EXPT NO: 8 DESIGN AND ANALYSIS OF IIR FILTERS - II

AIM:

i) To design a digital Butterworth IIR band pass filter using bilinear transformation for the given specification: lower stop band edge frequency = 0.1π rad; lower pass band edge frequency = 0.4π rad; upper stop band edge frequency = 0.9π rad; upper pass band edge frequency = 0.6π rad, Ap = 3 dB and A_s = 18 dB with F_s = 500Hz.

ii) To design a digital Chebyshev IIR band stop filter using impulse invariant transformation for the given specification: lower stop band edge frequency = 0.4π rad; lower pass band edge frequency = 0.1π rad; upper stop band edge frequency = 0.6π rad; upper pass band edge frequency = 0.9π rad, Ap = 3 dB and A_s = 18 dB with F_s = 500Hz.

HARDWARE REQUIREMENTS:

Computer Desktop

SOFTWARE REQUIREMENTS:

MATLAB 7 or later version

THEORY:

A digital filter is a linear time invariant discrete time system. The digital filters are classified into two, based on their lengths of impulse response

1. Finite Impulse response (FIR)

   They are of non-recursive type and h [n] has finite number of samples

2. Infinite Impulse response (IIR)

   h[n] has finite number of samples. They are of recursive type. Hence, their transfer function is of the form

$$H(z) = \sum_{n=0}^{\alpha} h(n) z^{-n}$$

$$H(Z) = \frac{\sum_{K=0}^{M-1} b_k Z^{-k}}{1 + \sum_{j=1}^{N-1} a_j Z^{-j}}$$

The digital filters are designed from analog filters. The two widely used methods for digitizing the analog filters include

1. Bilinear transformation
2. Impulse Invariant transformation

The bilinear transformation maps the s-plane into the z-plane by

$$H(Z) = H(S)|_{S} = \frac{2}{T} \times \frac{1 - Z^{-1}}{1 + Z^{-1}}$$

This transformation maps the jΩ axis (from Ω = -∞ to +∞) repeatedly around the unit circle (exp (jw), from w=-π to π) by

$$\Omega = \frac{2}{T} \tan\left(\frac{\omega}{2}\right)$$

Bandpass is realized by cascading high pass and low pass filter. Similarly, band stop filter is realized by cascading low pass followed by high pass filter. In general, the normalized low pass filter can be transformed to any other filter.

For each pole of the low-pass filter, two poles result for the bandpass filter. Consequently, the order of complexity based on poles is 2N for the bandpass filter.

$\omega_p = [\omega_{p1}\ \omega_{p2}]$

$\omega_s = [\omega_{s1}\ \omega_{s2}]$

If ω_n is a two-element vector, $\omega_n = [\omega_1\ \omega_2]$, BUTTER returns an order 2N bandpass filter with passband $\omega_1 < \omega_n < \omega_2$.

BILINEAR TRANSFORMATION:

DESIGN STEPS:

1. From the given specifications, find pre-warped analog frequencies using

$$\Omega = \frac{2}{T}\tan\left(\frac{\omega}{2}\right)$$

2. Using the analog frequencies, find H(s) of the analog filter
3. Substitute $S = \frac{2}{T} \times \frac{1-Z^{-1}}{1+Z^{-1}}$ in the H(s) of Step:2

IMPULSE INVARIANT TRANSFORMATION:

DESIGN STEPS:

1. Find the analog frequency using $\Omega = {}^{\omega}/_{T}$
2. Find the transfer function of analog filter $H_a(s)$
3. Express the analog filter transfer function as a sum of single pole filters
4. Compute H(Z) of digital filter using the formula

$$H(Z) = \sum_{k=1}^{N} \frac{C_k}{1 - e^{-TP_k Z^{-1}}}$$

BUTTERWORTH FILTER:

Low pass Analog Butterworth filters are all pole filters characterised by magnitude frequency response by

$$|H(j\omega)|^2 = \frac{1}{1 + \left(\frac{\Omega}{\Omega c}\right)^{2N}}$$

where N is the order of the filter and Ωc is the cut-off frequency.

