# DIGITAL SIGNAL PROCESSING LAB

# (EL-302)

# LABORATORY MANUAL

**ENGR. MUHAMMAD IBRAR KHAN**

## IIR Filter Design

**(LAB # 11)**

Student Name: ___________________________________

Roll No: _______________ Section: ___

Date performed: ____________, 2018

**NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES, ISLAMABAD**

Prepared by: Engr. Muhammad Asim
Last Edited by: Engr. Muhammad Asim, Mar 29, 2016
Verified by: Dr. Shahzad Saleem

Updated: Spring 2016

# Lab # 11: IIR Filter Design

## Learning Objectives

a. To design IIR digital filter using Butterworth analog filter

**b.** To design IIR digital filter using Chebyshev (Type I and II) analog filter

## Equipment Required

1. PC
2. Matlab

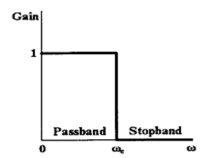
# 1. Introduction:

Filters are frequency selective network and are used for signal separation and restoration.
Filters can be classified into:
Analog filters- Realized by R, C and active devices, they are cheap, fast and large dynamic range, but poor performance.
Digital filters- Realized by digital system like memory, adder and multipliers; they have better performance in low frequency as well as high frequency operation.
There are two types of filters: IIR (Infinite Impulse Response) and FIR (Finite Impulse Response)

Ideal frequency response of filters:

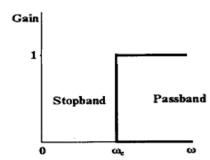
LPF

$$H(\omega) = \begin{cases} 1 & \text{for } 0 \le \omega < \omega_c \\ 0 & \text{for } \omega_c < \omega < \infty \end{cases}$$



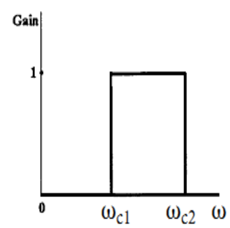

HPF

$$H(\omega) = \begin{cases} 0 & \text{for } 0 \le \omega < \omega_c \\ 1 & \text{for } \omega_c < \omega < \infty \end{cases}$$



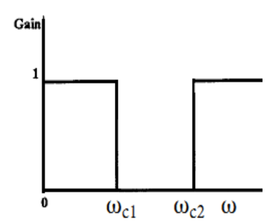

BPF

$$H(\omega) = \begin{cases} 0 & \text{for } 0 \le \omega < \omega_{c1} \\ 1 & \text{for } \omega_{c1} < \omega < \omega_{c2} \\ 0 & \text{for } \omega_{c2} \le \omega < \infty \end{cases}$$

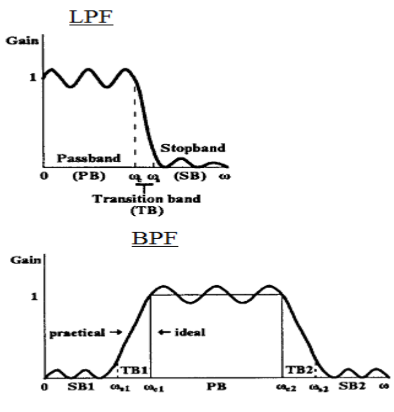



BSF

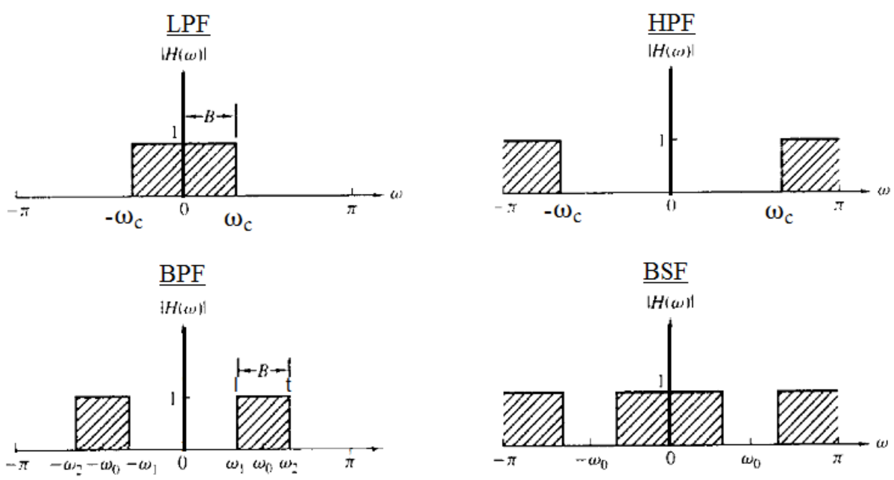
$$H(\omega) = \begin{cases} 1 & \text{for } 0 \le \omega < \omega_{c1} \\ 0 & \text{for } \omega_{c1} < \omega < \omega_{c2} \\ 1 & \text{for } \omega_{c2} \le \omega < \infty \end{cases}$$



Practical frequency response of filters:




Ideal frequency response of digital filters:




**a. IIR Filter Basics:**

IIR digital filters which can be realized and characterized by the recursive equation,

$$
\begin{aligned}
y[n] &= \sum_{k=0}^{\infty} h[k]x[n-k] \\
&= \sum_{k=0}^{N} b_k x[n-k] - \sum_{k=0}^{M} a_k y[n-k]
\end{aligned}
$$

Where h[k] is the impulse response of the filter, $b_k$ and $a_k$ are the filter coefficients of the filter, and x[n] and y[n] are the input and output to the filter. The transfer function for the IIR filter is,

$$
\begin{aligned}
H(z) &= \frac{b_0 + b_1 z^{-1} + \ldots + b_N z^{-N}}{1 + a_1 z^{-1} + \ldots + a_M z^{-M}} \\
&= \frac{\sum_{k=0}^{N} b_k z^{-k}}{1 + \sum_{k=1}^{M} a_k z^{-k}}
\end{aligned}
$$

The transfer function of the IIR filter, H(z) can be represented as

$$H(z) = \frac{K(z-z_1)(z-z_2)...(z-z_N)}{(z-p_1)(z-p_2)...(z-p_M)}$$

Where K is the gain, z1, z2 ….are the zeros and p1, p2 ….are the poles of the transfer function, H(z).

## b. IIR filter design through analog filters:

IIR filters are generally designed by first designing its analog counterparts and then transforming it to digital filters. This method is preferred as analog filter design procedures like Butterworth and Chebyshev.
In case of IIR filter design, the most common practice is to convert the digital filter specifications into analog low pass prototype filter specifications. Then to transform it into the desired digital filter transfer function H(z) by
a) Impulse invariant.
b) Bilinear transformation.
Here we will consider only Bilinear Transformation.

**Bilinear Transformation**

Bilinear transformation is much more important and useful than Impulse Invariant Transformation method, especially in high pass, band pass and stop filters. The method is derived by approximating the first order differential equation with a difference equation. This transformation maps the analog transfer function, H(s) from the s-plane onto the discrete transfer function, H(z) in the z-plane. The entire jΩ axis in the s-plane is mapped onto the z-plane. The left half s-plane is mapped inside the unit circle and the right half s-plane is mapped outside the unit circle in the z-plane.

The mapping formulae is

$$\begin{aligned} H(s) &\rightarrow H(z) \\ s &\rightarrow \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) \end{aligned}$$

Relation between Analog and Digital Frequencies:

$$\omega = 2\tan^{-1}\left(\frac{T\Omega}{2}\right)$$

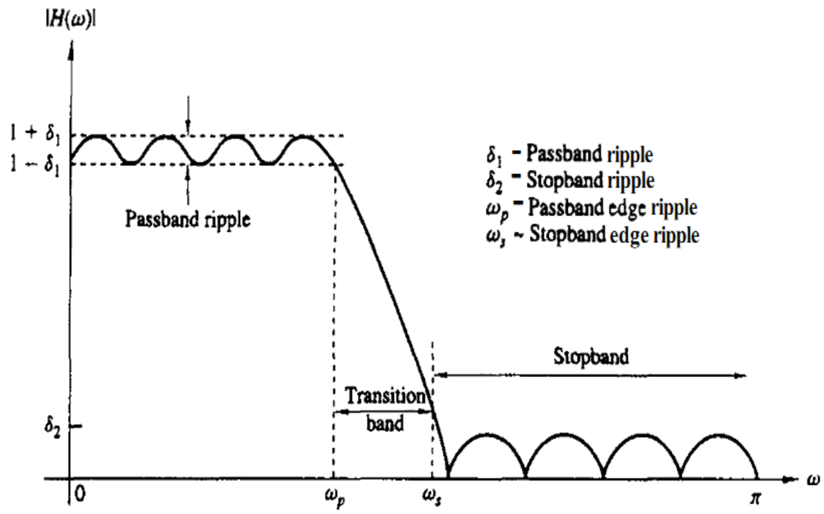
**MATLAB Functions:**

***BILINEAR*** *Bilinear transformation with optional frequency prewarping.*
***[Zd,Pd,Kd] = BILINEAR(Z,P,K,Fs)*** *converts the s-domain transfer function specified by Z, P, and K to a z-transform discrete equivalent obtained from the bilinear transformation: where column vectors Z and P specify the zeros and poles, scalar K specifies the gain, and Fs is the sample frequency in Hz.*

