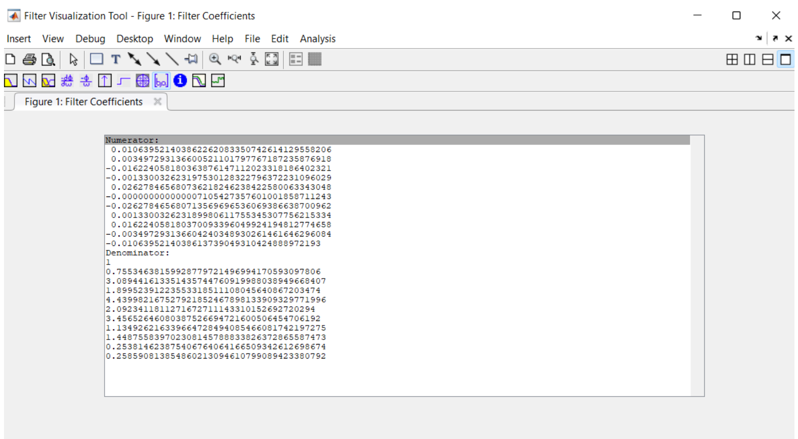
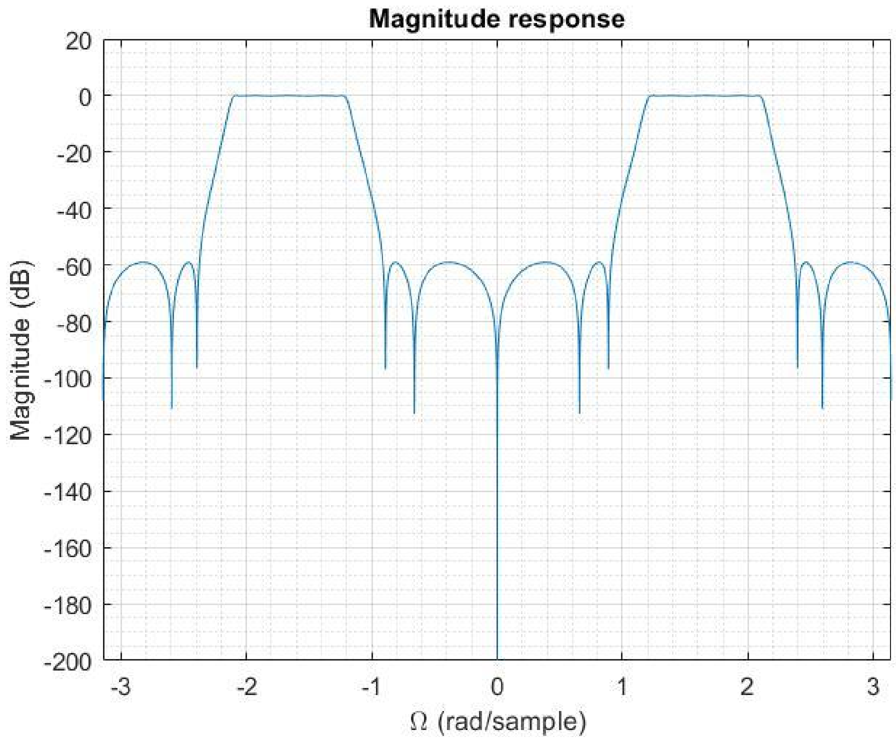


Figure: (taken from the fvtool in matlab)