As $N \rightarrow \alpha$, the low pass filter is said to be normalized. All types of filters namely-Low pass, High pass, Band pass and Band elimination filters can be derived from the Normalized Low pass filter.

Chebyshev Filters:

There are two types of Chebyshev filters.

Type I are all-pole filters that exhibit equi-ripple behaviour in pass band and monotonic characteristics in stop band.

Type II are having both poles and zeros and exhibit monotonic behavior in pass band and equi-ripple behaviour in stop band. The zero lies on the imaginary axis.

The magnitude-squared function is given as

$$\left|H(j\omega)\right|^2 = \frac{1}{1+\varepsilon^2 C_N^2(\Omega/\Omega p)}$$

ε is the ripple parameter in pass band

$C_N(x)$ is the N^{th} order Chebyshev polynomial defined as

$$C_N(x) = \begin{cases} Cos(N\cosh^{-1}x), & |x| \leq 1 \\ Cos(N\cosh^{-1}x), & |x| > 1 \end{cases}$$

LIBRARY FUNCTIONS

- **butter: Butterworth** digital and analog filter design.

[B, A] = butter (N,Wn) designs an Nth order Low pass digital Butterworth filter and returns the filter coefficient vectors B (numerator) and A (denominator) in length N+1. The coefficients are listed in descending powers of z. The cut-off frequency Wn must be in the range 0.0 < Wn < 1.0, with 1.0 corresponding to half the sample rate. butter (N, Wn,'s'), butter (N, ω_n, ‘bandpass’, ‘s’) designs an Nth order Butterworth analog bandpass filter.

butter (N, ω_n, ‘stop','s’) designs an Nth order Butterworth analog band stop filter. In this case, Wn is in [rad/s] and it can be greater than 1.0.

- **buttord**: Butterworth filter order selection.

[N, Wn] = buttord (Wp, Ws, Rp, Rs) returns the order N of the lowest order digital Butterworth filter that loses no more than Rp dB in the pass band and has at least Rs dB of attenuation in the stop band. Wp and Ws are the pass band and stop band edge frequencies, normalized from 0 to 1 (where 1 corresponds to pi radians/sample).

buttord: also returns Wn, the Butterworth natural frequency (or) the "3 dB frequency" to be used with BUTTER to achieve the specifications.

[N, Wn] = buttord (Wp, Ws, Rp, Rs, 's') does the computation for an analog filter, in which case Wp and Ws are in radians/second. When Rp is chosen as 3 dB, the Wn in BUTTER is equal to Wp in BUTTORD.

- **angle:** Phase angle.

Theta=angle (H) returns the phase angles, in radians, of a matrix with complex elements.

- **freqs:** Laplace-transform (s-domain) frequency response.

H = freqs(B, A,W) returns the complex frequency response vector H of the filter B/A:

Given the numerator and denominator coefficients in vectors B and A. The frequency response is evaluated at the points specified in vector W (in rad/s). The magnitude and phase can be graphed by calling freqs(B, A,W) with no output arguments.

- **cheb1ord: Chebyshev** Type I filter order selection.

[N, Wn] = cheb1ord (Wp, Ws, Rp, Rs) returns the order N of the lowest order digital Chebyshev Type I filter that loses no more than Rp dB in the pass band and has at least Rs dB of attenuation in the stop band. Wp and Ws are the pass band and stop band edge frequencies, normalized from 0 to 1 (where 1 corresponds to pi radians/sample).

cheb1ord also returns Wn, the Chebyshev natural frequency to use with cheby1 to achieve the specifications.

[N, Wn] = cheb1ord (Wp, Ws, Rp, Rs, 's') does the computation for an analog filter, in which case Wp and Ws are in radians/second.

- **cheby1** Chebyshev Type I digital and analog filter design.

