NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Electrical and Electronic Engineering

DESIGNING OF IIR FILTERS USING MATLAB TECHNIQUES

GRADUATION PROJECT
EE-400

Student: Waqar Ahmed (960832)

Supervisor: Prof. Dr. Fakherddin Mamedov

NICOSIA-2000

ACKNOWLEDGEMENTS

I am grateful to all the people in my life who have, supported me, advised me, taught me and who have always encouraged me to follow my dreams and ambitions. My dearest parents, my brothers and sister, my friends and my tutors. They have taught me that no dream is unachievable. As in the words of Walt Disney "If you can dream it, you can do it."

I wish to thank my advisor, Prof. Dr. Fakhraddin Mamedov, for intellectual support, encouragement, and enthusiasm, which made this project possible, and his patience for correcting both my stylistic and scientific errors.

My sincerest thanks must go to my friends, Jamal Fatih, Rabyan Ghayas, Kashif Malik and Nauman Shaukat Chaudhry who shared their suggestions and evaluations throughout the completion of my project. The comments from these friends enabled me to present this project successfully.

And above, I thank God for giving me stamina and courage to achieve my objectives.

Abstract

The project attempts to put together the theories, techniques, and procedures which can be used to design infinite impulse response filters. The mathematical tools for the designing of digital filters are introduced briefly, and then the designs are carried out using matlab. It's divided into four sections.

Sections 1 gives a brief introduction of filtering operation, the ideal models that can be used to design any type of filter depending on the specifications of the application. At the end designing method for IIR filters is briefly explained, which is firstly to design the analog filter then transform to digital filter using analog to digital filter transformations, and then transform it to other frequency selective filters by using frequency transformations

Sections 2 describe the designing equations for analog prototype filters (chebyshev, butterworth 1 & 2 and elliptic). Then these prototypes are designed using matlab techniques. At the end a comparison is given to choose the best suitable filter under the specifications required.

Sections 3 deals with the transformation of filters from analog to digital domain, two techniques are explained, which are bilinear transformation and impulse invariance transformation. Both are then are then demonstrated by using matlab techniques. At the end lowpass digital filters are designed using matlab techniques.

Sections 4 explain the frequency band transformation, zmapping is explained for this purpose, and at the end matlab is used to design other frequency selective filters.

CONTENTS

1. INTRODUCTION

1.1 Filtering Process:

Filtering is a process by which the frequency spectrum of signal can be modified, reshaped, or manipulated according to some desired specification. It may entail amplifying or attenuating a range of frequency components, rejecting or isolating one specific or attenuating a range of frequency component, etc. The uses of filtering are manifold, e.g., to eliminate signal contamination such as noise to remove signal distortion brought about by an imperfect transmission channel or by inaccuracies in measurement, to separate two or more distinct signals which were purposely mixed in order to maximize channel utilization, to resolve signals into their frequency components, to demodulate signals, to convert discrete-time signals into continuous-time signals, and to band-limited signals.

The digital filter is a digital system that can be used to filter discrete-time signals. It can be implemented by mean of software (computer programs) or by means of dedicated hardware, and in either case it can be used to filter real-time signals or non-real-time (recorded) signals.

Software digital filters made their appearance along with the first digital computer in the late forties, although the name digital filter did not emerge until the midsixties. Early in the history of the digital computer many of the classical numerical analysis formulas of Newton, Starling, Everett, and others were used to carry out interpolation, differentiation, and integration of function (signals) represented by mean of sequences of numbers (discrete-time signals). Since interpolation, differentiation, or integration of a signal represents a manipulation of the frequency spectrum of the signal, the subroutines or programs constructed to carry out these operations were essentially digital filters. In subsequent years, many complex and highly sophisticated algorithms and programs were developed to perform a variety of filtering tasks in numerous applications, e.g., data smoothing and prediction, pattern recognition, electrocardiogram processing, and spectrum analysis. In fact, as time goes on, interest in the software digital filter is becoming progressively more intense while its applications are increasing at an exponential rate. band-limited continuous-time signals can be transformed into a discrete-time signals by means of sampling. Conversely, the discrete-time signals so generated can be used to regenerate the original continuous-time signals by means of interpolation, by virtue of Shannon's sampling theorem. As a consequence, hardware Digital's filters can be used to perform real-time filtering tasks, which in the not too distant past were performed almost exclusively by analog filters. The advantages to be gained are the traditional advantages associated with digital systems in general:

1. Component tolerances are uncritical.
2. Component drift and spurious environmental signals have no influence on the system performance.
3. Accuracy is high.
4. Physical size is small.
5. Reliability is high.

A very important additional advantage of digital filters is the ease with which filter parameters can be changed in order to change the filter characteristics. This feature allows one to design programmable filters, which can be used to perform a multiplicity of filtering tasks. Also one can design new types of filters such as adaptive filters. The main disadvantage of hardware Digital's at present is their relatively high cost. However, with the tremendous advancements in the domain of large-scale integration, the cost of hardware digital filters is likely to drop drastically in the not too distant future, When this happens, hardware Digital's filters will replace analog filters in many more applications.[1]

1.2 Applications of Digital Filters:

Digital filters in the form of the form of software have been used extensively in the past and will no doubt continue to be used in the future at a progressively increasing rate. Typical applications are:

1. Data smoothing and prediction
2. Image enhancement
3. Pattern recognition
4. Speech processing
5. Processing of telemetry signals
6. Processing of biomedical signals
7. Simulation of analog systems

Programmable hardware digital filters have already made their appearance in the form of digital-signal processors such as FFT processors, frequency synthesizers, and wave analyzers. Many other applications are anticipated for the future, especially in the domain of instrumentation.

Nonprogrammable digital filters are currently considers as possible replacement of analog filters in any communication subsystems [2,3] with present-day technology, digital filters are still more expenses then analog filters except for some low-frequency, where extensive multiplexing of hardware is possible. Never the less, the gains in accuracy and stability of operation may some times justify the extra expanse. With the present trends in the fabrication of LSI circuit continuing, the cost of digital hardware is bound to drop drastically in the not too distant future. At that time digital filters will become more attractive than analog filters in many more application. It is nit expected filters will replace analog filter altogether, e.g., microwave filters! Instead, like LC, crystal, mechanical, monolithic, and active filters, digital filters will become and invaluable addition to the gab of tricks available to the filter designer.

1.3 Ideal Designing Models:

The ideal designing models represent the designing of lowpass, high-pass, band-pass, band-stop, and all-pass filters. These are graphically interpreted in Figure 1-1. Their shape represents the steady-state magnitude-frequency response of a filter with a transfer function of H (ω) = H(s) $|_{s=j\Omega}$ where ω denote frequency measured in radians per second.

The mathematical specification of each ideal filter is summarized as,

Ideal Low-pass $$|H(\omega)| = \begin{cases} 1 & \text{if } \omega \in [\text{-B}, \text{B}] \\ 0 & \text{otherwise} \end{cases} \tag{1-1}$$

Ideal High-pass $$|H(\omega)| = \begin{cases} 0 & \text{if } \omega \in [\text{-B}, \text{B}] \\ 1 & \text{otherwise} \end{cases} \tag{1-2}$$

Ideal Band-pass $$|H(\omega)| = \begin{cases} 1 & \text{if } \omega \in [\text{-B}_2, \text{-B}_1] \text{ or } \omega \in [B_1, B_2] \\ 0 & \text{otherwise} \end{cases} \tag{1-3}$$

Ideal Band-stop $$|H(\omega)| = \begin{cases} 0 & \text{if } \omega \in [\text{-B}_2, \text{-B}_1] \text{ or } \omega \in [B_1, B_2] \\ 1 & \text{otherwise} \end{cases} \tag{1-4}$$

All-pass $$|H(\omega)| = 1 \text{ for all } \omega \in [-\infty, \infty] \tag{1-5}$$

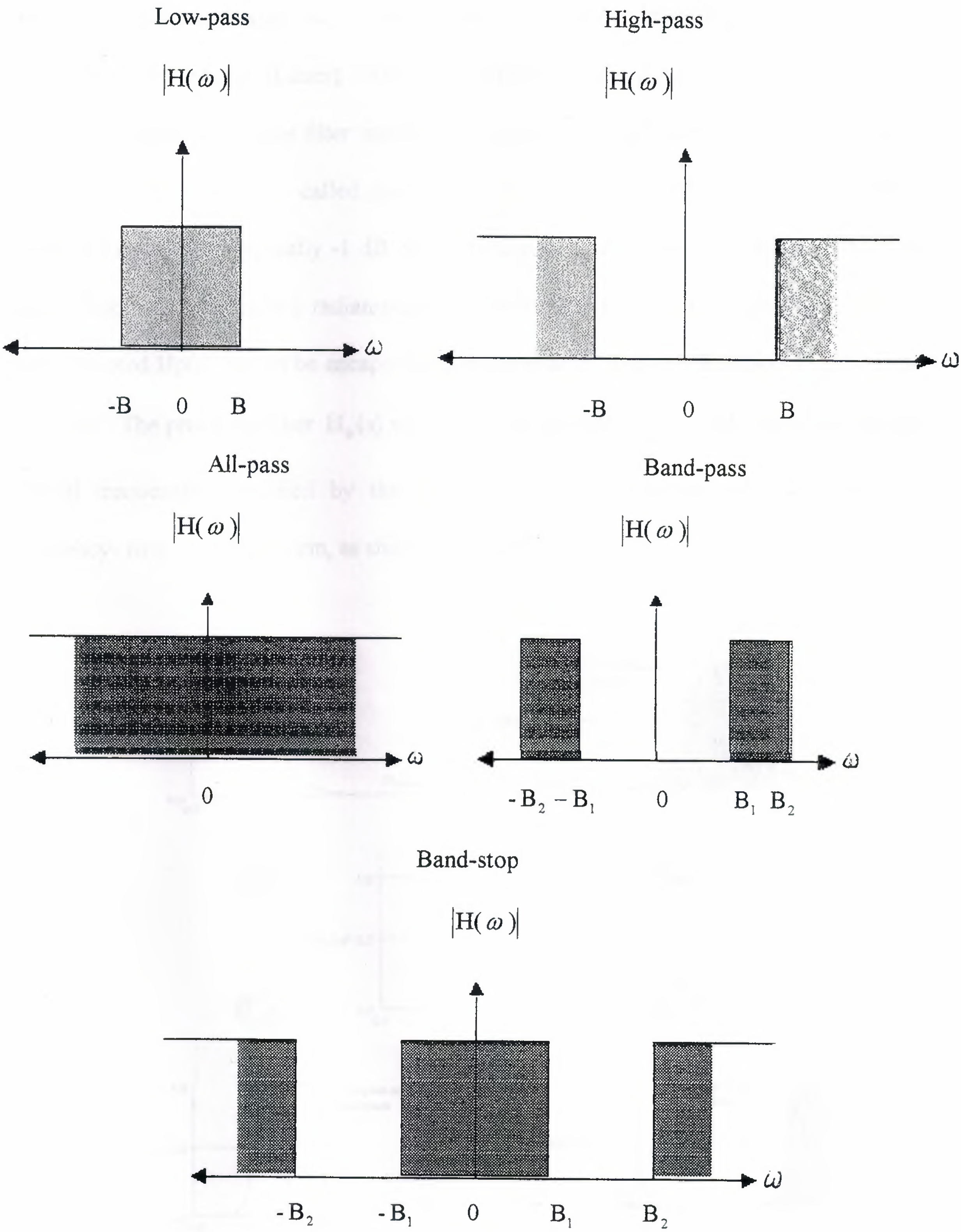


Figure 1-1 Basic ideal filter types.

Filter design is often based on the use of several well-known models called Butterworth, Chebyshev, and elliptic (Cauer) filters. To standardize the design procedure, a set of normalized analog prototype filter models was agreed upon and reduced to tables, charts, and graphs. These models, called prototypes, were all developed as low-pass systems having a known gain (typically -1 dB or -3 dB pass-band attenuation) at a known critical cut-off frequency (typically 1 radian/second). The transfer function of an analog prototype filter, denoted Hp(s), would be encapsulated in a standard table as a function of filter type and order. The prototype filter $H_p(s)$ would then be mapped into a final filter H(s) having critical frequencies specified by the designer [2]. The mapping rule is known as, frequency- frequency transform, as shown in figure.

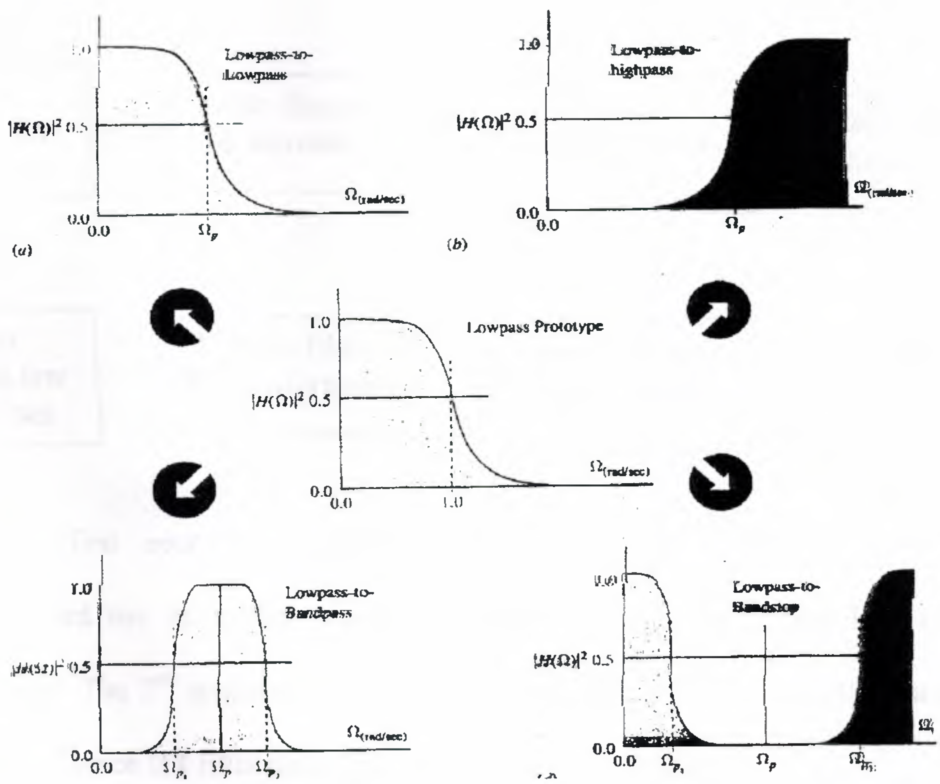
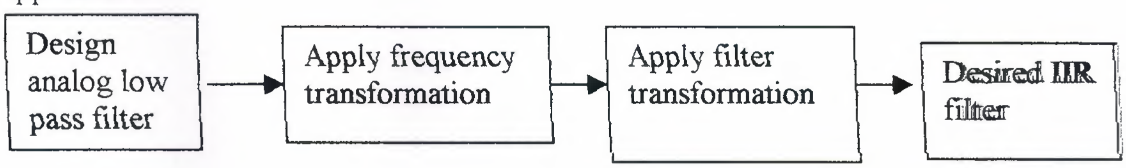


Figure 1-2. Frequency Band Transformation

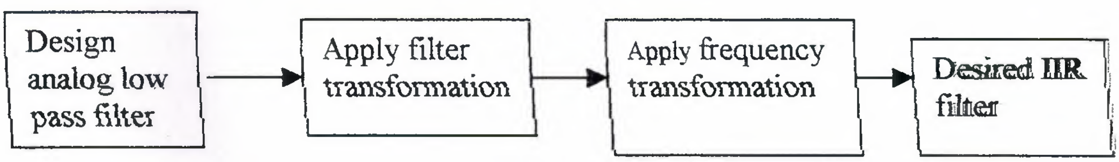
1.4 IIR Filter Design:

IIR filters has infinite duration impulse response, hence they can be matched to analog filters, all of which have generally infinitely long impulse response. Therefore the basic technique of IIR filter design is to transform well known analog filters to digital filters using complex valued mapping [4]. The advantage of this technique lies in the fact that both analog filter designs tables and mappings are extensively available. This basic technique is known as analog to digital filter transformation. However analog filter tables are available only for low pass filters. When the requirement is to design other frequency selective filters (high pass, band pass, band stop etc.), the band transformation is applied to low pass filters. The transformations are also complex valued mapping, and they are available in literature. There are two approaches to design IIR filters.

Approach 1:



Approach 2:



The first approach is used in matlab [5,6,7,8] to design IIR filters. A straightforward use of matlab functions does not provide an insight into the design methodology. The 2nd approach is used in following chapters to give mathematics behind their design. Hence IIR filter design will follow the following steps.

- Design analog low pass filter.

- Apply filter transformation to obtain digital low pass filter.
- Apply frequency band transformation to obtain other filters from digital low pass filter.

The main problem with these approaches is that we have no control over the phase characteristics of the IIR filter. Hence IIR filter is treated only as magnitude–only design. More sophisticated techniques, which can be used simultaneously to achieve magnitude and phase response. This requires advance optimization tools.

2. ANALOG PROTOTYPES

2.1 Overview:

IIR filter design techniques rely on existing analog filters to obtain digital filters. We designate these filters as analog *prototype* filters. Three prototypes are widely used in practice. In this chapter a brief summary of the low pass versions of these prototypes: Butterworth low pass, Chebyshev low pass (type I II), and Elliptic low pass is given. Then matlab functions are used for the designing of these filters. The phase response of all above filters is nonlinear. Bessel filter is implemented for linear phase response, so this filter is also introduced. [3,5,6,8]

2.2 Butterworth Approximation:

The magnitude-squared response of an analog low-pass Butterworth filter $H_a(s)$ of Nth order is given by,

$$|H_a(j\Omega)|^2 = \frac{1}{1+(\Omega/\Omega_c)^{2N}} \qquad (2\text{-}1)$$

It can be easily shown that the first 2N-1 derivatives of $|H_a(j\Omega)|^2$ at $\Omega = 0$ are equal to zero, and as a result, the Butterworth filter is said to have a maximally-flat magnitude at $\Omega = 0$. The gain of the Butterworth filter in dB is given by,

$g(\Omega) = 10 \log_{10} |H_a(j\Omega)|^2$ dB.

A dc i.e., at $\Omega = 0$, the gain in dB is equal to zero, and at $\Omega = \Omega c$, the gain is,

$g(\Omega c) = 10 \log_{10} (1/2) = -3.0103 \cong -3$ dB

and therefore, it is often called the 3-dB cutoff frequency. Since the derivative of the squared-magnitude response, or equivalently, of the magnitude response is always negative for positive values of Ω, the magnitude response, is monotonically decreasing with increasing Ω. For $\Omega >> \Omega_c$, the squared-magnitude function can be approximated by:

$$|H_a(j\Omega)|^2 = \frac{1}{1+(\Omega/\Omega_c)^{2N}}$$

The gain $g(\Omega_2)$ in dB at $\Omega_2 = 2\Omega_1$ with $\Omega_1 >> \Omega c$ is given by,

$$g(\Omega_c) = -20\log_{10}\left(\frac{\Omega_2}{\Omega_c}\right)^{2N} = g(\Omega_1) - 6N \text{ dB},$$

where $g(\Omega_1)$ is the gain in dB at Ω_1. As a result, the gain roll-off per octave in the stop-band decreases by 6 dB, or equivalently, by 20 dB per decade for an increase of the filter order by one. In other words, the pass-band and the stop-band behaviors of the magnitude response improve with a corresponding decrease in the transition band as the fitter order N increases..

The two parameters completely characterizing a Butterworth filter are therefore the 3-dB cut-off frequency Ω_c and the order N. These are determined from the specified pass-band edge Ω_p, the minimum pass-band magnitude $1/\sqrt{1+\varepsilon^2}$, the stop-band edge Ω_s, and the maximum stop-band ripple 1/A. From Eq. (2-1) we get,

$$|H_a(j\Omega_p)|^2 = \frac{1}{1+(\Omega_p/\Omega_c)^{2N}} = \frac{1}{1+\varepsilon^2} \qquad (2\text{-}2a)$$

$$|H_a(j\Omega_s)|^2 = \frac{1}{1+(\Omega_s/\Omega_c)^{2N}} = \frac{1}{A^2} \qquad (2\text{-}2b)$$

By solving the above we get the expression for the order N as,

$$N = \frac{1}{2}\frac{\log_{10}\left[\left(A^2 - 1\right)/\varepsilon^2\right]}{\log_{10}\left(\Omega_s/\Omega_p\right)} = \frac{\log_{10}\left(1/k_1\right)}{\log_{10}\left(1/k\right)} \quad (2\text{-}3)$$

Since the order N of the filter must be an integer, the value of N computed using the above expression is rounded up to the next higher integer. This value of N cam be used next in either Eq. (2-2a) or (2-2b) to solve for the 3-dB cutoff frequency Ω_c. If it is used in Eq. (2-2a), the pass-band specification is met exactly, whereas the stop-band specification is exceeded. On the Other hand, if it is used in Eq. (2-2b), the stop-band specification is met exactly, whereas the pass-band specification is exceeded.

The expression for the transfer function of the Butterworth low-pass filter is given by,

$$H_a(s) = \frac{C}{D_N(s)} = \frac{\Omega_c^N}{s^N + \sum_{\lambda=0}^{N-1} d_\lambda s^\lambda} = \frac{\Omega_c^N}{\prod_{\lambda=1}^{N}(s - p_\lambda)} \quad (2\text{-}4)$$

Where,

$$p\lambda = \Omega_c e^{j[\pi(N+2\lambda-1)/2N]}, \quad \lambda = 1,2,\ldots,N \quad (2\text{-}5)$$

The denominator $D_N(s)$ of Eq. (2-4) is known as the Butterworth polynomial of order N and is easy to compute.

2.3 Chebyshev Approximation:

The approximation error is defined as the difference between the ideal brickwall characteristic and the actual response, is minimized over a prescribed band of frequencies. In fact, the magnitude error is equiripple in the band. There are two types of Chebyshev transfer functions. In the approximation 1, the magnitude characteristic is equiripple in the pass-band and monotonic in the stop-band, whereas in type 2 approximation2,the magnitude response is monotonic in the pass-band and equiripple in the stop-band.

2.3.1 Type 1 Chebyshev Approximation:

The type 1 Chebyshev transfer function $H_a(s)$ has a magnitude response given by,

$$|H_a(j\Omega)|^2 = \frac{1}{1+\varepsilon^2 T_N^2(\Omega/\Omega_p)}, \qquad (2\text{-}6)$$

Where $T_N(\Omega)$ is the Chebyshev polynomial of order N:

$$T_N(\Omega) = \begin{cases} \cos(N\cos^{-1}\Omega), & |\Omega| \le 1, \\ \cosh(N\cosh^{-1}\Omega), & |\Omega| > 1, \end{cases} \qquad (2\text{-}7)$$

The above polynomial can also be derived by recurrence relation given by,

$$T_r(\Omega) = 2\Omega\, T_{r-1}(\Omega) - T_{r-2}(\Omega), \qquad r \ge 2, \qquad (2\text{-}8)$$

with $T_0(\Omega) = 1$ and $T_1(\Omega) = \Omega$

The zeros are on the $j\Omega$-axis and are given by,

$$z_\lambda = j\frac{\Omega_s}{\cos\left[\frac{(2\lambda-1)\pi}{2N}\right]}, \qquad \lambda = 1,2,\ldots,N. \qquad (2\text{-}9)$$

If N is odd, then for $\lambda = (N+1)/2$, the zero is at $s = \infty$. The poles are located at,

$$p_\lambda = \sigma_\lambda + j\Omega_\lambda, \qquad \lambda = 1,2,\ldots,N, \qquad (2\text{-}10)$$

Where:

$$\sigma_\lambda = \frac{\Omega_s \alpha_\lambda}{\alpha_\lambda^2 + \beta_\lambda^2}, \qquad \Omega_\lambda = \frac{\Omega_s \beta_\lambda}{\alpha_\lambda^2 + \beta_\lambda^2}, \tag{2-11a}$$

$$\alpha_\lambda = -\Omega_p \varsigma \sin\left[\frac{(2\lambda-1)\pi}{2N}\right], \qquad \beta_\lambda = \Omega_p \xi \cos\left[\frac{(2\lambda-1)\pi}{2N}\right], \tag{2-11b}$$

$$\varsigma = \frac{\gamma^2 - 1}{2\gamma}, \qquad \xi = \frac{\gamma^2 + 1}{2\gamma}, \qquad \gamma = \left(A + \sqrt{A^2 - 1}\right)^{1/N}. \tag{2-11c}$$

The order N of the Type 2 Chebyshev low-pass filter is determined from given ε, Ω_s, and A using Eq. (2- 11).

2.3.2 Type 2 Chebyshev Approximation:

The Type 2 Chebyshev magnitude response, also known as the inverse Chebyshev response, exhibits a monotonic behavior in the pass-band with a maximally flat response at $\Omega = 0$ and an equiripple behavior in the stop-band. The square-magnitude response expression here is given by,

$$|H_a(j\Omega)|^2 = \frac{1}{1 + \varepsilon^2 \left[\frac{T_N(\Omega_s/\Omega_p)}{T_N(\Omega_s/\Omega)}\right]^2} \tag{2-12}$$

The transfer function of a Type 2 Chebyshev low-pass filter is no longer an all-pole function and has both poles and zeros. If we write,

$$H_a(s) = C_0 \frac{\prod_{\lambda=1}^{N} (s - z_\lambda)}{\prod_{\lambda=1}^{N} (s - p_\lambda)} \tag{2-13}$$

2.4 Elliptic Approximation:

An elliptic filter, also known as a Cauer filter, has an equiripple pass-band and an equiripple stop-band magnitude response. The transfer function of an elliptic filter meets a given set of filter specifications, pass-band edge frequency Ω_p, stop-band edge frequency, pass-band ripple Ω_s, and minimum stop-band attenuation A, with the lowest filter order N. The theory of elliptic filter approximation is mathematically quite involved. The square-magnitude response of an elliptic low-pass filter is given by,

$$|H_a(j\Omega)|^2 = \frac{1}{1+\varepsilon^2 R_N^2(\Omega/\Omega_p)} \qquad (2\text{-}14)$$

where $R_N(\Omega)$ is a rational function of order N satisfying the property $R_N(1/\Omega)= 1/R_N(\Omega)$, with the roots of its numerator lying within the interval $0 < \Omega < 1$ and the roots of its denominator lying in the interval $1 < \Omega < \infty$. For most applications, the filter order meeting a given set of specifications of pass-band edge frequency Ω_p, pass-band ripple ε, stop-band edge frequency Ω_s, and the minimum stop-band ripple A can be, estimated by using the approximate formula:

$$N \cong \frac{2\log_{10}(4/k_1)}{\log_{10}(1/p)} \qquad (2\text{-}15)$$

where k_1 is the discrimination parameter and ρ is computed as follows :

$$k' = \sqrt{1-k^2} \qquad (2\text{-}16a)$$

$$\rho_0 = \frac{1-\sqrt{k'}}{2(1+\sqrt{k'})} \qquad (2\text{-}16b)$$

$$\rho = \rho_0 + 2(\rho_0)^5 + 15(\rho_0)^9 + 150(\rho_0)^{13}. \qquad (2\text{-}16c)$$

in Eq.(2-16a), k is the selective parameter.

2.5 Linear-Phase Approximation:

In previous three approximations the techniques are for developing analog low-pass transfer functions meeting specified magnitude or gain response specifications without any concern for their phase responses. In a number of applications it is desirable that the analog low-pass, filter being designed have a linear-phase characteristic in the pass-band, in addition to approximating the magnitude specifications. One way to achieve this goal is to cascade an analog all-pass filter with the filter designed to meet the magnitude specifications, so that the phase response of the overall cascade realization approximates linear-phase response in the pass-band. This approach increases the overall hardware complexity of the analog filter and may not be desirable for designing an analog anti-aliasing filter in some A/D conversion or designing an analog reconstruction filter in D/A conversion applications. It is possible to design a low-pass filter that approximates a linear-phase characteristic in the pass-band but with a poorer magnitude response than that can be achieved by the previous three techniques. Such a filter has an all-pole transfer function of the form:

$$H(s) = \frac{d_0}{B_N(s)} = \frac{d_0}{d_0 + d_1 s + \ldots\ldots + d_{N-1} s + s^N} \qquad \text{(2-17)}$$

and provides a maximally flat approximation to the linear-phase characteristic at $\Omega = 0$,

I.e., has a maximally flat constant group delay at dc ($\Omega = 0$). For a normalized group delay of unity at dc, the denominator polynomial fi $B_N(s)$ of the transfer function, called the Bessel polynomial, can be derived via the recursion relation:

$$B_N(s) = (2N-1)\,B_N - 1(s) + s^2 B_{N-2}(s) \qquad \textbf{(2-18)}$$

starting with $B_1(s) = s + 1$ and $B_2(s) = s^2 + 3s + 3$. Alternatively, the coefficients of the Bessel polynomial $B_N(s)$ can be found from:

$$d_\lambda = \frac{(2N-\lambda)!}{2^{N-\lambda}\lambda!(n-\lambda)!}, \qquad \lambda = 0,1,....,N-1 \qquad \textbf{(2-19)}$$

These filters are often referred to as Bessel filters. [2]

2.6 Analog filter design using matlab:

Butterworth filter:

The M-file functions for butterworth analog filters are

```
  [z,p,k]=buttap (N)
[num,den]=butter (N,Wn,'s')
[num,den]=butter (N,Wn,filtr type,'s')
[N,Wn]=buttord (Wp,Ws,Rp,Rs,'s')
```

The buttap(N) determines the pole zero and gain factor of transfer function. Alternatively we can use butter (N,Wn,'s') to design an order N transfer function with prescribed cut-off frequency of 3-dB at Wn rad/sec, the outputs are numerator and denominator polynomials coefficient vectors in descending order or s. If we wish to design a filter other than lowpass we can use the function butter(N,.Wn,filter type,'s') .The function buttord (Wp,Ws,Rp,Rs,'s') computes the lowest order of the filter under given specifications.

Type 1 Chebyshev Filter:

The matlab function for the designing of chebyshev type 1 filters are:

[z,p,k]=cheb1ap(N,Rp)

[num,den]=cheby1(N,Rp,Wn,'s')

[num,den]=cheby1 (N,Rp,Wn,filtertype,'s')

[N,Wn]=cheb1ord(Wp,Ws,Rp,Rs,'s')

The cheby1(N,Rp) determines the pole zero and gain factor of transfer function with order N and passband ripples Rp dB. Alternatively we can use cheby1 (N, Wn,'s') to design an order N transfer function with prescribed cut-off frequency of 3-dB at Wn rad/sec, the outputs are numerator and denominator polynomials coefficient vectors in descending order or s. If we wish to design a filter other than lowpass we can use the function cheby1 (N,Wn,filter type,'s') .The function cheb1(Wp,Ws,Rp,Rs,'s') computes the lowest order of the filter under given specifications.

Chebyshev type2 filters:

The matlab functions type2 chebyshev filters are:

[z,p,k]=cheby2ap(N,Rs)

[num,den]=cheby2(n,Rs,Wn,'s')

[num,den]=cheby2(n,Rs,Wn,filtertype,'s')

[N,Wn]=cheb2ord(Wp,Ws,Rp,Rs,'s')

The cheby2ap (N,Rs) determines the pole zero and gain factor of transfer function with order N and stopband ripples Rs. Alternatively we can use butter (N,Wn,'s') to design an order N transfer function with prescribed cut-off frequency of 3-dB at Wn rad/sec, the outputs are numerator and denominator polynomials coefficient vectors in descending order or s. If we wish to design a filter other than lowpass we can use the function cheby2(N,.Wn,filter type,'s') .The function cheb2ord(Wp,Ws,Rp,Rs,'s') computes the lowest order of the filter under given specifications.

Elliptic (causer) filters:

The matlab functions for the implementation elliptic filters are:

[z,p,k]=ellipap(N,Rp,Rs)

[num,den]=ellip(n,Rp,Rs,Wn,'s')

[num,den]=ellip(n,Rp,Rs,Wn,filtertype,'s')

[N,Wn]=ellip(Wp,Ws,Rp,Rs,'s')

The ellipap (N) determines the pole zero and gain factor of transfer function with order N passband ripples Rp and stopband ripples Rs dB. Alternatively we can use ellip (N,Wn,'s') to design an order N transfer function with prescribed cut-off frequency of 3-dB at Wn rad/sec, the outputs are numerator and denominator polynomials coefficient vectors in descending order or s. If we wish to design a filter other than lowpass we can use the function ellip(N,.Wn,filter type,'s') .The function ellipord (Wp,Ws,Rp,Rs,'s') computes the lowest order of the filter under given specifications.

2.6.1 Matlab implementation:

Here is the design of analog lowpass filters (butterworth, chebyshev, elliptic) using **matlab** under the following specification.

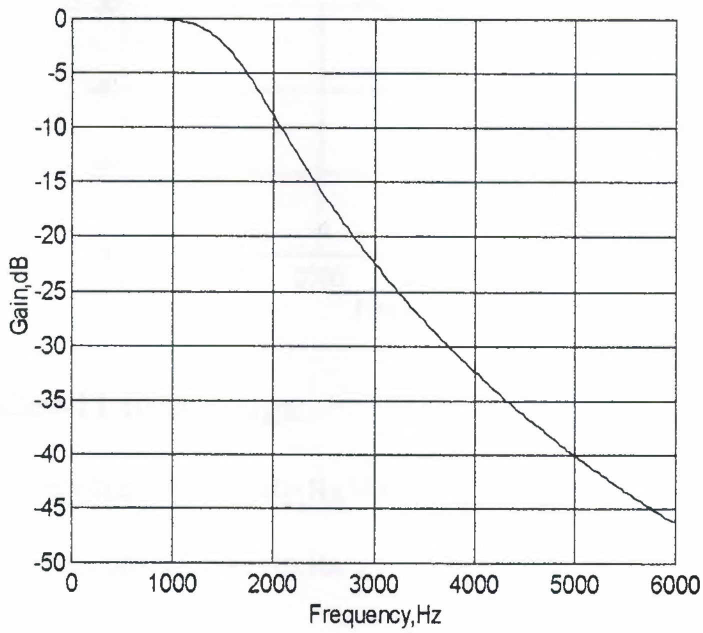
Passband edge frequency is 1kHz, stopband edge frequency is 5kHz, ripples in passband are 1dB and stopband attenuation is 40dB.

» %Designing specifications

» Wp=1000*2*pi;

```
» Ws=5000*2*pi;

» Rp=1;

» Rs=40;

» % A matlab program to design butterworth lowpass filter

» [N,Wn]=buttord(Wp,Ws,Rp,Rs,'s');

» disp('The order of butterworth filter');    disp(N);

The order of butterworth filter

    4

.» disp('Wn=');disp(Wn);

Wn=

  9.9347e+003

» % Determine transfer function

» [num,den]=butter(N,Wn,'s');

» disp('numerator polynomial is');   disp(num);

Numerator polynomial is

  1.0e+015 *

        0         0    0.0000    0.0000    9.7414

» disp('denominator polynomial is');  disp(den);

denominator polynomial is

  1.0e+015 *

    0.0000    0.0000    0.0000    0.0026    9.7414

» % Plot of frequency response

» omega=0:200:12000*pi;
```

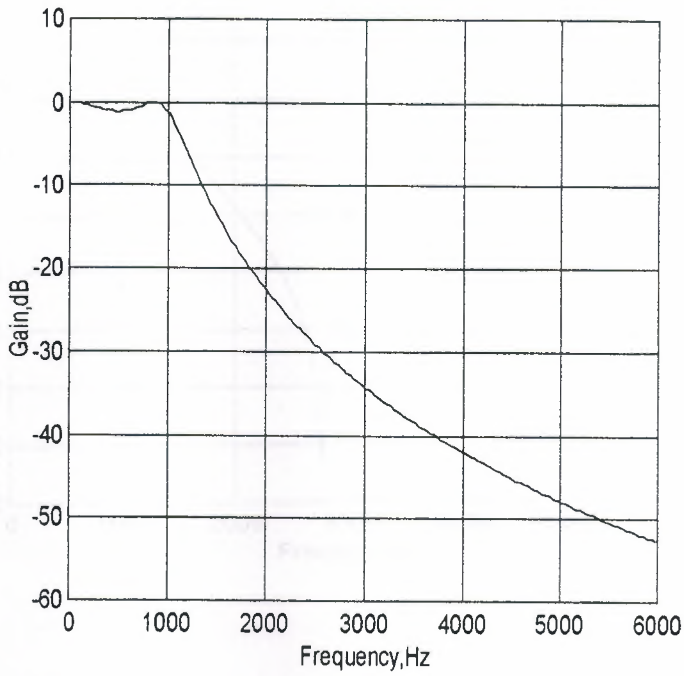
```
» h=freqs(num,den,omega);

» gain=20*log10(abs(h));

» plot(omega/(2*pi),gain,'r'); grid

»xlabel(' Frequency,Hz')

»ylabel(' Gain,dB')
```



Chebyshev type-1 filter design:

```
» [N,Wn]=cheb1ord(Wp,Ws,Rp,Rs,'s');

» [num,den]=cheby1(N,Rp,Wn,'s');

» omega=0:200:12000*pi;

» h=freqs(num,den,omega);

» gain=20*log10(abs(h));

» plot(omega/(2*pi),gain,'r'); grid

»xlabel(' Frequency,Hz')
```

»ylabel(' Gain,dB')



Chebyshev-11 filter design:

» [N,Wn]=cheb2ord(Wp,Ws,Rp,Rs,'s');

» [N,Wn]=cheb2ord(Wp,Ws,Rp,Rs,'s')

N =   3

Wn = 2.3441e+004

» [den,num]=cheby2(N,Rs,Wn,'s');

» omega = [0:200:12000*pi;];

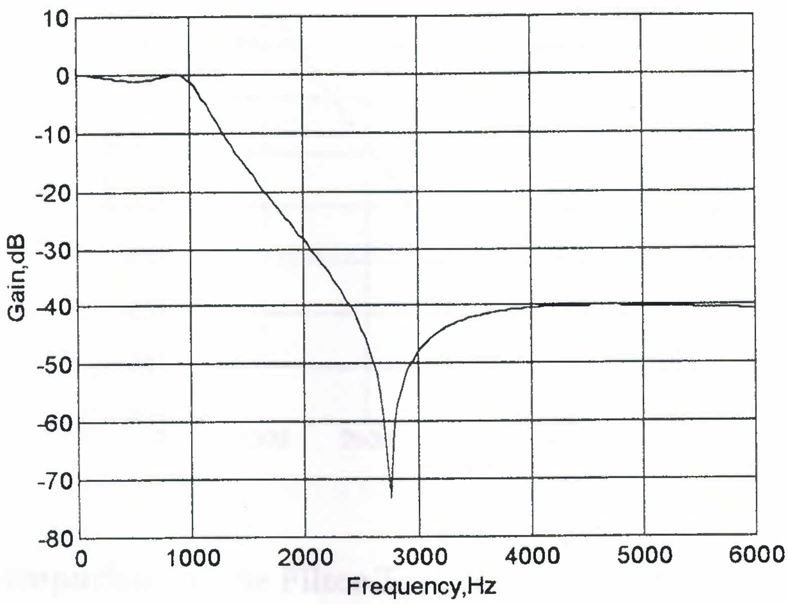
» h = freqs(num,den,omega);

» plot (omega/(2*pi),gain,'r')

» grid

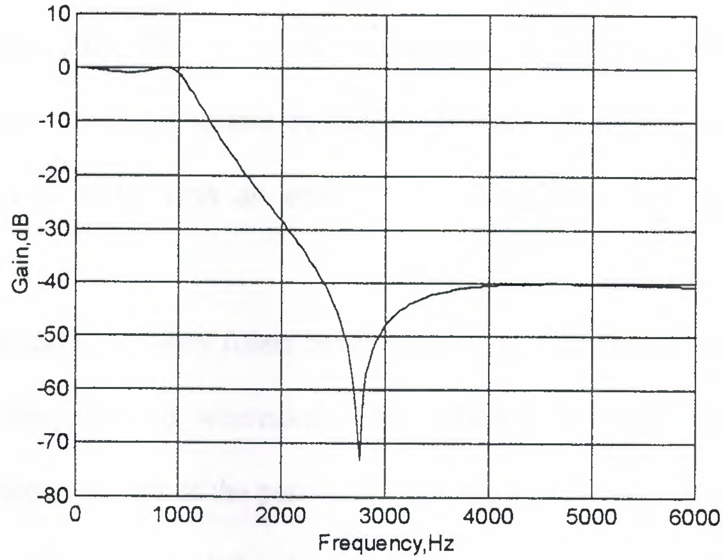
» xlabel('Frequency,Hz')

» ylabel('Gain,dB')



Elliptic filter design:

```
» [N,Wn]=ellipord(Wp,Ws,Rp,Rs,'s');

» [N,Wn]=ellipord(Wp,Ws,Rp,Rs,'s')

N =   3

Wn =    6.2832e+003

» [num,den]=ellip(N,Rp,Rs,Wn,'s')

» omega = [0:200:12000*pi;];

» h = freqs(num,den,omega);

» plot (omega/(2*pi),gain,'r')

» xlabel('Frequency,Hz')

» ylabel('Gain,dB')
```



2.7 A Comparison of the Filter Types:

We have discussed four types of analog low-pass fitter approximations, three of which have been developed primarily to meet the magnitude response specifications while the fourth has been developed primarily to provide a linear-phase approximation. In order to determine which filter type to choose to meet a given magnitude response specification, we need to compare the performances of the four types of approximations. To this end, we compare here the frequency responses of the normalized Butterworth, Chebyshev, and elliptic analog low-pass filters of same order. The pass-band ripple of the Type 1 Chebyshev and the equiripple filters are assumed to be the same, while the minimum stop-band attenuation of the Type 2 Chebyshev and the equiripple filters are assumed to be the same.

The Butterworth filter has the widest transition band, with a monotonically decreasing gain response. Both types of Chebyshev filters have a transition band of equal width that is smaller than that of the Butterworth filter but greater than that of the elliptic filter. The Type 1 Chebyshev filter provides a slightly faster roll-off in the transition band than the

Type 2 Chebyshev filter. The magnitude response of the Type 2 Chebyshev filter in the pass-band is nearly identical to that of the Butterworth filter. The elliptic filter has the narrowest transition band, with an equiripple pass-band and an equiripple stop band response.

The Butterworth and Chebyshev fillers have a nearly linear-phase response over about three-fourths of the pass-band, whereas the elliptic filter has a nearly linear-phase response over about one-half of the pass-band. One the other hand, the Bessel filter may be more attractive if the linearity of the phase response over a larger portion of the pass-band is desired at the expense of a poorer gain response. However, the Bessel filter provides a minimum attenuation at the largest transition band as compared other three types. [2]

3. ANALOG -TO-DIGITAL FILTER TRANSFORMATIONS

3.1 Overview:

After discussing different approaches to the design of analog filters, we are now ready to transform them into digital filters. These transsformations are complex-valued mappings that are extensively studied in the literature. These transformations are derived by preserving different aspects of anallog and digital filters. If we want to preserve the shape of the impulse response from analog to digital filter, then we obtain a techniqne called *impulse invariance trcansformation*. If we want to convert a differential eqnation representation into a corresponding difference equation representation, then we obtain a *finite differcnce* approximation technique. Numerous other techniques are also possible. One techniqne called *step invanance*, preserves the shape of the step response. The most poppular technique used in practice is called a *Bilinear transformation*, which preserves the system function representation from analog to digital domain.Here is a stuudy of impulse invariance and bilinear transformations, both of which can be easily implemented in MATLAB. [3,5,6,7]

3.2 Impulse Invariance Transform:

In this design method we want the digital filter impulse response to look "similar" to that of a frequency-selective analog filter. Hence we sample ha(t) at some sampling interval *T*

to obtain *h(n);* that is:

$$h(n) = ha(nT)$$

The parameter T is chosen so that the shape of $ha(t)$ is "captured" by the samples. Since this is a sampling operation, the analog and digital frequencies are related by

$$\omega = \Omega T \quad \text{or} \quad e = e^{j\omega T}$$

Since $z=e^\wedge(j\omega)$ on the unit circle and $s=j\Omega$ on the imaginary axis, we have the following transformation from the s-plane to the z-plane.

$$z = e^{sT} \qquad (3.1)$$

The system functions $H(z)$ and $Ha(s)$ are related through the frequency domain aliasing formula

$$H(z) = \frac{1}{T}\sum_{k=-\infty}^{\infty} H_a(s - j\frac{2\Pi}{T}k)$$

The complex plane transformation under the mapping under (3.1) is shown in Figure 3.1 from which we have the following observations

1.Using $\delta = \text{Re}(s)$, we note that

$\delta < 0$ maps into $|z| < 1$ (inside of the UC)

$\delta = 0$ niaps onto $|z| = 1$ (ott the UC)

$\delta > 0$ maps into $|z| > 1$ (outside of the UC)

2.All semi-infinite strips (shown above) of width $2\Pi/T$ map into $z < 1$. Thus this mapping is not unique but a *many-to-one* mapping.

3. Since the entire left half of the s-plane maps into the unit circle, a causal and stable analog fiter maps into a causal and stable digital filter.

4. If   Ha(jω)=Ha(jω/T)=0 for |Ω|≥λ/T, then

$$H(e^{j\omega}) = \frac{1}{T} Ha(\frac{j\omega}{T}) \ , \ |\omega| \le \lambda$$

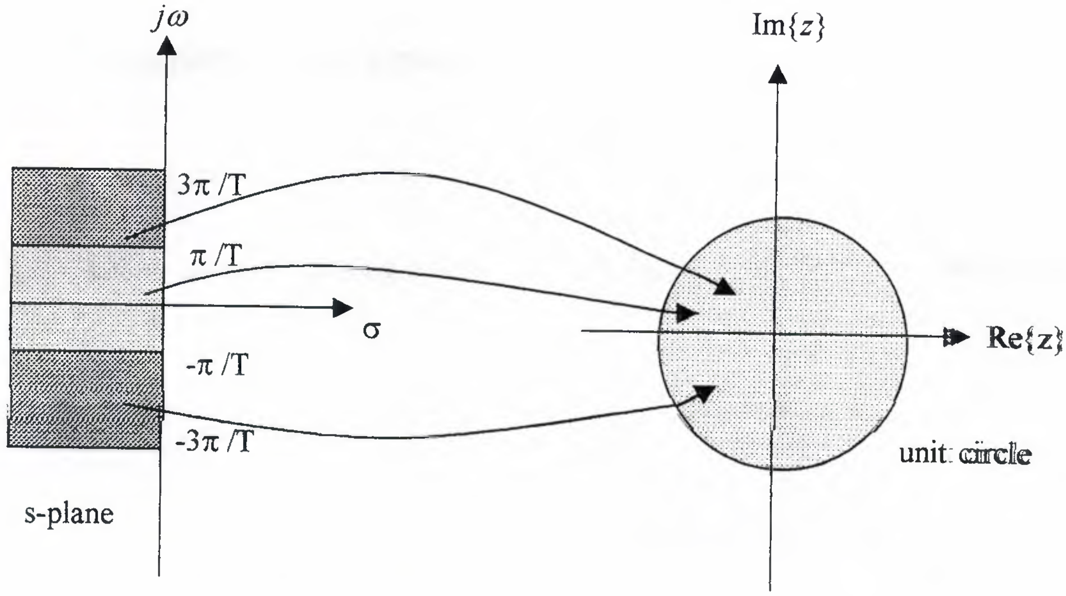


FIGURE 3.1 *CompLex-plane mapping in imulse invariance transfonmation*

and there will be no aliasing. However, no analog filter of finite order can be exactly band-limited. Therefore Some aliasing error will occur in this design procedure and hence the sampling interval *T* plays a minor role in this design method.

3.2.1 Design procedure:

Given the digital lowpass filter specifications ωp, ωs, Rp and As, and we want to determine H *(z)* by first designing an equivalent analog filter and then mapping it into the desired digital filter. The steps required for this procedure are

1. Choose *T* and determine the analog frequencies

Ωp=ωp/Tp and Ωs=ωs/T

2.Design an analog filter *Ha(s)* using the specifications ωp,ωs,Rpand As. This can be done using any one of the three (Butterworth, Chebyshev or elliptic) ¡prototypes .

3.Using partial fraction expansion, expand Ha (s) into

$$Ha(s) = \sum_{k=1}^{N} \frac{R_k}{s - p_k}$$

4.Now transform analog poles { p_k }into digital poles *{*$e^{p_k T}$ *}* to obtain the digital filter

$$H(z) = \sum_{k=1}^{N} \frac{R_k}{1 - e^{pkT} z^{-1}}$$

(3.2)

3.2.2 Matlab implemantation:

Matlab provide fuction "impinvar" to transform analog prototype transfer function to a digital filter.

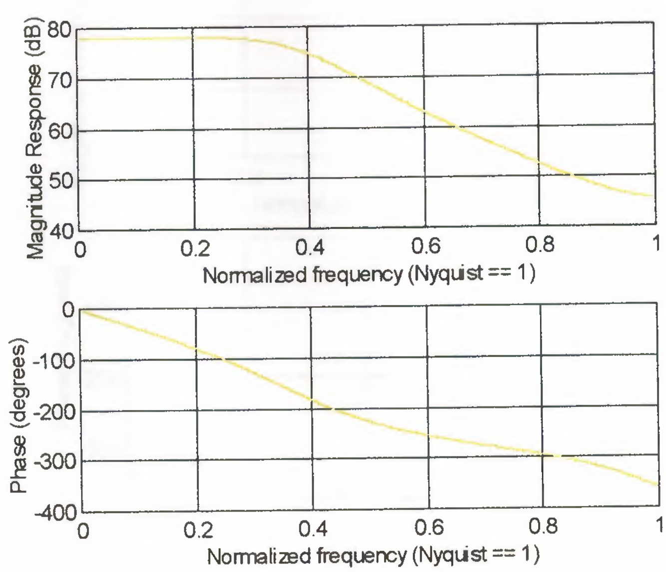
Here is implementation of above function in order to transform analog prototypes to a digital filter .The specification are:

Fp=1000KHz Fs=5000KHz Rp=1dB Rs=40dB

Butterworth filter:

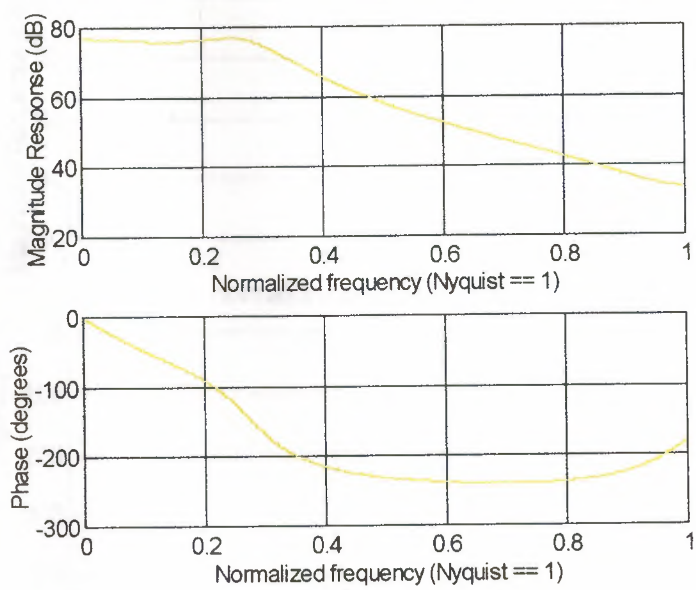
» Wp=1000*2*pi;

```
» Ws=5000*2*pi;        Rp=1;     Rs=40;

» [N,Wn]=buttord(Wp,Ws,Rp,Rs,'s');

» [N,Wn]=buttord(Wp,Ws,Rp,Rs,'s')

N =

     4

Wn =

   9.9347e+003

» [num,den]=butter(N,Wn,'s');

» %Digital transform using impulse invariance method

» Fp=8000;

» [b,a]=impinvar(num,den,Fp);

» freqz(b,a,512)
```



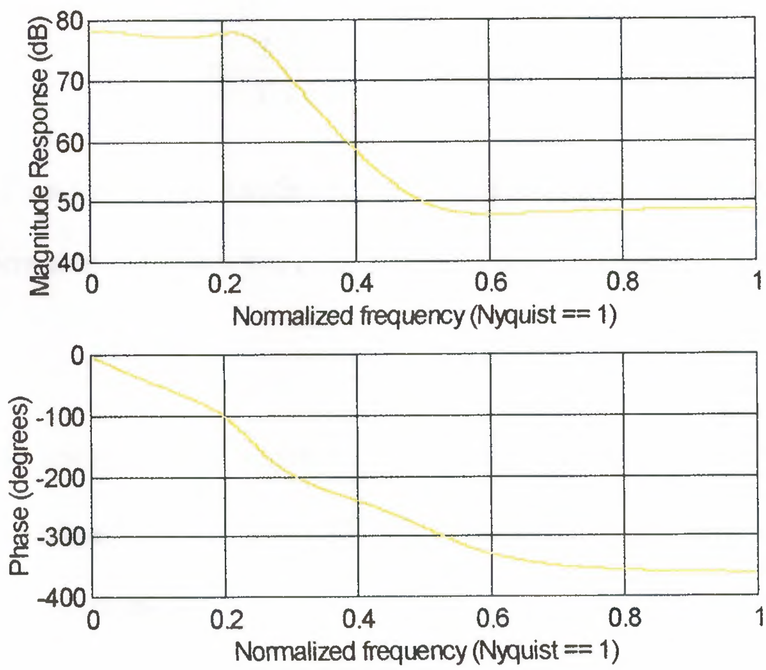
Chebyshev type 1 filter:

```
» [N,Wn]=cheb1ord(Wp,Ws,Rp,Rs,'s')

N =

     3

Wn =

  6.2832e+003

» [num,den]=cheby1(N,Rp,Wn,'s');

» %Digital transform using impulse invariance method

» Fp=8000;

» [b,a]=impinvar(num,den,Fp);

» freqz(b,a,512)

»
```



Elliptic filter:

```
» [N,Wn]=ellipord(Wp,Ws,Rp,Rs,'s')

N =

    3

Wn =

  6.2832e+003

» [num,den]=ellip(N,Rp,Rs,Wn,'s');

» %Digital transform using impulse invariance method

»  Fp=8000;

» [b,a]=impinvar(num,den,Fp);

» freqz(b,a,512)

»
```



The **advantages** of the impulse invariance mapping are that it is a stable design and that

frequencies ω and Ω are linearly related.And the **disadvantage** is that we should expect some aliasing of the analog frequency response, and in some cases this aliasing is intolerable. Consequently, this design method is useful only when the analog filter is essentially band-limited to a lowpass or bandpass filter in which there are no scillations in the stopband.

3.3 Bilinear transform:

This mapping is the best transformation method; it involves a well-known function

$$s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \Rightarrow z = \frac{1+sT/2}{1-sT/2} \qquad (3.3)$$

where T isa parameter. Another name for this transformation is the *ilinear frectional* transformation because when cleared of fractions, we obtain

$$\frac{T}{2}sz + \frac{T}{2}s - z - 1 = 0$$

which is linear in each variable if the other is fixed, or *bilinear* iin s and z. The complex plane mapping under (3.3) is shown in Figure, from which we have the following observations: [3]

I. Using s= δ +jΩ in (3.3), we obtain

$$z = \left(1 + \frac{\delta T}{2} + j\frac{\Omega T}{2}\right) \Big/ \left(1 - \frac{\delta T}{2} - j\frac{\Omega T}{2}\right) \quad (3.4)$$

Hence

$$\delta < 0 \Rightarrow |z| = \left| \frac{1 + \frac{\sigma T}{2} + j\frac{\Omega T}{2}}{1 - \frac{\sigma T}{2} - j\frac{\Omega T}{2}} \right| < 1$$

$$\sigma = 0 \Rightarrow |z| = \left| \frac{1 + j\frac{\Omega T}{2}}{1 - j\frac{\Omega T}{2}} \right| = 1$$

$$\sigma > 1 \Rightarrow |z| = \left| \frac{1 + \frac{\sigma T}{2} + j\frac{\Omega T}{2}}{1 - \frac{\sigma T}{2} - j\frac{\Omega T}{2}} \right| > 0$$

2. The entire left half-plane maps into the inside of the unit circle. *Hence* this is a stable transformation.

3. The imaginary axis maps onto the unit circle in a on-to-one fashion. Hence there is no aliasing in the frequency domain.

substituting $\sigma = 0$ in equation (3.4) we obtain

$$z = \frac{1 + j\frac{\Omega T}{2}}{1 - \frac{\Omega T}{2}} = e^{j\omega}$$

since the magnitude is 1. Solving for ω as a function of Ω, we obtain

$$\omega = 2\tan^{-1}\left(\frac{\Omega T}{2}\right), \quad \Omega_s = \frac{2}{T}\tan\left(\frac{\omega}{T}\right) \quad (3.5)$$

This shows that Ω is nonlinearly related to (or warped into) ω but that there is no aliasing. Hence in (3.5) we will say that ω is prewarped into Ω.

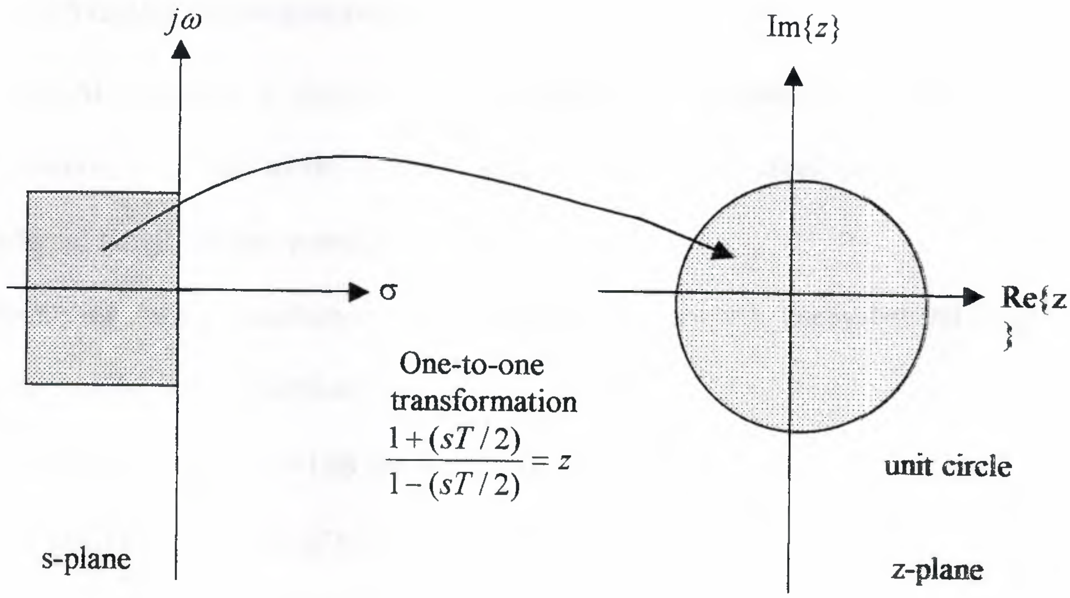


FIGURE 3.2 *Complex plane mapping using bilinear transform*

3.3.1 Design procedure:

Given the digital filter specification Wp, Ws, Rp and Rs we want to determine H(z). The design steps in the procedure are following.

1. Choose a value of T. This is arbitrary, and we may set T=1.
2. Prewarp the cutoff frequencies Wp and Ws; that is, calculatedl as Ωp and Ωs using (3.5):

$$\Omega_s = \frac{2}{T}\tan\left(\frac{\omega_s}{2}\right) \quad , \Omega_p = \frac{2}{T}\left(\frac{\omega_p}{2}\right) \qquad (3.6)$$

3. Design an analog filter Ha(s) to meet the specifications Ωp, Ωs, Rp and Rs.

4. Finally, set

$$H(z) = Ha\left(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right)$$

3.3.2 Matlab implementation:

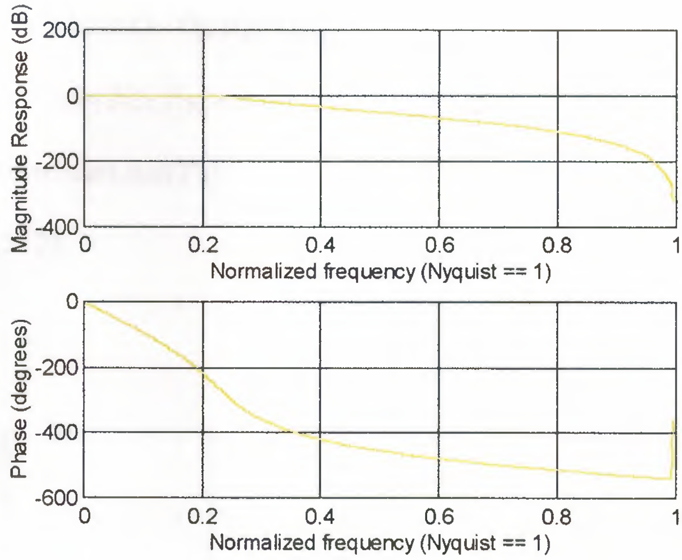
MATLAB provides a function called bilinear to implement this mapping. Its invocation is similar to the impinvr function, but it also takes several forms for different in-put out-put quantities.

Here is the design procedure of digital IIR filters(butterworth, chebyshevand elliptic) under the following specifications:

Wp=0.2π, Ws=0.3π, Rp=1dB and Rs=15dB.

```
» % DIGITAL FILTER SPECIFICATIONS
» Wp=0.2*pi;    Ws=0.3*pi;    Rp=1;    Rs=15;
» % inverse mapping for freq:
» T=1;    Fs=1/T;
» Op=(2/T)*tan(Wp/2);        % prewrwap prototype passband freq
» Os=(2/T)*tan(Ws/2);        % prewrwap prototype stopband freq
» %Butterworth filter order calculation:
» [N,Wn]=buttord(Op,Os,Rp,Rs,'s');
» [num,den]=butter(N,Wn,'s');
» %bilinear transformation
» [b,a]=bilinear(num,den,Fs);
» %magnitude and phase response
```

» freqz(b,a,512);



»% **Chebyshev1** filter order calculation:

» [N,Wn]=cheb1ord(Op,Os,Rp,Rs,'s');

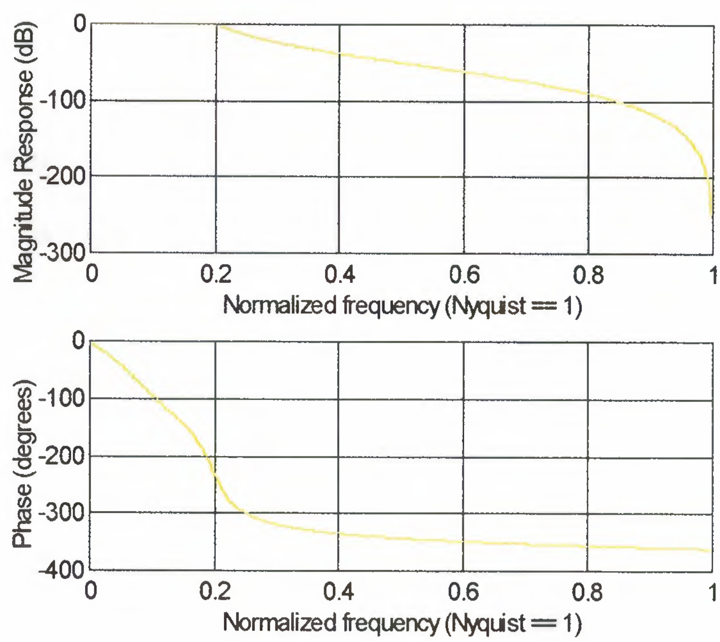
» [num,den]=cheby1(N,Rp,Wn,'s');

» %bilinear transformation

» [b,a]=bilinear(num,den,Fs);

» %magnitude and phase response

» freqz(b,a,512);



»%**Chebyshev type 2** order calculation:

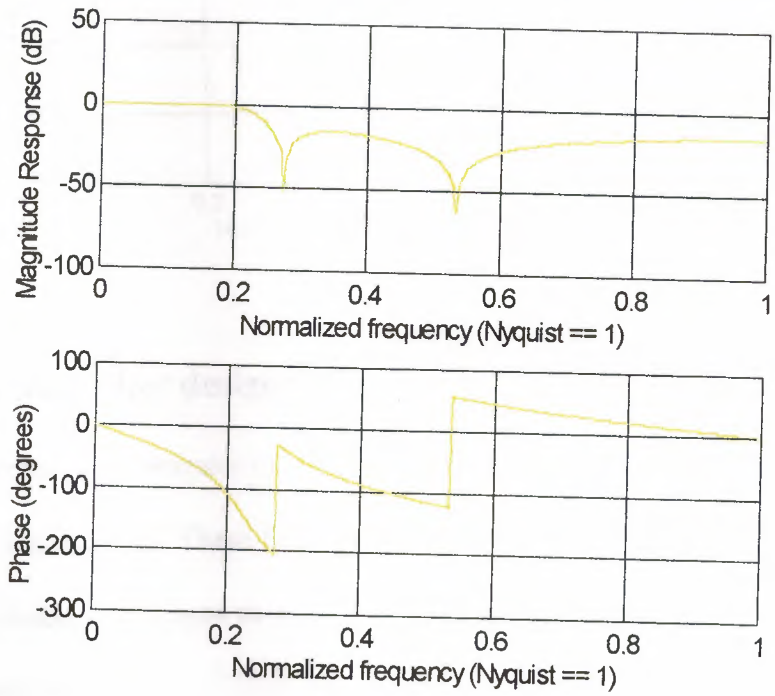
» [N,Wn]=cheb2ord(Op,Os,Rp,Rs,'s');

» [num,den]=cheby2(N,Rs,Wn,'s');

» [b,a]=bilinear(num,den,Fs);

» freqz(b,a,512);

»



»%**Elliptic** filter order calculation:

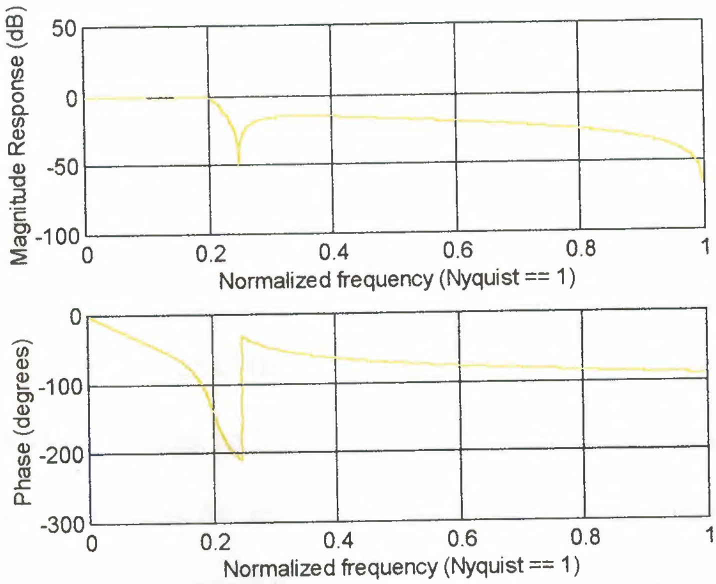
» [N,Wn]=ellipord(Op,Os,Rp,Rs,'s');

» [num,den]=ellip(N,Rp,Rs,Wn,'s');

» [b,a]=bilinear(num,den,Fs);

» freqz(b,a,512);

»



3.4 Lowpass filter design:

In this section demonstrates the use of matlab filter design routines to design digital lowpass filters. These functions use the bilinear transformation because of its desirable advantages as discussed in the previous section. These functions are as follows:

1. [b,a]=bntter(N,wn)

This function designs an Nth-order lowpass digital Butterworth filter and returns the filter coefficients in length N+1 vectors b and a. In MATLAB all digital frequencies are given in unit of λ . Hence wn is computed by using the following relation:

$$\omega_n = \frac{2}{\Pi} \tan^{-1}\left(\frac{\Omega_c T}{2}\right)$$

2. [b,a]=cheby1(N.Rp,wn)

This function designs an Nth-order lowpass digital Chebyshev-I filter with Rp decibels of ripple in the passband. It returns the filter coefficients in length $N + 1$ vectors b and a. The cutoff frequency wn is the digital pasaband frequency in units of λ; that is,

$$\omega_n = \omega_{p/\Pi}$$

3. [b,a]=cheby2[N,As,Wn)

This function designs an Nth-order lowpass digital Chebyshev-II filter with the stopband attenuation As decibels. It returns the filter coefficients in length $N + 1$ vectors b and a. The cutoff frequency wn is the digital stopband frequency in units of λ; that is,

$$\omega_n = \omega_{s/\Pi}$$

4. [b,al=ellip[N,Rp,As,Wn)

This function designs an Nth-order lawpass digital elliptic alter with the passband ripple of Ap decibels and a stopbsnd attenuation of As dccibels. It returns the filter coefficients in length $N + 1$ vectors b and a. The cutoff frequency *wn* is the digital pass band frequency in units of λ; that is,

$$\omega_n = \omega_{p/\Pi}$$

Here is matlab implementation of above functions for the designiing of lowpass Butterworth, Chebyshev and Elliptic filters under the following specifications.Passband normalized edde frequency Wp=0.2*pi ,stopband edge frequency Wp=0.2*pi ,stopband edge frequency Ws=0.3*pi ,passbancd ripples Rp=1dB and stopband attanuation Rs=15dB.

At the end a comperison is given to choose a best one, in terms of the order of the filter and stop band attanuation.

Butterworth lowpass filter:

```
» Wp=0.2*pi;        Ws=0.3*pi;        Rp=1;        Rs=15;

% analog prototype specification

» T=1;

» Op=(2/T)*tan(Wp/2);                    %prewrwap prototype passband
freq.

» Os=(2/T)*tan(Ws/2);                    %ptewrap prototype stopband
freq.

» % Analog butterworth prototype order calculation

» N=ceil((log10((10^(Rp/10)-1)/(10^(Rs/10)- 1)))/(2*log10(Op/Os)));;

» fprintf('**Butterworth filter order=%2.0f\n',N);

**Butterworth filter order= 6

» % analog Butterworth prototype cuttoff frequency

» Oc=Op/((10^(Rp/10)-1)^(1/(2*N))); %analog cuttoff freq.

» Wn=2*atan((Oc*T)/2);   %digital cuttoff freq.

» %Digital filter design

»Wn=Wn/pi;    %cuttoff freq. In pi unit

» [b,a]=butter(N,Wn);

» disp(a);
```

```
 1.0000   2.3505   2.8579   2.0069   0.8539   0.2034   0.0211

» disp(b);

  0.1452   0.8713   2.1782   2.9043   2.1782   0.8713   0.1452


»
```

Here is the design of chebyshev type 2 digital filter under the specifications mentioned above.

Chebyshev –1 lowpass filter :

```
» % Analog chebyshev-1  prototype order calculation

» ep=sqrt(10^(Rp/10)-1);           %passband ripple factor

» Oc=Op;                         %analog cutoff freq.

» Or=Os/Op;                      %Transition ratio

» A=10^(Rs/20);                   %stopbanb attenuation

» g=sqrt(A*A-1)/ep;           %Intermediate cal.

» N=ceil(log10(g+sqrt(g*g-1))/log10(Op+sqrt(Or*Or-1)));

» fprintf('chebyshev-1 filter order =%2.0f\n',N);

chebyshev-1 filter order = 4

» % digital chebyshev-1 filter design

» Wn=Wp/pi;            %passband freq. In unit of pi.

» [b,a]=cheby1(N,Rp,Wn);

» disp(a);

  1.0000  -3.9634   6.6990  -5.9815   2.8111  -0.5558
```

```
» disp(b);

   0.0003    0.0015    0.0029    0.0029    0.0015    0.0003

»
```

Chebyshev-2 lowpass filter design:

```
» % analog chebyshev-2 order calculation

» ep=sqrt(10^(Rp/10)-1);

» A=10^(Rs/20);

» g=sqrt(A*A-1)/ep;

» Oc=Op;

» Or=Os/Op;

» N=ceil(log10(g+sqrt(g*g-1))/log10(Or+sqrt(Or*Or-1)));

» fprintf('***chebyshev-2 filter order is=%2.0f',N);

***chebyshev-2 filter order is= 4

» % digital chebyshev-2 filter design

» Wn=Ws/pi;

» [b,a]=cheby2(N,Rs,Wn);

» disp(a);

   1.0000   -1.5508    1.3423   -0.4707    0.1079

» disp(b);

   0.1797   -0.0916    0.2525   -0.0916    0.1797
```

Elliptic lowpass filter:

```
» % analog ellip prototype order calculation
» ep=sqrt(10^(Rp/10)-1);
» A=10^(Rs/20);
» g=sqrt(A*A-1)/ep;
» k=Op/Os;
» k1=ep/sqrt(A*A-1);
» capk=ellipke([k.^2 1-k.^2]);
» capk1=ellipke([(k1 .^2) 1-(k1 .^2)]);
» N=ceil(capk(1)*capk1(2)/(capk(2)*capk1(1)));
» fprintf('***\n ellip filter order=%2.0f',N);
*** ellip filter order= 3
» % Digital Elliptic filter design
» Wn=Wp\pi;
» [b,a]=ellip(N,Rp,Rs,Wn);
» disp(a);
   1.0000    3.0000    3.0000    1.0000
» disp(b);
   1.0000    3.0000    3.0000    1.00
```

3.4 Comparison of three filters:

The filters compared in terms of order N and stopband attenuation Rs under the specification they were designed, the comparison is shown in table beloɔw

Prototype	Order N	Stoppband Attan.
Butterworth	6	15
Chebyshev-1	4	25
Elliptic	3	27

Clearly the Elliptic prototype gives the best design.However, if we compere their phase response ,elliptic design has the most non linear phase response in the passband.

4. FREQUENCY-BAND TRANSFORMATIONS

4.1 Overview:

So far we designed digital lowpass filters from their correspondhig analog filters. Here we are designing other types of freqnency-selective filters, such as higlipass, bandpass, and band-stop. This is accomplished by transforming the frequency axis (or band) of a lowpass filter so that it behaves as another frequency-selective filter. These transformations on the complex variable z are very similar to bilinear transformations, and the design equations are algebraic. [3]

4.2 Design procedure:

The procedure to design a general frequency-selective filter is to first design a *digital prototype* (of fixed bandwidth, say unit bandwidth) lowpass filter and then to apply algebraic transforniations. MATLAB provides fnctions that incorporate frequency-band transformation in the s-plane. We will first demonstrate the use of the z-plane mapping and then illustrate tlie use of MATLAB functions. Typical specifications for most commonly used types of frequency-selective digital filtters are shown in Figure 4.1.

Let *Hlp(Z)* be the given prototype lowpass digital filter, and let *H(z)* be the desired frequency-selective digital filter. We are using two different frequency variables, *Z* and z, with Hlp and *H*, respectively. Define a mapping of the form:

$$Z^{-1} = G(z^{-1})$$

Such that

$$H(z) = H_{LP}(z)\Big|_{z^{-1}=G(z^{-1})}$$

To do this, we simply replace Z^{-1} everywhere in *Hlp* by the function $G(z^{-1})$. Given that Hlp(Z*)* is a stable and causal filter, we also want*H(z)* to be stable and causal This imposes the following requirements:

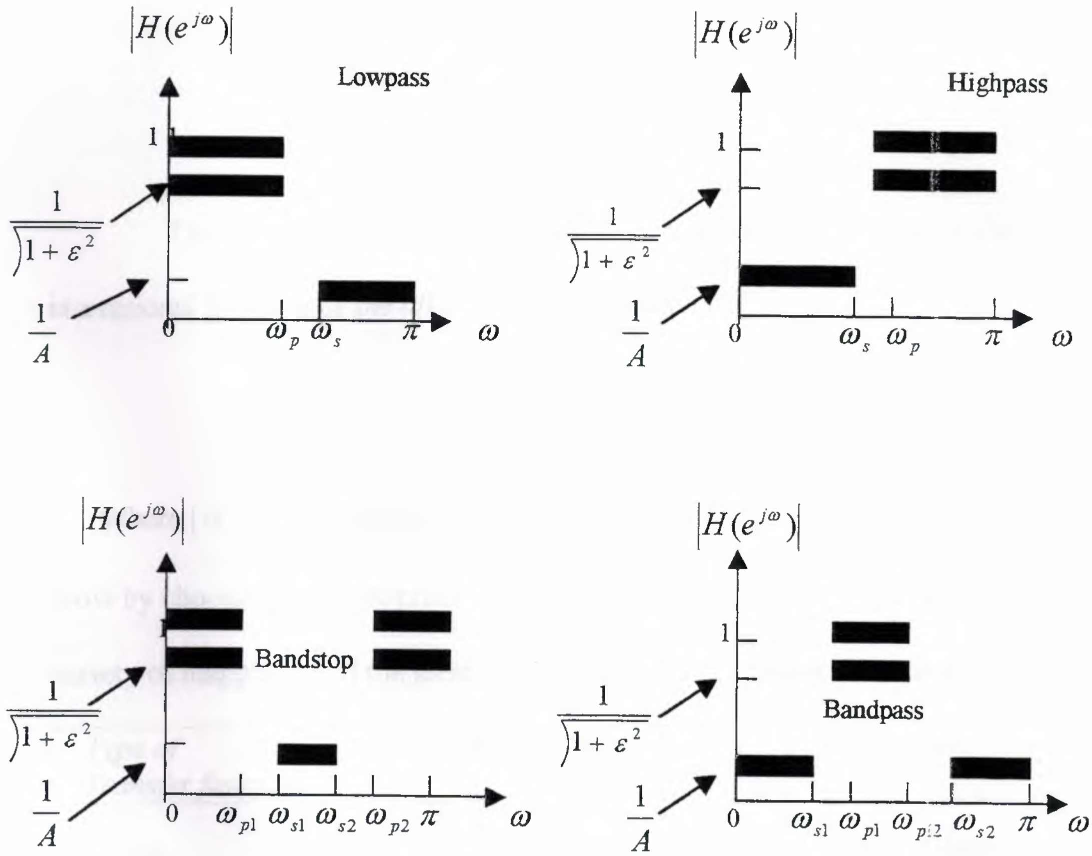


FIGURE 4.1 *Specifications of frequency-selective filters*

1. G (.) must be a rational function in z^{-1} so that *H (z)* is implementable.

2. The unit circle of the Z-plane must map onto the unit circle of the z-plane

3. For stable filters, the inside of the unit circle of the Z-plane musst also map onto the inside of the unit circle of the z-plane.

Let w' and w be the frequency variables of Z and z, respectively that is Z = $e^{j\omega'}$ and

z $= e^{j\omega}$ on their respective unit circles. Then requirement 2 above implies that

$$|Z^{-1}| = |G(e^{-j\omega})| = |G(z^{-1})| = 1$$

And

$$e^{j\omega'} = |G(e^{-j\omega})| e^{j\angle G(e^{-j\omega})}$$

The general form of the function *C (.)* that satisfies the above requirements is a rational function of the *all pass* type given by

$$Z^{-1} = G(z^{-1}) = \pm \prod \frac{z^{-1} - \alpha_k}{1 - \alpha_k z^{-1}}$$

Where | α |< 1 for stability and to satisfy requirement 3.

Now by choosing an appropriate order *n* and the coefficients {αk}, We can obtain a variety of mappings, [3] the most widely used transformations are given in Table 4.2

Type of Transfer function	*Transformation*	*Parameters*
Lowpass filter	$z^{-1} \rightarrow \dfrac{z^{-1} - \alpha}{1 - \alpha z^{-1}}$	ω_c = cutoff frequency of new $\alpha = \dfrac{\sin\left[(\omega_c' - \omega_c)/2\right]}{\sin\left[(\omega_c' + \omega_c)/2\right]}$
Highpass	$z^{-1} \rightarrow \dfrac{z^{-1} + \alpha}{1 + \alpha z^{-1}}$	ω_c = cutoff frequency of new filter

		$\alpha = -\dfrac{\cos\left[(\omega'_c + \omega_c)/2\right]}{\cos\left[(\omega'_c - \omega_c)/2\right]}$
Bandpass	$z^{-1} \rightarrow -\dfrac{z^{-2} - \alpha_1 z^{-1} + \alpha_2}{\alpha_2 z^{-2} - \alpha_1 z^{-1} + 1}$	ω_l = Lower cutoff frequency ω_u = upper cutoff frequency $\alpha_1 = -2k\beta / (k+1)$ $\alpha_2 = (k+1)/(k-1)$ $\beta = \dfrac{\cos\left[(\omega_u + \omega_l)/2\right]}{\cos\left[(\omega_u - \omega_l)/2\right]}$ $k = \cot\dfrac{\omega_u - \omega_l}{2} \tan\dfrac{\omega'_c}{2}$
Bandstop	$z^{-1} \rightarrow \dfrac{z^{-2} - \alpha_1 z^{-1} + \alpha_2}{\alpha_2 z^{-2} - \alpha_1 z^{-1} + 1}$	ω_l = Lower cutoff frequency ω_u = upper cutoff frequency $\alpha_1 = -2k\beta / (k+1)$ $\alpha_2 = (k+1)/(k-1)$ $\beta = \dfrac{\cos\left[(\omega_u + \omega_l)/2\right]}{\cos\left[(\omega_u - \omega_l)/2\right]}$ $k = \tan\dfrac{\omega_u - \omega_l}{2} \tan\dfrac{\omega'_c}{2}$

TABLE 4.2-*frequency transformation for digital filters*

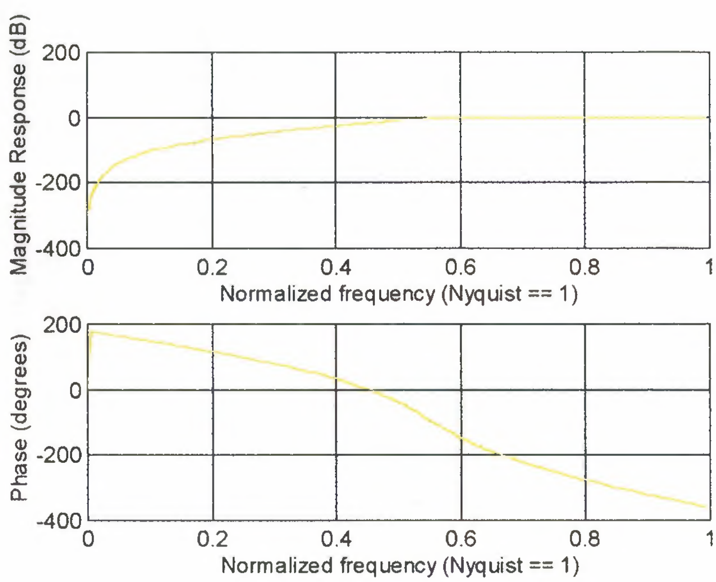
4.3 Matlab Implementation:

Here is the design of **Chebyshev type 1 high-pass** filter under the following specifications.

Wp=0.6pi, Ws=0.4586, Rp=1dB and Rs=15dB. The plots of phase response and magnitude response (in dB) are also sketched.

Matlab script:

```
» [N,Wn]=buttord(0.6,0.4586,1,15)

N =

   6

Wn =

  0.5491

» [b,a]=butter(N,Wn,'high')

b =

  0.0182  -0.1091   0.2728  -0.3637   0.2728  -0.1091   0.0182

a =

  1.0000   0.5822   0.9043   0.2882   0.1477   0.0200   0.0024

» freqz(b,a,128)

»
```

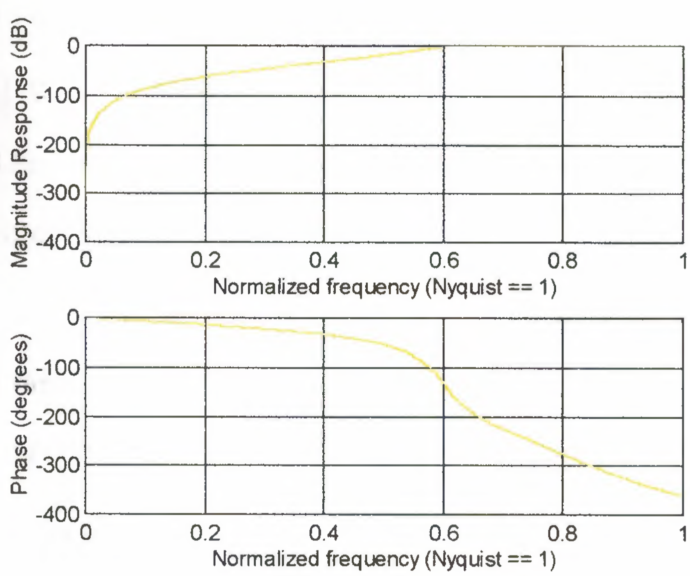


Design of **Chebyshev type 1 high pass** filter under following specification:

Wp=0.6pi, Ws=0.4586, Rp=1dB and Rs=15dB. The plots of phase response and magnitude response (in dB) are also sketched.

Matlab script

```
» [N,Wn]=cheb1ord(0.6,0.4586,1,15)

N = 4

Wn =0.6000

» [b,a]=cheby1(N,1,Wn,'high')

b =   0.0243   -0.0970   0.1456   -0.0970   0.0243

a =   1.0000   1.5977   1.7459   1.0200   0.3074

» freqz(b,a,128)

» freqz(b,a,512)
```



Here is the design of **elliptic high pass** filter under the following specifications:

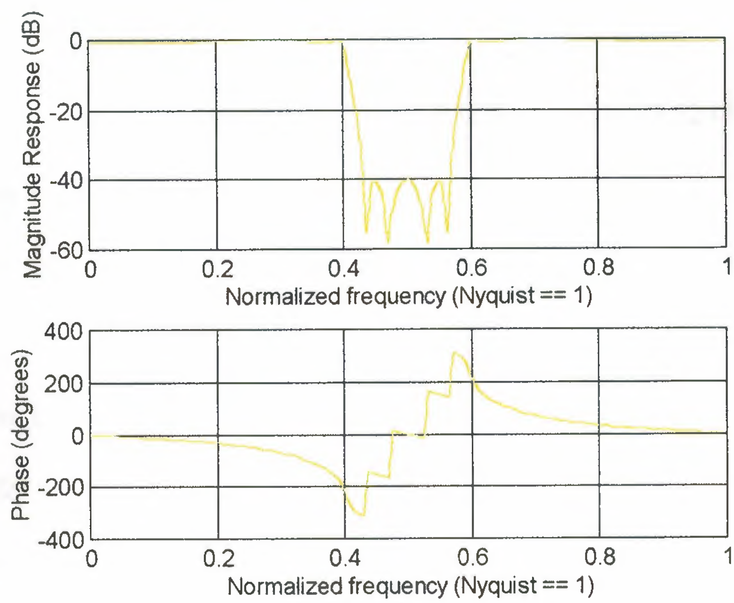
» Ws=[0.3pi 0.75pi] Wp=[0.4pi 0.6pi] Rp=1 Rs=40

The phase response and the magnitude response are also sketched.

Matlab script:

```
» Ws=[0.3*pi    0.75*pi];        Wp=[0.4*pi    0.6*pi];

» Rp=1;     Rs=40;

» [N,Wn]=ellipord(Wp/pi,Ws/pi,Rp,Rs)

N =

     4

Wn =

    0.4000    0.6000

» [b,a]=ellip(N,1,40,Wn,'pass')

b =

  Columns 1 through 7

    0.4093    0.0000    1.5594    0.0000    2.3024    0.0000    1.5594

  Columns 8 through 9

    0.0000    0.4093

a =

  Columns 1 through 7

    1.0000    0.0000    2.2509    0.0000    2.3194    0.0000    1.1417

  Columns 8 through 9

    0.0000    0.2893

» freqz(b,a,128)
```

»



Design of Chebyshev type-2 band **stops** filter under following specifications:

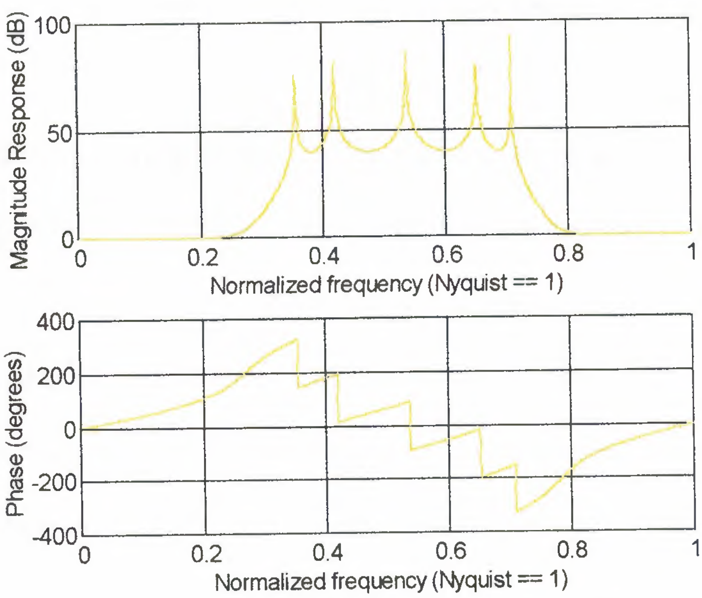
Ws=[0.4pi 0.7pi] Wp=[0.25pi 0.8pi] Rp=1 Rs=40

The phase and magnitude responses are also sketched.

Matlab script :

```
» Ws=[0.4*pi   0.7*pi];      Wp=[0.25*pi   0.8*pi];

» Rp=1;    Rs=40;

» [N,Wn]=cheb2ord(Wp/pi,Ws/pi,Rp,Rs)

N =

     5

Wn =

    0.3490    0.7157

» [b,a]=cheby2(N,40,Wn,'stop')
```

```
b =

 Columns 1 through 7

  0.1068   0.1082   0.4074   0.3325   0.7243   0.4611   0.7243

 Columns 8 through 11

  0.3325   0.4074   0.1082   0.1068

a =

 Columns 1 through 7

  1.0000   0.6233   0.3254   0.2841   0.8480   0.3345   0.1859

 Columns 8 through 11

  0.0832   0.1087   0.0175   0.0090

» freqz(a,b,1278)

»
```



CONCLUSION

The digital filter design problem is concerned with the development of suitable transfer function meeting the freqquency response specifications. The specifications are usually given in the term of pass band and stopband edge frequencies and allowable deviations from passband and stopband magnitude level. In this project the main objective was to obtain linear magnitude response. The IIR filters are best for this purpose.

IIR filter design is usually carried out by transforming a prototype analog filter function by means of a suitable mapping of complex frequency variable s to complex variable z. The impulse invariance method and bilinear transform are discussed. Of these two the later is less restricted and is more widely used than the former.

Here is a **comparison** of the IIR filters, designed in the project using matlab techniques.

Elliptic filters provide optimal performance in the magnitude-squared respouse but have highly nonline& phase response in the psssband (which is undesirable in many applications). Even though we are not concerened about phase response in our designs, phase is still an important issue in the overall system. At the other end of the performance scale are the Butter-worth filters, which have maximally flat magnitude response and reqnire a higher-order N (more poles) to achieve the same stophand specification. However, they exhibit a fairly linear phase response in their passband. The Chebyshev filters have phase characteristics that lie somewhere in between. Therefore in practical applications we do consider Butterworth as well as

Cbebyshev filters, in addition to elliptic filters. The choice depends on both the filter order (which influences processing speed and implementation complexity) and the phase characteristics (which control the distortion).

The advantage of IIR filters over the FIR filters is their order is less, so they require least implementation size. But their phase response is not linear, where as FIR filters shows linear phase response, but they require high order, and as a result their implementation require big size.

References

[1]. Andereas Anthoniou (1998), Digital Filters Analysis and Design, New York, McGraw-Hill Company.

[2]. Kishan Shenoi. (1997) Digital Signal processing in Telecommunication, San Jose, California. Telecom Solutions, Inc.,

[3]. L.B.Jackson, J.F.Kaiser, and H.S.McDonald, (1988) An Approach to the Implementation of Digital Filters, New Jersey, IEEE Trans., Audio Electroacoust,

[4]. S.K.Tewksbury, (1973) Special Purpose Hardware Implementation of Digital Filter, NJ USA proc. IEEE int. Symp. Circuits syst.,

[5]. James McClellan, Roland Schafer and Mark Yoder, (1998) DSP First: A Multimedia Approach,

Georgia, the Georgia Institute of Technology.

[6]. James.McClellan, C.Sidney Burrus, Thomas W. Park and Alan V.Oppenheim, (1998) Computer Based Exercise for Signal Processing Using Matlab, New York. Schafer/Schuessler.

[7]. Ashok Ambardar and Craig Borghesani, (1998) Mastering DSP Concepts Using Matlab, Michigan.Terasoft, Inc

[8]. E. Kamen and Bonnie Heck, (1997) Fundamentals of Signals and Systems: with Matlab, Georgia, the Georgia Institute of Technology.

[9]. Matlab Signal Processing tool box (User Guide). (1999), New York, The Mathworks.Inc.

[10]. Web site: http://www.mathworks.com

MATLAB COMMANDS FOR IIR FILTER DESIGN:

Filter type	Design Function
Butterworth	[b,a] = butter(n,Wn,option)
	[z,p,k] = butter(n,Wn,option)
	[A,B,C,D] = butter(n,Wn,option)
Chebyshev Type 1	[b,a] = cheby1(n,Rp,Wn,option)
	[z,p,k] = cheby1(n,Rp,Wn,option)
	[A,B,C,D] = cheby1(n,Rp,Wn,option)
Chebyshev Type 2	[b,a] = cheby2(n,Rs,Wn,option)
	[z,p,k] = cheby2(n,Rs,Wn,option)
	[A,B,C,D] = cheby2(n,Rs,Wn,option)
Elliptic	[b,a] = ellip(n,Rp,Rs,Wn,option)
	[z,p,k] = ellip(n,Rp,Rs,Wn,option)
	[A,B,C,D] =ellip(n,Rp,Rs,Wn,option)

By default each of these function return lowpass filters, we only need to specify the value of Wn in normalized frequency. For highpass, bandpass and bandstop filters we have to add high, pass and stop at the place of option respectively, and also to specify the value of Wn accordingly.

IIR Filter Order Calculation:

Filter Type	Order Estimation Function
Butterworth	[n,Wn] = butter(Wp,Ws,Rp,Rs)
Chebyshev type 1	[n,Wn] = cheb1ord(Wp,Ws,Rp,Rs)
Chebyshev type 2	[n,Wn] = cheb2ord(Wp,Ws,Rp,Rs)
Elliptic	[n,Wn] = ellipord(Wp,Ws,Rp,Rs)

Analog to Digital Transformation:

Bilinear Transformation: [b,a] = bilinear(num,den,Fs)

Impulse Invariance Transformation: [b,a] = impinvar(num,den,Fs)

Where num and den are the polynomials of transfer function of analog filter, Fs is the sampling frequency. b And a are the polynomials of digital filter.

INDEX