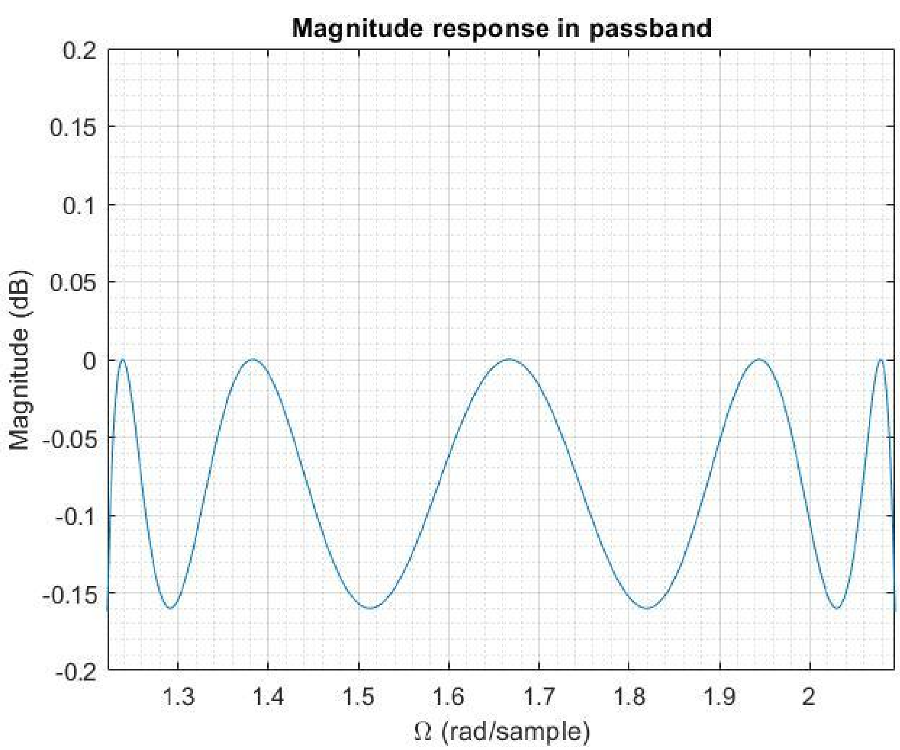
k	Numerator (ak)	Denominator (bk)
0	0.0106395214	1
1	0.003497293137	0.7553463816
2	-0.01622405818	3.089441613
3	-0.001330032623	1.899523912
4	0.02627846568	4.439982168
5	-0.00000000000000071054273 5760100185 8711243	2.092341181
6	-0.02627846568	3.456526461
7	0.001330032623	1.134926216
8	0.01622405818	1.44875584
9	-0.003497293137	0.2538146239
10	-0.0106395214	0.2585908139

Table 1: Numerator and denominator Coefficients of Z domain transfer function

2.2 Magnitude response of the digital filter for $-\pi \le \omega < \pi$ rad/sample



2.3 Magnitude response in the passband



COMPARING THE ORDER AND THE NUMBER OF MULTIPLICATIONS AND ADDITIONS REQUIRED TO PROCESS A SAMPLE BY THE DESIGNED FIR & IIR FILTERS.

Transfer Function

$$H(z) = \frac{\sum_{k=0}^{M} a_k z^{-k}}{1+\sum_{k=1}^{N} b_k z^{-k}} = \frac{a_0 z^0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_N z^{-M}}{1+b_1 z^{-1} + b_2 z^{-2} \ldots + b_N z^{-N}}$$

Order

- **Order of the FIR filter is 65 and the order of IIR filter is 5**.

Multiplications and Additions required to process a sample

- FIR is a non-recursive filter. Therefore, its outputs only depend on the given inputs. Output of the FIR filter is given as follows using the difference equation.

$$H(z) = \sum_{k=0}^{65} a_k z^{-k}$$

$$y[n] = \sum_{k=0}^{65} a_k x[n-k]$$

$$y[n] = \sum_{k=0}^{33} a_k x[n-k] + a_k x[n-(65-k)]$$

To process a sample using this FIR filter;
Assuming the symmetry of coefficients can be exploited to reduce the number of multiplications

Number of multiplications =33

Number of additions=65

- IIR is a recursive filter. Therefore, its output depends on both input values and output values. The difference equation of the IIR filter is,
As mentioned in Table 1, at $k = 0,\ b_k = 1$ & $z^0 = 1$. Therefore, denominator of the H[z] can be written as, $1 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} \ldots + b_N z^{-N}$.
In this transfer function, the degree of numerator and denominator should be equal in order to do bilinear transformation. (i.e M=N)

$$y[n] = \sum_{k=0}^{10} a_k x[n-k] - \sum_{k=1}^{10} b_k y[n-k]$$

$$H(z) = \frac{\sum_{k=0}^{10} a_k z^{-k}}{1 + \sum_{k=1}^{10} b_k z^{-k}}$$

Therefore, to process a sample from this IIR filter, 11 and 10 multiplications should be done in numerator and denominator respectively and 20 additions should be done in both numerator and denominator.

To process a sample using this IIR filter

Assuming the symmetry of coefficients can be exploited to reduce the number of multiplications

Number of multiplications = 21

Number of additions=20

APPENDIX

1. FIR BANDPASS FILTER

```
clc;
clear;
close all;

%Index number  = 200693D
A = 6;
B = 9;
C = 3;

%Filter Specifications
Ap = 0.1 + (0.01*A); %Maximum passband ripple in dB
Aa = 50 + B; %Minimum stopband attenuation in dB
w_p1 = (C*100) + 400; %Lower passband edge in rad/s
w_p2 = (C*100) + 900; %Upper passband edge in rad/s
w_a1 = (C*100) + 100; %Lower stopband edge in rad/s
w_a2 = (C*100) + 1100; %Upper stopband edge in rad/s
w_s = 2*((C*100) + 1500); %Sampling frequency in rad/s
Ts = 2*pi/w_s; %Sampling period in s

Bt = min((w_p1-w_a1),(w_a2-w_p2)); %Transition width in rad/s
w_c1 = w_p1 - Bt/2; %Lower cutoff frequency in rad/s
w_c2 = w_p2 + Bt/2; %Upper cutoff frequency in rad/s
```

```
%Determining Kaiser Window Parameters

%Calculating delta_p
dp = (10^(0.05*Ap)-1)/((10^(0.05*Ap)+1));
%Calculating delta_a
da = 10^(-0.05*Aa);
%Selecting the minimum delta
delta = min(dp,da);

%Actual passband ripple in dB
Ap = 20*log10((1+delta)/(1-delta));
%Actual stopband attenuation in dB
Aa = -20*log10(delta);

%Choosing alpha
if Aa <= 21
    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

%Choosing D
if Aa <= 21
    D = 0.9222;
else
    D = (Aa - 7.95)/14.36;
end

%Choosing the length of the filter
N = ceil((w_s*D/Bt)+1);
if rem(N,2) == 0
    N = N+1;
end

%Filter duration
range = 0:1:N-1;
M = (N-1)/2;
n = 0:1:M;

%Calculating beta
```

```
beta = alpha*sqrt(1-((2*n)/(N-1)).^2);

%No.of iterations
Iter =  100;

%Generating I(alpha)
I_alpha = 1;
for i = 1:Iter
    I_alpha = I_alpha + ((1/factorial(i))*((alpha/2)^i))^2;
end

%Generating I(beta)
I_beta = 1;
for i = 1:Iter
    I_beta = I_beta + ((1/factorial(i))*((beta/2).^i)).^2;
end

%defining the kaiser window
kaiser_win = I_beta/I_alpha;
kaiser_win = [fliplr(kaiser_win(2:end)) kaiser_win];

%assuming an idealized frequency response
hnT = zeros(1,N);
for i = 0:M
    if i == 0
        hnT(i+M+1) = (2/w_s)*(w_c2 - w_c1);
    else
        hnT(i+M+1) = (1/(i*pi))*(sin(w_c2*i*Ts) - sin(w_c1*i*Ts));
    end
end

hnT(1:M) = fliplr(hnT(M+2:end));

%Defining the filter
filter = kaiser_win.*hnT;

%Plotting the impulse response of the filter - time domain
stem(range, filter)
axis([0 N-1 -0.2 0.3])
grid on
xlabel('n')
ylabel('Amplitude h[n]')
title('Impulse Response - Time Domain')
```

```
%magnitude response of the digital filter for -π ≤ ω < π rad/sample
[H_dB,w] = freqz(filter);
H_dB = 20*log10(abs(H_dB));
figure;
plot([flip(-w); w], [flip(H_dB); H_dB])
xlabel('\Omega (rad/sample)')
ylabel('Magnitude (dB)')
title('Magnitude response')
axis([-pi pi -120 20])
grid on;

%magnitude response of pass band
figure;
plot(w/pi,db(H_dB))
xlim([w_p1*2/w_s w_p2*2/w_s])
xlabel('\Omega (rad/sample)')
ylabel('Magnitude (dB)')
title('Magnitude response in passband')
grid on;
```

2. FIR BANDPASS FILTER USING fvtool()

```
clc;
clear all;
close all;

%Index number  = 200693D
A = 6;
B = 9;
C = 3;

%Calculating filter specifications
Ap = 0.1+(0.01*A);            %Maximum passband ripple in dB
Aa = 50+B;                    %Minimum stopband attenuation in dB
w_p1 = (C*100)+400;           %Lower passband edge in rad/s
w_p2 = (C*100)+900;           %Upper passband edge in rad/s
w_s1 = (C*100)+100;           %Lower stopband edge in rad/s
w_s2 = (C*100)+1100;          %Upper stopband edge in rad/s
w_sm = 2*((C*100)+1500);      %Sampling frequency in rad/s
```

```
Ts = (2*pi)/w_sm;           %Sampling period in s
fs=1/Ts;                    %Sampling frequency in Hz

%Determining Kaiser Window Parameters
Bt = min((w_p1-w_s1), (w_s2 - w_p2));         %critical transition width
in rad/s
Oc1 = w_p1 - (Bt/2);                          %lower cutoff frequency in
rad/s
Oc2 = w_p2 + (Bt/2);                          %upper cutoff frequency in
rad/s
dp = (10^(0.05*Ap) - 1)/(10^(0.05*Ap) + 1);     %Calculating delta_p
da = 10^(-0.05*Aa);                           %Calculating delta_a

%Corresponding discrete frequencies
wp1 = w_p1 * Ts;
wp2 = w_p2 * Ts;
ws1 = w_s1 * Ts;
ws2 = w_s2 * Ts;

%Window coefficients
f_edges = [w_s1 w_p1 w_p2 w_s2] / (2 * pi);
amplitudes = [0 1 0];
deviations = [da dp da];
[N,Wn,beta,ftype] = kaiserord(f_edges, amplitudes, deviations, fs);

%Kaiser window
kaiser_window = kaiser(N+1, beta);  % n+1 is no of points in window

%FIR filter
h = fir1(N, Wn, ftype, kaiser_window, 'noscale');  %Impulse response
fvtool(h);

% Magnitude and phase response
[H,w] = freqz(h, 1, 2001);
Hdb = 20 * log10(abs(H));
grid on;
grid minor;

figure;
plot([flip(-w); w], [flip(Hdb); Hdb])
xlabel('\Omega (rad/sample)')
ylabel('Magnitude (dB)')
title('Magnitude response')
```

```
ax = gca;
ax.YLim = [-120 20];
ax.XLim = [-pi pi];
grid on;
grid minor;

figure;
plot(w, Hdb);
xlabel('\Omega (rad/sample)')
ylabel('Magnitude (dB)')
title('Magnitude response in passband')
ax = gca;
ax.YLim = [-0.05 0.05];
ax.XLim = [wp1 wp2];
grid on;
grid minor;
```

3. IIR BANDPASS FILTER ELLIPTIC APPROXIMATION

```
clc;
clear all;
close all;

%Index number  = 200693D
A = 6;
B = 9;
C = 3;

% Design Specifications
Ap  =   0.1+(0.01* A);          % Maximum passband ripple
Aa  =   50 + B;                 % Minimum stopband attenuation
w_p1 =   (C * 100) + 400;        % Lower passband edge
w_p2 =   (C * 100) + 900;        % Upper passband edge
w_s1 =   (C * 100) + 100;        % Lower stopband edge
w_s2 =   (C * 100) + 1100;        % Upper stopband edge
w_sm  =    2*(( C * 100) + 1500) ; % Sampling frequency

Ts=2*pi/w_sm; %sampling rate
fs=1/Ts;
```

```
wp=[w_p1 w_p2]; %passband range
w_sm=[w_s1 w_s2];   %stopband range

%warping
wp_warped=2*tan(wp*Ts/2)/Ts;
ws_warped=2*tan(w_sm*Ts/2)/Ts;

%calculating order and passband range
[n,Wp] = ellipord(wp_warped, ws_warped , Ap, Aa,"s");
[num,den] = ellip(n,Ap,Aa,Wp,"bandpass","s");

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

%fvtool for coefficients of the transfer function of the IIR filter
fvtool(dnum,dden);

[H,w]=freqz(dnum,dden,4001);
H_db=20*log10(abs(H));

%magnitude response of the digital filter for π ≤ ω < π rad/sample
figure;
plot([flip(-w); w], [flip(H_db); H_db])
xlabel('\Omega (rad/sample)')
ylabel('Magnitude (dB)')
title('Magnitude response')
ax = gca;
ax.YLim = [-200 20];
ax.XLim = [-pi pi];
grid on;
grid minor;

%magnitude response of passband
figure;
plot(w, H_db);
xlabel('\Omega (rad/sample)')
ylabel('Magnitude (dB)')
title('Magnitude response in passband')
ax = gca;
ax.YLim = [-0.2 0.2];
ax.XLim = [w_p1*Ts w_p2*Ts];
grid on;
grid minor;
```