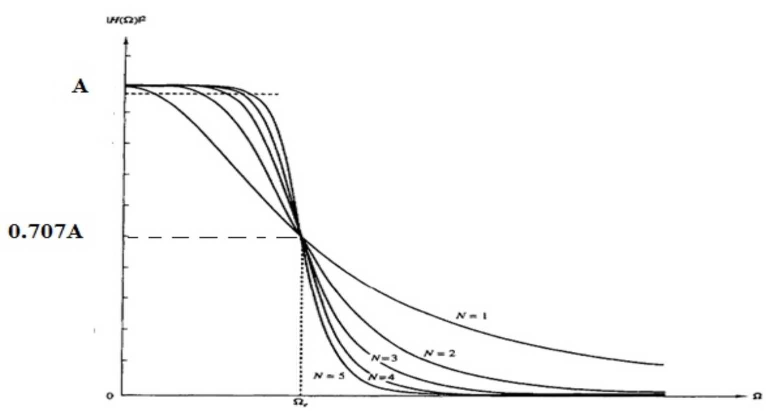
***[NUMd,DENd] = BILINEAR(NUM,DEN,Fs),*** *where NUM and DEN are row vectors containing numerator and denominator transfer function coefficients, NUM(s)/DEN(s), in descending powers of s, transforms to z-transform coefficients NUMd(z)/DENd(z).*

### c. Butterworth Approximation

The magnitude response of a practical low pass filter, Where δ is gain and ω is frequency in radians/sec




The magnitude response of Butterworth low pass filter,




The low pass Butterworth filter is characterized by the following magnitude squared magnitude response,

$$|H(j\Omega)|^2 = \frac{\mathbf{A}}{1 + \left(\frac{\Omega}{\Omega_C}\right)^{2N}}$$

Where N is the order of the filter, Ω is analog frequency in radians; ΩC is 3dB cut-off frequency in radians.

$$N \geq \frac{1}{2}\frac{\log_{10}\left\{\left(\frac{1}{\delta_2{}^2}-1\right)\Big/\left(\frac{1}{\delta_1{}^2}-1\right)\right\}}{\log_{10}\left(\Omega_s/\Omega_p\right)}$$

$$\Omega_C = \frac{\Omega}{\left[\frac{1}{\delta_1{}^2}-1\right]^{\frac{1}{2N}}}$$

The advantage of this approximation is flat response over stop band and pass band. The disadvantage is wide transition band.
The transfer function of the filter is:

For N is even

$$H(s) = \prod_{k=1}^{N/2}\frac{B_k\Omega_C{}^2}{s^2 + b_k\Omega_C s + c_k\Omega_C{}^2}$$

For N is odd

$$H(s) = \frac{B_0\Omega_C}{s + c_0\Omega_C}\prod_{k=1}^{(N-1)/2}\frac{B_k\Omega_C{}^2}{s^2 + b_k\Omega_C s + c_k\Omega_C{}^2}$$

The coefficients bk and ck are given by

$$b_k = 2\sin\left[\frac{(2k-1)\pi}{2\text{N}}\right]$$

and ck=1
The parameter Bk can be obtained from:
for N is even

$$A = \prod_{k=1}^{N/2} B_k$$

for N is odd

$$A = \prod_{k=0}^{(N-1)/2} B_k$$

**MATLAB functions:**

1) ***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 passband and has at least Rs dB of attenuation in the stopband.*

*Wp and Ws are the passband and stopband edge frequencies normalized from 0 to 1 (where 1 corresponds to pi radians/sample). For example,*
*Lowpass: Wp = .1, Ws = .2*
*Highpass: Wp = .2, Ws = .1*
*Bandpass: Wp = [.2 .7], Ws = [.1 .8]*
*Bandstop: Wp = [.1 .8], Ws = [.2 .7]*
*BUTTORD also returns Wn, the Butterworth natural frequency (or, the "3 dB frequency") to use 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.*

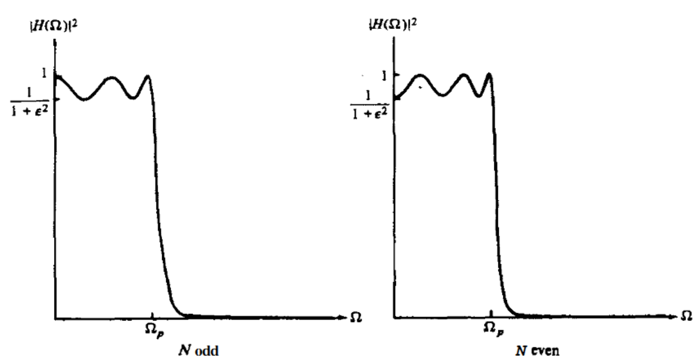
2) ***BUTTER*** *Butterworth digital and analog filter design.*
***[B,A] = BUTTER(N,Wn)*** *designs an Nth order lowpass digital Butterworth filter and returns the filter coefficients in length N+1 vectors B (numerator) and A (denominator). The coefficients are listed in descending powers of z. The cutoff frequency Wn must be 0.0 < Wn < 1.0, with 1.0 corresponding to half the sample rate.*
*If Wn is a two-element vector, Wn = [W1 W2], BUTTER returns an order 2N bandpass filter with passband W1 < W < W2.*
***[B,A] = BUTTER(N,Wn,'high')*** *designs a highpass filter.*
***[B,A] = BUTTER(N,Wn,'low')*** *designs a lowpass filter.*
***[B,A] = BUTTER(N,Wn,'stop')*** *is a bandstop filter if Wn = [W1 W2].*
*When used with three left-hand arguments, as in [Z,P,K] = BUTTER(...), the zeros and poles are returned in length N column vectors Z and P, and the gain in scalar K.BUTTER(N,Wn,'s'), BUTTER(N,Wn,'high','s') and BUTTER(N,Wn,'stop','s') design analog Butterworth filters. In this case, Wn is in [rad/s] and it can be greater than 1.0.*

## d. Chebyshev Approximation

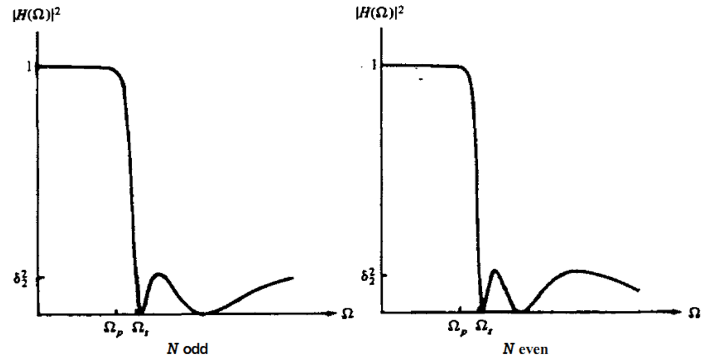
The Butterworth approximation, which is monotonic in both the pass band and stopband, gives us a magnitude response which exceeds the specifications for the pass band ripple. The problem is that the order of the filter will be high. Usually, we can tolerate some amount of ripple, and it would be advantageous to have a trade-off between the ripple and the filter order, so that we can obtain a filter which meets the specifications and has a lower order than its Butterworth counterpart. For this we use the Chebychev approximation. The characteristic of Chebyshev approximation is that it minimizes the error between the idealized and the actual filter over the range of filter allowing some ripples in pass band.

- Type I, with equal ripple in the pass band, monotonic in the stop band.
- Type II, with equal ripple in the stop band, monotonic in the pass band

Magnitude response of Type-I filter:



Magnitude response of Type-II filters:




**Type I Chebyshev** Filter is characterized by the magnitude squared response is given by

$$|H(\Omega)|^2 = \frac{1}{1+\varepsilon^2 C_N^2\left(\frac{\Omega}{\Omega_p}\right)}$$

Where $C_N\left(\frac{\Omega}{\Omega_p}\right)$ is a Chebyshev polynomial which exhibits equal ripple in the pass band, N is the order of the polynomial as well as that of the filter, and ε determines the pass band ripple, which in decibels is given by, $passband\,ripple \le 10\log_{10}(1+\varepsilon^2)$

Chebyshev polynomial is given by

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

The expression for filter parameter ε is given by

$$\varepsilon = \left[\frac{1}{{\delta_1}^2} - 1\right]^{0.5}$$

The order of the filter

$$N \ge \frac{cosh^{-1}\left\{\frac{1}{\varepsilon}\left[\frac{1}{{\delta_2}^2} - 1\right]^{0.5}\right\}}{cosh^{-1}\left(\frac{\Omega_s}{\Omega_p}\right)}$$

Assume $\Omega_C = \Omega_p$

The transfer function H(s) is similar to Butterworth filter, but the coefficients bk and ck is given by

$$b_k = 2y_N \sin\left[\frac{(2k-1)\pi}{2N}\right]$$
$$c_k = {y_N}^2 + cos^2\left[\frac{(2k-1)\pi}{2N}\right]$$
$$c_0 = y_N$$

The parameter $y_N$ is given by

$$y_N = \frac{1}{2}\left\{\left[\left(\frac{1}{\varepsilon^2}+1\right)^{0.5} + \frac{1}{\varepsilon}\right]^{\frac{1}{N}} - \left[\left(\frac{1}{\varepsilon^2}+1\right)^{0.5} + \frac{1}{\varepsilon}\right]^{\frac{-1}{N}}\right\}$$

The parameter $B_k$ can be obtained from:

For N is even

$$\frac{A}{(1+\varepsilon^2)^{0.5}} = \prod_{k=1}^{N/2} \frac{B_k}{c_k}$$

For N is odd

$$A = \prod_{k=0}^{(N-1)/2} \frac{B_k}{c_k}$$

**Type II Chebyshev** Filter is characterized by the magnitude squared response is given by

$$|H(\Omega)|^2 = \frac{\varepsilon C_N\left(\frac{\Omega_s}{\Omega}\right)}{\left[1+\varepsilon^2 {C_N}^2\left(\frac{\Omega_s}{\Omega}\right)\right]^{0.5}}$$

Chebyshev polynomial is give as Type-II

The filter parameter ε is given by

$$\varepsilon = \frac{\delta_2}{\left(1-{\delta_2}^2\right)^{0.5}}$$

The order N of the filter

$$N \geq \frac{cosh^{-1}\left[\frac{1}{{\delta_2}^2}-1\right]^{0.5}}{cosh^{-1}\left(\frac{\Omega_s}{\Omega_p}\right)}$$

All other parameters are similar to Type-I filter.

## MATLAB functions:

***1) CHEB1ORD*** *Chebyshev Type I filter order selection.*
***[N, Wp] = 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 passband and has at leastRs dB of attenuation in the stopband. Wp and Ws are the passband and stopband edgefrequencies normalized from 0 to 1 (where 1 corresponds to pi radians/sample). For example,*
*Lowpass: Wp = .1, Ws = .2*
*Highpass: Wp = .2, Ws = .1*
*Bandpass: Wp = [.2 .7], Ws = [.1 .8]*
*Bandstop: Wp = [.1 .8], Ws = [.2 .7]*
*CHEB1ORD also returns Wp, the Chebyshev natural frequency to use with CHEBY1 to achieve the specifications.*
***[N, Wp] = CHEB1ORD(Wp, Ws, Rp, Rs, 's')*** *does the computation for an analog filter, in which case Wp and Ws are in radians/second.*

***2) CHEB2ORD*** *Chebyshev Type II filter order selection.*
***[N, Ws] = CHEB2ORD(Wp, Ws, Rp, Rs)*** *returns the order N of the lowest order digital Chebyshev Type II filter that loses no more than Rp dB in the passband and has at least Rs dB of attenuation in the stopband. Wp and Ws are the passband and stopband edge frequencies normalized from 0 to 1 (where 1 corresponds to pi radians/sample). For example,*

*Lowpass: Wp = .1, Ws = .2*
*Highpass: Wp = .2, Ws = .1*
*Bandpass: Wp = [.2 .7], Ws = [.1 .8]*
*Bandstop: Wp = [.1 .8], Ws = [.2 .7]*
*CHEB2ORD also returns Wst, the Chebyshev natural frequency to use with CHEBY2 to achieve the specifications.*
***[N, Ws] = CHEB2ORD(Wp, Ws, Rp, Rs, 's')** does the computation for an analog filter, in which case Wp and Ws are in radians/second.*

***3) CHEBY1** Chebyshev Type I digital and analog filter design.*
***[B,A] = CHEBY1(N,R,Wp)** designs an Nth order lowpass digital Chebyshev filter with R decibels of peak-to-peak ripple in the passband. CHEBY1 returns the filter coefficients in length N+1 vectors B (numerator) and A (denominator). The passband-edge frequency Wp must be 0.0 < Wp < 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.*
*If Wp is a two-element vector, Wp = [W1 W2], CHEBY1 returns an order 2N bandpass filter with passband W1 < W < W2.*
***[B,A] = CHEBY1(N,R,Wp,'high')** designs a highpass filter.*
***[B,A] = CHEBY1(N,R,Wp,'low')** designs a lowpass filter.*
***[B,A] = CHEBY1(N,R,Wp,'stop')** is a bandstop filter if Wp = [W1 W2].*
*When used with three left-hand arguments, as in [Z,P,K] = CHEBY1(...), the zeros and poles are returned in length N column vectors Z and P, and the gain in scalar K.*
*CHEBY1(N,R,Wp,'s'), CHEBY1(N,R,Wp,'high','s') and CHEBY1(N,R,Wp,'stop','s') design analog Chebyshev Type I filters. In this case, Wp is in [rad/s] and it can be greater than 1.0.*

*4) **CHEBY2** Chebyshev Type II digital and analog filter design.*
***[B,A] = CHEBY2(N,R,Wst)** designs an Nth order lowpass digital Chebyshev filter with the stopband ripple R decibels down and stopband-edge frequency Wst. CHEBY2 returns the filter coefficients in length N+1vectors B (numerator) and A (denominator).*
*The stopband-edge frequency Wst must be 0.0 < Wst < 1.0, with 1.0 corresponding to half the sample rate.*
*Use R = 20 as a starting point, if you are unsure about choosing R.*
*If Wst is a two-element vector, Wst = [W1 W2], CHEBY2 returns an order 2N bandpass filter with passband W1 < W < W2.*
***[B,A] = CHEBY2(N,R,Wst,'high')** designs a highpass filter.*
***[B,A] = CHEBY2(N,R,Wst,'low')** designs a lowpass filter.*
***[B,A] = CHEBY2(N,R,Wst,'stop')** is a bandstop filter if Wst = [W1 W2].*
*When used with three left-hand arguments, as in [Z,P,K] = CHEBY2(...), the zeros and poles are returned in length N column vectors Z and P, and the gain in scalar K.*
*CHEBY2(N,R,Wst,'s'), CHEBY2(N,R,Wst,'high','s') and CHEBY2(N,R,Wst,'stop','s') design analog Chebyshev Type II filters. In this case, Wst is in [rad/s] and it can be greater than 1.0.*

# 2. Procedure:

**Step1:** Get the digital filter specifications **(** $\delta_p \;\; \delta_s \;\; \omega_p \;\; \omega_s$ **)**

**Step2:** Convert to analog filter specifications with bilinear transformation.

**Step3:** Design analog transfer function Ha(s)

**Step4:** Transfer Ha(s) to H(z) since $s = \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)$

**Analog Butterworth approximation**

1. Start

2. Enter pass band ripple a1 and stop band ripple a2 in dB
3. Enter pass band edge and stop band edge frequencies (fp and fs) in Hz (or in nnormalized form in radians)
4. Enter sampling frequency in samples/second
5. Check the entered values are correct
6. Normalize the frequencies and compute analog frequencies (Ωp and Ωs)

   Find orders and cut-off frequencies of the analog filters using *buttord( )* function and display the values for verification
8. Compute the transfer functions in s-domain using *butter( )*
9. Transform the transfer functions to z-domain by *impinvar( )* and *bilinear( )*
10. Compute the frequency responses of each type using *freqz( )* function
11. Plot the each magnitude and phase responses
12. Stop

**MATLAB**

```
%Butterworth filter
clc;
clear all;
close all;
disp('IIR filter design using Butterworth Approximation');
%Inputs
ftype=input('Enter filter type (LPF, HPF, BPF or BSF): ');
switch ftype
case 'LPF'
fp=input('Enter pass band frequency fp in Hz ');
fs=input('Enter stop band frequency fs in Hz ');
%Error message
if fp<=0
error('Pass band edge must be larger than zero');
end
if fs<=fp
error('Stop band edge must be larger than pass band edge');
end
case 'HPF'
fp=input('Enter pass band frequency fp in Hz ');
fs=input('Enter stop band frequency fs in Hz ');
%Error message
if fs<=0
error('Pass band edge must be larger than 0');
end
if fs>=fp
error('Stop band edge must be smaller than pass band edge');
end
case 'BPF'
fs=input('Enter stop band frequencies fs1 and fs2 (fs2 >fs1) in Hz ');
fp=input('Enter pass band frequencies fp1 and fs2 (fp1 >fs1 and fp2<fp2) in Hz ');
if (fp(1) || fp(2))<=0
error('Pass band edge must be larger than 0');
end
if fp(1)<=fs(1)

error('Pass band edge must be larger than stop band edge');
end
if fp(2)>=fs(2)
error('Pass band edge must be smaller than stop band edge');
```

```
end
case 'BSF'
fs=input('Enter stop band frequencies fs1 and fs2 (fs2 >fs1) in Hz ');
fp=input('Enter pass band frequencies fp1 and fs2 (fp1 < fs1 and fp2 > fs2) in Hz ');
if (fs(1) || fs(2))<=0
error('Stop band edge must be larger than zero');
end
if fs(1)<=fp(1)
error('Stop band edge must be larger than pass band edge');
end
if fs(2)>=fp(2)
error('Stop band edge must be smaller than pass band edge');
end
end
Fs=input('Enter sampling frequency F in samples/sec ');
a1=input('Enter pass band ripple in dB ');
a2=input('Enter stop band ripple in dB ');
%conversion and normalization of frequencies
%pi radians/second
wp=2*fp/Fs;
ws=2*fs/Fs;
%conversion of frequency for transformation
trans=input('Enter transformation type (IIT or BT): ');
switch trans
case 'IIT'
%Analog frequencies by IIT
Wp=wp*Fs*pi;
Ws=ws*Fs*pi;
case 'BT'
%Analog frequencies by BT
Wp=2*Fs*tan(wp*pi/2);
Ws=2*Fs*tan(ws*pi/2);
end
%Order and cut-off frequencies of filter
[N,Wc]=buttord(Wp,Ws,a1,a2,'s');
%Analog Filter response
switch ftype
case 'LPF'
[b,a]=butter(N,Wc,'s');
case 'HPF'
[b,a]=butter(N,Wc,'high','s');
case 'BPF'
[b,a]=butter(N,Wc,'bandpass','s');

case 'BSF'
[b,a]=butter(N,Wc,'stop','s');
end
%Transformation, Digital filter response & Frequency response
switch trans
case 'IIT'
[bz,az]=impinvar(b,a,Fs);
[H,f]=freqz(bz,az,512,Fs);
case 'BT'
[bz,az]=bilinear(b,a,Fs);
[H,f]=freqz(bz,az,512,Fs);
end
```

```
%display N and H(z)
disp('Order of the filter is:');
disp(N);
disp('Filter coefficients are:');
disp('a');
disp(az);
disp('b');
disp(bz);
%Plotting responses
subplot(2,1,1);
plot(f,abs(H));
xlabel('f in Hz');
ylabel('|H(w)|');
title(['Magnitude response of IIR ',ftype,' filter with Butterworth approximation using ',trans]);
grid on;
subplot(2,1,2);
plot(f,angle(H));
xlabel('f in Hz');
ylabel('Phase of H(w)');
title(['Phase response of IIR ',ftype,' filter with Butterworth approximation using ',trans]);
grid on;
```

**Analog Chebyshev approximation**

1. Start
2. Enter pass band ripple a1 and stop band ripple a2 in dB
3. Enter pass band edge and stop band edge frequencies (fp and fs) in Hz (or in normalized form in radians)
4. Enter sampling frequency in samples/second
5. Check the entered values are correct
6. Normalize the frequencies and compute analog frequencies ($\Omega$p and $\Omega$s)
7. Find orders and cut-off frequencies of the analog filters using *cheb1ord( )* and *cheb2ord( )* function and display the values for verification
8. Compute the transfer functions in s-domain using *cheby1( )* and *cheby2( )*
9. Transform the transfer functions to z-domain by *impinvar( )* and *bilinear( )*
10. Compute the frequency responses of each type using *freqz( )* function
11. Plot the each magnitude and phase responses
12. Stop

**MATLAB**

```
%Chebychev filter
clc;
clear all;
close all;
disp('IIR filter design using Chebyshev Approximation');
%Inputs
ftype=input('Enter filter type (LPF, HPF, BPF or BSF): ');
switch ftype
case 'LPF'
fp=input('Enter pass band frequency fp in Hz ');
fs=input('Enter stop band frequency fs in Hz ');
%Error message
```

```
if fp<=0
error('Pass band edge must be larger than 0');
end
if fs<=fp
error('Stop band edge must be larger than pass band edge');
end
case 'HPF'
fp=input('Enter pass band frequency fp in Hz ');
fs=input('Enter stop band frequency fs in Hz ');
%Error message
if fs<=0
error('Pass band edge must be larger than 0');
end
if fs>=fp
error('Stop band edge must be smaller than pass band edge');
end
case 'BPF'
fs=input('Enter stop band frequencies fs1 and fs2 (fs2 >fs1) in Hz ');
fp=input('Enter pass band frequencies fp1 and fs2 (fp1 >fs1 and fp2<fp2) in Hz ');
if (fp(1) || fp(2))<=0
error('Pass band edge must be larger than 0');
end
if fp(1)<=fs(1)
error('Pass band edge must be larger than stop band edge');
end
if fp(2)>=fs(2)
error('Pass band edge must be smaller than stop band edge');
end
case 'BSF'
fs=input('Enter stop band frequencies fs1 and fs2 (fs2 >fs1) in Hz ');
fp=input('Enter pass band frequencies fp1 and fs2 (fp1 < fs1 and fp2 > fs2) in Hz ');
if (fs(1) || fs(2))<=0
error('Stop band edge must be larger than 0');
end
if fs(1)<=fp(1)
error('Stop band edge must be larger than pass band edge');
end
if fs(2)>=fp(2)
error('Stop band edge must be smaller than pass band edge');
end
end
Fs=input('Enter sampling frequency Fs in samples/sec ');
a1=input('Enter pass band ripple in dB ');
a2=input('Enter stop band ripple in dB ');
%conversion and normalization of frequencies
%pi radians/second
wp=2*fp/Fs;

ws=2*fs/Fs;
%conversion of frequency for transformation
trans=input('Enter transformation type (IIT or BT): ');
switch trans
case 'IIT'
%Analog frequencies by IIT
Wp=wp*Fs*pi;
Ws=ws*Fs*pi;
```

```
case 'BT'
%Analog frequencies by BT
Wp=2*Fs*tan(wp*pi/2);
Ws=2*Fs*tan(ws*pi/2);
end
%Cheby1 or Cheby2
type=input('Enter Chebyshev filter type (type-1 or type-2): ');
switch type
case 'type-1'
%Order and cut-off frequencies of filter
[N,Wc]=cheb1ord(Wp,Ws,a1,a2,'s');
switch ftype
case 'LPF' %Analog Filter response
[b,a]=cheby1(N,a1,Wc,'s');
case 'HPF'
[b,a]=cheby1(N,a1,Wc,'high','s');
case 'BPF'
[b,a]=cheby1(N,a1,Wc,'bandpass','s');
case 'BSF'
[b,a]=cheby1(N,a1,Wc,'stop','s');
end
case 'type-2'
%Order and cut-off frequencies of filter
[N,Wc]=cheb2ord(Wp,Ws,a1,a2,'s');
switch ftype
case 'LPF'
%Analog Filter response
[b,a]=cheby2(N,a2,Wc,'s');
case 'HPF'
[b,a]=cheby2(N,a2,Wc,'high','s');
case 'BPF'
[b,a]=cheby2(N,a2,Wc,'bandpass','s');
case 'BSF'
[b,a]=cheby2(N,a2,Wc,'stop','s');
end
end
%Transformation, Digital filter response & Frequency response
switch trans
case 'IIT'

[bz,az]=impinvar(b,a,Fs);
[H,f]=freqz(bz,az,512,Fs);
case 'BT'
[bz,az]=bilinear(b,a,Fs);
[H,f]=freqz(bz,az,512,Fs);
end
%display N and H(z)
disp('Order of the filter is:');
disp(N);
disp('Filter coefficients are:');
disp('a');
disp(az);
disp('b');
disp(bz);
%Plotting responses
subplot(2,1,1);
```

```
plot(f,20*log10(abs(H))); % Gain in dB
xlabel('f in Hz');
ylabel('|H(w)|');
title(['Magnitude response of IIR ',ftype,' filter with Chebyshev ',type,' approximation using
',trans]);
grid on;
subplot(2,1,2);
plot(f,angle(H));
xlabel('f in Hz');
ylabel('Phase of H(w)');
title(['Phase response of IIR ',ftype,' filter with Chebyshev ',type,' approximation using',trans]);
```

# 5) Exercise:

**Task#1 IIR Lowpass Filter Design:**

Consider the low-pass discrete-time filter specification,

$$0.99 \le |H(e^{jw})| \le 1.01 \quad , \quad |w| \le 0.4\pi,$$
$$|H(e^{jw})| \le 0.001 \quad , \quad 0.6\pi \le |w| \le \pi$$

Design using MATLAB this low pass filter by butterworth, Chebyshev( I and II), elliptic approximation. Plot the gain response, detailted plot of magnitude in pass-band and pole-zero plot for each approximation.

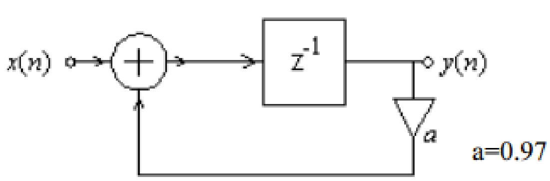
**Task#2 IIR bandpass Filter Design:**

Consider the band-pass discrete time filter specification,

$$0.99 \le |H(e^{jw})| \le 1.01 \quad , \quad 0.45\,\pi \le |w| \le 0.6\pi,$$
$$|H(e^{jw})| \le 0.001 \quad , \quad 0.75\pi \le |w| \le \pi,\ 0 \le |w| \le 0.3\,\pi$$

Design using MATLAB this band-pass filter by butterworth, Chebyshev( I and II), elliptic approximation. Plot the gain response, detailted plot of magnitude in pass-band and pole-zero plot for each approximation

**Task#3:** Generate the code for the given IIR filter, having the coefficient b0=0, b1= 1 and a0=1, a1=0.97. Taking impulse signal as x(n) having 80 samples.




$$y(n) = x(n\text{-}1) + ay(n\text{-}1)$$

**Task#04:**

**Given a=[1 3 5 7 9 11 13 15];b=[2 4 6 8 10 12 14]**

a) **Implement SOP(Sum of Products) in 6713 Simulator**
b) **Implement SOP on DSK6713.**

**Student's feedback:** Purpose of feedback is to know the strengths and weaknesses of the system for future improvements. This feedback is for the 'current lab session'. Circle your choice:

[-3 = Extremely Poor, -2 = Very Poor, -1 = Poor, 0 = Average, 1 = Good, 2 = Very Good, 3 = Excellent]:

The following table should describe your experience with:

| **S#** | **Field** | **Rating** | **Describe in words if required** |
|---|---|---|---|
| 1 | Overall Session | -3 -2 -1 0 1 2 3 | |
| 2 | Lab Instructor | -3 -2 -1 0 1 2 3 | |
| 3 | Lab Staff | -3 -2 -1 0 1 2 3 | |
| 4 | Equipment | -3 -2 -1 0 1 2 3 | |
| 5 | Atmosphere | -3 -2 -1 0 1 2 3 | |

Any other valuable feedback: ____________________________________________

____________________________________________

____________________________________________

____________________________________________

Student's Signature: ______________________________

| **MARKS AWARDED** | **Attitude** | **Neatness** | **Correctness of results** | **Initiative** | **Originality** | **Conclusion** | **TOTAL** |
|---|---|---|---|---|---|---|---|
| TOTAL | **10** | **10** | **10** | **20** | **20** | **30** | **100** |
| EARNED | | | | | | | |

Lab Instructor's Comments:____________________________________________

____________________________________________

____________________________________________

____________________________________________

Lab Instructor's Signature: ______________________________