Expt. No. 5. DESIGN OF ANALOG IIR FILTERs

Expt. No. 5a. DESIGN OF ANALOG IIR BUTTERWORTH LOW PASS FILTER

AIM:

To write a program in MATLAB to plot the magnitude and phase response of analog IIR Butterworth low pass filter.

SOFTWARE REQUIRED:

MATLAB Software

ALGORITHM:

1. Clear the command window.
2. Get the passband edge frequency and stopband edge frequency
3. Get sampling frequency greater than the nyquist frequency(fs_sf)
4. Get the attenuation values for passband and stop band.
5. Compute the digital frequency 'ω_p' and 'ω_s'by using the formula ω=2*pi*f/fs_sf 6. Compute the order of the analog filter using 'buttord' function with 's' option 7. Compute IIR analog lowpass filter coefficients using 'butter' command. 8. Compute frequency response of analog filter using 'freqs' command for different values of ω
9. Compute the magnitude using abs command.
10. Compute the phase using angle command.
11. Plot the magnitude and phase response.

FLOWCHART:

Start

Get the passband edge frequency and stopband edge frequency

Get sampling frequency greater than the nyquist frequency (fs_sf)

Get the attenuation values for passband and stop band

Compute the digital frequency 'ω_p' and 'ω_s'by using the formula ω=2*pi*f/fs_sf

Compute the order of the analog filter using 'buttord' function with 's' option

Compute IIR analog lowpass filter coefficients using 'butter' command

Compute frequency response of analog filter using 'freqs' command for different values of ω

Compute the magnitude using abs command

Compute the phase using angle command

Plot the magnitude and phase response

Stop

PROGRAM:

```
%Program for Butterworth IIR Lowpass analog filter
clc;
close all;
clear all;
fprintf('Program for Butterworth IIR Lowpass analog filter\n\n');

%We get the passedge, stopedge and sampling frequencies in
Hz fp=input('Enter the pass edge frequency: ');
fs=input('Enter the stop edge frequency: ');

%fs_min should be twice the maximum frequency. Here, fs_min =
2*fs. fs_min = 2*fs;
fprintf('\nEnter the sampling frequency greater than
%d\n',fs_min'); fs_sf=input('Enter the sampling frequency: ');

%We get the attenuation in dB. rp will be around 0 to 3
dB %rs will be around 30 to 50 dB
rp = input('\nEnter the passband ripple in dB: ');
rs = input('Enter the stopband attenuation in dB: ');
rp1=20*log10(rp)
rs1=20*log10(rs)


%We need to normalise wp,ws to pi. It is similar to finding the digital
%frequency; digital omega = analog omega* Ts = analog
omega/fs_sf wp=2*pi*fp/fs_sf;
ws=2*pi*fs/fs_sf;
%The normalised frequencies are wp and ws
fprintf('\nwp is %d\n',wp);
fprintf('ws is %d\n',ws);

%Computing the order(N) and cutoff frequency(wc) using wp,ws,rp,rs
using %the buttord command with 's' option for analog filter.
%Finding the coefficients of filter [b a] using butter command with 's'
```

```
%option [N wc]=buttord(wp,ws,rp1,rs1,'s');
[b a]=butter(N,wc,'s');

%Computing the frequency response using freqs command
%Computing h for specific values of w i.e. 0, pi/100, 2*pi/100... till
pi %and storing the corresponding the w values
w=0:(pi/100):pi;
[h omega] = freqs(b,a,w);
hf=tf(b,a)
disp(hf)
[z p]=tf2zp(b,a)


%Finding the magnitude response. Note: log10 should be
used. %Plotting magnitude versus omega.
mag_h=abs(h);

figure(1);
subplot(3,1,1);
plot(omega/pi,mag_h);

xlabel('frequency normalised to 1 -->');
ylabel('Gain in dB-->');
title('Magnitude Response of LPF');

%Finding the phase response.
%Plotting phase versus omega.
angle_h = angle(h);
subplot(3,1,2);
plot(omega/pi,angle_h)
xlabel('frequency normalised to 1 -->');
ylabel('Phase in radian-->');
title('Phase Response of LPF');

subplot(3,1,3)
zplane(z,p,'k')
title('Pole Zero Plot of LPF')
```

OUTPUT:

Enter the pass edge frequency: 1500
Enter the stop edge frequency: 3000
Enter the sampling frequency greater than 6000
Enter the sampling frequency: 7000
Enter the passband ripple in dB: 0.15
Enter the stopband attenuation in dB: 60

RESULT:

Thus, the magnitude and phase response of the analog IIR Butterworth low pass filter is plotted using MATLAB.

Expt. No. 5b. DESIGN OF ANALOG IIR BUTTERWORTH HIGH PASS FILTER

AIM:
To write a program in MATLAB to plot the magnitude and phase response of analog IIR Butterworth high pass filter.

SOFTWARE REQUIRED:
MATLAB Software

ALGORITHM:
1. Clear the command window.
2. Get the passband edge frequency and stopband edge frequency
3. Get sampling frequency greater than the nyquist frequency(fs_sf)
4. Get the attenuation values for passband and stop band.
5. Compute the digital frequency 'ω_p' and 'ω_s'by using the formula ω=2*pi*f/fs_sf 6. Compute the order of the analog filter using 'buttord' function with 's' option 7. Compute IIR analog highpass filter coefficients using 'butter' command with 'high' option. 8. Compute frequency response of analog filter using 'freqs' command for different values of ω
9. Compute the magnitude using abs command.
10. Compute the phase using angle command.
11. Plot the magnitude and phase response.

FLOWCHART:

Start

Get the stopband edge frequency and passband edge frequency

Get sampling frequency greater than the nyquist frequency (fs_sf)

Get the attenuation values for passband and stop band

Compute the digital frequency 'ω_p' and 'ω_s'by using the formula ω=2*pi*f/fs_sf

Compute the order of the analog filter using 'buttord' function with 's' option

Compute IIR analog highpass filter coefficients using 'butter' command with 'high' option

Compute frequency response of analog filter using 'freqs' command for different values of ω

Compute the magnitude using abs command

Compute the phase using angle command

Plot the magnitude and phase response

Stop

PROGRAM:

```
%Program for Butterworth IIR Highpass analog filter
clc;
close all;
clear all;
fprintf('Program for Butterworth IIR Highpass analog filter\n\n');

%We get the passedge, stopedge and sampling frequencies in
Hz fs=input('Enter the stop edge frequency: ');
fp=input('Enter the pass edge frequency: ');

%fs_min should be twice the maximum frequency. Here, fs_min =
2*fp. fs_min = 2*fp;
fprintf('\nEnter the sampling frequency greater than
%d\n',fs_min'); fs_sf=input('Enter the sampling frequency: ');

%We get the attenuation in dB. rp will be around 0 to 3 dB
%rs will be around 30 to 50 dB
rp = input('\nEnter the passband ripple in dB: ');
rs = input('Enter the stopband attenuation in dB: ');
rp1=20*log10(rp)
rs1=20*log10(rs)

%We need to normalise wp,ws to pi. It is similar to finding the digital
%frequency digital omega = analog omega* Ts = analog
omega/fs_sf wp=2*pi*fp/fs_sf;
ws=2*pi*fs/fs_sf;
%The normalised frequencies are wp and ws
fprintf('\nws is %d\n',ws);
fprintf('wp is %d\n',wp);


%Computing the order(N) and cutoff frequency(wc) using wp,ws,rp,rs
using %the buttord command with 's' option for analog filter.
%Finding the coefficients of filter [b a] using butter command with 'high' and 's'
option. [N wc]=buttord(wp,ws,rp1,rs1,'s');
[b a]=butter(N,wc,'high','s');

%Computing the frequency response using freqs command
%Computing h for specific values of w i.e. 0, pi/100, 2*pi/100... till
pi %and storing the corresponding the w values
```

```
w=0:(pi/100):pi;
[h omega] = freqs(b,a,w);
hf=tf(b,a)
disp(hf)
[z p]=tf2zp(b,a)


%Finding the magnitude response. Note: log10 should be used.
%Plotting magnitude versus omega.
mag_h=abs(h);
figure(1);
subplot(3,1,1);
plot(omega/pi,mag_h);

xlabel('frequency normalised to 1 -->');
ylabel('Gain in dB-->');
title('Magnitude Response of HPF');

%Finding the phase response.
%Plotting phase versus omega.
angle_h = angle(h);
subplot(3,1,2);
plot(omega/pi,angle_h)
xlabel('frequency normalised to 1 -->');
ylabel('Phase in radian-->');
title('Phase Response of HPF');

subplot(3,1,3)
zplane(z,p,'k')
title('Pole Zero Plot of HPF')
```

OUTPUT:

Enter the stop edge frequency: 1500
Enter the pass edge frequency: 3000
Enter the sampling frequency greater than 6000
Enter the sampling frequency: 7000
Enter the passband ripple in dB: 0.15
Enter the stopband attenuation in dB: 60

RESULT:

Thus, the magnitude and phase response of the analog IIR Butterworth high pass filter is plotted using MATLAB.

Expt. No. 5c. DESIGN OF ANALOG IIR CHEBYSHEV BAND PASS FILTER

AIM:

To write a program in MATLAB to plot the magnitude and phase response of analog IIR Chebyshev band pass filter.

SOFTWARE REQUIRED:

MATLAB Software

ALGORITHM:

1. Clear the command window.

2. Get the first stopband edge frequency (fs1), first passband edge frequency (fp1), second passband edge frequency (fp2) and second stopband edge frequency(fs2). 3. Get sampling frequency greater than the nyquist frequency
4. Get the attenuation values for passband and stop band.
5. Compute the digital frequency 'ω_p' and 'ω_s'by using the formula ω=2*pi*f/fs_sf
6. Group the passband and stopband edge frequencies as
   wp = [wp1 wp2] and ws = [ws1 ws2]
7. Compute the order of the analog filter using 'cheb1ord' function with 's' option 8. Compute IIR analog bandpass filter coefficients using 'cheby1' command. 9. Compute frequency response of analog filter using 'freqs' command for different values of ω
10. Compute the magnitude using abs command.
11. Compute the phase using angle command.
12. Plot the magnitude and phase response.

FLOWCHART:

Start

Get first stopband edge frequency (fs1), first passband edge frequency (fp1), second passband edge frequency (fp2) and second stopband edge frequency (fs2)

Get sampling frequency greater than the nyquist frequency (fs_sf)

Get the attenuation values for passband and stop band

Compute the digital frequency 'ω_p' and 'ω_s'by using the formula ω=2*pi*f/fs_sf

Group the passband and stopband edge frequencies as wp = [wp1 wp2] and ws = [ws1 ws2]

Compute the order of the analog filter using 'cheb1ord' function with 's' option

Compute IIR analog bandpass filter coefficients using 'cheby1' command

Compute frequency response of analog filter using 'freqs' command for different values of ω

Compute the magnitude using abs command

Compute the phase using angle command

Plot the magnitude and phase response

Stop

PROGRAM:

```
%Program for Chebtshev IIR Bandpass analog filter
clc;
close all;
clear all;
fprintf('Program for Butterworth IIR Bandpass analog filter\n\n');

%We get the passedge, stopedge and sampling frequencies in
Hz fs1=input('Enter the stop edge frequency1: ');
fp1=input('Enter the pass edge frequency1: ');
fp2=input('Enter the pass edge frequency2: ');
fs2=input('Enter the stop edge frequency2: ');

%fs_min should be twice the maximum frequency. Here, fs_min =
2*fs2. fs_min = 2*fs2;
fprintf('\nEnter the sampling frequency greater than
%d\n',fs_min'); fs_sf=input('Enter the sampling frequency: ');

%We get the attenuation in dB. rp will be around 0 to 3
dB %rs will be around 30 to 50 dB
rp = input('\nEnter the passband ripple in dB: ');
rs = input('Enter the stopband attenuation in dB: ');
rp1=20*log10(rp)
rs1=20*log10(rs)

%We need to normalise wp,ws to pi. It is similar to finding the digital
%frequency digital omega = analog omega* Ts = analog
omega/fs_sf ws1=2*pi*fs1/fs_sf;
wp1=2*pi*fp1/fs_sf;
wp2=2*pi*fp2/fs_sf;
ws2=2*pi*fs2/fs_sf;

%The normalised frequencies are wp and ws
fprintf('\nws1 is %d\n',ws1);
fprintf('wp1 is %d\n',wp1);
fprintf('wp2 is %d\n',wp2);
fprintf('ws2 is %d\n',ws2);

%Computing the order(N) and cutoff frequency(wc) using wp,ws,rp,rs
using %the cheb1ord command with 's' option for analog filter.
%Finding the coefficients of filter [b a] using cheby command with 's'
option. wp = [wp1 wp2];
ws = [ws1 ws2];
[N wc]=cheb1ord(wp,ws,rp1,rs1,'s');
[b a]=cheby1(N,rp,wc,'s');
```

```
%Computing the frequency response using freqs command
%Computing h for specific values of w i.e. 0, pi/100, 2*pi/100... till
pi %and storing the corresponding the w values
w=0:(pi/100):pi;
[h omega] = freqs(b,a,w);
hf=tf(b,a)
disp(hf)
[z p]=tf2zp(b,a)

%Finding the magnitude response. Note: log10 should be
used. %Plotting magnitude versus omega.
mag_h=abs(h);
figure(1);
subplot(3,1,1);
plot(omega/pi,mag_h);

xlabel('frequency normalised to 1 -->');
ylabel('Gain in dB-->');
title('Magnitude Response of BPF');

%Finding the phase response.
%Plotting phase versus omega.
angle_h = angle(h);
subplot(3,1,2);
plot(omega/pi,angle_h)
xlabel('frequency normalised to 1 -->');
ylabel('Phase in radian-->');
title('Phase Response of BPF');

subplot(3,1,3)
zplane(z,p,'k')
title('Pole Zero Plot of BPF')
```

OUTPUT:

Enter the stop edge frequency1: 1000
Enter the pass edge frequency1: 4000
Enter the pass edge frequency2: 6000
Enter the stop edge frequency2: 9000
Enter the sampling frequency greater than 18000
Enter the sampling frequency: 20000
Enter the passband ripple in dB: 3
Enter the stopband attenuation in dB: 50

RESULT:

Thus, the magnitude and phase response of the analog IIR Chebyshev band pass filter is plotted using MATLAB.

Expt. No. 5d. DESIGN OF ANALOG IIR CHEBYSHEV BAND STOP FILTER

AIM:
To write a program in MATLAB to plot the magnitude and phase response of analog IIR

Chebyshev band stop filter.

SOFTWARE REQUIRED:

MATLAB Software

ALGORITHM:

1. Clear the command window.
2. Get the first stopband edge frequency (fs1), first passband edge frequency (fp1), second passband edge frequency (fp2) and second stopband edge frequency(fs2). 3. Get sampling frequency greater than the nyquist frequency
4. Get the attenuation values for passband and stop band.
5. Compute the digital frequency 'ω_p' and 'ω_s'by using the formula ω=2*pi*f/fs_sf
6. Group the passband and stopband edge frequencies as
   wp = [wp1 wp2] and ws = [ws1 ws2]
7. Compute the order of the analog filter using 'cheb1ord' function with 's' option 8. Compute IIR analog bandpass filter coefficients using 'cheby1' command with 'stop' option.
9. Compute frequency response of analog filter using 'freqs' command for different values of ω
10. Compute the magnitude using abs command.
11. Compute the phase using angle command.
12. Plot the magnitude and phase response.

FLOWCHART:

Start

Get the first passband edge frequency (fp1), first stopband edge frequency (fs1), second stopband edge frequency (fs2) and second passband edge frequency(fp2)

Get sampling frequency greater than the nyquist frequency (fs_sf)

Get the attenuation values for passband and stop band

Compute the digital frequency 'ω_p' and 'ω_s'by using the formula ω=2*pi*f/fs_sf

Group the passband and stopband edge frequencies as wp = [wp1 wp2] and ws = [ws1 ws2]

Compute the order of the analog filter using 'cheb1ord' function with 's' option

Compute IIR analog bandstop filter coefficients using 'butter' command with ‘stop’ option.

Compute frequency response of analog filter using ‘freqs’ command for different values of ω

Compute the magnitude using abs command

Compute the phase using angle command

Plot the magnitude and phase response

Stop

PROGRAM:

```
%Program for Butterworth IIR Bandstop analog filter
clc;
close all;
clear all;
fprintf('Program for Butterworth IIR Bandstop analog filter\n\n');

%We get the passedge, stopedge and sampling frequencies in
Hz fp1=input('Enter the pass edge frequency1: ');
fs1=input('Enter the stop edge frequency1: ');
fs2=input('Enter the stop edge frequency2: ');
fp2=input('Enter the pass edge frequency2: ');

%fs_min should be twice the maximum frequency. Here, fs_min =
2*fp2. fs_min = 2*fp2;
fprintf('\nEnter the sampling frequency greater than
%d\n',fs_min'); fs_sf=input('Enter the sampling frequency: ');

%We get the attenuation in dB. rp will be around 0 to 3 dB
%rs will be around 30 to 50 dB
rp = input('\nEnter the passband ripple in dB: ');
rs = input('Enter the stopband attenuation in dB: ');
rp1=20*log10(rp)
rs1=20*log10(rs)

%We need to normalise wp,ws to pi. It is similar to finding the digital
%frequency digital omega = analog omega* Ts = analog
omega/fs_sf wp1=2*pi*fp1/fs_sf;
ws1=2*pi*fs1/fs_sf;
ws2=2*pi*fs2/fs_sf;
wp2=2*pi*fp2/fs_sf;

%The normalised frequencies are wp and ws
fprintf('\nwp1 is %d\n',wp1);
fprintf('ws1 is %d\n',ws1);
fprintf('ws2 is %d\n',ws2);
fprintf('wp2 is %d\n',wp2);
```

```
%Computing the order(N) and cutoff frequency(wc) using wp,ws,rp,rs
using %the cheb1ord command with 's' option for analog filter.
%Finding the coefficients of filter [b a] using cheby command with 'stop' and 's'
option. wp = [wp1 wp2];
ws = [ws1 ws2];
[N wc]=cheb1ord(wp,ws,rp1,rs1,'s');
[b a]=cheby1(N,rp,wc,'stop','s');

 %Computing the frequency response using freqs command
%Computing h for specific values of w i.e. 0, pi/100, 2*pi/100... till
pi %and storing the corresponding the w values
w=0:(pi/100):pi;
[h omega] = freqs(b,a,w);
hf=tf(b,a)
disp(hf)
[z p]=tf2zp(b,a)

%Finding the magnitude response. Note: log10 should be used.
%Plotting magnitude versus omega.
mag_h=abs(h);
figure(1);
subplot(3,1,1);
plot(omega/pi,mag_h);

xlabel('frequency normalised to 1 -->');
ylabel('Gain in dB-->');
title('Magnitude Response of BRF');

%Finding the phase response.
%Plotting phase versus omega.
angle_h = angle(h);
subplot(3,1,2);
plot(omega/pi,angle_h)
xlabel('frequency normalised to 1 -->');
ylabel('Phase in radian-->');
title('Phase Response of BRF ');

subplot(3,1,3)
zplane(z,p,'k')
title('Pole Zero Plot of BRF')
```

OUTPUT:

Enter the pass edge frequency1: 1000
Enter the stop edge frequency1: 4000
Enter the stop edge frequency2: 6000
Enter the pass edge frequency2: 9000
Enter the sampling frequency greater than 18000
Enter the sampling frequency: 20000
Enter the passband ripple in dB: 3
Enter the stopband attenuation in dB: 50

RESULT:

Thus, the magnitude and phase response of the analog IIR Chebyshev band stop filter is plotted using MATLAB.

Expt. No. 6. DESIGN OF DIGITAL IIR FILTERS

Expt. No. 6a. DESIGN OF DIGITAL IIR BUTTERWORTH LOW PASS FILTER USING IMPULSE INVARIANT TRANSFORMATION

AIM:

To write a program in MATLAB to plot the magnitude and phase response of digital IIR Butterworth low pass filter using Impulse Invariant Transformation.

SOFTWARE REQUIRED:

MATLAB Software

ALGORITHM:

1. Clear the command window.
2. Get the passband edge frequency and stopband edge frequency
3. Get sampling frequency greater than the nyquist frequency(fs_sf)
4. Get the attenuation values for passband and stop band.
5. Compute the digital frequency 'ω_p' and 'ω_s'by using the formula ω=2*pi*f/fs_sf
6. Compute the order of the analog filter using 'buttord' function with 's' option 7. Compute IIR analog lowpass filter coefficients using 'butter' command. 8. Compute the IIR digital lowpass filter using 'impinvar' function.
9. Compute frequency response of digital filter using 'freqz' command for different values of ω
10. Compute the magnitude using abs command.
11. Compute the phase using angle command.
12. Plot the magnitude and phase response.

NOTE: INCLUDE STEP 8 IN THE FLOW CHART FOR ANALOG LPF (EXPT. No. 5) IN APPROPRIATE PLACE

PROGRAM:

```
clc;
close all;
clear all;
fprintf('Program for Digital IIR Butterworth Low Pass Filter using Impulse Invariant
Transformation\n\n');

%We get the passedge, stopedge and sampling frequencies in
Hz fp=input('Enter the passband edge frequency: ');
fs=input('Enter the stopband edge frequency: ');

%fs_min should be twice the maximum frequency. Here, fs_min =
2*fs. fs_min = 2*fs;
fprintf('\nEnter the sampling frequency greater than
%d\n',fs_min'); fs_sf=input('Enter the sampling frequency: ');

%We get the attenuation in dB. rp will be around 0 to 3 dB
%rs will be around 30 to 50 dB
rp = input('\nEnter the passband ripple in dB: ');
rs = input('Enter the stopband attenuation in dB: ');
```

```
rp1=20*log10(rp)
rs1=20*log10(rs)

%We need to normalise wp,ws to pi. It is similar to finding the digital
%frequency; digital omega = analog omega* Ts = analog
omega/fs_sf wp=2*pi*fp/fs_sf;
ws=2*pi*fs/fs_sf;
analog_wp=wp*fs_sf;
analog_ws=ws*fs_sf;

%The normalised frequencies are wp and ws
fprintf('\nwp is %d\n',wp);
fprintf('ws is %d\n',ws);

%Computing the order(N) and cutoff frequency(wc) using wp,ws,rp,rs
using %the buttord command with 's' option for analog filter.
%Finding the coefficients [b a] of filter using butter command with 's'
option. [N wc]=buttord(analog_wp,analog_ws,rp1,rs1,'s')
[b a]=butter(N,wc,'s');

%Finding the digital filter coefficients [c d] using impinvar
 command. [c d]=impinvar(b,a,fs_sf);

%Computing the frequency response using freqz command
%Computing h for specific values of w i.e. 0, pi/100, 2*pi/100... till
pi %and storing the corresponding the w values
w=0:(pi/100):pi;
[h omega] = freqz(c,d,w);

disp('Analog Filter Transfer Function - Unnormlised: ')
hf=tf(b,a)
disp('Digital Filter Transfer Function: ')
hf1=tf(c,d,fs_sf)
disp(hf)
disp(hf1)

[z p]=tf2zp(c,d)

%Plotting magnitude versus omega.
mag_h=abs(h);

figure(1);
subplot(3,1,1);
plot(omega/pi,mag_h);

xlabel('frequency normalised to 1 -->');
ylabel('Gain in dB-->');
title('Magnitude Response of LPF');

%Finding the phase response.
%Plotting phase versus omega.
angle_h = angle(h);
subplot(3,1,2);
plot(omega/pi,angle_h)
xlabel('frequency normalised to 1 -->');
```

```
ylabel('Phase in radian-->');
title('Phase Response of LPF');

%Plotting poles in the Z plane.
subplot(3,1,3)
zplane(z,p)
title('Pole Zero Plot of LPF')
```

OUTPUT:

```
Enter the passband edge frequency: .2*pi
Enter the stopband edge frequency: .7*pi

Enter the sampling frequency greater than 4.398230e+000
Enter the sampling frequency: 5

Enter the passband ripple in dB: .707
Enter the stopband attenuation in dB:. 2
```

RESULT:

Thus, the magnitude and phase response of the digital IIR Butterworth low pass filter using Impulse Invariant Transformation is plotted using MATLAB.

Expt. No. 6b. DESIGN OF DIGITAL IIR CHEBYSHEV HIGH PASS FILTER USING BILINEAR TRANSFORMATION

AIM:

To write a program in MATLAB to plot the magnitude and phase response of digital IIR Chebyshev High pass filter using Bilinear Transformation.

SOFTWARE REQUIRED:

MATLAB Software

ALGORITHM:

1. Clear the command window.
2. Get the passband edge frequency and stopband edge frequency
3. Get sampling frequency greater than the nyquist frequency(fs_sf)
4. Get the attenuation values for passband and stop band.
5. Compute the digital frequency 'ω_p' and 'ω_s'by using the formula ω=2*pi*f/fs_sf 6. Compute the order of the analog filter using 'cheb1ord' function with 's' option 7. Compute IIR analog highpass filter coefficients using 'cheby1' command with 'high' option.
8. Compute the IIR digital highpass filter using 'bilinear' function.
9. Compute frequency response of digital filter using 'freqz' command for different values of ω
10. Compute the magnitude using abs command.
11. Compute the phase using angle command.
12. Plot the magnitude and phase response.

NOTE: INCLUDE STEP 8 IN THE FLOW CHART FOR ANALOG HPF (EXPT. No. 5) IN APPROPRIATE PLACE

PROGRAM:

```
clc;
close all;
clear all;
fprintf('Program for Digital IIR Chebyshev High Pass Filter using Bilinear Transformation\n\n');

%We get the passedge, stopedge and sampling frequencies in Hz
fs=input('Enter the stopband edge frequency: ');
fp=input('Enter the passband edge frequency: ');


%fs_min should be twice the maximum frequency. Here, fs_min =
2*fs. fs_min = 2*fp;
fprintf('\nEnter the sampling frequency greater than %d\n',fs_min');
fs_sf=input('Enter the sampling frequency: ');

%We get the attenuation in dB. rp will be around 0 to 3 dB
%rs will be around 30 to 50 dB
rp = input('\nEnter the passband ripple in dB: ');
rs = input('Enter the stopband attenuation in dB: ');
rp1=20*log10(rp)
rs1=20*log10(rs)

%We need to normalise wp,ws to pi. It is similar to finding the digital
%frequency; digital omega = analog omega* Ts = analog omega/fs_sf
wp=2*pi*fp/fs_sf;
ws=2*pi*fs/fs_sf;
analog_wp=2*fs_sf*(tan(wp/2));
analog_ws=2*fs_sf*(tan(ws/2));

%The normalised frequencies are wp and ws
fprintf('\nwp is %d\n',wp);
fprintf('ws is %d\n',ws);

%Computing the order(N) and cutoff frequency(wc) using wp,ws,rp,rs
using %thecheb1ord command with 's' option for analog filter.
%Finding the coefficients of filter [b a] using cheby1 command with 's' and ‘high’
options. [N wc]=cheb1ord(analog_wp,analog_ws,rp1,rs1,'s')
[b a]=cheby1(N,rp,wc,'high','s');

%Finding the digital filter coefficients [c d] using bilinear command.
[c d]=bilinear(b,a,fs_sf);

%Computing the frequency response using freqz command
%Computing h for specific values of w i.e. 0, pi/100, 2*pi/100... till pi
```

```
%and storing the corresponding the w values
w=0:(pi/100):pi;
[h omega] = freqz(c,d,w);
 disp('Analog Filter Transfer Function - Unnormlised: ')
hf=tf(b,a)
disp('Digital Filter Transfer Function: ')
hf1=tf(c,d,fs_sf)
disp(hf)
disp(hf1)
 [z p]=tf2zp(c,d)

%Plotting magnitude versus omega.
mag_h=abs(h);
figure(1);
subplot(3,1,1);
plot(omega/pi,mag_h);

xlabel('frequency normalised to 1 -->');
ylabel('Gain in dB-->');
title('Magnitude Response of HPF');

%Finding the phase response.
%Plotting phase versus omega.
angle_h = angle(h);
subplot(3,1,2);
plot(omega/pi,angle_h)
xlabel('frequency normalised to 1 -->');
ylabel('Phase in radian-->');
title('Phase Response of HPF');

%Plotting poles in the Z plane.
subplot(3,1,3)
zplane(z,p)
title('Pole Zero Plot of HPF')
```

OUTPUT:

Enter the stopband edge frequency: 1500
Enter the passband edge frequency: 3000

Enter the sampling frequency greater than 6000
Enter the sampling frequency: 7000

Enter the passband ripple in dB: 4
Enter the stopband attenuation in dB: 50

RESULT:

Thus, the magnitude and phase response of the digital IIR Chebyshev High pass filter using Bilinear Transformation is plotted using MATLAB.

Expt. No. 6c. DESIGN OF DIGITAL IIR BUTTERWORTH BAND PASS FILTER USING IMPULSE INVARIANT TRANSFORMATION

AIM:

To write a program in MATLAB to plot the magnitude and phase response of analog IIR Chebyshev band pass analog IIR filter.

SOFTWARE REQUIRED:

MATLAB Software

ALGORITHM:

1. Clear the command window.
2. Get the first stopband edge frequency (fs1), first passband edge frequency (fp1), second passband edge frequency (fp2) and second stopband edge frequency(fs2). 3. Get sampling frequency greater than the nyquist frequency
4. Get the attenuation values for passband and stop band.
5. Compute the digital frequency 'ω_p' and 'ω_s'by using the formula ω=2*pi*f/fs_sf
6. Group the passband and stopband edge frequencies as
   wp = [wp1 wp2] and ws = [ws1 ws2]
7. Compute the order of the analog filter using 'buttord' function with 's' option
8. Compute IIR analog bandpass filter coefficients using 'butter' command. 9. Compute the IIR digital band pass filter using 'impinvar' function.
10. Compute frequency response of digital filter using 'freqz' command for different values of ω
11. Compute the magnitude using abs command.
12. Compute the phase using angle command.
13. Plot the magnitude and phase response.

NOTE: INCLUDE STEP 9 IN THE FLOW CHART FOR ANALOG BPF (EXPT. NO. 5) IN APPROPRIATE PLACE

PROGRAM:

```
clc;
close all;
clear all;
fprintf('Program for Digital IIR Butterworth Band Pass Filter using Impulse Invariant
Transformation\n\n');

%We get the passedge, stopedge and sampling frequencies in
Hz fs1=input('Enter the stop edge frequency1: ');
fp1=input('Enter the pass edge frequency1: ');
fp2=input('Enter the pass edge frequency2: ');
fs2=input('Enter the stop edge frequency2: ');

%fs_min should be twice the maximum frequency. Here, fs_min =
2*fs2. fs_min = 2*fs2;
fprintf('\nEnter the sampling frequency greater than
%d\n',fs_min'); fs_sf=input('Enter the sampling frequency: ');

%We get the attenuation in dB. rp will be around 0 to 3 dB
%rs will be around 30 to 50 dB
rp = input('\nEnter the passband ripple in dB: ');
rs = input('Enter the stopband attenuation in dB: ');
rp1=20*log10(rp)
rs1=20*log10(rs)
```

```
%We need to normalise wp,ws to pi. It is similar to finding the digital
%frequency digital omega = analog omega* Ts = analog
omega/fs_sf ws1=2*pi*fs1/fs_sf;
wp1=2*pi*fp1/fs_sf;
wp2=2*pi*fp2/fs_sf;
ws2=2*pi*fs2/fs_sf;

analog_ws1=ws1*fs_sf;
analog_wp1=wp1*fs_sf;
analog_wp2=wp2*fs_sf;
analog_ws2=ws2*fs_sf;

 %The normalised frequencies are wp and ws
fprintf('\nws1 is %d\n',ws1);
fprintf('wp1 is %d\n',wp1);
fprintf('wp2 is %d\n',wp2);
fprintf('ws2 is %d\n',ws2);

 %Computing the order(N) and cutoff frequency(wc) using wp,ws,rp,rs
using %the buttord command with 's' option for analog filter.
%Finding the coefficients of filter [b a] using butter command with 's'
option. wp = [analog_wp1 analog_wp2];
ws = [analog_ws1 analog_ws2];

[N wc]=buttord(wp,ws,rp1,rs1,'s');
[b a]=butter(N,wc,'s');

%Finding the digital filter coefficients [c d] using impinvar
 command. [c d]=impinvar(b,a,fs_sf);

%Computing the frequency response using freqz command
%Computing h for specific values of w i.e. 0, pi/100, 2*pi/100... till
pi %and storing the corresponding the w values
w=0:(pi/100):pi;
[h omega] = freqz(c,d,w);

disp('Analog Filter Transfer Function - Unnormlised: ')
hf=tf(b,a)
disp('Digital Filter Transfer Function: ')
hf1=tf(c,d,fs_sf)
disp(hf)
disp(hf1)

[z p]=tf2zp(c,d)

%Plotting magnitude versus omega.
mag_h=abs(h);

figure(1);
subplot(3,1,1);
plot(omega/pi,mag_h);

xlabel('frequency normalised to 1 -->');
ylabel('Gain in dB-->');
```

```
title('Magnitude Response of BPF');

%Finding the phase response.
%Plotting phase versus omega.
angle_h = angle(h);
subplot(3,1,2);
plot(omega/pi,angle_h)
xlabel('frequency normalised to 1 -->');
ylabel('Phase in radian-->');
title('Phase Response of BPF');

%Plotting poles in the Z plane.
subplot(3,1,3)
zplane(z,p)
title('Pole Zero Plot of BPF')
```

OUTPUT:

Enter the stop edge frequency1: 1000
Enter the pass edge frequency1: 4000
Enter the pass edge frequency2: 6000
Enter the stop edge frequency2: 9000

Enter the sampling frequency greater than 18000
Enter the sampling frequency: 20000

Enter the passband ripple in dB: 3
Enter the stopband attenuation in dB: 30

RESULT:

Thus, the magnitude and phase response of the digital IIR Butterworth Band Pass filter using Impulse Invariant Transformation is plotted using MATLAB.

Expt. No. 6d. DESIGN OF DIGITAL IIR CHEBYSHEV BAND STOP FILTER USING BILINEAR TRANSFORMATION

AIM:

To write a program in MATLAB to plot the magnitude and phase response of digital IIR Chebyshev Band Stop filter using Bilinear Transformation.

SOFTWARE REQUIRED:

MATLAB Software

ALGORITHM:

1. Clear the command window.
2. Get the first stopband edge frequency (fs1), first passband edge frequency (fp1), second passband edge frequency (fp2) and second stopband edge frequency(fs2). 3. Get sampling frequency greater than the nyquist frequency
4. Get the attenuation values for passband and stop band.
5. Compute the digital frequency 'ω_p' and 'ω_s'by using the formula ω=2*pi*f/fs_sf
6. Group the passband and stopband edge frequencies as
   wp = [wp1 wp2] and ws = [ws1 ws2]
7. Compute the order of the analog filter using 'cheb1ord' function with 's' option 8. Compute

IIR analog bandpass filter coefficients using ‘cheby1’ command with ‘stop’ option.
9. Compute the IIR digital band stop filter using ‘bilinear’ function.
10. Compute frequency response of digital filter using ‘freqz’ command for different values of ω
11. Compute the magnitude using abs command.
12. Compute the phase using angle command.
13. Plot the magnitude and phase response.

NOTE: INCLUDE STEP 9 IN THE FLOW CHART FOR ANALOG BPF (EXPT. NO. 5) IN APPROPRIATE PLACE

PROGRAM:

```
clc;
close all;
clear all;
fprintf('Program for Digital IIR Chebyshev Band Stop Filter using Bilinear
Transformation\n\n'); %We get the passedge, stopedge and sampling frequencies in Hz
fp1=input('Enter the pass edge frequency1: ');
fs1=input('Enter the stop edge frequency1: ');
fs2=input('Enter the stop edge frequency2: ');
fp2=input('Enter the pass edge frequency2: ');

%fs_min should be twice the maximum frequency. Here, fs_min =
2*fs2. fs_min = 2*fp2;
fprintf('\nEnter the sampling frequency greater than %d\n',fs_min');
fs_sf=input('Enter the sampling frequency: ');

%We get the attenuation in dB. rp will be around 0 to 3 dB
%rs will be around 30 to 50 dB
rp = input('\nEnter the passband ripple in dB: ');
rs = input('Enter the stopband attenuation in dB: ');
rp1=20*log10(rp)
rs1=20*log10(rs)

%We need to normalise wp,ws to pi. It is similar to finding the digital
%frequency digital omega = analog omega* Ts = analog omega/fs_sf
wp1=2*pi*fp1/fs_sf;
ws1=2*pi*fs1/fs_sf;
ws2=2*pi*fs2/fs_sf;
wp2=2*pi*fp2/fs_sf;

analog_wp1=2*fs_sf*(tan(wp1/2));
analog_ws1=2*fs_sf*(tan(ws1/2));
analog_ws2=2*fs_sf*(tan(ws2/2));
analog_wp2=2*fs_sf*(tan(wp2/2));


 %The normalised frequencies are wp and ws
fprintf('\nws1 is %d\n',ws1);
fprintf('wp1 is %d\n',wp1);
fprintf('wp2 is %d\n',wp2);
fprintf('ws2 is %d\n',ws2);

 %Computing the order(N) and cutoff frequency(wc) using wp,ws,rp,rs
```

```
using %the cheb1ord command with 's' option for analog filter.
%Finding the coefficients of filter [b a] using cheby command with 's' and ‘stop’
option. wp = [analog_wp1 analog_wp2];
ws = [analog_ws1 analog_ws2];

[N wc]=cheb1ord(wp,ws,rp1,rs1,'s');
[b a]=cheby1(N,rp,wc,'stop','s');

%Finding the digital filter coefficients [c d] using bilinear command.
[c d]=bilinear(b,a,fs_sf);

%Computing the frequency response using freqz command
%Computing h for specific values of w i.e. 0, pi/100, 2*pi/100... till pi
%and storing the corresponding the w values
w=0:(pi/100):pi;
[h omega] = freqz(c,d,w);

disp('Analog Filter Transfer Function - Unnormlised: ')
hf=tf(b,a)
disp('Digital Filter Transfer Function: ')
hf1=tf(c,d,fs_sf)
disp(hf)
disp(hf1)

[z p]=tf2zp(c,d)

%Plotting magnitude versus omega.
mag_h=abs(h);

figure(1);
subplot(3,1,1);
plot(omega/pi,mag_h);

xlabel('frequency normalised to 1 -->');
ylabel('Gain in dB-->');
title('Magnitude Response of BRF');

%Finding the phase response.
%Plotting phase versus omega.
angle_h = angle(h);
subplot(3,1,2);
plot(omega/pi,angle_h)
xlabel('frequency normalised to 1 -->');
ylabel('Phase in radian-->');
title('Phase Response of BRF');

%Plotting poles in the Z plane.
subplot(3,1,3)
zplane(z,p)
title('Pole Zero Plot of BRF')
```

OUTPUT:

Enter the pass edge frequency1: 1000

Enter the stop edge frequency1: 2000
Enter the stop edge frequency2: 3000
Enter the pass edge frequency2: 4000

Enter the sampling frequency greater than 8000
Enter the sampling frequency: 10000

Enter the passband ripple in dB: 3
Enter the stopband attenuation in dB: 30

RESULT:

Thus, the magnitude and phase response of the digital IIR Chebyshev Band Stop filter using Bilinear Transformation is plotted using MATLAB.