Experiment No: 7
Experiment Name: Implementation of LP IIR Filter

Objective: The objective is to design and implement a Low-Pass Infinite Impulse Response (LP IIR) filter using suitable design methods. The experiment aims to demonstrate the application of IIR filter principles, analyze its frequency response, and assess its behavior in theoretical and practical contexts. This involves examining the impulse response, stability, and frequency characteristics of the filter, and implementing it through digital signal processing tools or simulation software.

Software:

- MATLAB

Theory: IIR filters are digital filters characterized by an infinite impulse response. Unlike FIR filters, they incorporate feedback (a recursive component), earning the name recursive digital filters. Due to this feedback mechanism, IIR filters achieve superior frequency response compared to FIR filters of the same order. However, they exhibit a non-linear phase characteristic, which can pose challenges for systems requiring phase linearity. Hence, IIR filters are less favored for applications where phase accuracy is crucial but are highly effective when linear phase response is not critical.

A potential drawback of IIR filters is their susceptibility to instability due to feedback, unlike FIR filters, which are inherently stable. Thus, ensuring the stability of IIR filters is a critical step during the design process. IIR filter design involves creating an analog filter (e.g., Butterworth or Chebyshev) based on the specifications and transforming it into a digital filter through methods such as bilinear transformation or impulse invariant transformation.

Algorithm:

Step I : Enter the pass band ripple (rp) and stop band ripple (rs).

Step II : Enter the pass band frequency (wp) and stop band frequency (ws).

Step III : Get the sampling frequency (fs).

Step IV : Calculate normalized pass band frequency, and normalized stop band frequency w1 and w2 respectively. w1 = 2 * wp /fs, w2 = 2 * ws /fs

Step V : Make use of the following function to calculate order of filter

Butterworth filter order [n,wn]=buttord(w1,w2,rp,rs)

Chebyshev filter order [n,wn] = cheblord(w1,w2,rp,rs)

Step VI : Design an nth order digital low pass Butterworth or Chebyshev filter using the following statements.

Butterworth filter

[b, a]=butter (n, wn)

Chebyshev filter [b,a]=cheby1 (n, 0.5, wn)

Step VII : Find the digital frequency response of the filter by using 'freqz( )' function

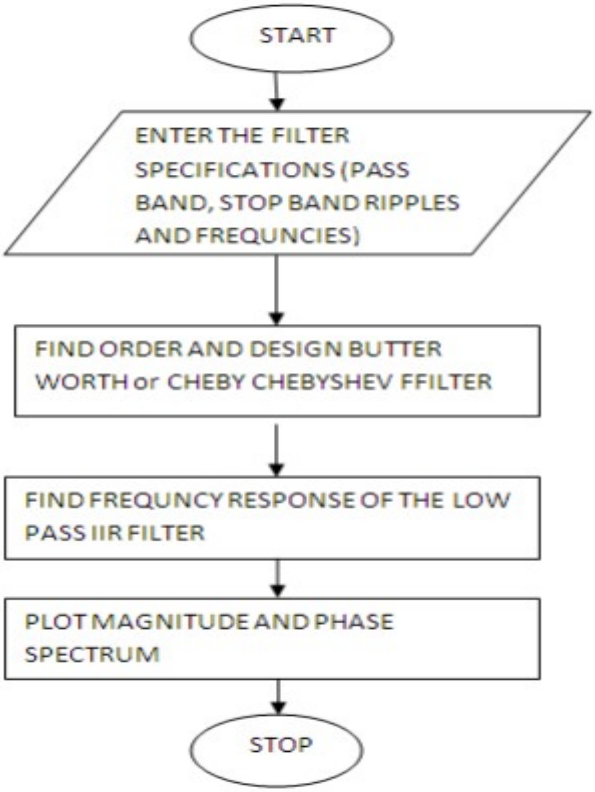
Step VIII : Calculate the magnitude of the frequency response in decibels (dB) mag=20*log10 (abs (H))

Step IX : Plot the magnitude response [magnitude in dB Vs normalized frequency]

Step X : Calculate the phase response using angle (H)

Step XI : Plot the phase response [phase in radians Vs normalized frequency (Hz)].

Flow Chart:



Code:

```
clc; clear all; close all;
disp('Enter the IIR filter design specifications');
rp=input('Enter the passband ripple: ');
rs = input('Enter the stopband ripple: ');
wp = input('Enter the passband freq: ');
ws = input('Enter the stopband freq: ');
fs = input('Enter the sampling freq: ');

w1 = 2*wp/fs;
w2 = 2*ws/fs;
[n,wn] = buttord(w1,w2,rp,rs,'s');
disp('Frequency response of IIR LPF is: ');
[b,a] = butter(n,wn,'low','s');
w = 0:.01:pi;
[h,om] = freqs(b,a,w);
m = 20*log10(abs(h));
an = angle(h);

figure,subplot(2,1,1);
```

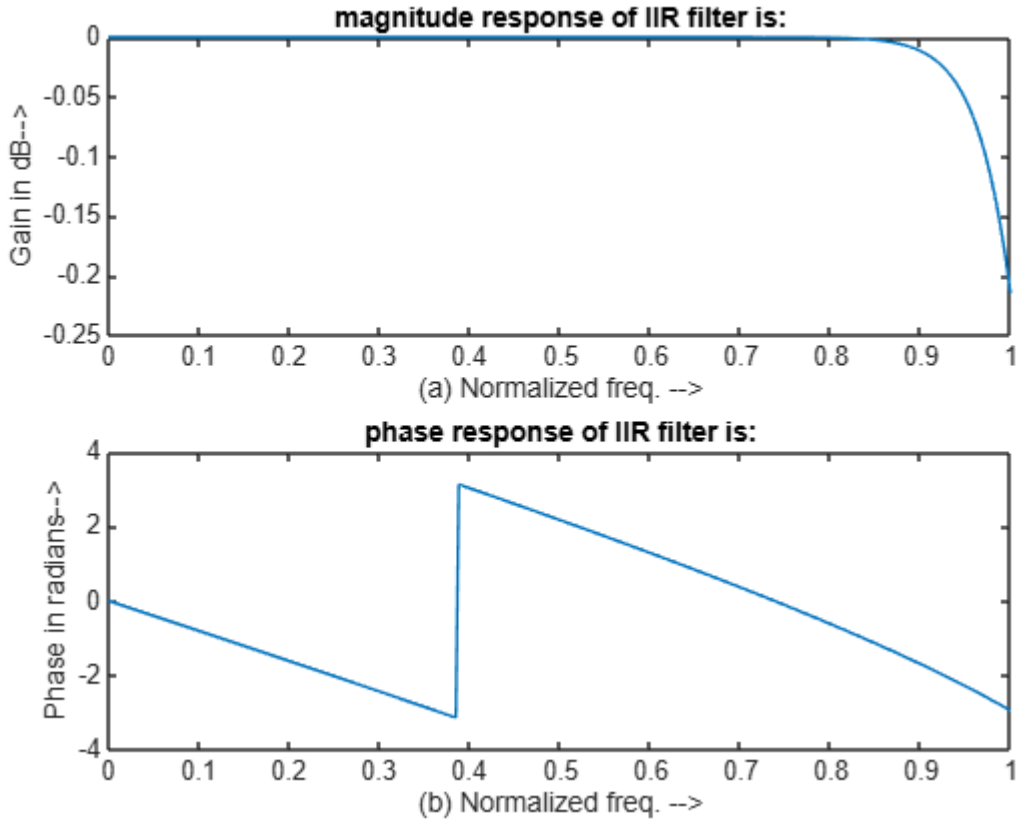
```
plot(om/pi,m);
title('magnitude response of IIR filter is:');
xlabel('(a) Normalized freq. -->');
ylabel('Gain in dB-->');

subplot(2,1,2);
plot(om/pi,an);
title('Phase response of IIR filter is:');
xlabel('(b) Normalized freq. -→');
ylabel('Phase in radians-→');
```

Input:

Enter the IIR filter design specifications
Enter the passband ripple: 5
Enter the stopband ripple: 200
Enter the passband freq: 9900
Enter the stopband freq: 1900
Enter the sampling freq: 5635

Output:



Discussion: The MATLAB implementation of the IIR low-pass filter illustrates its efficiency in digital signal processing. The filter effectively attenuates high frequencies while preserving low frequencies, meeting the design specifications. Though IIR filters introduce phase distortion, they are computationally efficient due to fewer required coefficients compared to FIR filters. Frequency and time-domain analyses confirmed its performance, highlighting the importance of proper parameter selection and MATLAB's utility in designing robust signal processing systems.