[B, A] = cheby1 (N,R,Wn) designs an Nth order Low pass digital Chebyshev filter with R decibels of peak-to-peak ripple in the pass band. cheby1 returns the filter coefficient vectors B (numerator) and A (denominator) of length N+1. The cut-off frequency Wn must be in the range 0.0 < Wn < 1.0, with 1.0 corresponding to half the sample rate. Use R=0.5 as a starting point, if you are unsure about choosing R. cheby1 (N,R,Wn,'s'), cheby1 (N,R,Wn,'bandpass','s'), cheby1 (N,R,Wn,'stop,'s') design analog Chebyshev Type I bandpass and bandstop filters. In this case, Wn is in [rad/s] and it can be greater than 1.0.

- **Impinvar:** Impulse Invariant method for analog-to-digital filter conversion [bz,az] = impinvar(b,a,fs) creates a digital filter with numerator and denominator coefficients bz and az, respectively, whose impulse response is equal to the impulse response of the analog filter with coefficients b and a, scaled by 1/fs. If you leave out the argument fs (or) specify fs as an empty vector [ ], it takes the default value of 1 Hz.
- **Bilinear:** Bilinear transformation method for analog-to-digital filter conversion. The bilinear transformation is a mathematical mapping of variables. In digital filtering, it is a standard method of mapping the s or analog plane into the z or digital plane. It transforms analog filters, designed using classical filter design

ALGORITHM

1. Get the passband ripple and stopband attenuation.
2. Get the lower and upper passband and lower and upper stopband edge frequencies.
3. Get the sampling period, $T_{s.}$
4. Specify ω_p and ω_s as a two-element vector.
5. Compute the corresponding analog frequency
6. Compute the order and cut-off frequency of Butterworth or Chebyshev filters for the given specifications.
7. Design the analog filter.
8. Using Impulse Invariant /Bilinear transformation obtain the digital filter.
9. Display the transfer function. Plot the magnitude response and phase response.

QUESTION:

1. **To design a digital Butterworth IIR bandpass filter using bilinear transformation for the given specification:**

Lower stop band edge frequency = 0.1π rad, Lower pass band edge frequency = 0.4π rad

Upper stop band edge frequency = 0.9π rad, Upper pass band edge frequency = 0.6π rad

Ap = 3 dB and A_s = 18 dB with F_s = 500Hz.

PROGRAM:

```
%IIR- BUTTERWORTH BANDPASS FILTER USING BILINEAR
TRANSFORMATION
close all;
clc;

fprintf("\t\t\t\t\t\t\t\t<strong>IIR- BUTTERWORTH BANDPASS
FILTER USING BILINEAR TRANSFORMATION </strong>\n");
Ap=input('Enter pass band ripple in dB,Ap : ');  %Get the
input
As=input('Enter stop band attenuation in dB,As : ');
Wpl=input('Enter lower passband edge frequency in rad : ');
Wpu=input('Enter upper passband edge frequency in rad : ');
Wsl=input('Enter lower stopband edge frequency in rad : ');
Wsu=input('Enter upper stopband edge frequency in rad : ');
Fs=input('Enter the sampling frequency in Hz,Fs : ');

Wp=[Wpl Wpu]; %Vector of passband edge frequencies
Ws=[Wsl Wsu]; %Vector of stopband edge frequencies

omgp=(2*Fs)*tan(Wp/2); %Compute pre-warped analog frequency
omgs=(2*Fs)*tan(Ws/2);

%Butterworth IIR BPF Filter
[N,Wn]=buttord(omgp,omgs,Ap,As,'s'); %Compute Cutoff edge
frequencies and Order
fprintf("\n");
disp("Butterworth IIR BPF Filter");
fprintf("Order, N = ");disp(N);
fprintf("Cutoff edge frequencies, Wn = ");disp(Wn);

%Design a analog Butterworth BPF of Order N and Cutoff edge
frequencies Wn
[num,den]=butter(N,Wn,'bandpass','s');
```

```
%Covert analog filter into digital using Bilinear
Transformation
[B,A]=bilinear(num,den,Fs);
W=0:0.01:pi;
[h,ph]=freqz(B,A,W,'whole'); %Obtain the frequency response
m=abs(h);
an=angle(h);

subplot(2,1,1); %Plot the response graphs
plot(ph/pi,20*log(m));
title('Butterworth IIR BPF Filter - Magnitude Response');
xlabel('Normalised frequency');
ylabel('Gain in dB');
subplot(2,1,2);
plot(ph/pi,an);
title('Butterworth IIR BPF Filter - Phase Response');
xlabel('Normalised frequency');
ylabel('Angle in Radians');
```

