

Analysis of Digital IIR Filter with LabVIEW

Yaduvir Singh
Associate professor
Thapar University, Patiala
Punjab, India

Sweta Tripathi
Assistant professor
Mewar University, Chittorgarh
Rajasthan, India

Manoj Pandey
Assistant professor
St. Margaret Engineering College
Neemrana, Rajasthan, India

ABSTRACT

Aiming at importance of virtual instruments in the field of Digital Signal Processing, a digital IIR Filter system is developed using National Instruments (NI) data LabVIEW software package. All the types of IIR filters like Butterworth filters, Chebyshev filters, inverse Chebyshev filters, and Elliptic filters are designed to generate their magnitude response and filter coefficients. The LabVIEW software is used to develop virtual instrument (VI) that includes a front panel and a functional diagram. The VI reads the desired parameters of the filters entered by the user on the front panel and determines its magnitude response and filter coefficients.

Keywords

Digital IIR filter, LabVIEW, Virtual Instruments.

1. 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) [7]. The LabVIEW based digital filter system involves the concept of Virtual Instrumentation. A virtual instrumentation system is computer software that a user would employ to develop a computerized test and measurement system, for controlling from a computer desktop an external measurement hardware device, and for displaying test or measurement data collected by the external device on instrument-like panels on a computer screen.

Two types of digital filters exist – the IIR (Infinite Impulse Response) and the FIR (Finite Impulse Response). IIR filter possess certain properties, which make them the preferred design choices in numerous situations over FIR 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 [9].

The design of IIR filters proceeds through a vastly different set of steps than those followed by FIR filter design algorithms. 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 exist – the impulse invariant transformation and the bilinear transformation. In this paper, the focus is on designing minimum order IIR filters to meet a set of specifications using LabVIEW functions. Each design is accompanied by a plot of its frequency response, impulse response and pole-zero diagrams.

The responses of IIR filters using LabVIEW are compared with the responses from MATLAB with the same specifications. The main goal of this paper is to obtain an optimized filter response along with the filter coefficients.

2. DIGITAL IIR FILTERS

In signal processing, the function of a filter is to remove unwanted parts of the signal, such as random noise, or to extract useful parts of the signal, such as the components lying within a certain frequency range. There are two main kinds of filter, analog and digital.

There are some considerable advantages of digital over analog filters which make digital filters unavoidable [8]. Some of these are as follows

1. A digital filter is programmable, i.e. its operation is determined by a program stored in the processor's memory. This means the digital filter can easily be changed without affecting the circuitry (hardware).
2. Digital filters are easily designed, tested and implemented on a general-purpose computer or workstation.
3. Unlike their analog counterparts, digital filters can handle low frequency signals accurately. As the speed of DSP technology continues to increase, digital filters are being applied to high frequency signals in the RF (radio frequency) domain, which in the past was the exclusive preserve of analog technology.

Further digital filters can be classified as FIR (Finite impulse response) filters and IIR filters. Finite impulse response (FIR) filter, also known as non-recursive filters (in a non-recursive filter the current output is calculated solely from the current and previous input values). Infinite impulse response (IIR) filter, also known as recursive filter (a recursive filter is one which in addition to input values also uses previous output values). IIR filters have the advantages of providing the higher selectivity for a particular order. IIR filters can achieve the same level of attenuation as FIR filters but with far fewer coefficients. Therefore, an IIR filter can provide a significantly faster and more efficient filtering operation than an FIR filter. FIR filters provide a linear-phase response [1]. IIR filters provide a nonlinear-phase response. FIR filters are used for applications that require linear-phase responses like high quality audio systems. IIR filters are used for applications that do not require phase information, such as signal monitoring applications.

Infinite impulse response (IIR) filters operates on current and past input values and current and past output values. Theoretically, the impulse response of an IIR filter never reaches zero and is an infinite response. A recursive filter is one which in addition to

input values also uses previous output values [5]. The expression for a recursive filter therefore contains not only terms involving the input values $(x_n, x_{n-1}, x_{n-2}, \dots)$ but also terms involving the past output values y_{n-1}, y_{n-2}, \dots .

The following general difference equation characterizes IIR filters

$$y_i = \frac{1}{a_0} \left(\sum_{j=0}^{N_b-1} b_j x_{i-j} - \sum_{k=1}^{N_a-1} a_k y_{i-k} \right) \quad (1)$$

where b_j the set of forward coefficients is, N_b is the number of forward coefficients, a_k is the set of reverse coefficients, and N_a is the number of reverse coefficients. Where x_i is the current input, x_{i-j} is the past inputs, and y_{i-k} is the past outputs.

From this explanation, recursive filters require more calculations to be performed, since there are previous output terms in the filter expression as well as input terms. In fact, the reverse is usually the case: to achieve a given frequency response characteristic using a recursive filter generally requires a much lower order filter (and therefore fewer terms to be evaluated by the processor) than the equivalent nonrecursive filter. IIR filters might have ripple in the passband, the stopband, or both. IIR filters have a nonlinear-phase response [9].

Equation 2 defines the direct-form transfer function of an IIR filter.

$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_{N_b-1} z^{-(N_b-1)}}{1 + a_1 z^{-1} + \dots + a_{N_a-1} z^{-(N_a-1)}} \quad (2)$$

A filter implemented by directly using the structure defined by Equation 3.12. Where a_n and b_n are the reverse and forward coefficients of the IIR filter.

Digital IIR filter designs come from the classical analog designs and include the following filter types:

- Butterworth filters
- Chebyshev filters
- Chebyshev II filters, also known as inverse Chebyshev and Type II Chebyshev filters
- Elliptic filters, also known as Cauer filters

The IIR filter designs differ in the sharpness of the transition between the passband and the stopband and where they exhibit their various characteristics—in the passband or the stopband [9].

2.1 Butterworth Filters

Butterworth filters have the following characteristics:

- Smooth response at all frequencies
- Monotonic decrease from the specified cut-off frequencies
- Maximal flatness, with the ideal response of unity in the passband and zero in the stopband
- Half-power frequency, or 3 dB down frequency, that corresponds to the specified cut-off frequencies.

The transfer function for Butterworth filter is given by

$$B(\omega) = \frac{1}{\left[1 + \left(\frac{\omega}{\omega_0} \right)^{2n} \right]^{1/2}} \quad (3)$$

Where n is the order of filter [6].

Butterworth filters do not always provide a good approximation of the ideal filter response because of the slow rolloff between the passband and the stopband.

2.2 Chebyshev Filters

Chebyshev filters have the following characteristics:

- Minimization of peak error in the passband
- Equiripple magnitude response in the passband
- Monotonically decreasing magnitude response in the stopband
- Sharper rolloff than Butterworth filters

Compared to a Butterworth filter, a Chebyshev filter can achieve a sharper transition between the passband and the stopband with a lower order filter. The sharp transition between the passband and the stopband of a Chebyshev filter produces smaller absolute errors and faster execution speeds than a Butterworth filter. The frequency response of the filter is given by

$$|H(\Omega)|^2 = \left(1 + \varepsilon^2 T_N^2 \left(\frac{\Omega}{\Omega_p} \right) \right)^{-1} \quad (4)$$

Where ε a parameter of the filter is related to ripple present in the passband and $T_N(x)$ is the Nth- order Chebyshev polynomial defined as

$$\begin{aligned} T_N &= \cos(N \cos^{-1} x) & |x| \leq 1 \\ &= \cosh(N \cosh^{-1} x) & |x| \geq 1 \end{aligned} \quad (5)$$

2.3 Chebyshev II Filters or Inverse Chebyshev Filters

Chebyshev II filters have the following characteristics:

- Minimization of peak error in the stopband

- Equiripple magnitude response in the stopband
- Monotonically decreasing magnitude response in the passband
- Sharper rolloff than Butterworth filters

Chebyshev II filters are similar to Chebyshev filters. However, Chebyshev II filters differ from Chebyshev filters in the following ways:

- Chebyshev II filters minimize peak error in the stopband instead of the passband. Minimizing peak error in the stopband instead of the passband is an advantage of Chebyshev II filters over Chebyshev filters [9].
- Chebyshev II filters have an equiripple magnitude response in the stopband instead of the passband.
- Chebyshev II filters have a monotonically decreasing magnitude response in the passband instead of the stopband.

2.4 Elliptic Filters

Elliptic filters have the following characteristics:

- Minimization of peak error in the passband and the stopband
- Equiripples in the passband and the stopband

Compared with the same order Butterworth or Chebyshev filters, the elliptic filters provide the sharpest transition between the passband and the stopband, which accounts for their widespread use [9].

The transfer function is given by

$$|H(\Omega)|^2 = \left(1 + \varepsilon^2 U_N \left(\frac{\Omega}{\Omega_c} \right) \right)^{-1} \quad (6)$$

Where $U_N(x)$ the Jacobian is elliptic function of order N and ε is a constant related to passband ripple. They provide a realization with the lowest order for a particular set of conditions.

Selection of a Digital filters for any particular application among all can be done by considering following points:

- Does the analysis require a linear-phase response?
- Can the analysis tolerate ripples?
- Does the analysis require a narrow transition band?

3. DESIGN OF DIGITAL IIR FILTER WITH LABVIEW

LabVIEW empowers to build own solutions for scientific and engineering systems. It gives the flexibility and performance of a powerful programming language without the associated difficulty and complexity. It gives thousands of successful users a faster way to program instrumentation, data acquisition, and digital signal processing [2]. By using LabVIEW to prototype, design, test, and implement your instrument systems, system development time can be reduced and productivity increases by a factor of 4 to 10 [10].

We design IIR filters by approximating the desired magnitude response of a discrete-time system.

3.1 Designing Butterworth Filter

Figure 1 shows the block diagram of a VI that returns the magnitude response of a butterworth IIR filter.

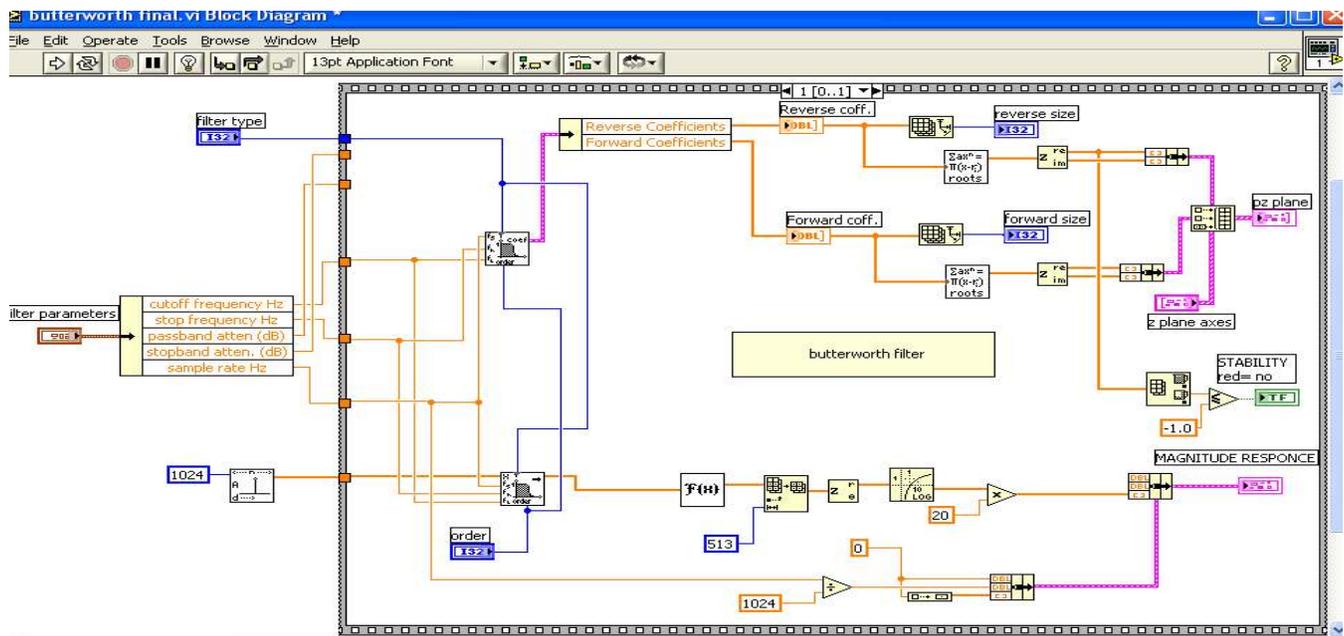


Figure 1 Butterworth.vi

The VI in Figure 1 completes the following steps to compute the magnitude response, filter coefficients and pole-zero plots to find the stability of the filter.

1. Pass all the parameters (cutoff frequency, stop frequency, passband attenuation, stopband attenuation, sample rate) with filter type—lowpass, highpass,

- bandpass, or bandstop to the Case Structure to calculate the order of filter [3].
2. Apply these parameters to the Butterworth Coefficient Function to generate the coefficients [3].
 3. Divides these coefficient arrays into two separate parts named as forward and reverse coefficients.
 4. Display these forward and reverse coefficients with coefficient size.
 5. Pass these coefficients to Complex Polynomial Roots VI and apply its output to Complex to Re/Im Function to display the pole-zero plots.
 6. Compare the output of Complex to Re/Im Function (only for reverse coefficients) with -1 to find out the stability. This is shown with the help of LED. If LED is red then the system is unstable otherwise stable.
 7. Pass an impulse signal through the butterworth filter function.
 8. Apply all the parameters and filter type—lowpass, highpass, bandpass, or bandstop to the butterworth filter function [3].
 9. The signal passed out from this function is the impulse response of the filter.
 10. Pass the filtered signal to the FFT VI. Use the FFT VI to perform a Fourier transform on the impulse response and to compute the frequency response of the filter, such that the impulse response and the frequency response comprise the Fourier transform pair

$h(t) \Leftrightarrow H(f)$. $h(t)$ is the impulse response. $H(f)$ is the frequency response [3].

11. Use the Array Subset function to reduce the data returned by the FFT VI. Half of the real FFT result is redundant so the VI needs to process only half of the data returned by the FFT VI [4].
12. Use the Complex To Polar function to obtain the magnitude-and-phase form of the data returned by the FFT VI. The magnitude-and-phase form of the complex output from the FFT VI is easier to interpret than the rectangular component of the FFT [4].
13. Convert the magnitude to decibels.
14. Display the magnitude response with the help of waveform graph.

3.2 Designing Chebyshev Filter, Inverse Chebyshev Filter and Elliptic Filter

Because the same mathematical theory applies to design other type of IIR filters, the block diagram in Figure 2, 3, and 4 of VI return the magnitude response of chebyshev, inverse chebyshev and elliptic IIR filter respectively [4].

