# DSP LAB – 11

NAME: Ritu

PRN: 20220802132  
BATCH: A1

## 1. LOW PASS IIR FILTER

### AIM:

To implement a Low Pass IIR filter for a given sequence.

### Software:

MATLAB

### Theory:

IIR filters have an infinite impulse response and use feedback, providing a superior frequency response compared to FIR filters. However, due to their recursive nature, they can be unstable and are not ideal when phase linearity is required. Common types of IIR filters include Butterworth and Chebyshev.

### Algorithm:

* Enter the passband ripple (rp) and stopband ripple (rs).
* Enter the passband frequency (wp) and stopband frequency (ws).
* Define the sampling frequency (fs).
* Normalize the passband and stopband frequencies: w1 = 2 \* wp / fs and w2 = 2 \* ws / fs.
* Determine the filter order using Butterworth or Chebyshev method:
* - Butterworth: [n, wn] = buttord(w1, w2, rp, rs);
* - Chebyshev: [n, wn] = cheb1ord(w1, w2, rp, rs);
* Design the filter:
* - Butterworth: [b, a] = butter(n, wn, 'low');
* - Chebyshev: [b, a] = cheby1(n, 0.5, wn, 'low');
* Calculate the frequency response using freqz(b, a).
* Plot the magnitude and phase responses:
* - Image Required: Magnitude response (Gain in dB vs Normalized Frequency)
* - Image Required: Phase response (Phase in Radians vs Normalized Frequency)

### Program:

clc;  
clear all;  
close all;  
  
% Enter filter specifications  
rp = input('Enter the passband ripple: ');  
rs = input('Enter the stopband ripple: ');  
wp = input('Enter the passband frequency: ');  
ws = input('Enter the stopband frequency: ');  
fs = input('Enter the sampling frequency: ');  
  
% Calculate normalized frequencies  
w1 = 2 \* wp / fs;  
w2 = 2 \* ws / fs;  
  
% Determine filter order  
[n, wn] = buttord(w1, w2, rp, rs);  
  
% Design Butterworth low-pass filter  
[b, a] = butter(n, wn, 'low');  
  
% Frequency response  
[H, F] = freqz(b, a, 512, fs);  
  
% Plot magnitude response  
subplot(2, 1, 1);  
plot(F, 20\*log10(abs(H)));  
title('Magnitude Response of Low Pass IIR Filter');  
xlabel('Frequency (Hz)');  
ylabel('Gain (dB)');  
  
% Plot phase response  
subplot(2, 1, 2);  
plot(F, angle(H));  
title('Phase Response of Low Pass IIR Filter');  
xlabel('Frequency (Hz)');  
ylabel('Phase (Radians)');

### Output:

* Image Required: Magnitude Response (Gain in dB vs Frequency)
* Image Required: Phase Response (Phase in Radians vs Frequency)

## 2. HIGH PASS IIR FILTER

### AIM:

To implement a High Pass IIR filter for a given sequence.

### Software:

MATLAB

### Algorithm:

* Enter the passband ripple (rp) and stopband ripple (rs).
* Enter the passband frequency (wp) and stopband frequency (ws).
* Define the sampling frequency (fs).
* Normalize the passband and stopband frequencies: w1 = 2 \* wp / fs and w2 = 2 \* ws / fs.
* Determine the filter order using Butterworth or Chebyshev method.
* Design the filter:
* - Butterworth: [b, a] = butter(n, wn, 'high');
* - Chebyshev: [b, a] = cheby1(n, 0.5, wn, 'high');
* Calculate the frequency response using freqz(b, a).
* Plot the magnitude and phase responses:
* - Image Required: Magnitude response (Gain in dB vs Normalized Frequency)
* - Image Required: Phase response (Phase in Radians vs Normalized Frequency)

### Program:

clc;  
clear all;  
close all;  
  
% Enter filter specifications  
rp = input('Enter the passband ripple: ');  
rs = input('Enter the stopband ripple: ');  
wp = input('Enter the passband frequency: ');  
ws = input('Enter the stopband frequency: ');  
fs = input('Enter the sampling frequency: ');  
  
% Calculate normalized frequencies  
w1 = 2 \* wp / fs;  
w2 = 2 \* ws / fs;  
  
% Determine filter order  
[n, wn] = buttord(w1, w2, rp, rs);  
  
% Design Butterworth high-pass filter  
[b, a] = butter(n, wn, 'high');  
  
% Frequency response  
[H, F] = freqz(b, a, 512, fs);  
  
% Plot magnitude response  
subplot(2, 1, 1);  
plot(F, 20\*log10(abs(H)));  
title('Magnitude Response of High Pass IIR Filter');  
xlabel('Frequency (Hz)');  
ylabel('Gain (dB)');  
  
% Plot phase response  
subplot(2, 1, 2);  
plot(F, angle(H));  
title('Phase Response of High Pass IIR Filter');  
xlabel('Frequency (Hz)');  
ylabel('Phase (Radians)');

### Output:

* Image Required: Magnitude Response (Gain in dB vs Frequency)
* Image Required: Phase Response (Phase in Radians vs Frequency)

## RESULT:

Successfully implemented Low Pass and High Pass IIR filters and observed their magnitude and phase responses.