OUTPUT:

2. **To design a digital Chebyshev IIR bandstop filter using impulse invariant transformation for the given specification:**

Lower stop band edge frequency = 0.4π rad, Lower pass band edge frequency = 0.1π rad

Upper stop band edge frequency = 0.6π rad, Upper pass band edge frequency = 0.9π rad

Ap = 3 dB and A_s = 18 dB with F_s = 500Hz.

PROGRAM:

```
%IIR- CHEBYSHEV BANDSTOP FILTER USING IMPULSE INVARIANT
TRANSFORMATION
close all;
clc;

fprintf("\t\t\t\t\t\t\t\t<strong>IIR- CHEBYSHEV BANDSTOP
FILTER USING IMPULSE INVARIANT TRANSFORMATION</strong>\n");
Ap=input('Enter pass band ripple in dB,Ap : ');  %Get the
input
As=input('Enter stop band attenuation in dB,As : ');
Wpl=input('Enter lower passband edge frequency in rad : ');
Wpu=input('Enter upper passband edge frequency in rad : ');
Wsl=input('Enter lower stopband edge frequency in rad : ');
Wsu=input('Enter upper stopband edge frequency in rad : ');
Fs=input('Enter the sampling time period in Hz,Fs : ');

Wp=[Wpl Wpu]; %Vector of passband edge frequencies
Ws=[Wsl Wsu]; %Vector of stopband edge frequencies

omgp=(Wp*Fs); %Compute pre-warped analog frequency
omgs=(Ws*Fs);

%Chebyshev IIR BSF Filter
[N,Wn] = cheb1ord(omgp,omgs,Ap,As,'s'); %Compute Cutoff edge
frequencies and Order
fprintf("\n");
disp("Chebyshev IIR BsF Filter");
fprintf("Order, N = ");disp(N);
fprintf("Cutoff edge frequencies, Wn = ");disp(Wn);

%Design a analog Chebyshev BSF of Order N and Cutoff
frequency Wn
[num,den] = cheby1(N,Ap,Wn,'stop','s');
[B,A]=impinvar(num,den,Fs);
%Covert analog filter into digital using Impulse Invariant
Transformation
```

```
W=0:0.001:pi;
[h,ph]=freqz(B,A,W,'whole'); %Obtain the frequency response
m=abs(h);
an=angle(h);

subplot(2,1,1); %Plot the response graphs
plot(ph/pi,20*log(m));
title('Chebyshev IIR BSF Filter - Magnitude Response');
xlabel('Normalised frequency');
ylabel('Gain in dB');
subplot(2,1,2);
plot(ph/pi,an);
title('Chebyshev IIR BSF Filter - Phase Response');
xlabel('Normalised frequency');
ylabel('Angle in Radians');
```

OUTPUT:

VIVA QUESTIONS

1. **Mention the steps to covert Analog filter into Digital Filter.**

   Analog filter is converted to Digital filter by using the following steps:

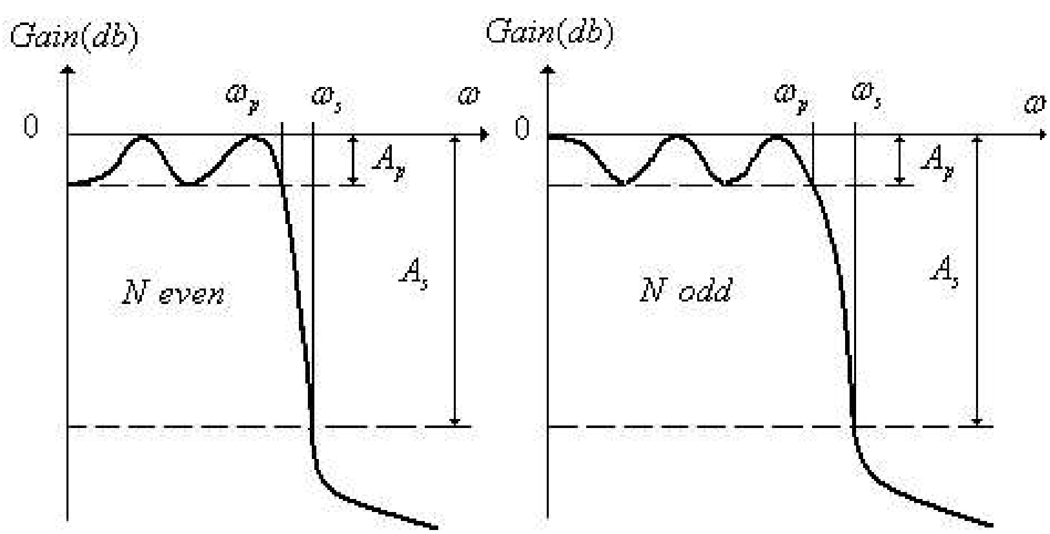
   a. Design a prototype analog filter with transfer function F(jω).
   b. Make a partial fraction expansion of F(jω) to obtain the NN values for K_i and s_i.
   c. Form the digital transfer function H(z) from to give the desired design.

   The two main methods used to convert an analog filter to digital filter are,

   Impulse-Invariant Transformation - This method results in a digital filter with an impulse response exactly equal to samples of the prototype analog filter.

   Bilinear Transformation - This method uses a frequency mapping to convert the analog filter to a digital filter.

2. **Draw typical responses of Chebyshev filter, when the order is (i) odd and (ii) even.**



3. Compare impulse invariant and bilinear technique.

S.NO	Impulse Invariance Method	Bilinear Transform Method
1.	It is a many to one mapping technique.	It is a one to one mapping technique.
2.	Frequency relation $\Omega = \omega / T_s$	Frequency relation $\Omega = \frac{2}{T} \tan(\omega/2)$
3.	No frequency warping exists.	Frequency warping exists.
4.	Frequency aliasing exists.	No Frequency aliasing.
5.	Best suitable for LPF and BPF design.	Suitable for all types of filter.
6.	It’s a linear mapping.	It’s a non-linear mapping.
7.	Pre warping is not required.	Pre warping is required.

4. Is IIR a linear phase filter?

No, IIR filters usually have a **highly non-linear phase response**, especially around the filter’s cut-off frequencies and it’s not possible to design an IIR to have linear phase. However, an IIR filter’s passband phase can be modified in order to achieve linear phase using **all-pass equalisation filters**.

5. **Why IIR filter is referred to as a recursive filter?**

   Digital filters with an infinite-duration impulse response are called Infinite – duration Impulse Response (IIR) filter. It is necessary to have a feedback loop and it's necessary in the implementation of the filter. Hence, Infinite Impulse Response (IIR) filter is also called a recursive filter or, sometimes, an autoregressive moving-average filter (ARMA).

RESULT:

Butterworth and Chebyshev IIR bandpass and band stop filters were designed using Bilinear and Impulse Invariant transformations, and manually verified the order, cut off frequency and filter co-efficient of the filters.

APPENDIX:

1. **BUTTERWORTH BPF USING IMPULSE INVARIANT TRANFORMATION METHOD**

PROGRAM:

```
%IIR- BUTTERWORTH BANDPASS FILTER USING IMPULSE INVARIANT
TRANSFORMATION
close all;
clc;

fprintf("\t\t\t\t\t\t\t\t<strong>IIR- BUTTERWORTH BANDPASS
FILTER USING IMPULSE INVARIANT TRANSFORMATION </strong>\n");
Ap=input('Enter pass band ripple in dB,Ap : ');  %Get the
input
As=input('Enter stop band attenuation in dB,As : ');
Wpl=input('Enter lower passband edge frequency in rad : ');
Wpu=input('Enter upper passband edge frequency in rad : ');
Wsl=input('Enter lower stopband edge frequency in rad : ');
Wsu=input('Enter upper stopband edge frequency in rad : ');
Fs=input('Enter the sampling frequency in Hz,Fs : ');

Wp=[Wpl Wpu]; %Vector of passband edge frequencies
Ws=[Wsl Wsu]; %Vector of stopband edge frequencies

omgp=(Wp*Fs); %Compute pre-warped analog frequency
omgs=(Ws*Fs);

%Butterworth IIR BPF Filter
[N,Wn]=buttord(omgp,omgs,Ap,As,'s'); %Compute Cutoff edge
frequencies and Order
fprintf("\n");
disp("Butterworth IIR BPF Filter");
fprintf("Order, N = ");disp(N);
fprintf("Cutoff edge frequencies, Wn = ");disp(Wn);

%Design a analog Butterworth BPF of Order N and Cutoff edge
frequencies Wn
[num,den]=butter(N,Wn,'bandpass','s');
%Covert analog filter into digital using Impulse Invariant
Transformation
[B,A]=impinvar(num,den,Fs);
W=0:0.01:pi;
[h,ph]=freqz(B,A,W,'whole'); %Obtain the frequency response
m=abs(h);
```

```
an=angle(h);

subplot(2,1,1); %Plot the response graphs
plot(ph/pi,20*log(m));
title('Butterworth IIR BPF Filter - Magnitude Response');
xlabel('Normalised frequency');
ylabel('Gain in dB');
subplot(2,1,2);
plot(ph/pi,an);
title('Butterworth IIR BPF Filter - Phase Response');
xlabel('Normalised frequency');
ylabel('Angle in Radians');
```

OUTPUT:

2. CHEBYSHEV BSF USING BILINEAR TRANSFORMATION METHOD

PROGRAM:

```
%IIR- CHEBYSHEV BANDSTOP FILTER USING BILINEAR TRANSFORMATION
close all;
clc;

fprintf("\t\t\t\t\t\t\t\t<strong>IIR- CHEBYSHEV BANDSTOP
FILTER USING BILINEAR TRANSFORMATION</strong>\n");
Ap=input('Enter pass band ripple in dB,Ap : ');  %Get the
input
As=input('Enter stop band attenuation in dB,As : ');
Wpl=input('Enter lower passband edge frequency in rad : ');
Wpu=input('Enter upper passband edge frequency in rad : ');
Wsl=input('Enter lower stopband edge frequency in rad : ');
Wsu=input('Enter upper stopband edge frequency in rad : ');
Fs=input('Enter the sampling time period in Hz,Fs : ');

Wp=[Wpl Wpu]; %Vector of passband edge frequencies
Ws=[Wsl Wsu]; %Vector of stopband edge frequencies

omgp=(2*Fs)*tan(Wp/2); %Compute pre-warped analog frequency
omgs=(2*Fs)*tan(Ws/2);

%Chebyshev IIR BSF Filter
[N,Wn] = cheb1ord(omgp,omgs,Ap,As,'s'); %Compute Cutoff edge
frequencies and Order
fprintf("\n");
disp("Chebyshev IIR BsF Filter");
fprintf("Order, N = ");disp(N);
fprintf("Cutoff edge frequencies, Wn = ");disp(Wn);

%Design a analog Chebyshev BSF of Order N and Cutoff
frequency Wn
[num,den] = cheby1(N,Ap,Wn,'stop','s');
[B,A]=bilinear(num,den,Fs);
%Covert analog filter into digital using Bilinear
Transformation
W=0:0.01:pi;
[h,ph]=freqz(B,A,W,'whole'); %Obtain the frequency response
m=abs(h);
an=angle(h);

subplot(2,1,1); %Plot the response graphs
plot(ph/pi,20*log(m));
title('Chebyshev IIR BSF Filter - Magnitude Response');
```

```
xlabel('Normalised frequency');
ylabel('Gain in dB');
subplot(2,1,2);
plot(ph/pi,an);
title('Chebyshev IIR BSF Filter - Phase Response');
xlabel('Normalised frequency');
ylabel('Angle in Radians');
```

OUTPUT: