EXPERIMENT NO. 6

Design and simulation of analog IIR Filters (Butterworth and Chebyshev)
Design and simulation of digital IIR Filters (Bilinear transformation)

Aim:

Equation used for computation:

MATLAB Commands used:

Code/Program

Program 1:

```
%generate filter coefficients for the given %order & cutoff Say N=2, fc=150Hz, %fs=1000 Hz,
%butterworth filter
[b,a]=butter(2, 150/(1000/2));
%generate simulated input of 100, 300 & 170 Hz, each of 30 points
n=1:30;
f1=100;f2=300;f3=170;fs=1000;
x=[];
x1=sin(2*pi*n*f1/fs);
x2=sin(2*pi*n*f2/fs);
x3=sin(2*pi*n*f3/fs);
x=[x1 x2 x3];
subplot(2,1,1);
stem(x);
title('input');
%generate o/p
y=filter(b,a,x);
subplot(2,1,2);
stem(y);
title('output');
```

Computations/Calculations

Input:

Computation (Clearly mention all steps involved in computation)

Output:

Graph/Display

Program 2:

```
% Butterworth filter: Given data: rp=1, rs=40, w1=800, w2=1200,ws=3600;
% Analog frequency
aw1=2*pi*w1/ws;
aw2=2*pi*w2/ws;
% Prewrapped frequency
pw1 = 2*tan(aw1/2);
```

```
pw2 = 2*tan(aw2/2);
%Calculate order and cutoff freq
[n,wc]= buttord (pw1,pw2,rp,rs,'s');
% analog filter transfer
[b,a] = butter(n,wc,'s');
% obtaining the digital filter using bilinear transformation
fs=1;
[num,den]= bilinear(b,a,fs);
%plot the frequency response
[mag,freq1]=freqz(num,den,128);
freq=freq1*ws/(2*pi);
m = 20*log10(abs(mag));
plot(freq,m);
grid;;
```

Computations/Calculations

Input:

Computation (Clearly mention all steps involved in computation)

Output:

Graph/DisplaY

Program 3:

To design a chebyshev filter for given specifications

```
%Given data
rp=1,rs=40,w1=800,w2=1200,ws=3600
%Analog frequencies
aw1= 2*pi*w1/ws;
aw2=2*pi*w2/ws;
% Prewrapped frequency assuming T=1/fs
pw1 = 2*tan(aw1/2);
pw2 = 2*tan(aw2/2);
[n,wc]= cheb1ord (pw1,pw2,rp,rs,'s');
[b,a] = cheby1(n,rp,wc,'s');
% obtaining the digital filter using bilinear transformation
fs=1;
[num,den]= bilinear(b,a,fs);
%plot the frequency response
[mag,freq1]=freqz(num,den,128);
freq=freq1*ws/(2*pi);
m = 20*log10(abs(mag));
plot(freq,m);
grid;
```

<u>Computations/Calculations</u>

Input:

Computation (Clearly mention all steps involved in computation)

Output:

Graph/Display

Inference:

EXPERIMENT NO. 7

Design and simulation of FIR Filters (LP, HP, BP, BS) by using window techniques

Aim:

Equation used for computation:

MATLAB Commands used:

Code/Program

```
%Design and implementation of FIR filter Method 1
%generate filter coefficients for the given %order & cutoff Say N=33, fc=150Hz, %fs=1000 Hz,
Hamming window
h=fir1(33, 150/(1000/2),hamming(34));
%generate simulated input of 50, 300 & 200 Hz, each of 30 points
n=1:30;
f1=50;f2=300;f3=200;fs=1000;
x=[];
x1=sin(2*pi*n*f1/fs);
x2=sin(2*pi*n*f2/fs);
x3=sin(2*pi*n*f3/fs);
x=[x1 x2 x3];
subplot(2,1,1);
stem(x);
title('input');
%generate o/p
%y=conv(h,x);
y=filter(h,1,x);
subplot(2,1,2);
stem(y);
title('output');
```

Computations/Calculations

Input:

Computation (Clearly mention all steps involved in computation)

Output:

Graph/Display

```
%Design and implementation of FIR filter Method 2
%Method 2: the following program gives only the design of the FIR filter- for implementation
continue with the next program (after h[n])
%input data to be given: Passband & Stopband frequency
% Data given: Passband ripple & stopband attenuation As. If As>40 dB, Choose hamming
clear
wpa=input('Enter passband edge frequency in Hz');
wsa= input('Enter stopband edge frequency in Hz');
ws1= input('Enter sampling frequency in Hz');
%Calculate transmission BW,Transition band tb,order of the filter
wpd=2*pi*wpa/ws1;
wsd=2*pi*wsa/ws1;
tb=wsd-wpd;
N=ceil(6.6*pi/tb)
```

```
wc=(wsd+wpd)/2;
%compute the normalized cut off frequency
wc=wc/pi;
%calculate & plot the window
hw=hamming(N+1);
stem(hw);
title('Fir filter window sequence- hamming window');
%find h(n) using FIR
h=fir1(N,wc,hamming(N+1));
%plot the frequency response
figure(2);
[m,w]=freqz(h,1,128);
mag=20*log10(abs(m));
plot(ws1*w/(2*pi),mag);
title('Fir filter frequency response');
grid;
```

<u>Computations/Calculations</u>

Input:

Computation (Clearly mention all steps involved in computation)

Output:

Graph/Display

Inference: