DESIGN OF DIGITAL IIR FILTER USING MATRIX OPERATION

A dissertation submitted
in partial fulfilment of the requirements for the award of the degree of

Master of Engineering
In
Electronics and Communication Engineering

By

Arvind Kumar

Roll No. 2853

Supervised By
O.P.Verma

Department of Electronics and Communication Engineering,
Delhi College of Engineering, University of Delhi
Session 2004-2006

ACKNOWLEDGEMENT

I wish to acknowledge our sincere thanks to my guide **Assistant Professor O.P.Verma**,(Department of Electronics and Communication Engineering) for his suggestions, excellent guidance and timely advice which have made my thesis work success. I am indebted to the entire faculty and non teaching staff of Electronics and communication department, who had very helpful and cooperative to me at all times.

Arvind Kumar (2853)

CERTIFICATE

Certified that the thesis work entitled

DESIGN OF DIGITAL IIR FILTER USING MATRIX OPERATION

is bonafied work carried by

Arvind Kumar (2853)

In partial fulfilment for the award of degree of Master of Engineering in Electronics and Communication Engineering of the University of Delhi during the year 2004-2006.It is certified that all corrections/suggestions indicated for internal assessment have been in corporate in the report deposited in the Departmental library. The project report has been approved as it satisfied the academic requirements in respect of thesis work prescribed for the Master of Engineering Degree.

Signature of Guide

Assistant Professor O.P.Verma

SYNOPSIS

DESIGN OF DIGITAL IIR FILTER USING MATRIX OPERATION

The transfer function H(z) of the desired Digital Infinite Impulse Response Filter can be obtained from the normalized transfer function $H_a(s)$ of the analog low pass filter .Recently it has been shown that Pascal Matrix allows the appropriate transformation for design of the low pass, high pass , and band pass digital filters . Unfortunately, this method is difficult to use in case of transforming $H_a(s)$ to the High order bandpass filter. This project discusses the design of digital IIR filter using frequency transformation by matrix operation . This method uses two matrices one for frequency transformation and another for bilinear transformation, obviously second matrix is the Pascal matrix. Combination of the matrices described here with the Pascal matrix allows the design of the digital IIR filters from the continuous time prototypes. After designing IIR Digital filter using above method we will design same filter by using other well known frequency transformation methods, after that we will try to apply some implementations , and at the last, we will compare these methods with the method given at beginning to see which one is better. All approaches described above will use matlab commands, a mathematical software package, to design, manipulate, and analyze digital filters.

Contents

INTRODUCTION

The Digital Filter Design problem involves the determination of a set of filter coefficients to meet a set of design specifications. These specifications typically consist of the width of the passband and the corresponding gain, the width of the stopband(s) and the attenuation therein; the band edge frequencies (which give an indication of the transition band) and the peak ripple tolerable in the passband and stopband(s).

Two types of digital filters exist – the IIR (Infinite Impulse Response) and the FIR (Finite Impulse Response). FIR filters possess certain properties ,which make them the preferred design choices in numerous situations over IIR filters. Most notably, FIR filters (all zero system function) are always stable, with a realization existing for each FIR filter. Another feature exclusive to FIR filters is that of a linear phase response.

The design of IIR filters is closely related to the design of analog filters, which is a widely studied topic. An analog filter is usually designed and a transformation is carried out into the digital domain. Two transformations – the **impulse invariant** transformation and the **bilinear transformation** are widely used till now. **This project 'Design of Digital IIR Filter using Matrix operation'** discusses the design of digital IIR filter using frequency transformation by matrix operation . This method uses two matrices one for frequency transformation and another for bilinear transformation, obviously second matrix is the Pascal matrix. Combination of the matrices described here with the Pascal matrix allows the design of the digital IIR filters from the continuous time prototypes. After designing IIR Digital filter using above method we will design same filter by using other well known frequency transformation methods, after that we will try to apply some implementations , and at the last, we will compare these methods with the method given at beginning to see which one is better. All approaches described above will use Matlab commands, a mathematical software package, to design, manipulate, and analyze digital filters .

CHAPTER 1

Introduction to Basic Digital Filter

1.1. IIR filters

The output from an IIR digital filter is made up from previous inputs and previous outputs, as described by the difference equation

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k]$$

1.11

In order to compute the impulse response two convolutions are involved: one with the previous inputs, and one with the previous outputs. In each case the convolving function is called the filter coefficients.

If such a filter is subjected to an impulse then its output need not necessarily become zero after the impulse has run through the summation. So the impulse response of such a filter can be infinite in duration. Such a filter is called an **Infinite Impulse Response** filter or **IIR** filter.

Note that the impulse response needs not necessarily be infinite: if it were, the filter would be unstable. In fact for most practical filters, the impulse response will converge to zero. One might argue that mathematically the response can go on for ever, getting smaller and smaller: but in a digital world once a level gets below one bit it might as well be zero. The **Infinite Impulse Response** refers to the ability of the filter to have an infinite impulse response and does not imply that it necessarily will have one: it serves as a warning that this type of filter is prone to feedback and instability.

Note that equation (1.11) describes a linear time-invariant system, which is causal and physically realizable. It is easy to demonstrate that its frequency response is

$$H(w) = \frac{\sum_{k=0}^{M} b_k e^{-jwk}}{1+\sum_{k=1}^{N} a_k e^{-jwk}}$$

1.12

While it is nice to be able to calculate the frequency response given the filter coefficients, in practice the inverse operation is done more frequently: that is, to calculate the filter coefficients (a_k and b_k) having first defined the desired frequency response, but there is one problem, so far, it has not been discovered a general inverse solution to the frequency response equation.
Usually, engineers design digital filters with the idea that they will be implemented on some piece of hardware. This means that the filter must meet some requirements by using the least possible amount of computation, in other words, using the smallest number of coefficients. So one is facing with an insoluble inverse problem, efficiency in term of hardware and the need of additional constraints.

Matlab has several design algorithms that can be used to create and analyze directly both IIR. The IIR filters that can be created in Matlab are Butterworth, Chebyshev type 1 and 2, and Elliptic.

1.2. FIR Filters

It is much easier to approach the problem of calculating the filter coefficients if one simplify the filter equation so that one only has to deal with previous inputs. The filter equation is then simplified:

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] \quad (1.21)$$

If such a filter is subjected to an impulse then its output must necessarily become zero after the impulse has run through the summation. So the impulse response of such a filter must necessarily be finite in duration. Such a filter is called a **Finite Impulse Response** filter or **FIR** filter.
The filter's frequency response is also simplified, because all the bottom half goes away:

$$H(w) = \sum_{k=0}^{M} b_k e^{-jwk} \quad (1.22)$$

Equation (1.22) is just the Fourier transform of the filter coefficients. So the coefficients for an FIR filter can be calculated simply by taking the inverse Fourier Transform of the desired frequency response.
There are several ways to design an FIR filter as described by the next two sections

Selection of Using FIR and IIR Filter

- If linear phase is required →FIR
- If stability is required → FIR (since it can be non-recursive technique)
- Finite word-length effect to FIR less than to IIR.
- If sharp cut-off frequencies are required → FIR requires more coefficients, processing time, and memory size than IIR. (However, FFT algorithm or multirate technique may be used for FIR to compensate these disadvantages.)
- IIR can be used analog filter as prototype, but FIR can be synthesized more easily for any required frequency response. (However, in general, to synthesize FIR is required CAD because its algebraic design technique is very difficult.)

CHAPTER 2

Analog to Digital Domain Mapping Techniques

Digital Filters are designed by using the values of both the past outputs and the present input, an operation brought about by convolution. If such a filter is subjected to an impulse then its output need not necessarily become zero. The impulse response of such a filter can be infinite in duration. Such a filter is called an *Infinite Impulse Response* filter or **IIR** filter. The infinite impulse response of such a filter implies the ability of the filter to have an infinite impulse response. This indicates that the system is prone to feedback and instability. The report studies several different types of IIR filters including the Butterworth Filter, Chebyshev I & II Filters and Elliptic Low, High and Bandpass filters. Here IIR filters are designed essentially by the Frequency transformation by using Matrix operations , Impulse Invariant and Bilinear Z-Transformation.

Reasons of Design of Discrete-Time IIR Filters from Continuous-Time Filters

- The art of continuous-time IIR filter design is highly advanced and, since useful results can be achieved, it is advantageous to use the design procedures already developed for continuous-time filters.
- Many useful continuous-time IIR design methods have relatively simple closed-form design formulas. Therefore, discrete-time IIR filter design methods based on such standard continuous-time design formulas are rather simple to carry out.
- The standard approximation methods that work well for continuous-time IIR filters do not lead to simple closed-form design formulas when these methods are applied directly to the discrete-time IIR case.

2.1 Frequency Transformation by Matrix operations.

It is known that the transfer function H (z)of the desired digital infinite-impulse response filter can be obtained from the normalized transfer function H_a(s) of the analog low-pass filter. It has been shown that Pascal matrix allows the appropriate transformation for design of the low-pass, high-pass, and bandpass digital filters. Unfortunately, this method is difficult to use in case of transforming H_a (s) to the high-order bandpass filter. This project is based on a similar method, especially suitable for design of the bandpass and bandstop filters without order limitation .We will also try to design lowpass and highpass digital IIR filters by this method.

Introduction:

One of the popular digital infinite impulse response (IIR) filters design method bases on the procedures for digitizing a continuous-time filter into discrete-time filter. Usually, two steps are performed. First, the frequency transformation is applied to the normalized analog low-pass transfer function . Second, the bilinear transformation is used to obtain the transfer function of the desired digital filter. Recently, it has been shown that Pascal matrix allows executing these two steps at the same time for design the low-pass, high-pass , and bandpass digital filters.

This project aims to obtain the same transformation from H_a (s) to H(z) by mean of two matrices; one for frequency transformation and second for bilinear transformation. The second matrix remains the Pascal matrix.

Let us assume the normalized transfer function of an analog low-pass filter is in the form

$$H_a(s) = \frac{B_0 + B_1 s + \ldots + B_n s^n}{A_0 + A_1 s + \ldots + A_n s^n}.$$

………………(1)

During the first step, the coefficients B_i and A_i are transformed to the coefficients $\hat{A}_i$ and $\hat{B}_i$ Which form the transfer function of the desired filter type

$$H_D(s) = \frac{\hat{B}_0 + \hat{B}_1 s + \ldots + \hat{B}_m s^m}{\hat{A}_0 + \hat{A}_1 s + \ldots + \hat{A}_m s^m}.$$

…………..(2)

If $H_D(s)$ describes the band-pass or band- stop filter then m = 2n, where for low-pass and high-pass filters m = n .

Finally , from $H_D(s)$, the transfer function H(z) of the digital filter must be obtained

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_m z^{-m}}{a_0 + a_1 z^{-1} + \ldots + a_m z^{-m}}$$

(3)

By using Pascal Matrix P, as follows:

$$\mathbf{a} = \mathbf{P}\widetilde{\mathbf{A}}$$
$$\mathbf{b} = \mathbf{P}\widetilde{\mathbf{B}}$$

(4)

where $\mathbf{a}$, $\mathbf{b}$, $\widetilde{\mathbf{A}}$, and $\widetilde{\mathbf{B}}$ are the vectors

$$\mathbf{a} = [a_0 \quad a_1 \quad a_2 \quad \cdots \quad a_m]^T$$
$$\mathbf{b} = [b_0 \quad b_1 \quad b_2 \quad \cdots \quad b_m]^T$$
$$\widetilde{\mathbf{A}} = [\hat{A}_0 \quad \hat{A}_1 c \quad \hat{A}_2 c^2 \quad \cdots \quad \hat{A}_m c^m]^T$$
$$\widetilde{\mathbf{B}} = [\hat{B}_0 \quad \hat{B}_1 c \quad \hat{B}_2 c^2 \quad \cdots \quad \hat{B}_m c^m]^T$$

(5)

The coefficient **c** depends on the sampling period T_s i.e. $c = 2/T_s$ and p is the $m+1 \times m+1$ matrix formed by the following algorithm.

- First row of the matrix contains only ones.
- The elements of the last column can be calculated by

$$P_{i,n} = (-1)^i m!/(m-i)!\, i! \qquad (6)$$

Where $i = 0,1\ldots\ldots\ldots\ldots,m$.

- All the others elements can be computed on the basis of previous one , as follows:

$$P_{i,j} = P_{i-1,j} + P_{i-1,j+1} + P_{i,j+1} \qquad (7)$$

Where $i = 1,2,3\ldots\ldots\ldots m$ and $j = m-1, m-2, \ldots\ldots..0$

Analog frequency transformations

The well-known analog frequency transformations from a normalized low-pass filter to the desired filter type are summarized in the second column of Table 1. Let us now consider the case n = 3 . After substituting of the appropriate expression for in (1) and comparing of the result obtained with (2), we observe
that the denominator coefficients can be calculated from by use of the matrices presented in the last column of the Table I. The same holds for the transformation in the numerator. The problem is how to construct the desired matrices for the arbitrary filter order. The analysis of the data for n = 3 leads to conclusion that the problem becomes trivial in case of the low-pass-to-low-pass and low-pass-to-high-pass transformations

$$\mathbf{Q}^{(L)} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \frac{1}{\Omega_u} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\Omega_u^n} \end{bmatrix}$$

(8)

And

$$\mathbf{Q}^{(H)} = \begin{bmatrix} 0 & \cdots & 0 & \Omega_l^n \\ 0 & \cdots & \Omega_l^{n-1} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix} \tag{9}$$

Let us notice that in case of the low-pass to band-pass transformation, all the nonzero elements in each column refer to the expansion of the expression $(K_1 + 1)^r$, where r is the column number (r= 0,1,2…………n) and $K_1 = \Omega_l\Omega_u$. Therefore, matrix $Q^{(B)}$ can be formed by means of the following simple algorithm.

- Create matrix with 2n + 1 rows and n + 1 columns.
- Calculate nonzero elements by use of equation (3) recurrently for j = 0,1,…n

$$Q^{(B)}_{2i+j,n-j} = \frac{(n-j)!k_1^{n-j-i}}{(n-j-i)!i!} \quad i = 0,1,\ldots,n-j \tag{10}$$

The similar algorithm is valid for the low-pass-to-bandstop transformation because matrix $Q^{(s)}$ can be obtained from $Q^{(B)}$ by the column order reversion. Thus, (3) must be replaced with (4)

$$Q^{(S)}_{2i+j,j} = \frac{(n-j)!k_1^{n-j-i}}{(n-j-i)!i!} \quad i = 0,1,\ldots,n-j. \tag{11}$$

Table for analog frequency transformation from a normalized low-pass filter to the desired filter type

TABLE I
COEFFICIENTS CALCULATION DEPENDING ON FREQUENCY TRANSFORMATION

Frequency transformation	Substitution	Coefficients calculation by matrix (n=3)
Lowpass-to-lowpass	$s \rightarrow \frac{s}{\Omega_u}$	$\begin{bmatrix} \hat{A}_0 \\ \hat{A}_1 \\ \hat{A}_2 \\ \hat{A}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/\Omega_u & 0 & 0 \\ 0 & 0 & 1/\Omega_u^2 & 0 \\ 0 & 0 & 0 & 1/\Omega_u^3 \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \end{bmatrix} = \mathbf{Q}^{(L)} \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \end{bmatrix}$
Lowpass-to highpass	$s \rightarrow \frac{\Omega_l}{s}$	$\begin{bmatrix} \hat{A}_0 \\ \hat{A}_1 \\ \hat{A}_2 \\ \hat{A}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \Omega_l^3 \\ 0 & 0 & \Omega_l^2 & 0 \\ 0 & \Omega_l & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \end{bmatrix} = \mathbf{Q}^{(H)} \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \end{bmatrix}$
Lowpass-to-bandpass	$s \rightarrow \frac{s^2 + \Omega_l \Omega_u}{s(\Omega_u - \Omega_l)} = \frac{s^2 + k_1}{s} c_1$	$\begin{bmatrix} \hat{A}_0 \\ \hat{A}_1 \\ \hat{A}_2 \\ \hat{A}_3 \\ \hat{A}_4 \\ \hat{A}_5 \\ \hat{A}_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & k_1^3 \\ 0 & 0 & k_1^2 & 0 \\ 0 & k_1 & 0 & 3k_1^2 \\ 1 & 0 & 2k_1 & 0 \\ 0 & 1 & 0 & 3k_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 c_1 \\ A_2 c_1^2 \\ A_3 c_1^3 \end{bmatrix} = \mathbf{Q}^{(B)} \begin{bmatrix} A_0 \\ A_1 c_1 \\ A_2 c_1^2 \\ A_3 c_1^3 \end{bmatrix}$
Lowpass-to-bandstop	$s \rightarrow \frac{s(\Omega_u - \Omega_l)}{s^2 + \Omega_l \Omega_u} = \frac{s\,c_2}{s^2 + k_1}$	$\begin{bmatrix} \hat{A}_0 \\ \hat{A}_1 \\ \hat{A}_2 \\ \hat{A}_3 \\ \hat{A}_4 \\ \hat{A}_5 \\ \hat{A}_6 \end{bmatrix} = \begin{bmatrix} k_1^3 & 0 & 0 & 0 \\ 0 & k_1^2 & 0 & 0 \\ 3k_1^2 & 0 & k_1 & 0 \\ 0 & 2k_1 & 0 & 1 \\ 3k_1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 c_2 \\ A_2 c_2^2 \\ A_3 c_2^3 \end{bmatrix} = \mathbf{Q}^{(S)} \begin{bmatrix} A_0 \\ A_1 c_2 \\ A_2 c_2^2 \\ A_3 c_2^3 \end{bmatrix}$

Ω_1 and Ω_u are the lower and upper cutoff frequencies, $k_1 = \Omega_l \Omega_u$, $c_1 = 1/c_2$, $c_2 = (\Omega_u - \Omega_l)$.

2.2 Impulse Invariant

This procedure involves choosing the response of the digital filter as an equi-spaced sampled version of the analog filter.

Impulse Invariant Algorithm:

- Step 1: define specifications of filter
  - Ripple in frequency bands
  - Critical frequencies: passband edge, stopband edge, and/or cutoff frequencies.
  - Filter band type: lowpass, highpass, bandpass, bandstop.
- Step 2: linear transform critical frequencies as follow

  $$\Omega = \omega/Ts$$
- Step 3: select filter structure type and its order: Butterworth, Chebyshev type I, Chebyshev type II or inverse Chebyshev, elliptic.
- Step 4: convert Ha(s) to H(z) using linear transform in step 2.
- Step 5: verify the result. If it does not meet requirement, return to step 3.

The impulse invariance method maps the left hand portion of the s-plane into the interior of the unit circle and the right hand portion of the s-plane to the exterior of the unit circle; hence each horizontal strip in the s-plane is overlaid onto the z-plane to form the digital system function from the analog system function. Since any practical analog filter can never be band limited interference is a major consideration. Due to the aliasing that arises in the sampling process the digital frequency response is distinct from the analog filter frequency response. Hence distortion in the frequency response is one of the major limiting factors of this implementation while its advantage lies in the fact that there is a linear relationship between the analog and digital frequency response. Hence in order to prevent sever distortion due to the band limiting this method is restricted to the design of Low and Bandpass Filters.

2.3 Bilinear Transformation:

The Billinear Transformation method overcomes the effect of aliasing that is caused to due the analog frequency response containing components at or beyond the *Nyquist* Frequency. The bilinear transform is a method of compressing the infinite, straight analogue frequency axis to a finite one long enough to wrap around the unit circle once only. This is also sometimes called frequency warping. This introduces a distortion in the frequency. This is undone by pre-warping the critical frequencies of the analog filter (cutoff frequency, center frequency) such that when the analog filter is transformed into the digital filter, the designed digital filter will meet the desired specifications. IIR digital filters can be designed by using bilinear transformation method which is as follows.

Design of Lowpass IIR Digital Filter:

First we illustrate the development of a lowpass IIR digital transfer meeting given specifications by the bilinear transformation method .To this end , we first obtain the specification for a prototype lowpass analog filter from the specifications of the lowpass digital filter using the inverse transformation. We then determine the analog transfer function H_a (s) meeting the specifications of the prototype analog filter . Finally, the analog transfer function H_a (s) is transformed into a digital transfer function H(z) using the bilinear transformation.

The bilinear transformation from the s – plane to the z – plane is given by

$$s = \frac{2\,(1- z^{-1})}{T(1+z^{-1})}$$

Here T represents the sampling interval, the parameter T has no effect on the expression for H(z).For convenience , we choose T=2.

The relation between the digital transfer H(z) and the parent analog transfer function $H_a(s)$ is then given by

$$H(z) = H_a(s)|_{s = \frac{2(1- z^{-1})}{T(1+ z^{-1})}}$$

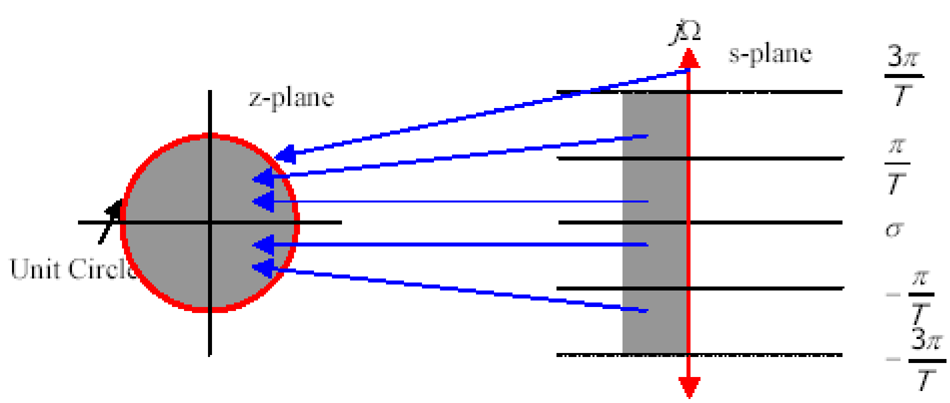


Fig:2.1 bilinear transformation mapping from s- plane to z- plane

The corresponding inverse transformation for T =2 is given by

$$z = \frac{1 + s}{1 - s}$$

The exact relation between the imaginary axis in the s-plane ($s = j\Omega$) and the unit circle in the z-plane ($z = e^{j\omega}$) is of interest, therefore the relation between 3db cut off frequency in digital and analog domain is given by

$$\Omega = \tan(\omega/2)$$

***Design of Highpass, Bandpass, and Bandstop IIR Digital Filters*:**

Follow are the steps to design Highpass, Bandpass, and Bandstop IIR digital Filters

Approach I

1.)Pre-wrap the specified digital frequency specification of the desired digital filter to analog frequency specifications using equation

$$\Omega = \tan(\omega/2)$$

2.)Convert the frequency specification into a prototype analog lowpass filter with appropriate frequency transformation as mention earlier.

3.)Design analog lowpass filter H_{Lp} (s) with Chebyshev, Elliptic, Butterworth or other appropriate methods.

4.)Convert the transfer function $H_{Lp}(s)$ into $H_D(s)$ using the inverse of the frequency transformation used in step 2.

5.) Transform the transfer function $H_D(s)$ using bilinear transformation to arrive the desire digital IIR transfer function**.**

Approach II

1)Prewarp the specified digital frequency specification of the desired digital filter to analog frequency specifications using equation .

$$\Omega = \tan(\omega/2)$$

- Frequency warping inherent in the bilinear transformation of a continuous-time lowpass filter into a discrete-time lowpass filter. To achieve the desired discrete-time cutoff frequencies, the continuous-time cutoff frequencies must be prewarped. Here Td=T=2

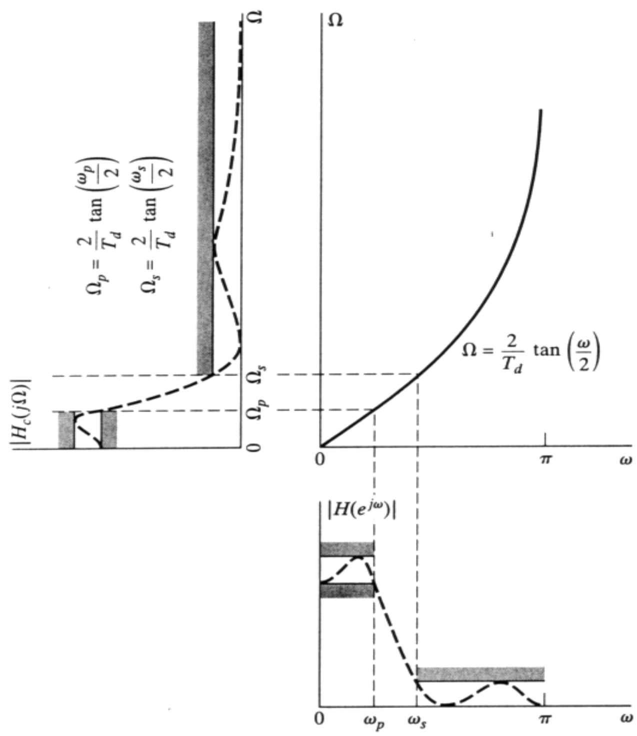


Fig.2.2: Illustration of the frequency warping effect

2) Convert the frequency specification into a prototype analog lowpass filter with appropriate frequency transformation as shows below.

3) Design analog lowpass H_{Lp} (s) filter with Chebyshev, Elliptic, Butterworth or other appropriate methods.

4) Convert transfer function H_{Lp} (s) into transfer function of IIR digital filter using bilinear transformation.

5) Transform transfer function of IIR digital filter to desired transfer function using appropriate spectral transformation as shown earlier.

CHAPTER 3

Filter Types

3.1 Butterworth Filters

Butterworth filters are causal in nature and of various orders, the lowest order being the best (shortest) in the time domain, and the higher orders being better in the frequency domain. Butterworth or maximally flat filters have a monotonic amplitude frequency response which is maximally flat at zero frequency response (Fig 3.1) and the amplitude frequency response decreases logarithmically with increasing frequency. The butterworth filter has minimal phase shift over the filter's band pass when compared to other conventional filters

$$\overline{B(\omega)}B(\omega) = \frac{1}{1+\left(\frac{\omega}{\omega_0}\right)^{2n}}$$

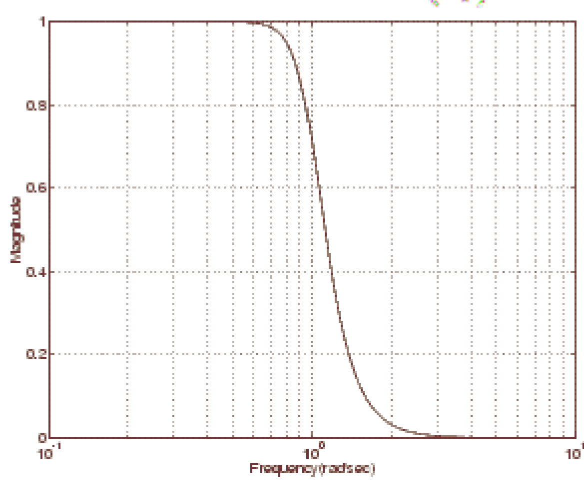


Fig.3.1

3.2 Chebyshev Filters

Chebyshev filters are of two types: Chebyshev I filters are all pole filters which are equiripple in the passband and are monotonic in the stopband. (Fig 3.2)

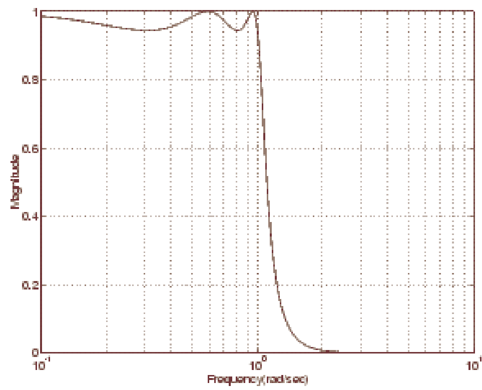


Fig.3.2

Chebyshev II filters contain both poles and zeros exhibiting a monotonic behavior in the passband and equi-ripple in the stopband.(Fig 3.3)

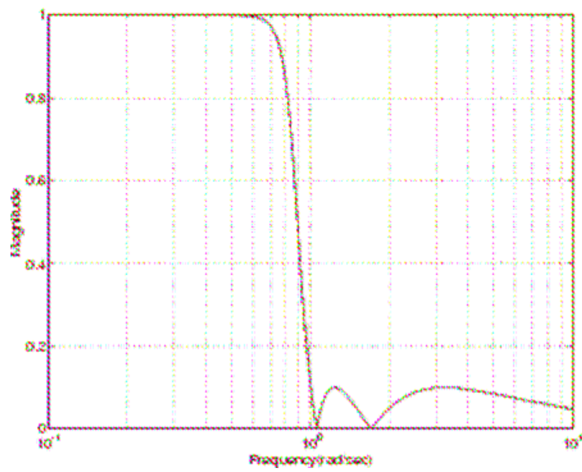


Fig.3.3

The frequency response of the filter is given by

$$|H(\Omega)|^2 = (1 + \epsilon^2 T_N^2(\Omega/\Omega_P))^{-1}$$

where ε is a parameter related to the ripple present in the passband

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

3.3 Elliptic Filters

Elliptic filters are characterized by equi-ripples in both their stop and their passbands. (Fig3.4) They provide a realization with the lowest order for a particular set of conditions.

$$|H(j\Omega)| = 10^{-Rp/20} \quad \Omega = 1$$

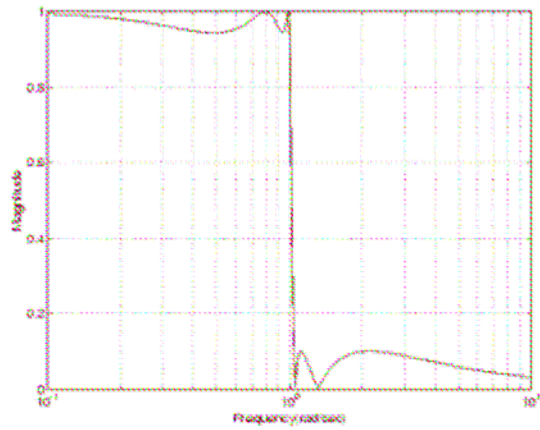


Fig.3.4

Frequency transformations:

This is one of the more common techniques employed in the design of filters. A low pass analog or digital filter may be designed first and then transformed into digital high or bandpass filters.

Analog frequency transformations:

The frequency transformations that can be used to obtain a high pass, low pass, bandpass or band reject filter are indicated below in Table 1.

Here $\Omega_0^2 = \Omega_1 * \Omega_2$ which is defined as the cutoff frequency for a low or highpass and the center frequency for the bandpass and band reject filter

$$Q = \frac{\Omega_0}{\Omega_2 - \Omega_1}$$

where Ω_1 and Ω_2 are the upper and lower cutoff frequencies respectively. $\Omega_2 - \Omega_1$ denotes the bandwidth.

Filter Type	Transformation
Low Pass	$S = s/\Omega_c$
High Pass	$S = \Omega_c/s$
Bandpass	$S = Q(s^2 + \Omega_0^2)/\Omega_0 s$

Table 1

CHAPTER 4

Design and Implementations

The Signal Processing Toolbox in MATLAB includes several useful functions for designing both FIR and IIR digital filters as well as traditional analog filters. The MATLAB functions used are listed *in the appendix*. The design techniques for the digital filters were briefly described in the previous chapters. The basic characteristics of the common analog filter types were also summarized .The filters considered in this project are the Butterworth and Chebyshev Type I .

All four types of design problems are now considered. Lowpass IIR filter, is the subject of **Chapter 4.1,**here I have designed three LP filters. As mentioned earlier, IIR filter design proceeds by conversion of digital specifications to analog specifications and an analog filter is then designed prior to conversion to the required digital filter using the appropriate frequency transformation. In this problem we have taken two IIR Lowpass Digital filters, making the problem a useful example for comparison of different design options.

In **Chapter 4.2**, Highpass Chebyshev Type I filter is designed. This design problems is approached in two ways. The first approach uses the prescribed MATLAB command and frequency transformation by matrix operation by constructing a low-pass analog prototype, followed by a transformation to the respective high-pass by using two matrices . Second approach designs same filter by using Bilinear Z-Transform.

Chapter 4.3 and **4.4** comprises of designing IIR Bandpass and Bandstop filters first by Matrix Operations then by Bilinear Z-Transform.

All the examples are accompanied by the corresponding frequency response characteristics (both magnitude and phase plots), with close-ups given wherever required, impulse response diagrams (truncated to a significant number of terms – in our examples 60 when infinite), and pole-zero diagrams. All the magnitude response plots are in dB and phase response plots in degree. The frequency axis is in terms of normalized frequency.

Assumptions:

- Here frequencies are taken in normalized form.
- Phase angle are taken in degree.
- c =2/Ts, here I have taken Ts=2 in the all the filter design problems, it simplifies the equation (5) given in chapter 2.1.

4.1. Design of Lowpass Filter

In this section we will design digital IIR lowpass filters by using different design techniques .Matlab code for these design techniques are also given after each design. The following Frequency Transformation techniques are used to design Digital IIR Lowpass Filters :

- Matrix operations
- Bilinear Z –transform
- Impulse Invariant.

First filter: Specifications(Matlab codes are given in A1 to A3)

Filter Type	**Digital IIR Butterworth Lowpass**
Sampling frequency	8000Hz
Passband edge frequency	0-500Hz
Stopband frequency	2000-4000Hz
Pasband Ripple(Rp)	3dB
Minimum Stopband Attenuation(Rs)	20dB
Filter order	2

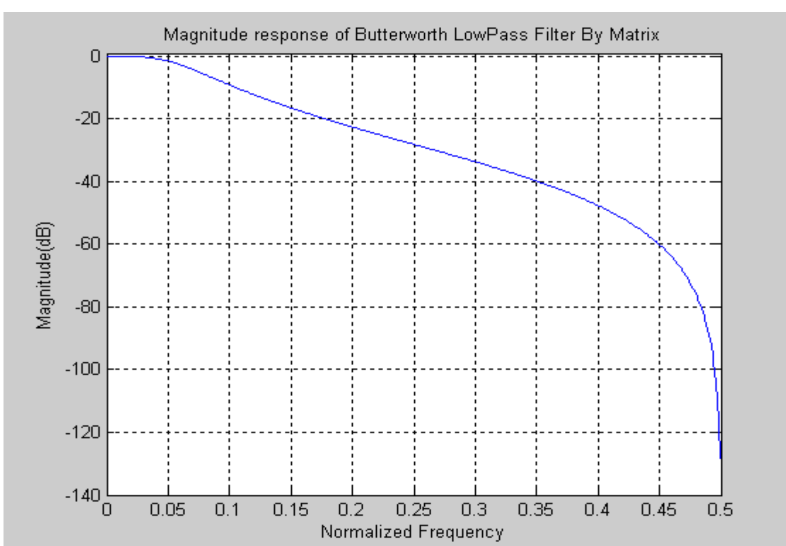


Fig 4.11 Magnitude response of Digital IIR Lowpass filter by Matrix

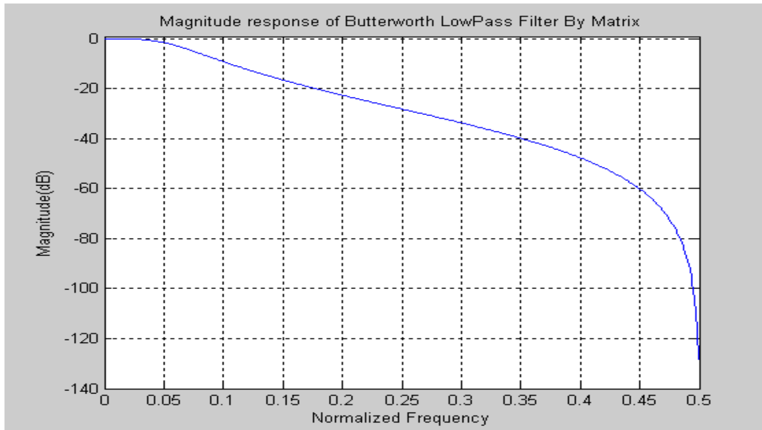
A1

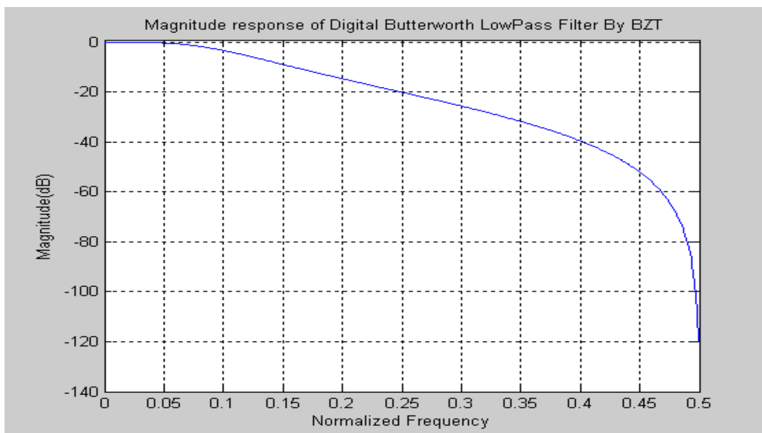
Matlab Code of Digital Butterworth IIR Lowpass filter by Matrix

```
%Filter specifications
Fs=8000;%sampling frequency
Fpb =500;fpb=Fpb/Fs;%Passband edge frequency
Fsb1=2000;fsb1=Fsb1/Fs;%Lower stopband edge frequency
Fsb2=4000;%Upper stopband edge frequency
Rp=3;%maximum passband ripple
Rs=20;%minimum stopband attenuation

[N,Wn]=buttord(fpb,fsb1,Rp,Rs);
disp(N);%Order of LP Filter is 2
[z,p,k1]=buttap(N);
disp(z);
disp(p);
%A=[1 1.414 1]  B=[1 0 0]

Opb=tan(pi*(Fpb)/Fs);
disp(Opb);
capA=[1 0 0;0 1/(Opb) 0;0 0 1/(Opb^2);]*[1 1.414 1]';
disp(capA);
capB=[1 0 0;0 1/(Opb) 0;0 0 1/(Opb^2);]*[1 0 0]';
disp(capB);
%-------------------------------------------------------------------------
m=2;%here m=n(order of  LP Filter)
 for i=0:m
    f(i+1)=(-1)^i*factorial(m)/(factorial(i)*factorial(m-i));
 end
 f
%-------------------------------------------------------------------------
%Calculating the Denominator and Numerator coefficients of Digital LowPass Filter
a1=[1 1 1;2 0 -2;1 -1 1]*[1 7.1087 25.274]';
disp(a1);
a=a1/33.3827;
disp(a);
b1=[1 1 1;2 0 -2;1 -1 1]*[1 0 0]';
b=b1/33.3827;
disp(b);
[h,f]=freqz(b,a,512,1);

 fvtool(b,a);
%Plotting the Magnitude Response
plot(f,20*log10(abs(h))),grid
ylabel('Magnitude(dB)');
xlabel('Normalized Frequency');
title('Magnitude response of Butterworth LowPass Filter By Matrix');

% Plotting the phase response
plot(f,angle(h));
```

```
title('Phase response of Butterworth LowPass Filter By Matrix');
xlabel('Normalized Frequency');
ylabel('Phase(rad)');
grid on;

%  Plotting the Impulse Response
[y,t] = impz(b,a,60);
figure;
stem(t,y);grid
xlabel('Samples');
ylabel('Amplitude');
title('Impulse response of Butterworth LowPass Filter By Matrix');

%  Plotting the Pole-Zero Plot
z = roots(b);%zeros
p = roots(a);%poles
figure;
zplane(z,p);grid
title('Pole-Zero plot of Butterworth LowPass Filter By Matrix');
```

COMPARISONS





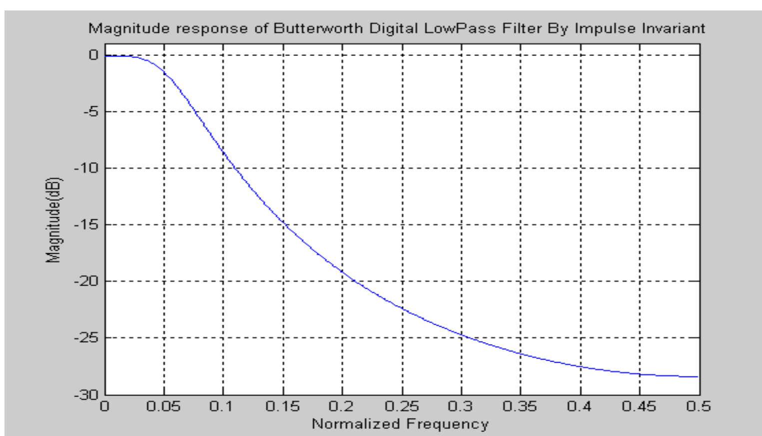
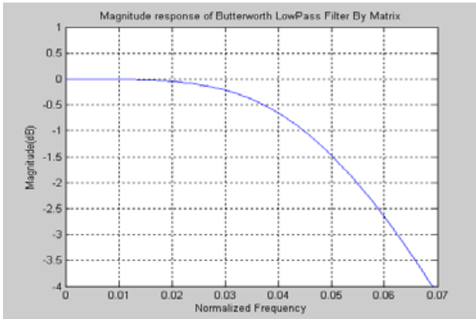
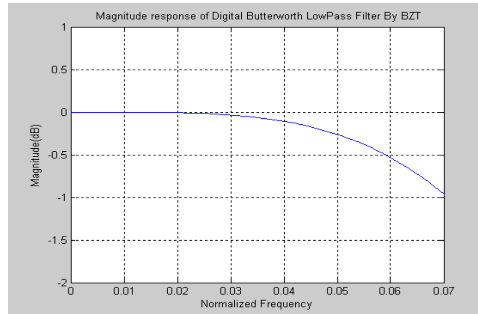


Fig:4.12.Comparison of the Magnitude responses of LPF designed by Matrix with Bilinear –Z Transform and Impulse Invariant.





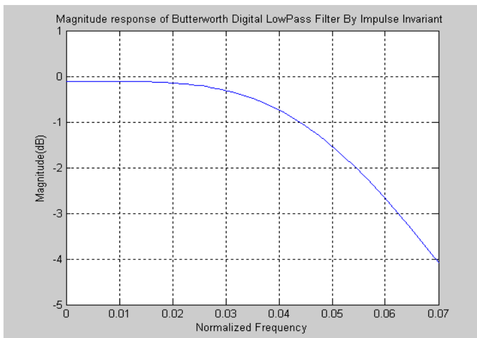
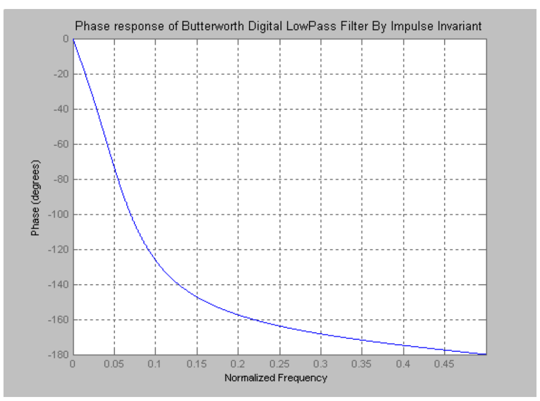
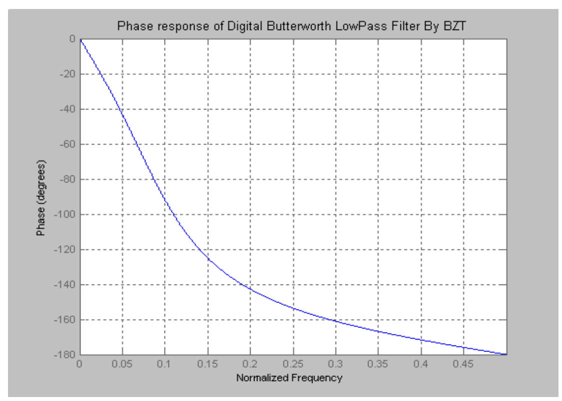


Fig:4.13. Zoom in of the passband





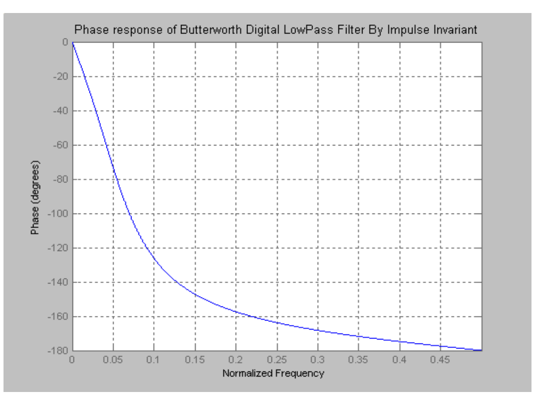


Fig.4.14.Phase response

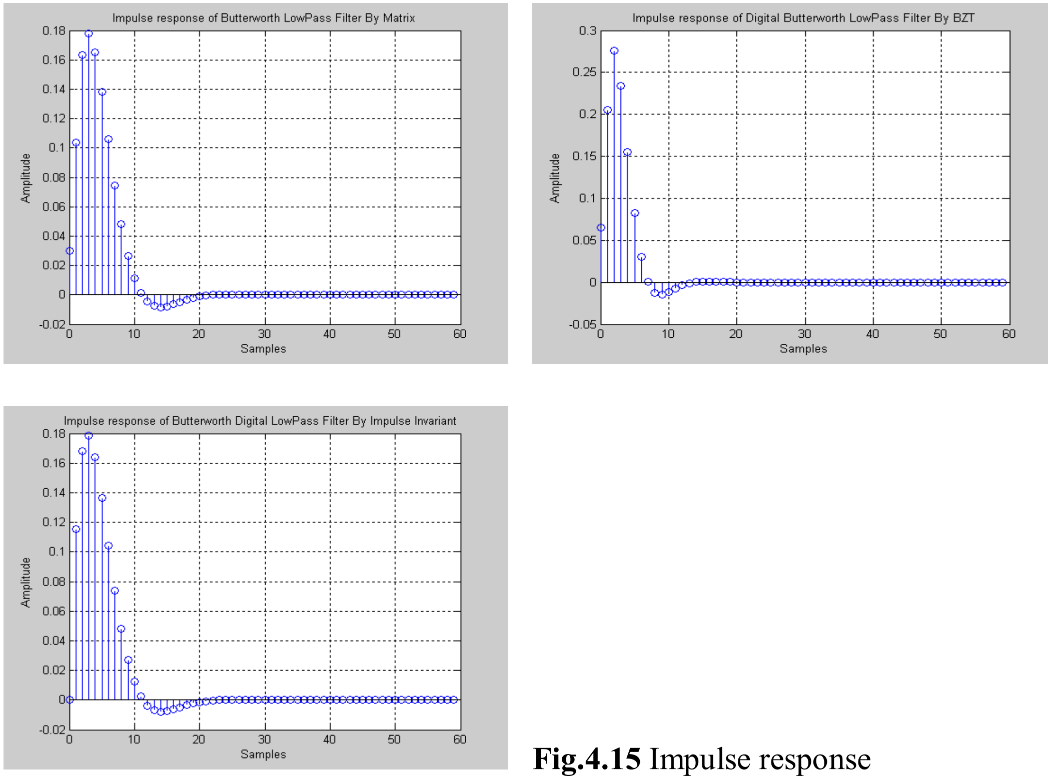


Fig.4.15 Impulse response

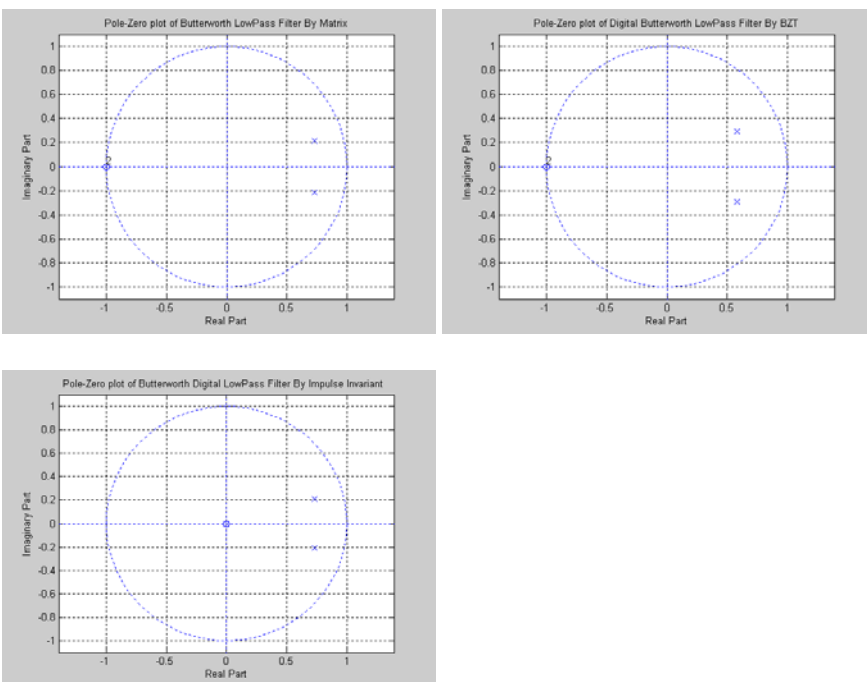


Fig.4.16. Pole-Zero plot

A2

Matlab Code for Digital IIR Lowpass filter by Bilinear Z-transform

```
%Filter Specifications

Fs = 8000; fs = Fs/2; %Sampling Frequency
Fpb = 500; fpb = Fpb/fs; %Passband edge frequency
Fsb = 2000; fsb = Fsb/fs; %Stopband edge frequency
Rp = 3; %Passband Ripple
Rs = 20; %Stopband Attenuation
%----------------------------------------------------------------------
%Butterworth Filter
[N,Wn] = buttord(fpb,fsb,Rp,Rs);
disp(N);

%Order of LPF is 2
[b,a] = butter(N,Wn);
%fvtool(b,a);
[h,f] = freqz(b,a,512,1);
figure;

% Plotting the Magnitude Response
plot(f, 20*log10(abs(h)));
grid on;
title(' Magnitude response of Digital Butterworth LowPass Filter By BZT');
xlabel('Normalized Frequency');
ylabel('Magnitude(dB)');

% Plotting the phase response
plot(f,angle(h));
title(' Phase response of Digital Butterworth LowPass Filter By BZT ');
xlabel('Normalized Frequency');
ylabel('Phase(rad)');
grid on;

%Plotting the Impulse Response
[y,t] = impz(b,a,60);
figure;
stem(t,y);grid
xlabel('Samples');
ylabel('Amplitude');
title('Impulse response of Digital Butterworth LowPass Filter By BZT');

% plotting the Pole-Zero Plot
z = roots(b);%zeros
p = roots(a);%poles
figure;
zplane(z,p);grid
title('Pole-Zero plot of Digital Butterworth LowPass Filter By BZT');
```

A3

Matlab Code for Digital Butterworth IIR Lowpass filter by Impulse Invariant

```
%Filter specifications
Fs=8000;%Sampling frequency
fc=500;%Passband edge frequency
Rp=3;%Maximum Passband Ripple
Rs=20;%Minimum Stopband Attenuation
Wc=2*pi*fc;%Cutoff frequency in radian
N=2;%Order of Low pass filter
[b1,a1]=butter(N,Wc,'s');%Create an analog filter
[z,p,k]=butter(N,Wc,'s');
[b,a]=impinvar(b1,a1,Fs);
 %fvtool(b,a);
%[h,f]=freqz(b,a,512,1);
[h,f]=freqz(b,a,512,1);

%Plotting the Magnitude Response
plot(f,20*log10(abs(h))),grid
xlabel('Normalized Frequency');
ylabel('Magnitude(dB)');
title('Magnitude response of Butterworth Digital LowPass Filter By Impulse Invariant ');

% Plotting the phase response
plot(f,angle(h));
title('Phase response of Butterworth Digital LowPass Filter By Impulse Invariant  ');
xlabel('Normalized Frequency');
ylabel('Phase(rad)');
grid on;

% Plotting the Impulse Response
[y,t] = impz(b,a,60);
figure;
stem(t,y);grid
xlabel('Samples');
ylabel('Amplitude');
title('Impulse response of Butterworth Digital LowPass Filter By Impulse Invariant  ');
% Plotting the Pole-Zero Plot
z = roots(b);%zeros
p = roots(a);%poles
figure;
zplane(z,p);grid
title('Pole-Zero plot of Butterworth Digital LowPass Filter By Impulse Invariant  ');
```

Second Filter: Filter specifications(Matlab codes are given in A4 to A6)

Filter Type	**Digital IIR Butterworth Lowpass**
Sampling frequency	20000Hz
Passband edge frequency	0-700Hz
Stopband frequency	4000-10000Hz
Pasband Ripple(Rp)	3dB
Minimum Stopband Attenuation(Rs)	20dB
Filter order	2

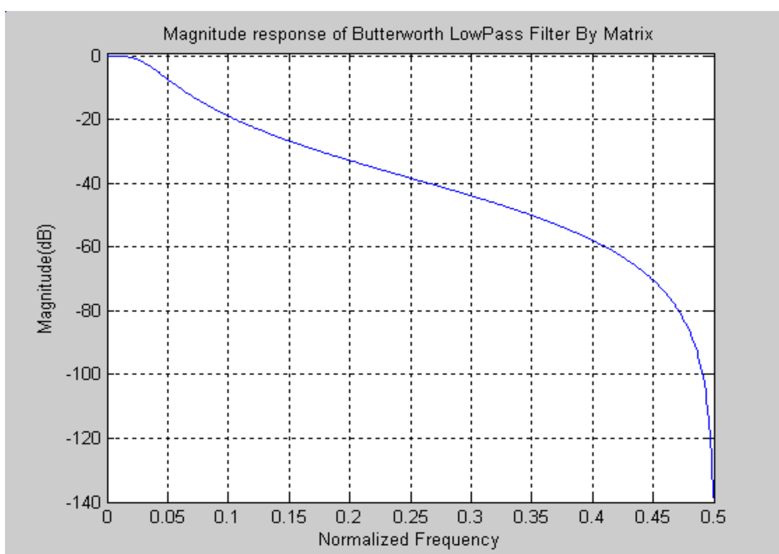


Fig.4.17. Magnitude response of Digital IIR Lowpass filter by Matrix

A4

Matlab Code for Digital Butterworth IIR Lowpass filter by Matrix

```
%Filter specifications

Fs=20000;%sampling frequency
Fpb =700;fpb=Fpb/Fs;%Passband edge frequency

Fsb=4000;fsb=Fsb/Fs;% Stopband edge frequency
Rp=3;%maximum passband ripple
Rs=20;%minimum stopband attenuation

[N,Wn]=buttord(fpb,fsb,Rp,Rs);
disp(N);
[z,p,k1]=buttap(N);
disp(z);
disp(p);
%A=[1 1.414 1] B=[1 0 0]
OmegaP=tan(pi*(Fpb)/Fs);
disp(OmegaP);
capA=[1 0 0;0 1/(OmegaP) 0;0 0 1/(OmegaP^2)]*[1 sqrt(2) 1]';
disp(capA);
%capA=[1.0000  12.8098 82.0453]

capB=[1 0 0;0 1/k 0;0 0 1/(k^2)]*[1 0 0 ]';
disp(capB);

%-----------------------------------------------------------
m=2;%here m=n(order of  LP Filter)
 for i=0:m
    f(i+1)=(-1)^i*factorial(m)/(factorial(i)*factorial(m-i));
 end
 f
%-----------------------------------------------------------
a1=[1 1 1;2 0 -2;1 -1 1]*[1 12.8098 82.0453]';
disp(a1);
a=a1/95.8551;
disp(a);
b1=[1 1 1;2 0 -2;1 -1 1]*[1 0 0]';
b=b1/95.8551;
disp(b);
[h,f]=freqz(b,a,512,1);

 %fvtool(b,a);
%Plotting Magnitude Response
plot(f,20*log10(abs(h))),grid
ylabel('Magnitude(dB)');
xlabel('Normalized Frequency');
title('Magnitude response of Butterworth LowPass Filter By Matrix');
```

```
% Plotting the phase response
plot(f,angle(h));
title('Phase response of Butterworth LowPass Filter By Matrix');
xlabel('Normalized Frequency');
ylabel('Phase(rad)');
grid on;

% Plotting the Impulse Response
[y,t] = impz(b,a,60);
figure;
stem(t,y);grid
xlabel('Samples');
ylabel('Amplitude');
title('Impulse response of Butterworth LowPass Filter By Matrix');

% Plotting the Pole-Zero Plot
z = roots(b);%zeros
p = roots(a);%poles
figure;
zplane(z,p);grid
title('Pole-Zero plot of Butterworth LowPass Filter By Matrix');
```

COMPARISONS

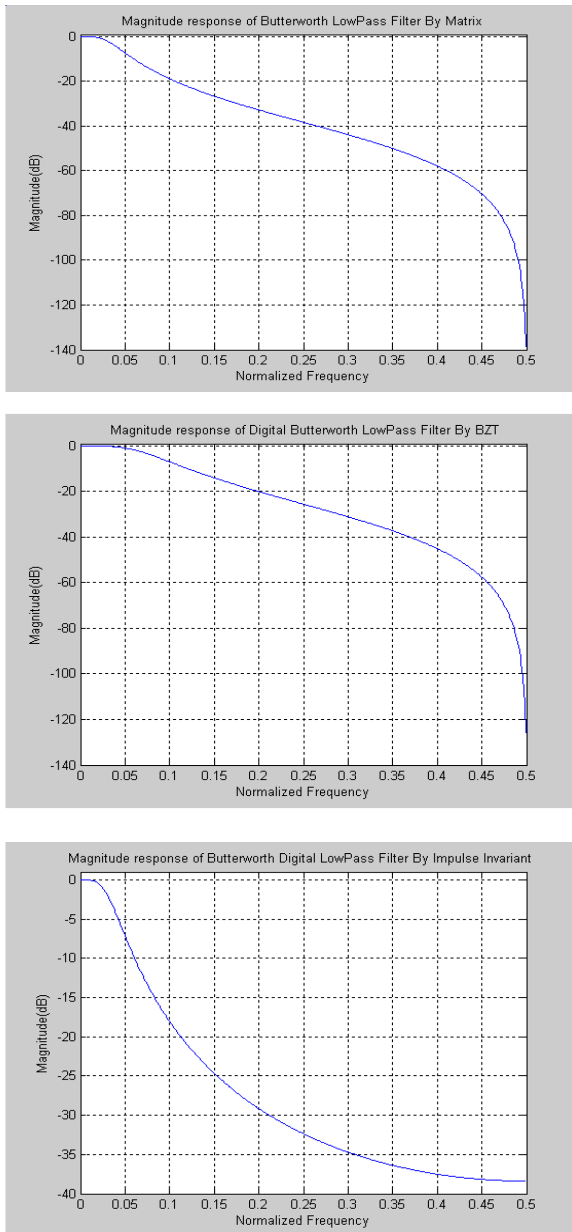


Fig:4.18.Comparison of Magnitude responses of LPF designed by Matrix with Bilinear – Z Transform and Impulse Invariant**.**

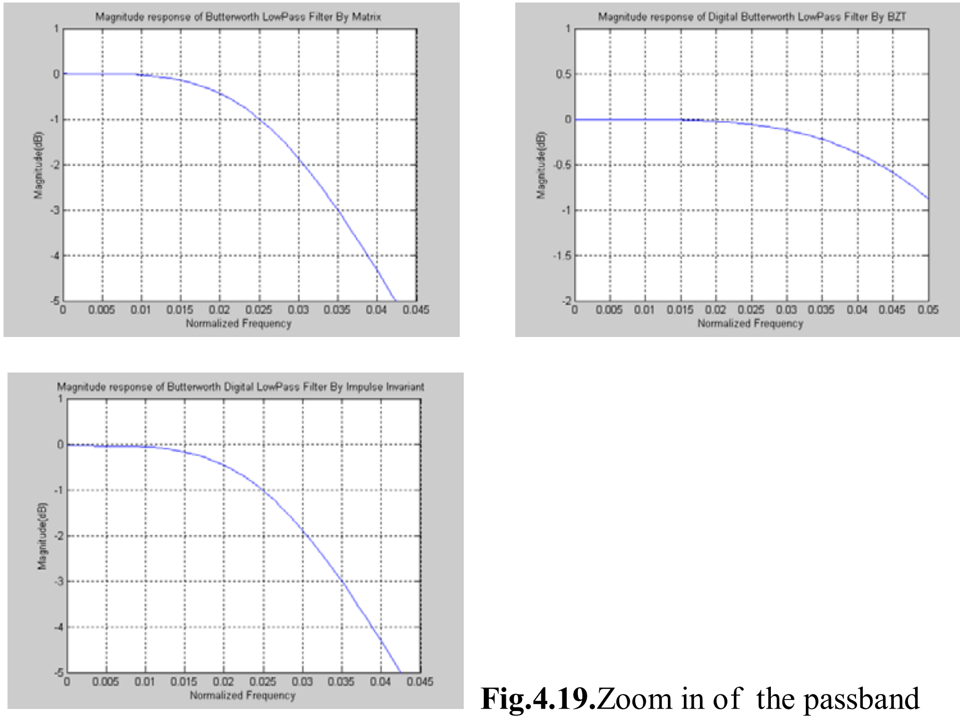


Fig.4.19. Zoom in of the passband

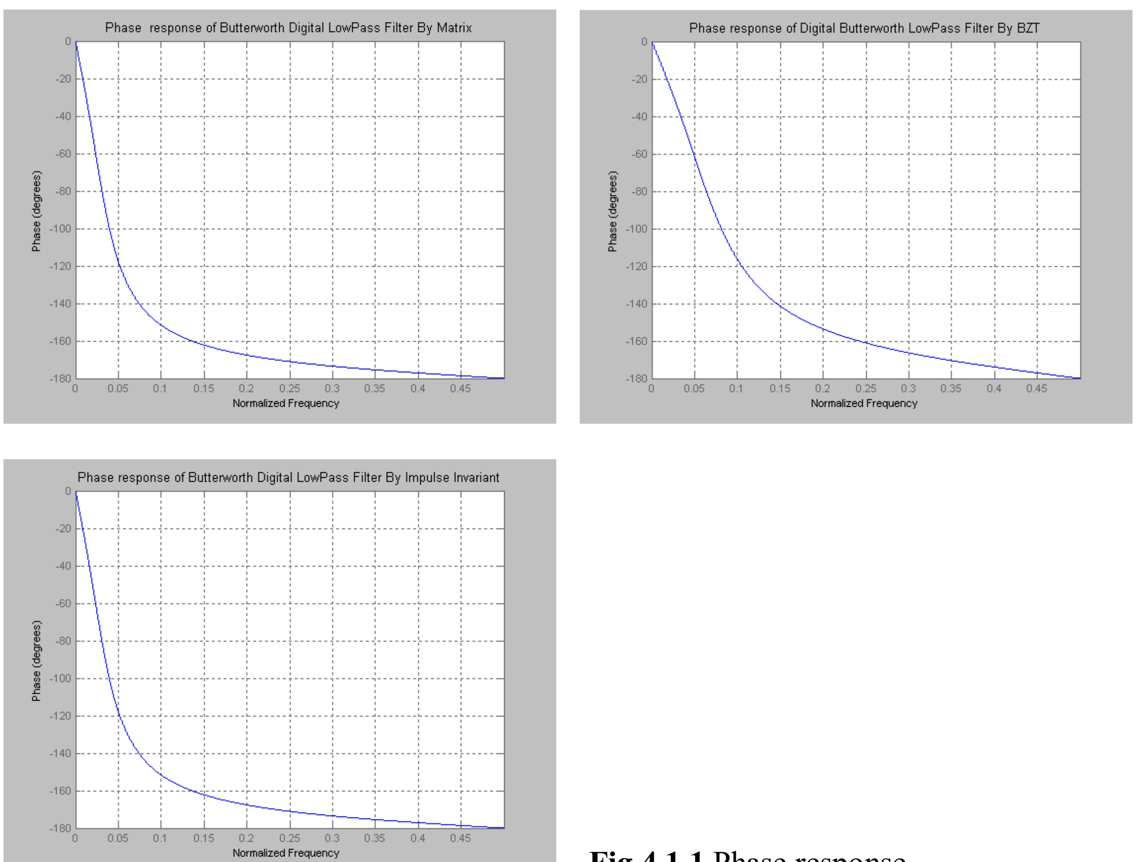
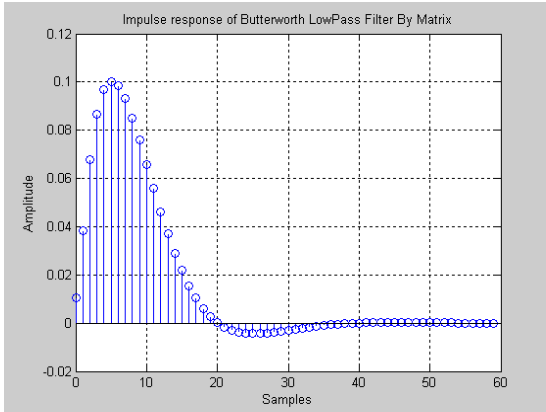
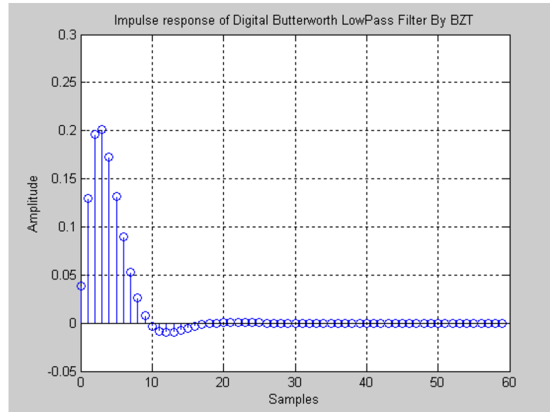


Fig.4.1.1. Phase response





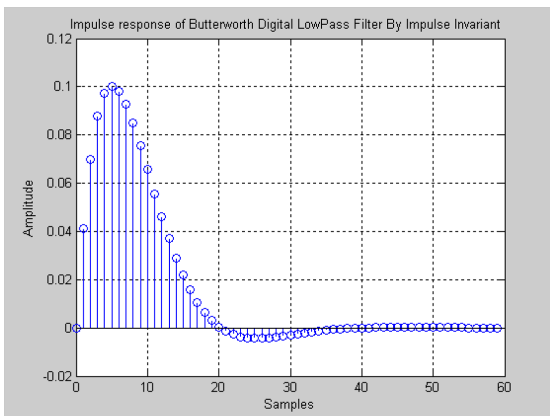
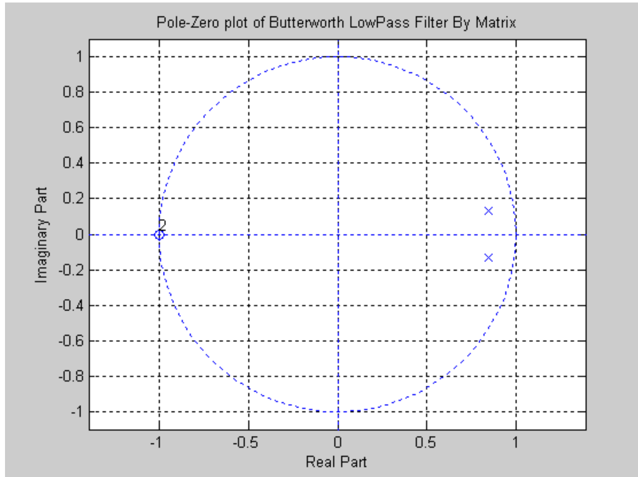
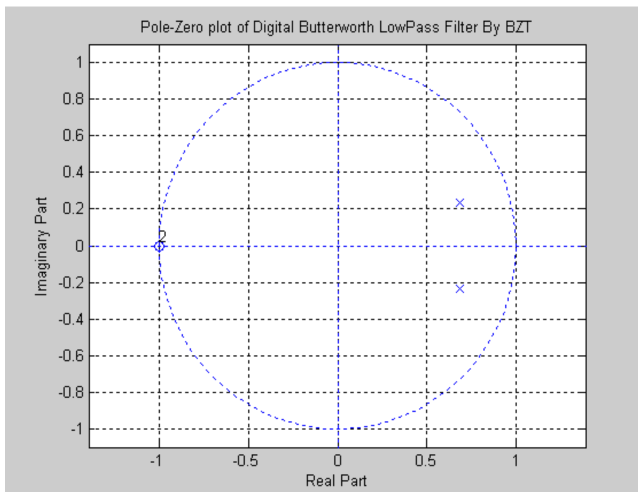


Fig.4.1.2. Impulse response





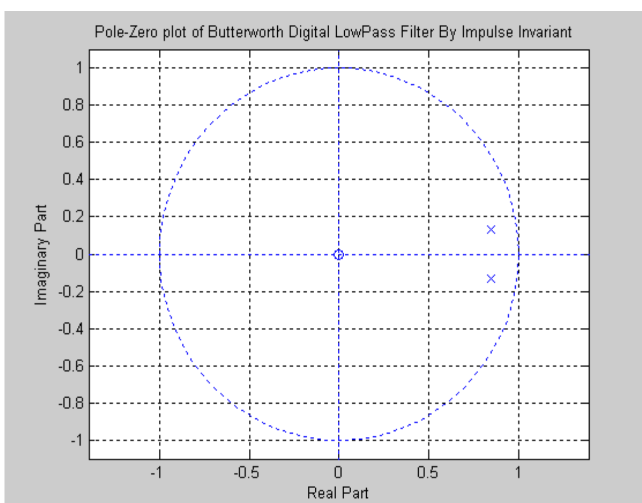


Fig.4.1.3. Pole-Zero plot

A5

Matlab Code for IIR Digital Lowpass filter by Bilinear Z-transform

```
%Filter Specifications

Fs = 20000; fs = Fs/2; %Sampling Frequency
Fpb =700; fpb = Fpb/fs; %Passband edge frequency
Fsb = 4000; fsb = Fsb/fs; %Stopband edge frequency
Rp = 3; %Passband Ripple
Rs = 20; %Stopband Attenuation

[N,Wn] = buttord(fpb,fsb,Rp,Rs);
disp(N);

%Order of LPF is 2
[b,a] = butter(N,Wn);
%fvtool(b,a);
[h,f] = freqz(b,a,512,1);
figure;

% Plotting the Magnitude Response
plot(f, 20*log10(abs(h)));
grid on;
title(' Magnitude response of Digital Butterworth LowPass Filter By BZT');
xlabel('Normalized Frequency');
ylabel('Magnitude(dB)');

% Plotting the phase response
plot(f,angle(h));
title(' Phase response of Digital Butterworth LowPass Filter By BZT ');
xlabel('Normalized Frequency');
ylabel('Phase(rad)');
grid on;

%Plotting the Impulse Response
[y,t] = impz(b,a,60);
figure;
stem(t,y);grid
xlabel('Samples');
ylabel('Amplitude');
title('Impulse response of Digital Butterworth LowPass Filter By BZT');

% plotting the Pole-Zero Plot
z = roots(b);%zeros
p = roots(a);%poles
figure;
zplane(z,p);grid
title('Pole-Zero plot of Digital Butterworth LowPass Filter By BZT');
```

A6

Matlab Code for Digital Butterworth IIR Lowpass filter by Impulse Invariant

```
%Filter specifications

Fs=20000;%Sampling frequency
fc=700;%Passband edge frequency
Rp=3;%Maximum Passband Ripple
Rs=20;%Minimum Stopband Attenuation
Wc=2*pi*fc;%Cutoff frequency in radian
N=2;%Order of Low pass filter
[b1,a1]=butter(N,Wc,'s');%Create an analog filter
[z,p,k]=butter(N,Wc,'s');
[b,a]=impinvar(b1,a1,Fs);
 fvtool(b,a);
%[h,f]=freqz(b,a,512,1);
[h,f]=freqz(b,a,512,1);

%Plotting the Magnitude Response
plot(f,20*log10(abs(h))),grid
xlabel('Normalized Frequency');
ylabel('Magnitude(dB)');
title('Magnitude response of Butterworth Digital LowPass Filter By Impulse Invariant ');

% Plotting the phase response
plot(f,angle(h));
title('Phase response of Butterworth Digital LowPass Filter By Impulse Invariant  ');
xlabel('Normalized Frequency');
ylabel('Phase(rad)');
grid on;

% Plotting the Impulse Response
[y,t] = impz(b,a,60);
figure;
stem(t,y);grid
xlabel('Samples');
ylabel('Amplitude');
title('Impulse response of Butterworth Digital LowPass Filter By Impulse Invariant  ');

% Plotting the Pole-Zero Plot
z = roots(b);%zeros
p = roots(a);%poles
figure;
zplane(z,p);grid
title('Pole-Zero plot of Butterworth Digital LowPass Filter By Impulse Invariant  ');
```

Third Filter: Filter Specifications (Matlab codes are given in A7 to A9)

Filter Type	**Butterworth LP Filter**
Passband edge frequency	0-60Hz
Stopband edge frequency	85-128Hz
Sampling frequency(Fs)	256Hz
Passband Ripple(Rp)	3dB
Stopband Attenuation(Rs)	15dB
Order(N)	3

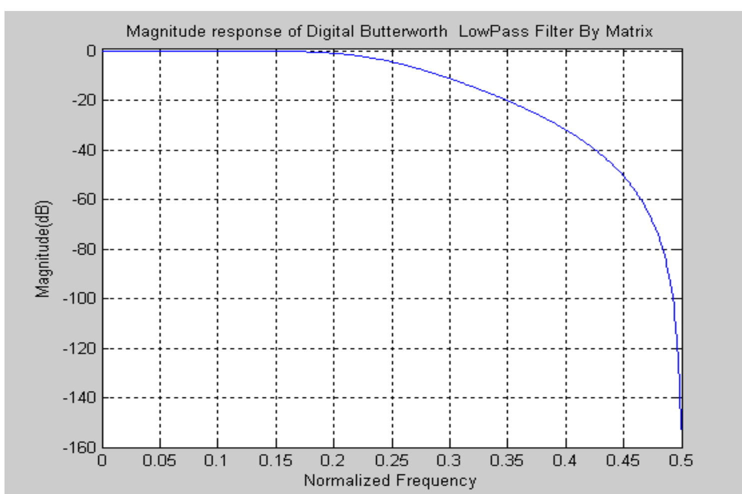


Fig.4.1.4 Magnitude response of Digital IIR Lowpass filter by Matrix

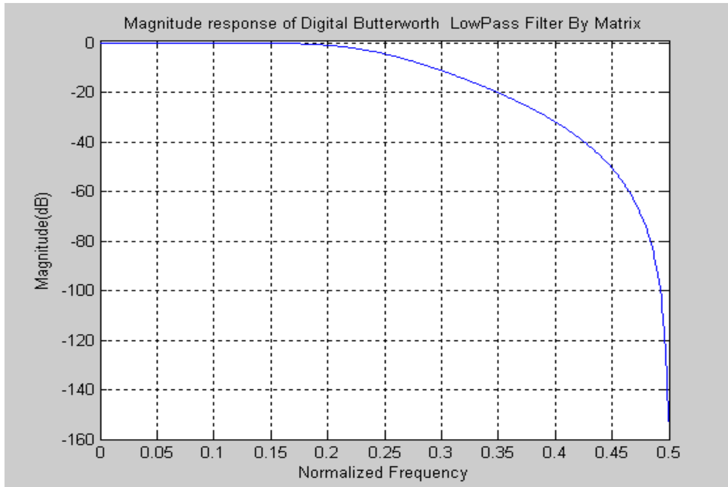
A7

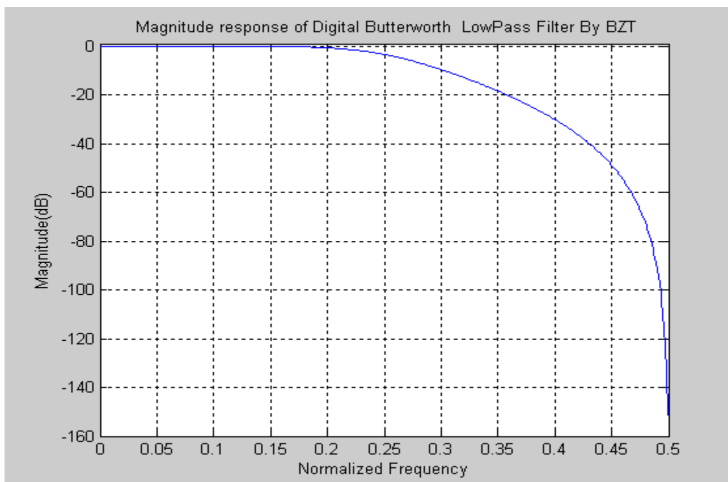
Matlab Code for Digital Butterworth IIR Lowpass filter by Matrix

```
 %Filter Specifications
Fs=256;%Sampling frequency
Fpb = 60;fpb=Fpb/Fs; % passband edge frequency
Fsb = 85;fsb=Fsb/Fs;% stopband edge frequency
Rp=3;%Maximum passband ripple
Rs=15;%Minimum stopband attenuation

[N,Wn]=buttord(.906347,1.7158,3,15,'s');
disp(N);%Order of LPF is 3
[z,p,k]=buttap(N);%Find zero,pole and gain of normalized analog lowpass filter
%B=[1 0 0 0]  Numerator coefficient of analog low pass filter
%here we are using Matrix(described in Table 1) for  frequency transformation from analog
%low pass to desired analog lowpass filter
capB=[1 0 0 0;0 1.103155 0 0;0 0 1.2173 0;0 0 0 1.342486]*[1 0 0 0]';
disp(capB);
%capB=[1    0     0     0]
%A=[1 2 2 1] Denominator coefficients of analog lowpass filter
capA=[1 0 0 0;0 1.103155 0 0;0 0 1.2173 0;0 0 0 1.342486]*[1 2 2 1]';
disp(capA);
%capA=[1.0000 2.2063 2.4346  1.3425]
%--------------------------------------------------------------------------
 m=3;%here m=n(order of  LP Filter)
 for i=0:m
    f(i+1)=(-1)^i*factorial(m)/(factorial(i)*factorial(m-i));
 end
 f
 %f =    1    -3     3    -1
%--------------------------------------------------------------------------
%Calculating the coefficients of Numerator and Denominator for digital IIR Lowpass filter. Here
we are
%using pascal matrix for analog frequency transformation from analog
%lowpass to desired digital lowpass filter
b1=[1 1 1 1;3 1 -1 -3;3 -1 -1 3;1 -1 1 -1]*[1 0 0 0]';
% disp(B);
b=b1/6.9834;
disp(b);
%b=[0.1432 0.4296  0.4296   0.1432] Numerator coefficients of desired digital LP filter

a1=[1 1 1 1;3 1 -1 -3;3 -1 -1 3;1 -1 1 -1]*[1 2.2063 2.4346 1.3425]';
% disp(A);
%  6.9834    -1.2558     2.3866    -0.1142
a=a1/6.9834;
disp(a);
%a=[1.0000  -0.1798   0.3418   -0.0164] Denominator coefficients of desired digital LP filter
fvtool(b,a);
```

```
%[h,f]=freqz(b,a,512,256);
[h,f]=freqz(b,a,512,1);

%Plotting the Magnitude Response
plot(f,20*log10(abs(h))),grid
xlabel('Normalized Frequency')
ylabel('Magnitude(dB)')
title('Magnitude response of Digital Butterworth  LowPass Filter By Matrix');

% Plotting the Phase response
plot(f,angle(h));
title('Phase response of Digital Butterworth  LowPass Filter By Matrix');
xlabel('Normalized Frequency');
ylabel('Phase(rad)');
grid on;

% Plotting the Impulse Response
[y,t] = impz(b,a,60);
figure;
stem(t,y);grid
xlabel('Samples');
ylabel('Amplitude');
title('Impulse response of Digital Butterworth  LowPass Filter By Matrix');

% Plotting the Pole-Zero Plot
z = roots(b);%zeros
p = roots(a);%poles
figure;
zplane(z,p);grid
title('Pole-Zero plot of Digital Butterworth  LowPass Filter By Matrix');
```

COMPARISONS





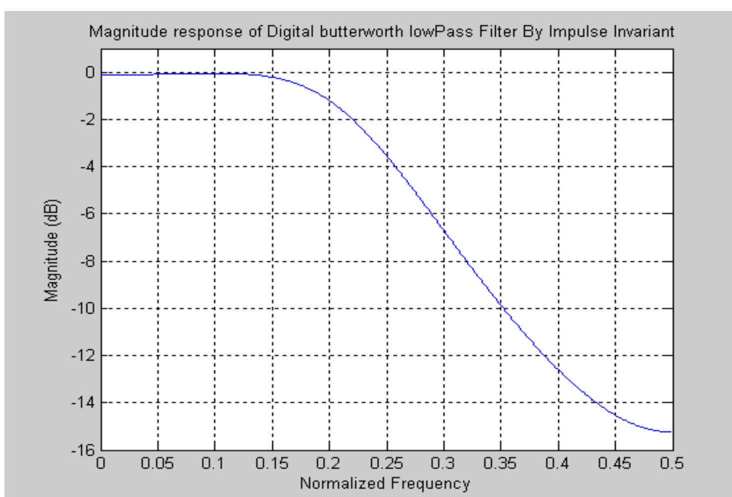
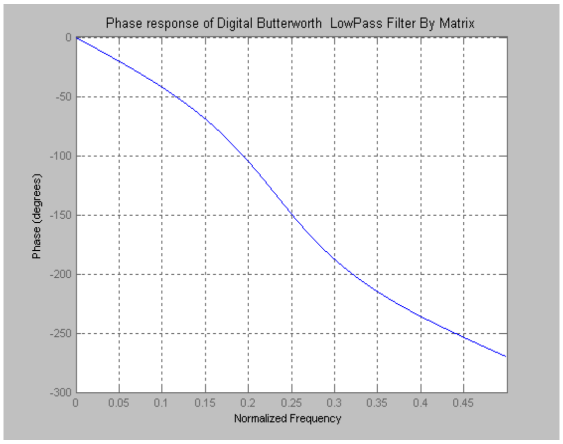
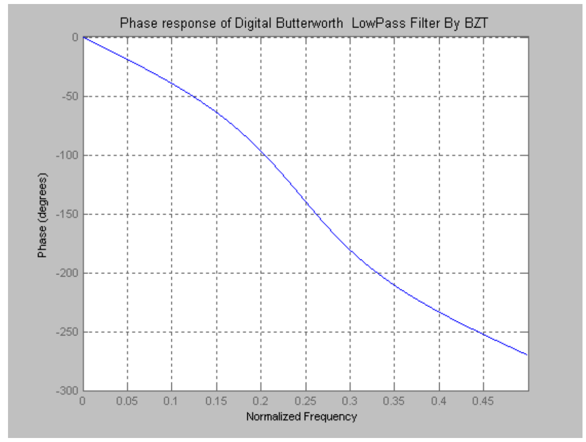


Fig.4.1.5 Comparison of Magnitude responses of LPF designed by Matrix with Bilinear – Z Transform and Impulse Invariant.





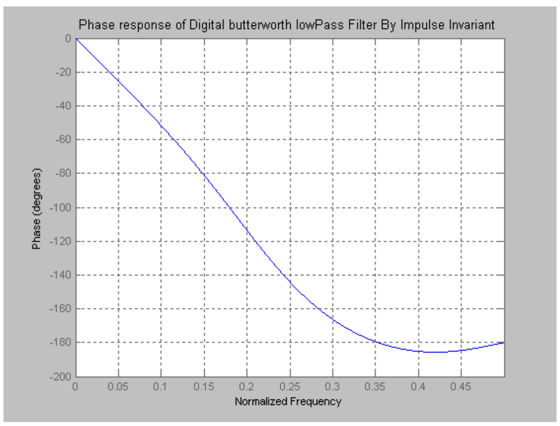
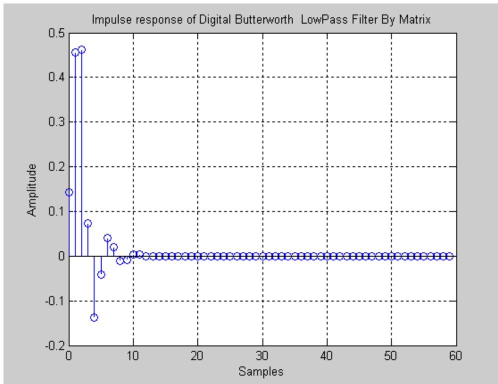
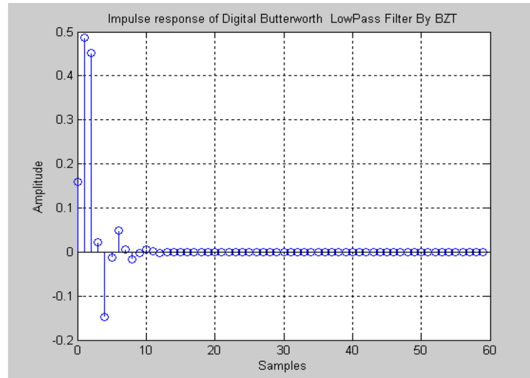


Fig.4.1.6 Phase response





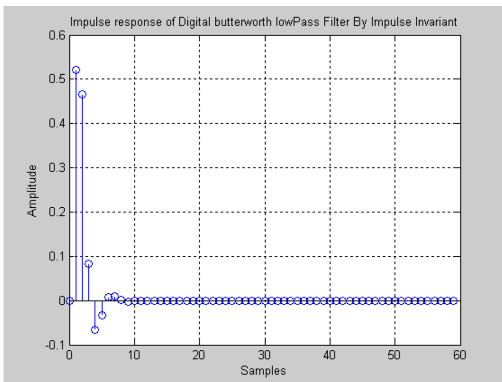


Fig.4.1.7 Impulse response

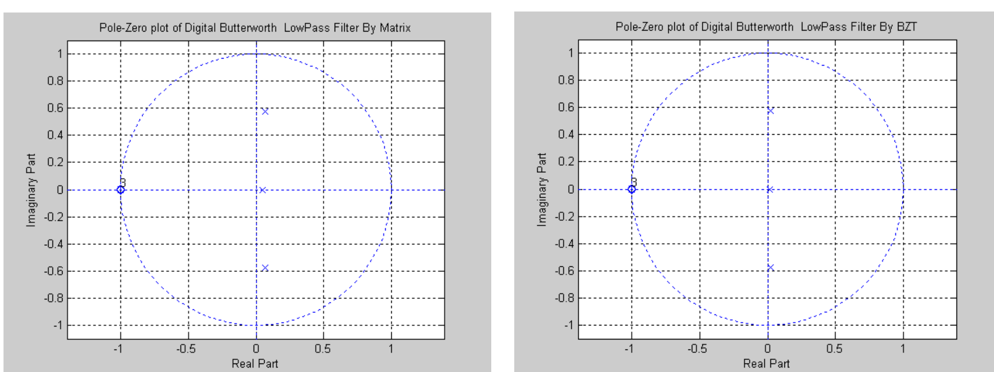


Fig.4.1.8 Pole-Zero plot

A8

Matlab Code for Digital Butterworth IIR Lowpass filter by Bilinear Z-transform

```
%Filter Specifications

Fs = 256; fs = Fs/2; %Sampling Frequency
Fpb = 60; fpb = Fpb/fs; %Passband edge frequency
Fsb = 85; fsb = Fsb/fs; %Stopband edge frequency
Rp = 3; %Passband Ripple
Rs = 15; %Stopband Attenuation
%------------------------------------------------------------------------
[N,Wn] = buttord(fpb,fsb,Rp,Rs);
disp(N);

%Order of LPF is 2
[b,a] = butter(N,Wn);sss
fvtool(b,a);
[h,f] = freqz(b,a,512,1);
figure;

% Plotting the magnitude Response
plot(f, 20*log10(abs(h)));
grid on;
title(' Magnitude response of Digital Butterworth  LowPass Filter By BZT');
xlabel('Normalized Frequency');
ylabel('Magnitude(dB)');

% Plotting the phase response
plot(f,angle(h));
title(' Phase response of Digital Butterworth  LowPass Filter By BZT');
xlabel('Normalized Frequency');
ylabel('Phase(rad)');
grid on;

%Plotting the Impulse Response
[y,t] = impz(b,a,60);
figure;
stem(t,y);grid
xlabel('Samples');
ylabel('Amplitude');
title('Impulse response of Digital Butterworth  LowPass Filter By BZT');
% plotting the Pole-Zero Plot
z = roots(b);%zeros
p = roots(a);%poles
figure;
zplane(z,p);grid
title('Pole-Zero plot of Digital Butterworth  LowPass Filter By BZT');
```

A9

Matlab Code for Digital Butterworth IIR Lowpass filter by Impulse Invariant

```
%Filter specifications

Fs=256;%Sampling frequency
fc=60;%Passband edge frequency
Rp=3;%Maximum Passband Ripple
Rs=15;%Minimum Stopband Attenuation
Wc=2*pi*fc;%Cutoff frequency in radian
N=3;%Order of Low pass filter
[b1,a1]=butter(N,Wc,'s');%Create an analog filter
[z,p,k]=butter(N,Wc,'s');
[b,a]=impinvar(b1,a1,Fs);

 fvtool(b,a);
%[h,f]=freqz(b,a,512,1);
[h,f]=freqz(b,a,512,1);

%Plotting the Magnitude Response
plot(f,20*log10(abs(h))),grid
xlabel('Normalized Frequency');
ylabel('Magnitude (dB)');
title('Magnitude response of Digital Butterworth LowPass Filter By Impulse Invariant ');

% Plotting the phase response
plot(f,angle(h));
title('Phase response of Digital Butterworth LowPass Filter By Impulse Invariant ');
xlabel('Normalized Frequency');
ylabel('Phase(rad)');
grid on;

% Plotting the Impulse Response
[y,t] = impz(b,a,60);
figure;
stem(t,y);grid
xlabel('Samples');
ylabel('Amplitude');
title('Impulse response of Digital Butterworth LowPass Filter By Impulse Invariant');

% Plotting the Pole-Zero Plot
z = roots(b);%zeros
p = roots(a);%poles
figure;
zplane(z,p);grid
title('Pole-Zero plot of Digital Butterworth LowPass Filter By Impulse Invariant');
```

4.2.Design of Highpass Filter

This section comprises , the design of a digital IIR highpass filter with BZT and Matrix method,after designing we will compare the results .Specification of chebyshev highpass filter are given below **.**Matlab codes for this filter are given in programs A10 and A11.

Filter	**Chebyshev Type 1 HPF**
Passband edge frequency	700Hz
Stopband edge frequency	500Hz
Passband ripple(Rp)	1dB
Stopband attenuation(Rs)	32dB
Sampling frequency(Fs)	2000Hz
Order(N)	4

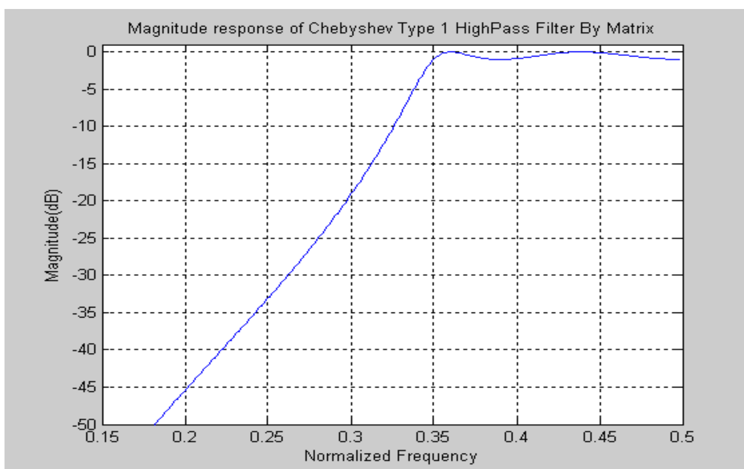


Fig.4.21 Magnitude response of Digital IIR Highpass filter by Matrix

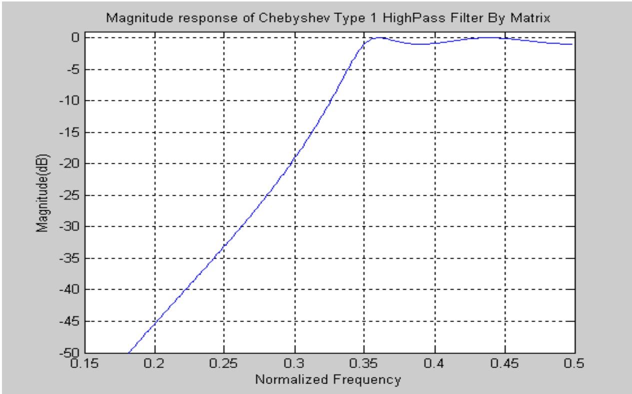
A10

Matlab Code for Digital IIR Chebyshev Type 1 Higpass filter by Matrix

```
%Filter specifications

Fs=2000; % Sampling Frequency
Rp=1; % Passband Ripple
Rs=32; % Stopband Attenuations
Fstop=500; % Stopband Cutoff Frequency
Fpass=700; % Passband Cutoff Frequency

%---------------------------------------------------------------------------
Wp=2*pi*Fpass/Fs;
Ws=2*pi*Fstop/Fs;
OmegaP=tan(Wp/2); % Normalized angular passband edge frequency of analog Highpass filter
%disp(OmegaP);
OmegaS=tan(Ws/2); % Normalized angular stopband edge frequency of analog Highpass filter

[N,Wn]=cheb1ord(OmegaS,OmegaP,1,32,'s');%Calculate order
[B,A]=cheby1(N,1,Wn,'s');
disp(A);
%A=[.2756 .7426 1.4539 .9528 1]
%B=[.2457 0 0 0 0]
disp(B);
disp(N);%Order of HPF is equal to LPF which is 4
w=1.9626%w=Omega

capA=[0 0 0 0 w^4;0 0 0 w^3 0;0 0 w^2 0 0;0 w 0 0 0;1 0 0 0 0]*[.2756 .7426 1.4539 .9528 1]';
capB=[0 0 0 0 w^4;0 0 0 w^3 0;0 0 w^2 0 0;0 w 0 0 0;1 0 0 0 0]*[.2457 0 0 0 0]';
disp(capA);
%capA=[14.8367 7.2028 5.6002 1.4574 0.2756]

disp(capB);

%capB=[ 0 0 0 0 .2457]

%.....................................................
m=4;%here m=n(order of  LP Filter)
 for i=0:m
    f(i+1)=(-1)^i*factorial(m)/(factorial(i)*factorial(m-i));
 end
 f
 %f =     1    -4     6    -4     1
 %..........................................................
 %Calculating the Numerator and Denominator coefficients of Digital Highpass Filter
 a1=[1 1 1 1 1;4 2 0 -2 -4;6 0 -2 0 6;4 -2 0 2 -4;1 -1 1 -1 1]*[14.8367 7.2028 5.6002 1.4574
0.2756]';
 disp(a1);
```

```
%a1=[29.3727 69.7352  79.4734  46.7536   12.0523]
a=a1/29.3727;
disp(a);
%a=[1.0000  2.3742   2.7057   1.5917   0.4103]
b1=[1 1 1 1 1;4 2 0 -2 -4;6 0 -2 0 6;4 -2 0 2 -4;1 -1 1 -1 1]*[ 0 0 0 0 .2457]';
b=b1/29.3727;
disp(b);
%b=[0.0836  -0.3346   0.5019  -0.3346   0.0836]
 fvtool(b,a);
[h,f] = freqz(b,a,512,1);
figure;

% Plotting the magnitude Response
plot(f, 20*log10(abs(h)));
grid on;
title('Magnitude response of Chebyshev Type 1 HighPass Filter By Matrix');
xlabel('Normalized Frequency');
ylabel('Magnitude(dB)');

% Plotting the phase response
plot(f,angle(h));
title('Phase response of Chebyshev Type 1 HighPass Filter By Matrix');
xlabel('Normalized Frequency');
ylabel('Phase(rad)');
grid on;

% Plotting the Impulse Response
[y,t] = impz(b,a,60);
figure;
stem(t,y);grid
xlabel('Samples');
ylabel('Amplitude');
title('Impulse response of Chebyshev Type 1 HighPass Filter By Matrix');

% Plotting the Pole-Zero Plot
z = roots(b);%zeros
p = roots(a);%poles
figure;
zplane(z,p);grid
title('Pole-Zero Plot of Chebyshev Type 1 HighPass Filter By Matrix');
```

COMPARISONS



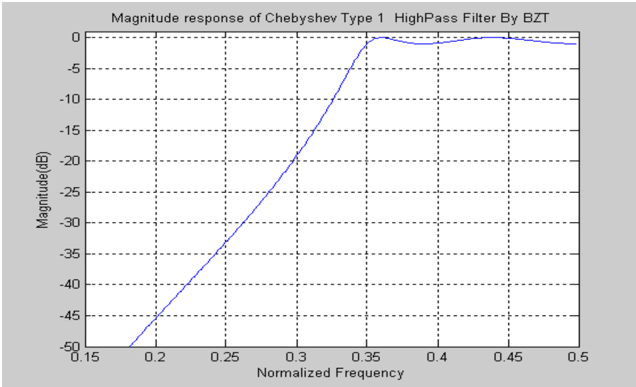
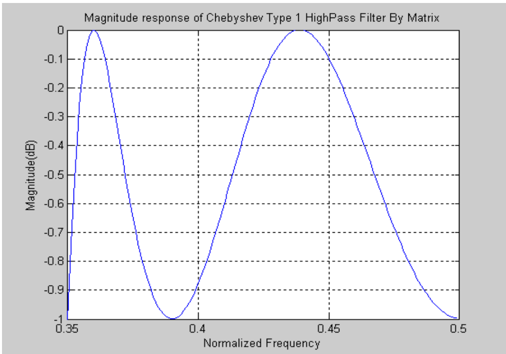


Fig.4.22. Magnitude response



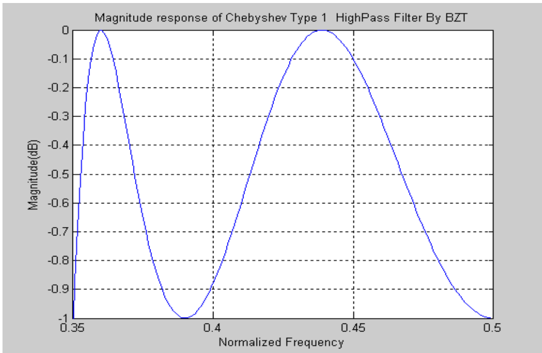


Fig.4.23. Zoom in of the passband

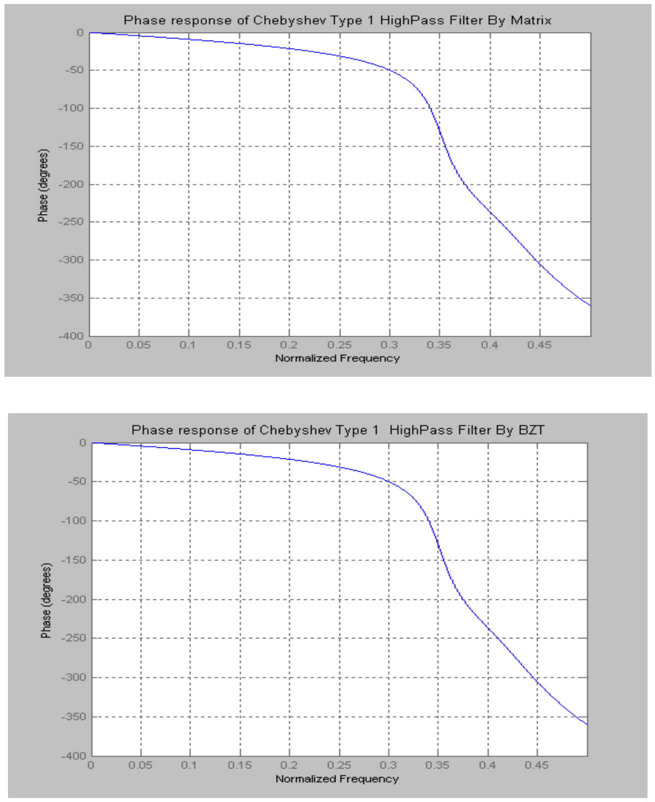


Fig.4.24. Phase response

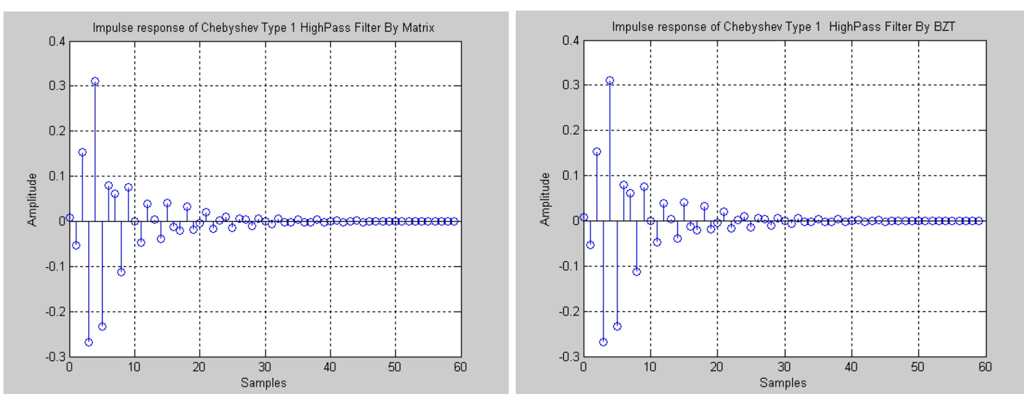
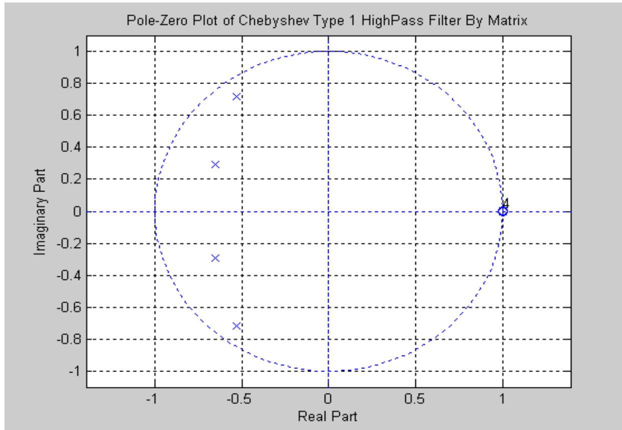


Fig.4.25. Impulse response



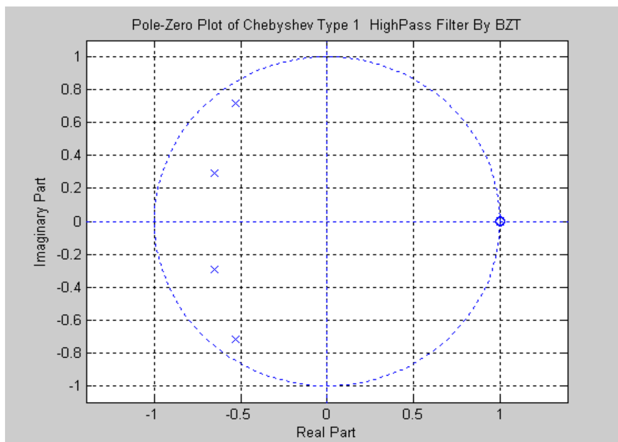


Fig.4.26. Pole-Zero plot

A 11

Matlab Code for Digital Chebyshev Type 1 IIR Highpass filter by BZT

```
% Filter specifications

Fs=2000; % Sampling Frequency
Rp=1; % Passband Ripple
Rs=32; % Stopband Attenuations
Fstop=500; % Stopband Cutoff Frequency
Fpass=700; % Passband Cutoff Frequency

%---------------------------------------------------------------------------
Wp=2*pi*Fpass/Fs;
Ws=2*pi*Fstop/Fs;
OmegaP=tan(Wp/2); % Normalized angular passband edge frequency of analog Highpass filter
OmegaS=tan(Ws/2); % Normalized angular stopband edge frequency of analog Highpass filter
% Wn=1-Fpass/(Fs/2); % Find Cutoff frequency

[N,Wn]=cheb1ord(OmegaS,OmegaP,1,32,'s');%Calculate order
[B,A]=cheby1(N,1,Wn,'s');
disp(N);%Order of HPF is equal to LPF which is 4
[BT,AT]=lp2hp(B,A,OmegaP); % Lowpass to Highpass Filter
[b,a]=bilinear(BT,AT,.5); % Bilinear Transformation

%fvtool(b,a);
%Plotting the Magnitude response
[h,f] =freqz(b,a,512,1); % Find Frequency Response
plot(f,20*log10(abs(h))),grid

grid on;
title('Magnitude response of Chebyshev Type 1  HighPass Filter By BZT');
xlabel('Normalized Frequency');
ylabel('Magnitude(dB)');

% Plotting the phase response
plot(f,angle(h));grid
title('Phase response of Chebyshev Type 1  HighPass Filter By BZT');
xlabel('Normalized Frequency');
ylabel('Phase(rad)');

% Plotting the Impulse Response
[y,t] = impz(b,a,60);
figure;
stem(t,y);grid
xlabel('Samples');
ylabel('Amplitude');
title('Impulse response of Chebyshev Type 1  HighPass Filter By BZT');
```

```
% Plotting the Pole-Zero Plot
z = roots(b);%zeros
p = roots(a);%poles
figure;
zplane(z,p);grid
title('Pole-Zero Plot of Chebyshev Type 1  HighPass Filter By BZT');
```

4.3 Design of Bandpass Filter

In this section we will design two IIR Bandpass filters by using both bilinear transformation and Pascal matrix operation. Then we will compare the magnitude response ,phase response , impulse response and location of poles and zeros by using both techniques.

First Filter: Specifications (Matlab codes are given in A12 to A13)

Filter Type	**Butterworth BP Filter**
Sampling frequency(Fs)	10 Hz
Lower Passband edge frequency	2Hz
Upper passband edge frequency	4Hz
Passband Ripple(Rp)	1dB
Order(N)	6

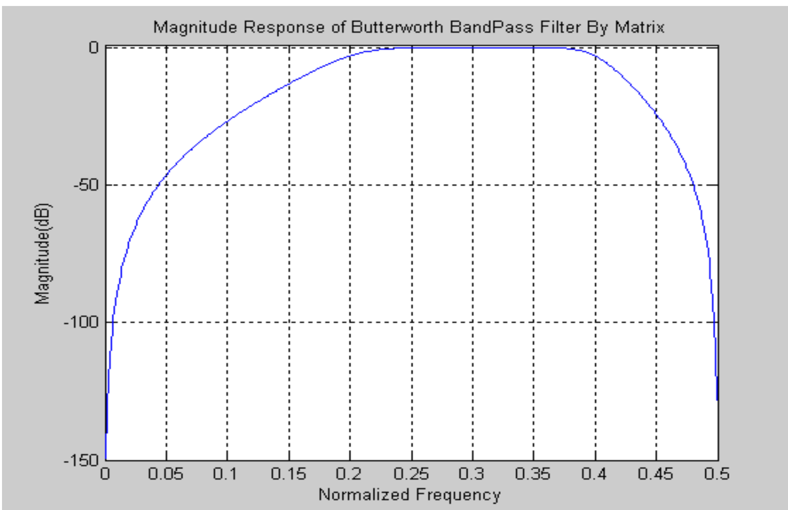


Fig.4.31. Magnitude response of Digital Butterworth Bandpass filter by Matrix

A12

Matlab Code for Digital Butterworth IIR Bandpass filter by Matrix

```
%Filter Specifications
Fs = 10;fs = Fs/2; %Sampling Frequency
Fpb1 = 2;fpb1 = Fpb1/fs; %Lower Passband edge frequency
Fpb2= 4;fpb2 = Fpb2/fs; %Upper passband edge frequency
Rp = 1; %Passband Ripple
N=3;  % For band pass filter m=2N=6

[z,p,k1]=buttap(3);%Calculate pole-zero of analog lowpass filter
disp(p);
disp(k1);
% B=[ 1 2 2 1],Denominator coefficients for normalized analog lowpass filter
% A=[1 0 0 0],Numerator coefficients for normalized analog lowpass filter
%k=tan(pi*(Fpb1)/Fs)*tan(pi*(Fpb2)/Fs);
%disp(k);
k=2.23593;
c=.42532;
capA=[0 0 0 k^3;0 0 k^2 0;0 k 0 3*k^2;1 0 2*k 0;0 1 0 3*k;0 0 1 0;0 0 0 1]*[1 2*c 2*c^2
1*c^3]';
disp(capA);
%capA =[0.8600  1.8087   3.0559   2.6179   1.3667  0.3618  0.0769]
capB=[0 0 0 k^3;0 0 k^2 0;0 k 0 3*k^2;1 0 2*k 0;0 1 0 3*k;0 0 1 0;0 0 0 1]*[1 0 0 0]';
disp(capB);
%capB=[0  0 0 1 0 0 0]
%------------------------------------------------------------------------
m=6;%here m=2N
 for i=0:m
    f(i+1)=(-1)^i*factorial(m)/(factorial(i)*factorial(m-i));
 end
 f
 %-----------------------------------------------------------------------

%Calculating the Numerator and Denominator coefficients of digital bandpass filter
a1=[1 1 1 1 1 1 1;6 4 2 0 -2 -4 -6;15 5 -1 -3 -1 5 15;
   20 0 -4 0 4 0 -20;15 -5 -1 3 -1 -5 15;
   6 -4 2 0 -2 4 -6;1 -1 1 -1 1 -1 1]*[0.8600  1.8087   3.0559   2.6179   1.3667  0.3618  0.0769]';
disp(a1);
a=a1/10.1479;
%a1=[10.1479  13.8646 12.6297  8.9052  6.6321   2.2894   0.5711]
b1=[1 1 1 1 1 1 1;6 4 2 0 -2 -4 -6;15 5 -1 -3 -1 5 15;
   20 0 -4 0 4 0 -20;15 -5 -1 3 -1 -5 15;
   6 -4 2 0 -2 4 -6;1 -1 1 -1 1 -1 1]*[0 0 0 1 0 0 0]';
disp(b1);
%b1=[1  0   -3    0    3    0   -1]
b=b1/10.1479;
%fvtool(b,a);
```

```
[h,f]=freqz(b,a,512,1);

%Plotting Magnitude Response
plot(f,20*log10(abs(h))),grid
title('Magnitude Response of Butterworth BandPass Filter By Matrix');
ylabel('Magnitude(dB)');
xlabel('Normalized Frequency');

% Plotting the phase response

plot(f,angle(h));
title('Phase Resopnse of Butterworth BandPass Filter  By Matrix');
xlabel('Normalized Frequency');
ylabel('Phase(rad)');
grid on;

% plotting the Impulse Response
[y,t] = impz(b,a,60);
figure;
stem(t,y);grid
xlabel('Samples');
ylabel('Amplitude');

title('Impulse Response of Butterworth BandPass Filter  By Matrix');

% Plotting the Pole-Zero Plot
z = roots(b);%zeros
p = roots(a);%poles
figure;
zplane(z,p);grid
title('Pole-Zero plot of Butterworth BandPass Filter By Matrix');
```

COMPARISONS

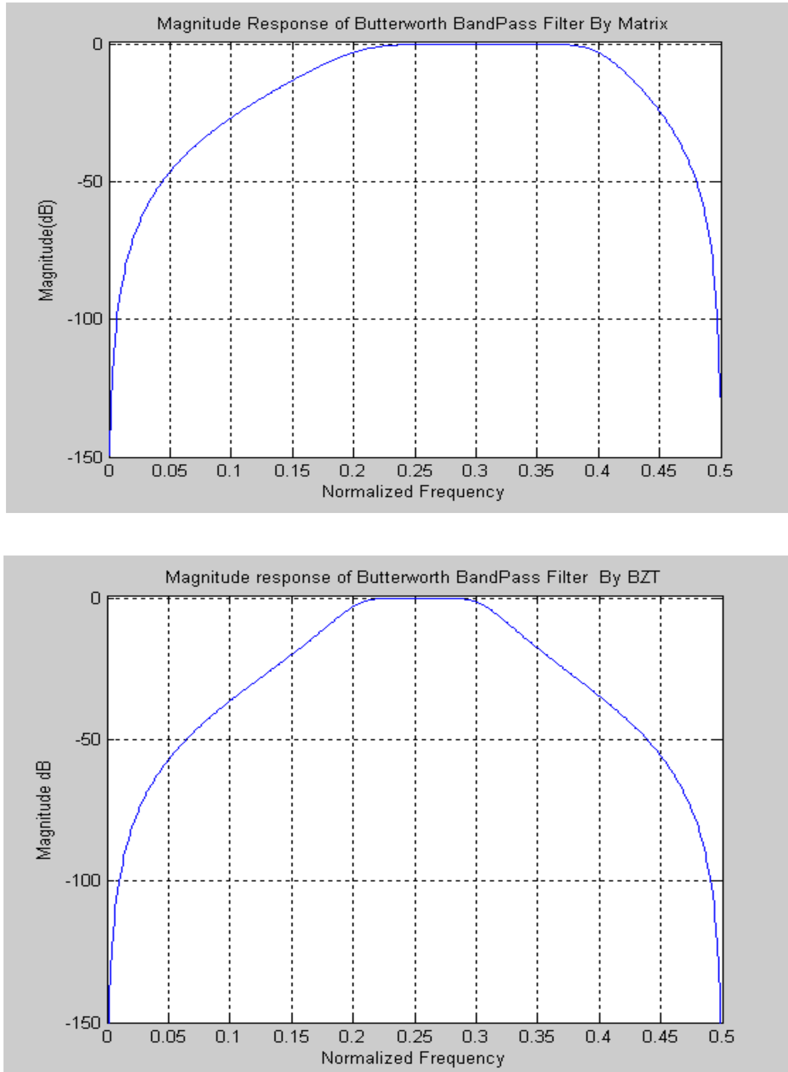


Fig.4.32. Magnitude response

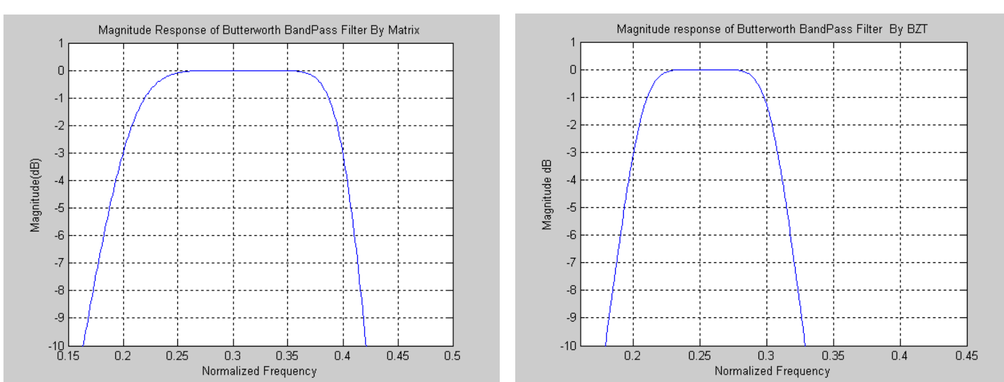


Fig.4.33. Zoom in of the passband

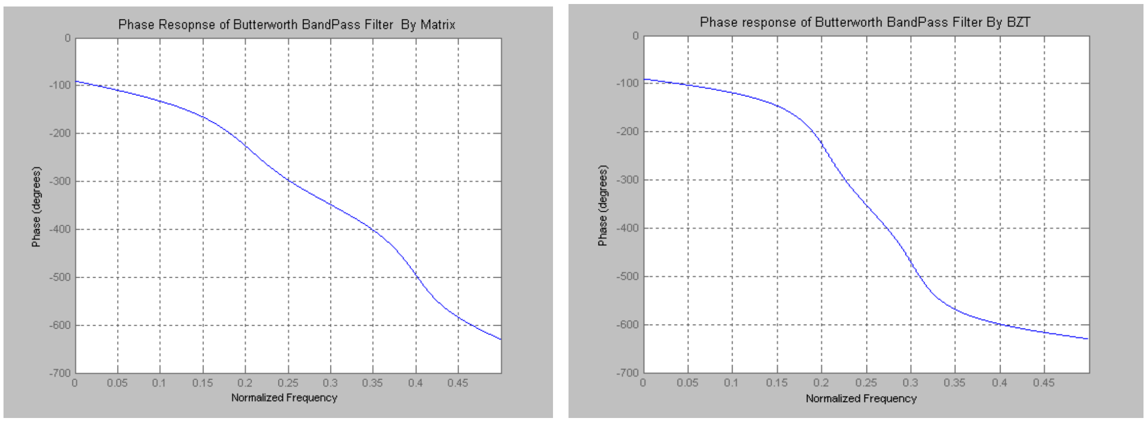


Fig.4.34. Phase response

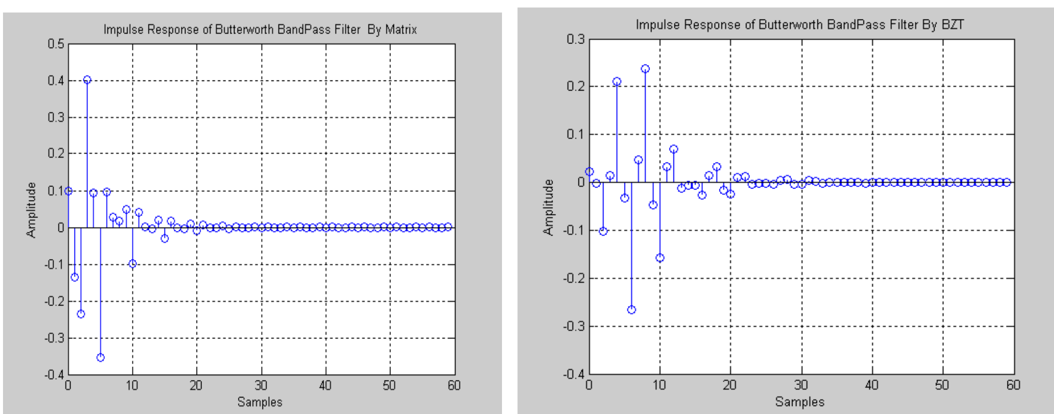


Fig.4.35 Impulse response

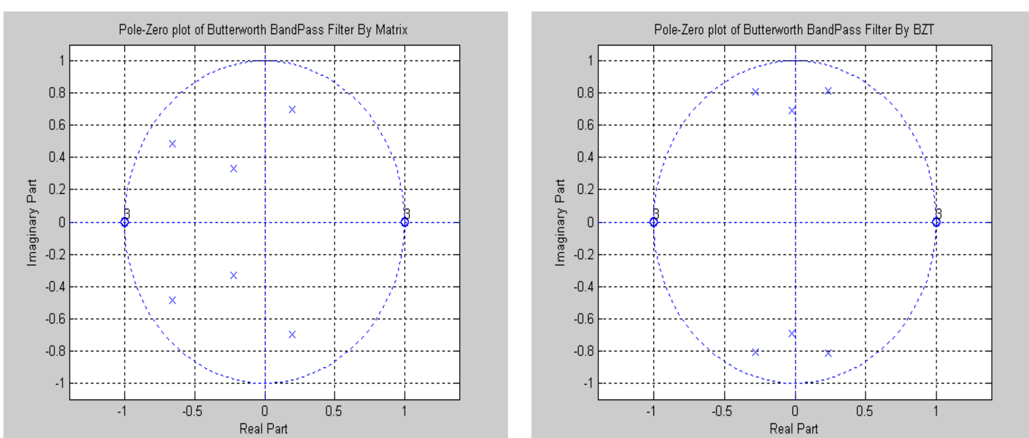


Fig.4.36. Pole – Zero plot

A13

Matlab Code for Butterworth IIR Bandpass filter by Bilinear Z -Transform

```
%Filter Specifications

Fs = 10;fs = Fs/2;            %Sampling Frequency
Fpb1 = 2;fpb1 = Fpb1/fs;      %Lower Passband edge frequency
Fpb2= 4;fpb2 = Fpb2/fs;       %Upper Passband edge frequency
Rp = 1;                       %Passband Ripple
N = 3;                        %Order of digital bandpass filter =6
wo = 2*pi*sqrt(Fpb1*Fpb2);    %Centre Frequency rad/s
bw = (Fpb2-Fpb1)*2*pi;        %Bandwidth rad/s

[z,p,k] = buttap(N);          %Calculate pole-zero of analog lowpass filter

b = k*poly(z);                % Numerator polynomial of analog lowpass filter
a = poly(p);                  % Denominator polynomial of analog lowpass filter
[bt,at] = lp2bp(b,a,wo,bw);   % Low Pass to Bandpass Transformation
[b2,a2] = bilinear(bt,at,Fs,Fpb1); % Bilinear Transformation matching lower pass frequency
%fvtool(b2,a2);
[H,w] = freqz(b2,a2,512,1);
figure;

% Plotting the magnitude Response
plot(w, 20*log10(abs(H)));
grid on;
title('Magnitude response of Butterworth BandPass Filter  By BZT');
xlabel('Normalized Frequency');
ylabel('Magnitude dB');

% Plotting the phase response

plot(w,angle(H));
title('Phase Response of Butterworth BandPass Filter By BZT');
xlabel('Normalized');
ylabel('Phase(rad)');
grid on;

% Plotting the Impulse Response
[y,t] = impz(b2,a2,60);
figure;
stem(t,y);grid
xlabel('Samples');
ylabel('Amplitude');

title('Impulse Response of Butterworth BandPass Filter By BZT');

% plotting  the Pole-Zero Plot
```

```
z = roots(b2);%zeros
p = roots(a2);%poles
figure;
zplane(z,p);grid
title('Pole-Zero plot of Butterworth BandPass Filter By BZT');
```

Second Filter:

Specification of second IIR Bandpass filter are as follow. Matlab code for this filter are given in A14 to A15

Filter Type	**Butterworth IIR BPF**
Passband edge frequency	200-300Hz
Lower stopband edge frequency	50Hz
Upper stopband edge frequency	450Hz
Sampling frequency(Fs)	1000Hz
Passband ripple(Rs)	3dB
Stopband attenuation(Rp)	20dB
Order(N)	2

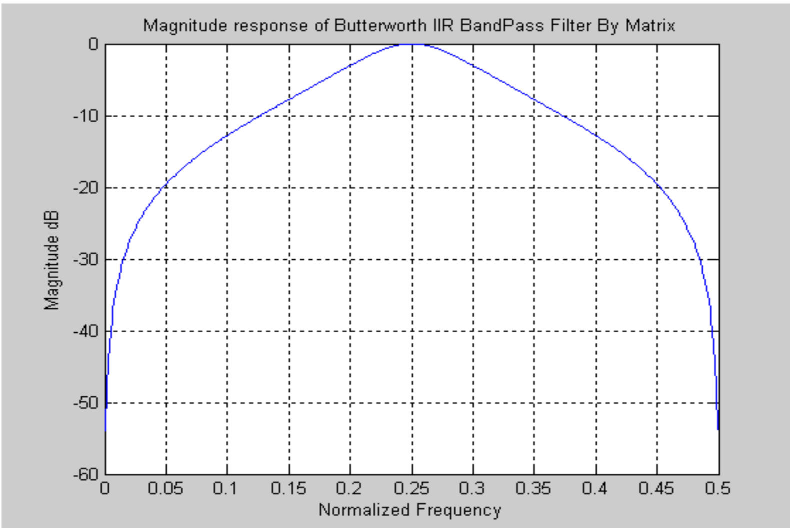


Fig.4.37. Magnitude response of Butterworth IIR Bandpass filter by Matrix

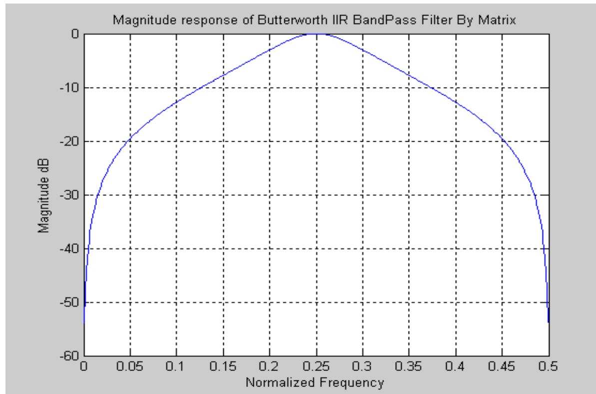
A14

Matlab code for Digital Butterworth IIR Bandpass Filter By using Matrix

```
%Filter Specifications
Fs = 1000;fs = Fs/2; %Sampling Frequency
Fpb1 = 200;fpb1 = Fpb1/fs; %Lower passband edge frequency
Fpb2= 300;fpb2 = Fpb2/fs; %Upper passband edge frequency
Fsb1=50;%Lower stopband edge frequency
Fsb2=450;%Upper stopband edge frequency
Rp = 3; %Passband Ripple in dB
Rs=20;%Stopband Attenuation in dB
N = 1;%For bandpass filter Order is 2
%Calculate edge frequencies for the bandpass analog filter by using
OmegaP1=tan(pi*(Fpb1)/Fs);
OmegaP2=tan(pi*(Fpb2)/Fs);
OmegaS1=tan(pi*(Fsb1)/Fs);
OmegaS2=tan(pi*(Fsb2)/Fs);

[z,p,k1]=buttap(N)
disp(p);
disp(z);
%We obtain analog low pass filter with Denominator coefficients A=[1 1] and
%numerator coefficients B=[1 0]
A=[1 1];
B=[1 0];
k=OmegaP1*OmegaP2;
c=1/(OmegaP2-OmegaP1);
disp(c);
capA=[0 1;1 0;0 1]*[1 1*c]';
disp(capA);
%capA=[1.5388  1.0000  1.5388]
capB=[0 1;1 0;0 1]*[1 0]';
disp(capB);
%capB=[0  1   0]
%------------------------------------------------------------------------
m=2;%here m=2N
 for i=0:m
    f(i+1)=(-1)^i*factorial(m)/(factorial(i)*factorial(m-i));
 end
 f
 %------------------------------------------------------------------------
%Calculate the Numerator and denominator coefficients of digital bandpass filter
a1=[1 1 1;2 0 -2;1 -1 1]*[1.5388  1.0000  1.5388]';
 disp(a1);
a=a1/4.0776;
disp(a);
b1=[1 1 1;2 0 -2;1 -1 1]* [0 1 0]';
% disp(b1);
b=b1/4.0776;
```

```
disp(b);
 %fvtool(b,a);
[h,f]=freqz(b,a,512,1);

% Plotting the magnitude Response
plot(f, 20*log10(abs(h)));grid on;
title('Magnitude response of Butterworth IIR BandPass Filter By Matrix');
xlabel('Normalized Frequency');
ylabel('Magnitude dB');

% Plotting the phase response
plot(f,angle(h));
title(' Phase response of Butterworth IIR BandPass Filter By Matrix');
xlabel('Normalized frequency');
ylabel('Phase(rad)');
grid on;

% Plotting the Impulse Response
[y,t] = impz(b,a,60);
figure;
stem(t,y);grid
xlabel('Amplitude');
ylabel('Samples');
title('Impulse response of Butterworth IIR BandPass Filter By Matrix');

% plotting the Pole-Zero Plot
z = roots(b);%zeros
p = roots(a);%poles
figure;
zplane(z,p);grid
title('Pole-Zero plot of Butterworth IIR BandPass Filter By Matrix');
```

COMPARISONS



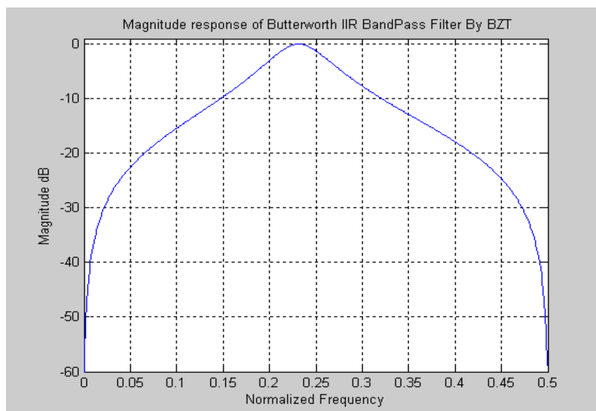


Fig.4.38. Magnitude response

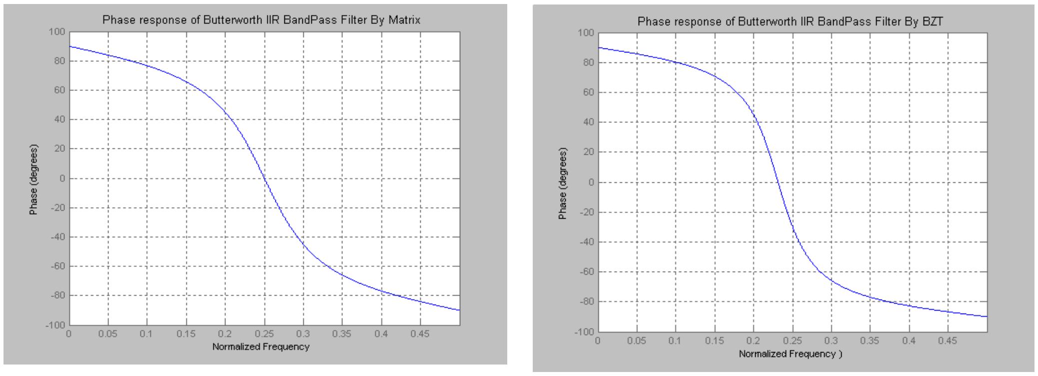


Fig.4.39. Phase response

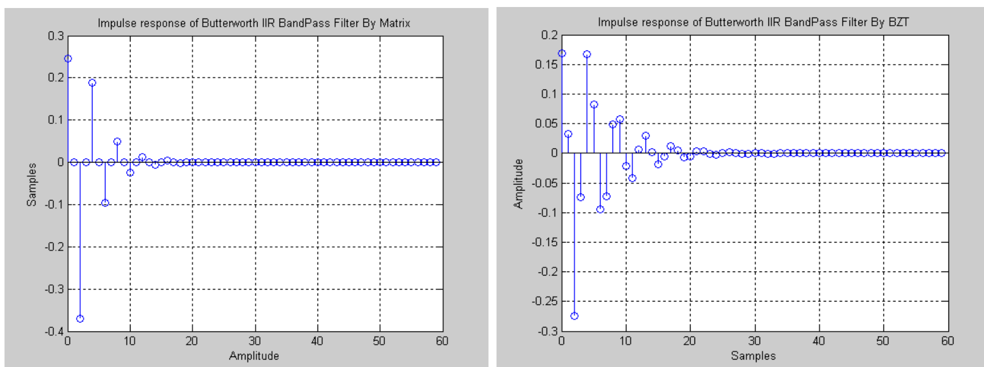


Fig.4.3.1. Impulse response

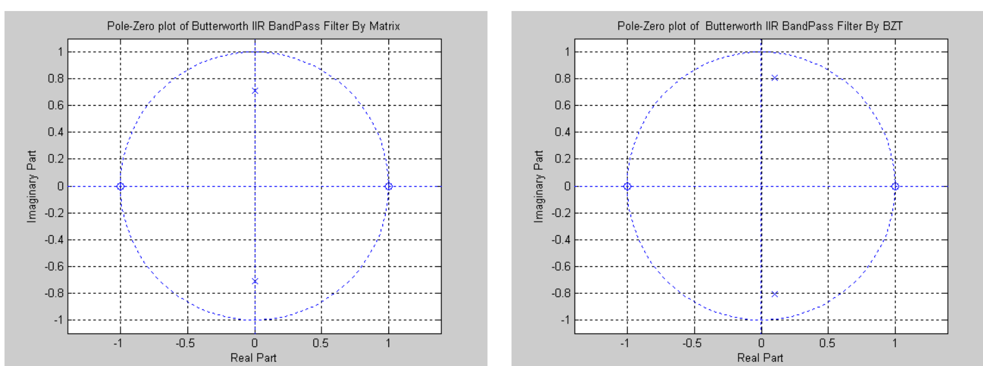


Fig.4.3.2. Pole-Zero plot

A 15

Matlab code for Digital IIR Bandpass Filter by Bilinear Z-transform

```
%Filter Specifications

Fs = 1000;fs = Fs/2; %Sampling Frequency
Fpb1 = 200;fpb1 = Fpb1/fs; %Lower passband edge frequency
Fpb2= 300;fpb2 = Fpb2/fs; %Upper assband edge frequency
Rp = 3; %Passband Ripple
N = 1;%For bandpass filter Order is 2
wo = 2*pi*sqrt(Fpb1*Fpb2);%Centre Frequency rad/s
bw = (Fpb2-Fpb1)*2*pi; %Bandwidth rad/s

[z,p,k] = buttap(N);%Calculate pole-zero for the analog lowpass filter
b = k*poly(z); % Numerator coefficients for analog lowpass filter
a = poly(p); % Denominator coefficients for analog lowpass filter
[bt,at] = lp2bp(b,a,wo,bw); % Low Pass to Bandpass Transformation
[b,a] = bilinear(bt,at,Fs,Fpb1); % Bilinear Transformation
%fvtool(b,a);
% [h,f] = freqz(b,a,512,1000);
[h,f] = freqz(b,a,512,1);
figure;
% Plotting the magnitude Response
plot(f, 20*log10(abs(h)));grid on;
title(' Magnitude response of Butterworth IIR BandPass Filter By BZT');
xlabel('Normalized Frequency');
ylabel('Magnitude dB');

% Plotting the phase response
 plot(f,angle(h));
title('Phase response of Butterworth IIR BandPass Filter By BZT');
xlabel('Normalized Frequency');
ylabel('Phase(rad)');
grid on;

% Plotting the Impulse Response
[y,t] = impz(b,a,60);
figure;
stem(t,y);grid
ylabel('Amplitude');
xlabel('Samples');
title('Impulse response of Butterworth IIR BandPass Filter By BZT');

% Plotting the Pole-Zero Plot
z = roots(b);%zeros
p = roots(a);%poles
figure;
zplane(z,p);grid
title('Pole-Zero plot of  Butterworth IIR BandPass Filter By BZT');
```

4.4. Design of Bandstop Filter

In this section we will design digital IIR Bandstop filters by using both bilinear transformation and matrix operation. Then we will compare the magnitude response ,phase response , impulse response and location of poles and zeros by using both techniques. Matlab code for these designs are given in A16 to A17.

Filter Type	**Chebyshev Type 1 Bandstop Filter**
Lower Passband edge frequency	.3Hz
Upper Passband edge frequency	.4Hz
Sampling frequency(Fs)	1Hz
Passband Ripple(Rp)	2dB
Order(N)	4

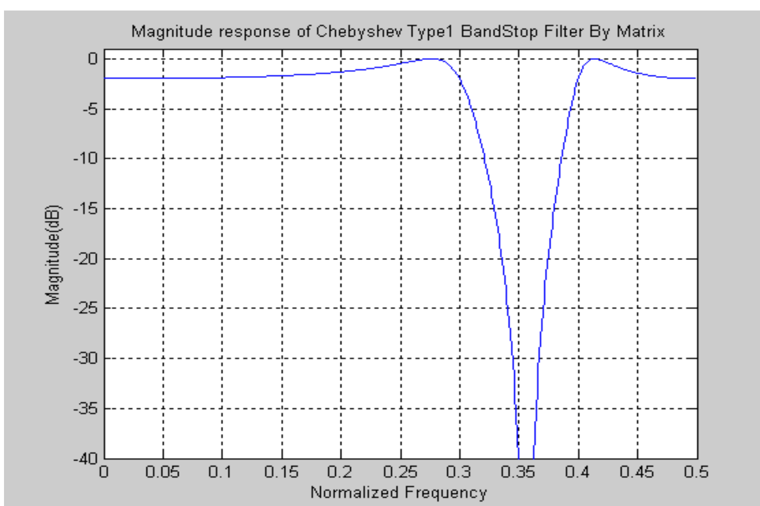


Fig.4.41. Magnitude response of Digital IIR Chebyshev Type1 Bandstop filter by Matrix

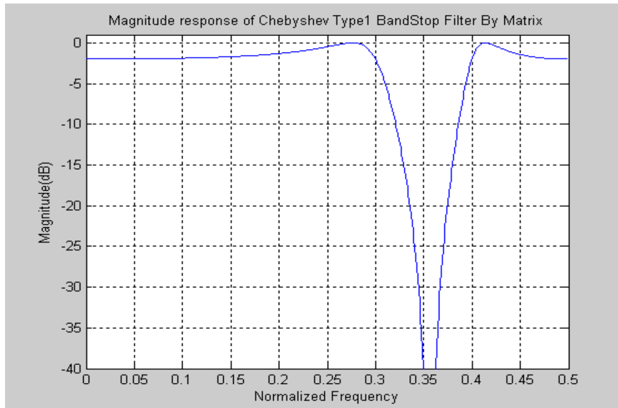
A16

Matlab Code for Digital Chebyshev Type 1 BandStop Filter by Matrix

```
%Filter specifications
Fs=1;%Sampling frequency
Fpb1=0.3;fpb1=Fpb1/Fs;%Lower passband edge frequency
Fpb2=0.4;fpb2=Fpb2/Fs;%Upper passband edge frequency
Rp=2;%Passband ripple
N=2;%Order of analog lowpass filter
[z,p,k1]=cheb1ap(N,Rp);%For Bandstop Filter oredr is 4

disp(k1);
disp(z);
disp(p);
%k1= 0.6538 p1=  -0.4019 + 0.8133i  p2=  -0.4019 - 0.8133i
OmegaP1=tan(pi*(Fpb1)/Fs);
OmegaP2=tan(pi*(Fpb2)/Fs);
c=(OmegaP2-OmegaP1);%here c=c1=1.7013
disp(c);
k=OmegaP1*OmegaP2;%here k=k1
disp(k);%k=  4.2361
%k1= 0.6538 p1=  -0.4019 + 0.8133i  p2=  -0.4019 - 0.8133i
%B=[.654 0 0] ,  A=[.823 .804  1]
capA=[k^2 0 0 ;0 k 0 ;2*k 0 1 ;0 1 0 ;1 0 0]*[.823 .804*c 1*c^2]';
disp(capA);
%capA=[14.7681 5.7943 9.8670 1.3678 0.8230]

capB= [k^2 0 0 ;0 k 0 ;2*k 0 1 ;0 1 0 ;1 0 0]*[.654 0 0]';
disp(capB);
%capB=[11.7355 0 5.5408 0 0.6540]
%-------------------------------------------------------------------------
m=4;%here m=2N
 for i=0:m
    f(i+1)=(-1)^i*factorial(m)/(factorial(i)*factorial(m-i));
 end
 f
 %-------------------------------------------------------------------------

%calculate the Numerator and Denominator coefficients for digital bandstop filter
a1=[1 1 1 1 1;4 2 0 -2 -4;6 0 -2 0 6;4 -2 0 2 -4;1 -1 1 -1 1]*[14.7681 5.7943 9.8670 1.3678
0.8230]';
a=a1/32.6202;
% coeff  of denominator   1.0000  1.9814  2.2628   1.4386   0.5609
disp(a);
b1=[1 1 1 1 1;4 2 0 -2 -4;6 0 -2 0 6;4 -2 0 2 -4;1 -1 1 -1 1]*[11.7355 0 5.5408 0 0.6540]';
b=b1/32.6202;
disp(b);
%fvtool(b,a);
[h,f]=freqz(b,a,512,1);
```

```
%Plotting Magnitude Response
plot(f,20*log10(abs(h))),grid
title('Magnitude response of Chebyshev Type1 BandStop Filter By Matrix');
ylabel('Magnitude(dB)');
xlabel('Normalized Frequency');

% Plotting the phase response

plot(f,angle(h));
title('Phase response of Chebyshev Type1 BandStop Filter By Matrix');
xlabel('Normalized Frequency');
ylabel('Phase(rad)');
grid on;

% Plotting the Impulse Response
[y,t] = impz(b,a,60);
figure;
stem(t,y);grid
xlabel('Samples');
ylabel('Amplitude');
title('Impulse response of Chebyshev Type1 BandStop Filter By Matrix');

% Plotting the Pole-Zero Plot
z = roots(b);%zeros
p = roots(a);%poles
figure;
zplane(z,p);grid
title('Pole-Zero plot of Chebyshev Type1 BandStop Filter By Matrix');
```

COMPARISONS



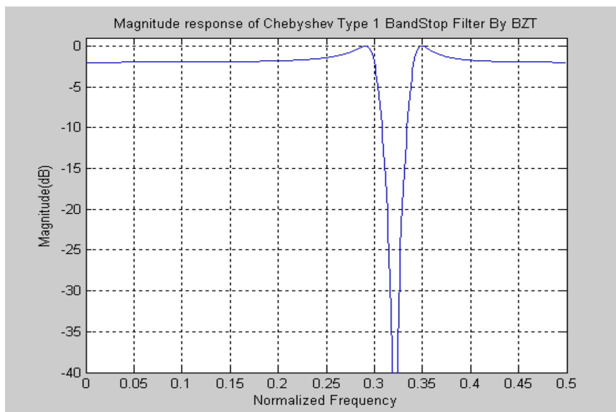
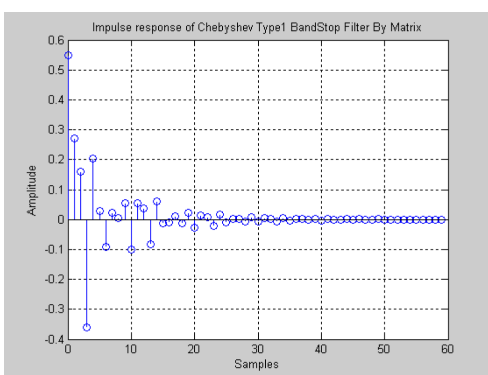


Fig.4.42. Magnitude response



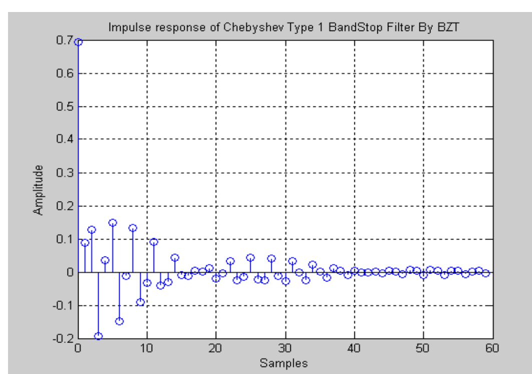


Fig.4.43. Impulse response

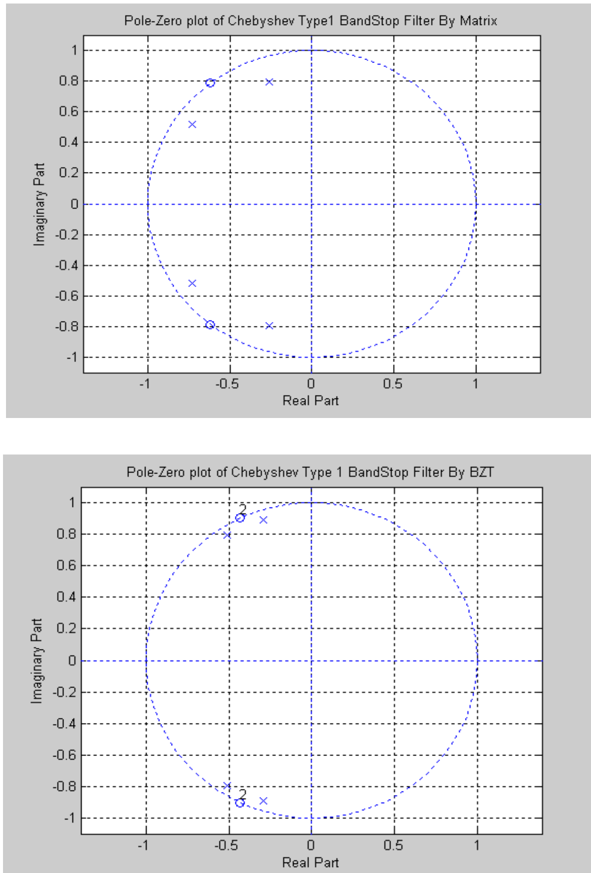


Fig.4.44. Pole- Zero response

A 17

Matlab Code for Digital Chebyshev Type1 Bandstop IIR filter by BZT

```
%Filter specifications
Fs = 1;fs = Fs/2; %Sampling Frequency
Fpb1 = .3;fpb1 = Fpb1/fs; %Lower Passband edge frequency
Fpb2= .4;fpb2 = Fpb2/fs; %Upper Passband edge frequency
Rp = 2; %Passband Ripple
wo = 2*pi*sqrt(Fpb1*Fpb2);%Centre Frequency rad/s
bw = (Fpb2-Fpb1)*2*pi; %Bandwidth rad/s
N = 2;%For Bandstop Filter order is 4
[z,p,k] = cheb1ap(N,Rp);
b = k*poly(z); % Numerator polynomial of normalized analog lowpass filter
a = poly(p); % Denominator polynomial of normalized analog lowpass filter
[bt,at] = lp2bs(b,a,wo,bw); % Low Pass to Bandstop Transformation
[b2,a2] = bilinear(bt,at,Fs,Fpb1); % Bilinear Transformation
% fvtool(b2,a2);
[h,f] = freqz(b2,a2,512,1);
figure;

% Plotting the magnitude Response
plot(f, 20*log10(abs(h)));
grid on;
title('Magnitude response of Chebyshev Type 1 BandStop Filter By BZT');
xlabel('Normalized Frequency');
ylabel('Magnitude(dB)');

% Plotting the phase response

plot(f,angle(h));
title('Phase response of Chebyshev Type 1 BandStop Filter By BZT');
xlabel('Normalized Frequency');
ylabel('Phase(rad)');
grid on;

% Plotting the Impulse Response
[y,t] = impz(b2,a2,60);
figure;
stem(t,y);grid
xlabel('Samples');
ylabel('Amplitude');
title('Impulse response of Chebyshev Type 1 BandStop Filter By BZT');

% Pole-Zero Plot
z = roots(b2);%zeros
p = roots(a2);%poles
figure;
zplane(z,p);grid
title('Pole-Zero plot of Chebyshev Type 1 BandStop Filter By BZT');
```

Conclusion

In this project, the design of IIR filters using analog frequency transformation by matrix operation was presented. The method is simple and can be implemented by means of any mathematical software .Combination of the matrices described here with the pascal matrix allows the design of the digital IIR filters from the continuous – time prototypes. Several results were verified in the design. It was mentioned by *Jacek Konopacki,"The Frequency transformation by Matrix Operation and Its Application in IIR Filters Design",IEEE Signal Processing Lett.vol.12,no.1,pp.5– 8,Jan.2005*, that ,this approach .ie the frequency transformation by using matrix operation is not highly suitable for the design of lowpass and highpass digital IIR filter .I have designed all four types of filters by using this approach ,I have given more attention to the lowpass and highpass filters. After designing several lowpass and highpass filters both at low and high frequency and comparing them with BZT and Impulse Invariant filters several significant observations were made:

(1)During the design of lowpass filters I found satisfactory results. when we increase or decrease the passband ,stopband and sampling frequency the magnitude response are not showing much attenuation as compared to the BZT.

(2) By this approach I got the attenuation at Nyquist frequency ,which is similar to the attenuation got by designing digital lowpass filter by the Bilinear Z-Transform method.

(3)Results of the phase, Impulse, and pole zero plot are also similar and sometimes better than the BZT method.. I also got satisfactory results when I designed highpass ,bandpass and bandstop digital filters by using frequency transformation by matrix operation approach.

Finally I came to the conclusion that using frequency transformation by matrix operation we can effectively design lowpass and highpass along with the bandpass and bandstop filters.

References

[1] Sanjit K. Mitra,2001,"Digital Signal Processing A Computer –Based Approach",3rd Edition ,Tata McGraw Hill.

[2]John G.Proakis and Dimitris G.Manolakis,1996,"Digital Signal Processing ,principles, algorithms, and applications,"3rd Edition, Pearson Education.

[3]Stephen J.Chapman ,2004,"Matlab Programming for Engineers",3rd Edition ,Thomson Learning.

[4]Andreas Antoniou,1993,"Digital Filters, analysis, design and applications",2nd Edition,Tata McGraw-Hill.

[5] Oppenheim A.V. and Schafer R.W., *Digital Signal Processing*, Prentice-Hall, 1975

[6]B.Psenicka, F.Garcia-Ugalde,and A.Herrera-Camacho,"The bilinear Z transform by Pascal matrix and its application in the design of digital filters," IEEE Signal Processing Lett.,vol.9,no. 11,pp.368-370,Nov.2002

[7]Jacek Konopacki,"The Frequency transformation by Matrix Operation and Its Application in IIR Filters Design",IEEE Signal Processing Lett.vol.12,no.1,pp.5–8,Jan.2005

[8]Anthony J.O'Conner," Introduction to Matlab"

[9] http://www.mathworks.com/matlabcentral/?s_cid=matlabcentral

[10] www.bores.com/courses/intro/iir*: Introduction to DSP - IIR filters: Digital filter equation*