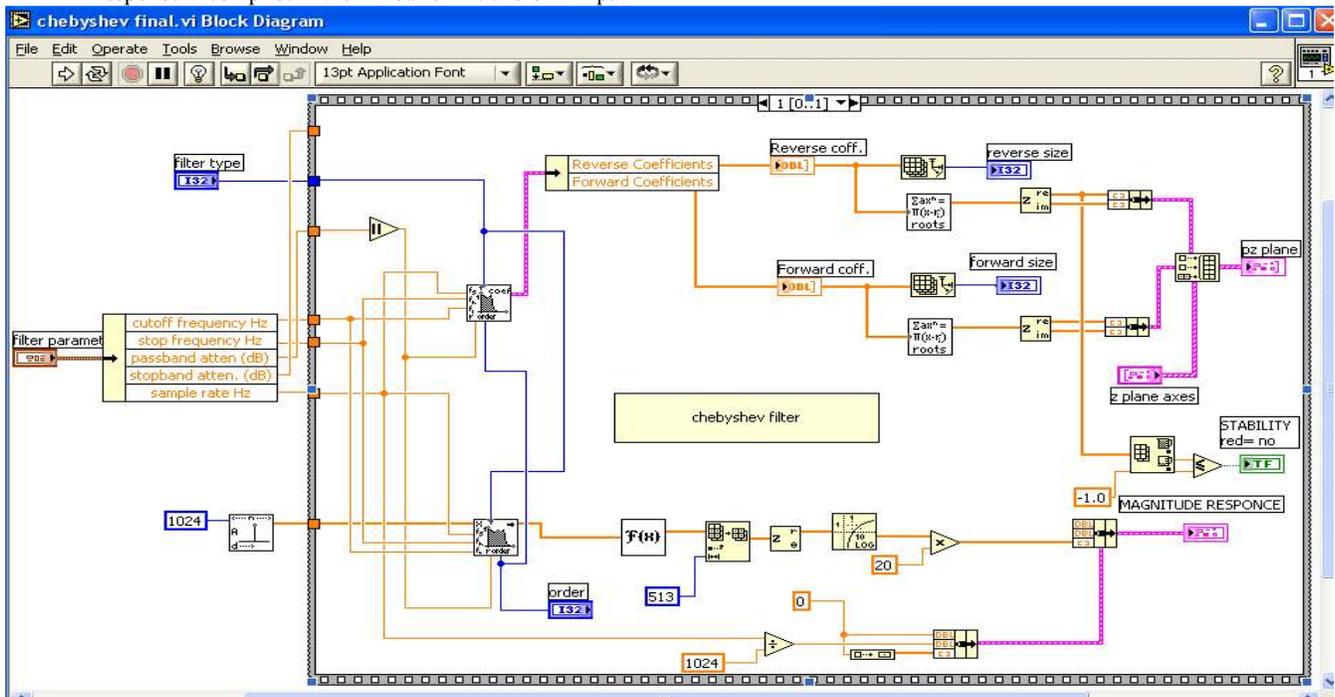


Figure 2 Chebyshev.vi

The design procedure is same for all type of filter as described for butterworth filter. The main difference between these VIs is that the chebyshev, inverse chebyshev and elliptic coefficient functions are used in step 2 in place of butterworth coefficient function and the chebyshev, inverse chebyshev and elliptic filter

function in place of butterworth filter function in step 7 and step 8 [11] .

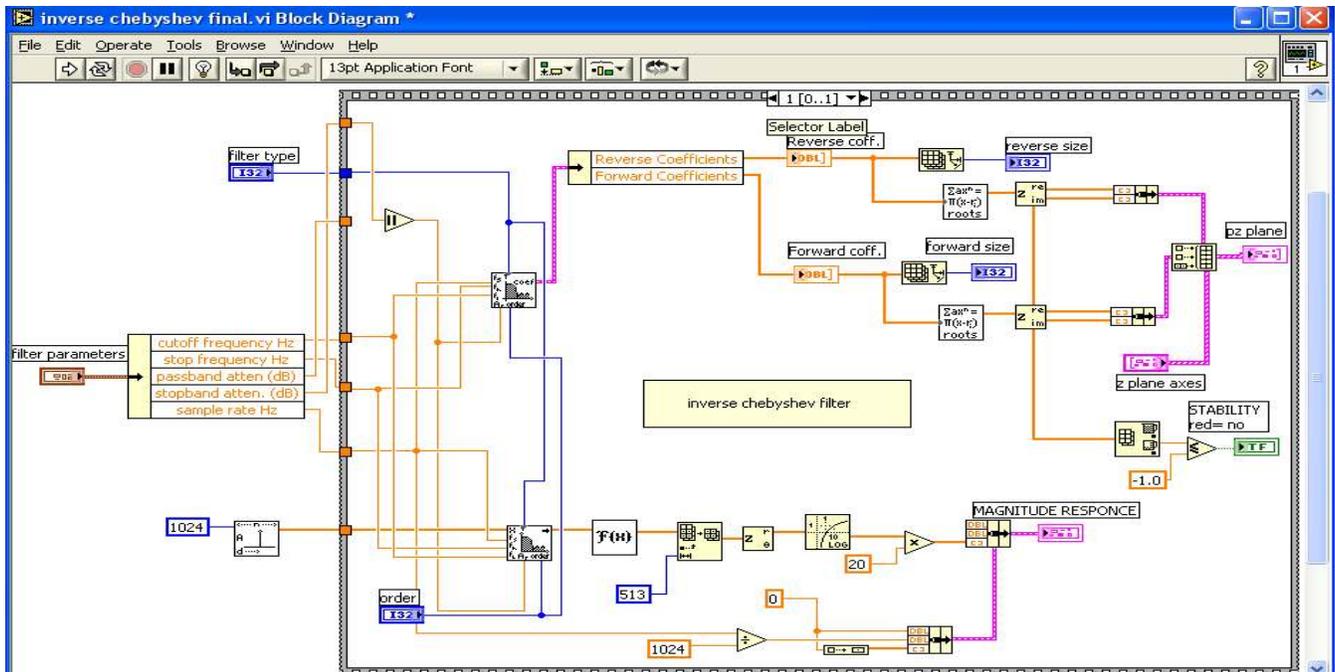


Figure 3 Inverse Chebyshev.vi

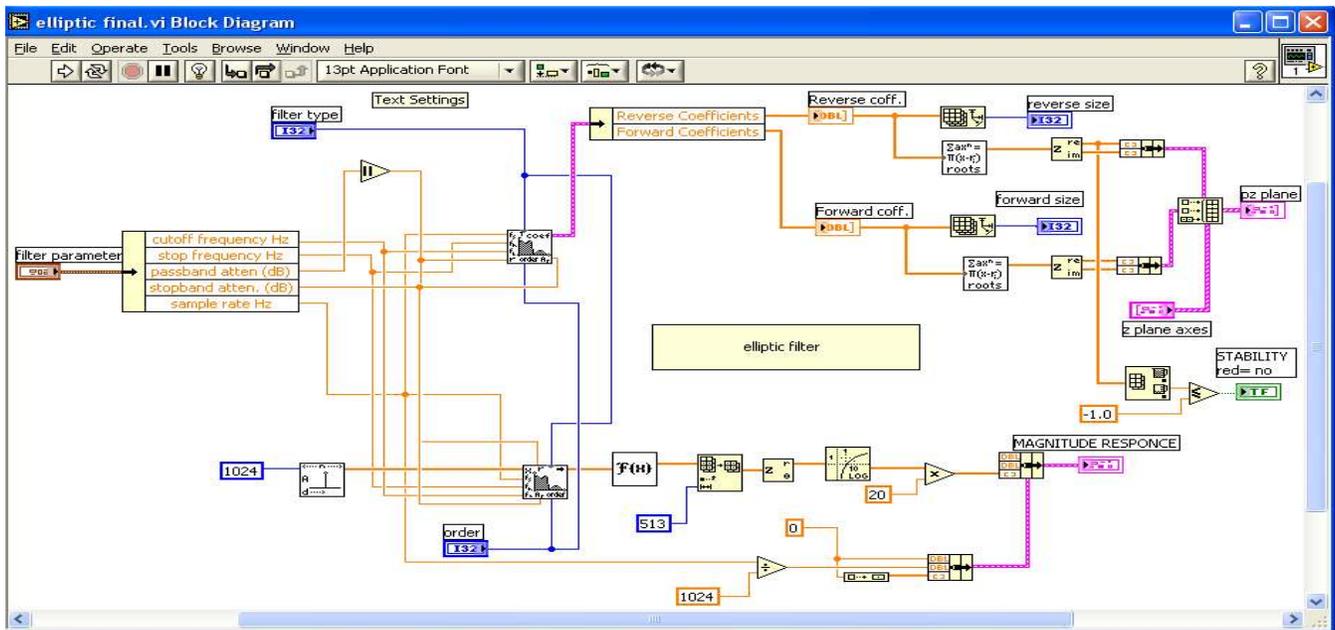


Figure 4 Elliptic.vi

4. RESULTS AND DISCUSSIONS

Figure 5 shows the response of 5th order butterworth lowpass filter. The specifications for this filter are given as:

cutoff frequency – 500 Hz

stop frequency– 1000Hz

passband attenuation- 15 dB

stopband attenuation- 100 dB

sample rate- 10000 Hz

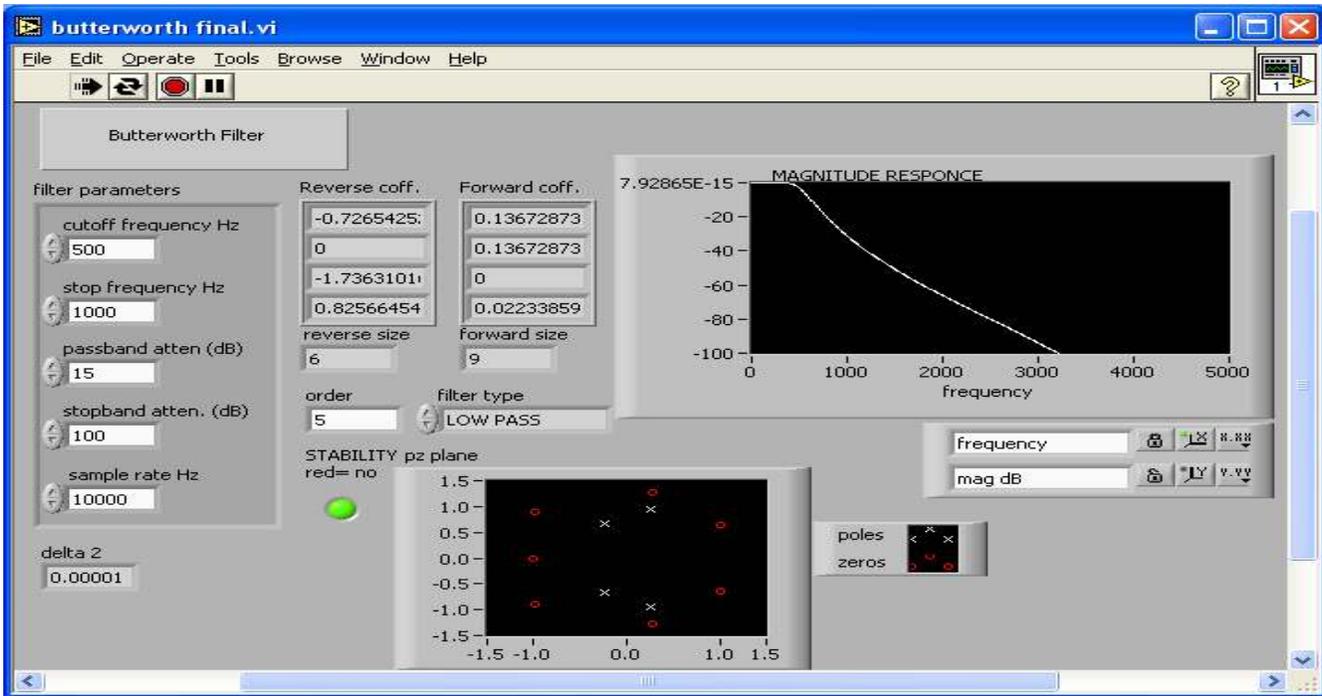


Figure 5 Response of Butterworth Lowpass Filter for 5th order

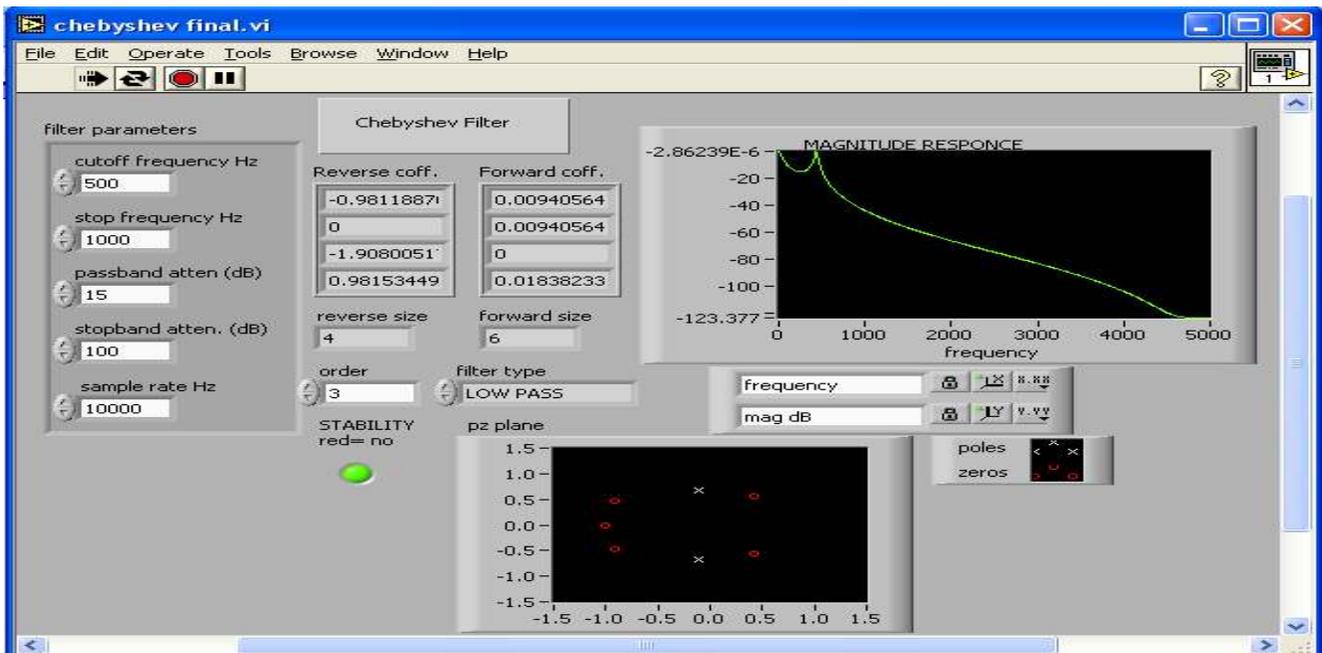


Figure 6 Response of Chebyshev Lowpass Filter for 3rd order

Figure 6 shows the response of 3rd order Chebyshev lowpass filter while all other specifications are kept same as above. Chebyshev filters are analog or digital filters having a steeper roll-off and more passband ripple (type I) or stopband ripple (type II) than Butterworth filters. Chebyshev filters have the property that they minimize the error between the idealized and the actual filter characteristic over the range of the filter, but with ripples in the passband. The pole-zero plot of Chebyshev lowpass filter for

above specification is also shown in figure 6. If we increase the order of the filter keeping the same specifications, the magnitude response plot slope becomes steep.

Figure 7 shows the response of 3rd order Inverse Chebyshev lowpass filter while all other specifications are kept same as above. If we increase the order of the filter keeping the same specifications, the magnitude response plot slope becomes steep.

Figure 8 shows the response of 3rd order Elliptic lowpass filter while all other specifications are kept same as above. If we increase the order of the filter keeping the same specifications, the magnitude response plot slope becomes steep.

From the above results we analyze that the transition band become sharper between passband and stopband on increasing the order of filter with other parameters having same value.

Compared to a Butterworth filter as shown in figure 5, Chebyshev filter as shown in figure 6, Inverse Chebyshev of figure 7 can achieve a sharper transition between the passband and the stopband with a lower order filter. The sharp transition between the passband and the stopband of these filter produces smaller absolute errors and faster execution speeds than a Butterworth filter.

Compared with the same order Butterworth or Chebyshev filters, the elliptic filters figure 8 provide the sharpest transition between the passband and the stopband, which accounts for their widespread use.

5. CONCLUSIONS

In this paper, the design of IIR filters was considered. Several results from theory were verified in the design. The

characteristics of a number of important approximations Butterworth, Chebyshev, and Elliptic were affirmed from the results obtained. Experimental results are very enthusiastic. In LabVIEW the parameters can be changed at the time of execution of the program but in case of MATLAB it is not possible. There is smooth Transition Band in LabVIEW Design and Least Square error in case of LabVIEW is also less than MATLAB Design. LabVIEW Design is based on G-Programming so that the analysis of the performance can be done very easily. In LabVIEW analysis of all types of Filters (LP, HP, BP, and BS) is possible in single program. In MATLAB all have separate programs. Design Constraints are more accurate in case of LabVIEW because we can take good approximation on LabVIEW but in MATLAB all real things are implemented there is no approximation in programming if requiring than added on program (there is no need of user definition on approximation). It is also verified that on increasing the order of any filter the transition band decreases for the same parameters.

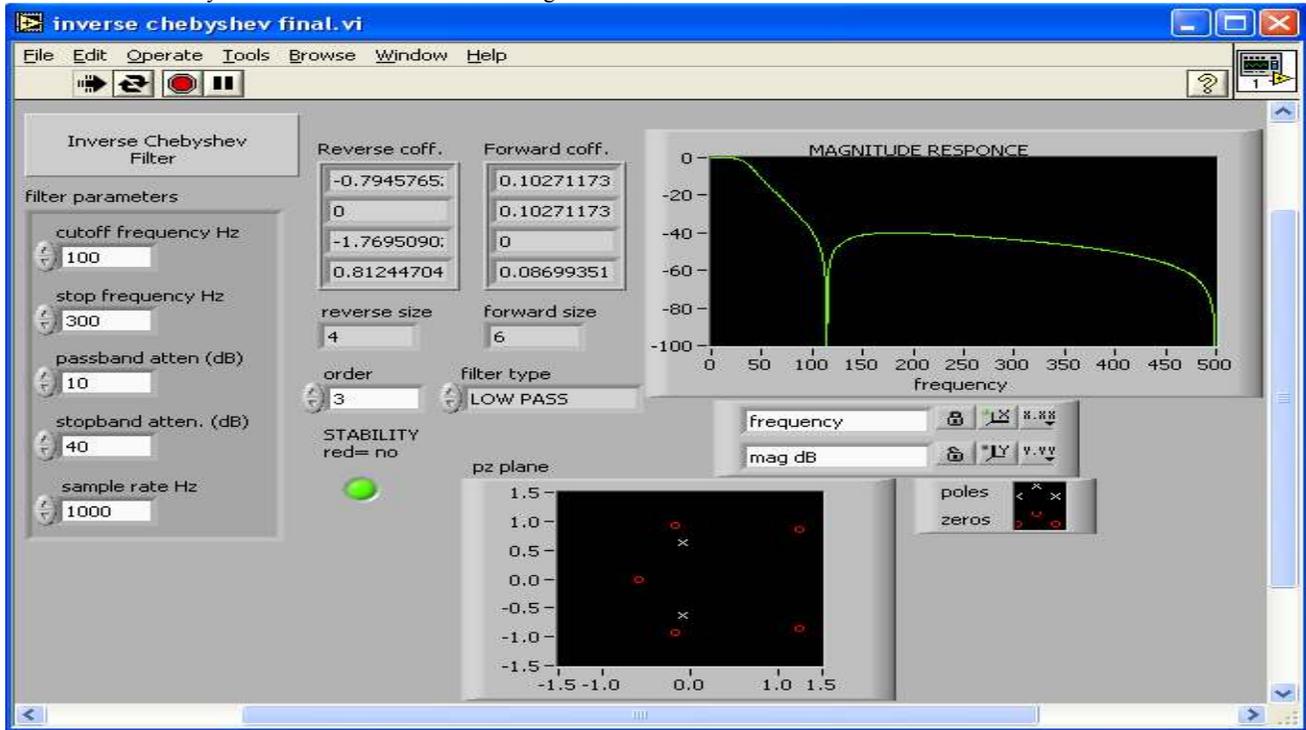


Figure 7 Response of Inverse Chebyshev Lowpass Filter for 3rd order

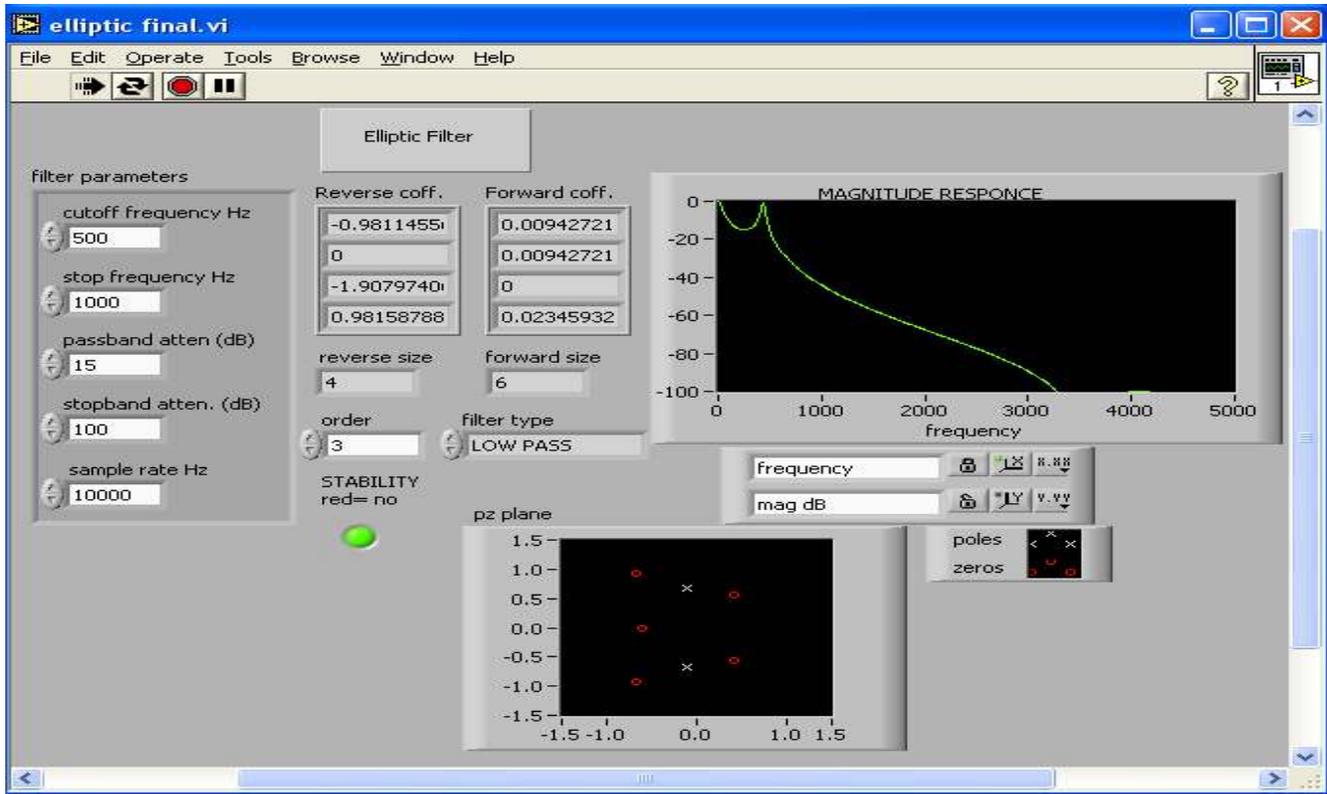


Figure 8 Response of Elliptic Lowpass Filter for 3rd order

6. REFERENCES

- [1] Aimin Jiang and Hon Keung Kwan, "IIR Digital Filter Design with Novel Stability Criterion Based on Argument Principle", Department of Electrical and Computer Engineering, University of Windsor, vol. 1, pp. 126-131, 2007
- [2] Beyon, J. Y., "Hands-On Exercise Manual for LabVIEW Programming, Data Acquisition and Analysis", Prentice Hall, Inc., New Jersey, 2001
- [3] Chugani, M. L., "LabVIEW Signal Processing", Prentice Hall, Inc., Upper Saddle River, New Jersey, 1998
- [4] Clark C. L., "LabVIEW Digital Signal Processing and Digital Communications", Tata McGRAW-HILL, 2005
- [5] Fahmy M.F., Abo-Zahhad M. and Shoby M.I., "Design of Selective Linear Phase Switched-Capacitor Filters with Equiripple Passband Amplitude Responses", IEEE Trans. on Circuits and Systems, CAS-35, no. 10, pp. 1220-1229, 1988
- [6] Jackson L. B., "Digital Filters and Signal Processing", 3rd ed., Kluwer Academic Publishers, 1996
- [7] Namjin Kim, "Digital Signal Processing System-Level Design Using LabVIEW", Elsevier Inc., vol. 1, 122-127, 2005
- [8] Oppenheim, A.V., R.W. Schaffer, "Discrete Time Signal Processing", 2nd ed., Pearson Education, 2005
- [9] Proakis J.G., and D.G. Manolakis, "Digital Signal Processing, Principles Algorithms, and Applications", 3rd ed., Pearson Education, Inc., 1996
- [10] Wells, L. K. and Travis, J., "LabVIEW for Everyone Graphical Programming Made Even Easier", Prentice Hall, Inc., Upper Saddle River, New Jersey, 1997