Mathematical & Computational Applications, Vol. 8, No. 3, pp. 287-294, 2003

A WINDOWS-BASED DIGITAL FILTER DESIGN

Yüksel ÖZBAY*, Bekir KARLIK**, A. Resit KAVSAOGLU*

*Selcuk University, Department of Electrical & Electronic Engineering, Turkey
**University of Bahrain, Department of Computer Engineering, Bahrain
yozbay@selcuk.edu.tr bkarlik@itc.uob.bh

Abstract- This study aims to describe a general digital filter, practically for Windows' users. For this purpose, an I/O interfacing circuits base on PIC16F877 was designed to receive analog signal into the PC and return filtered signals. This I/O module was then communicated with PC using parallel port protocol with EPP mode, and a digital filter program was introduced using C++. Various filters; such as LPF, HPF, BPF, and BSF were designed using the method of frequency transformation on normalized Butterworth and Chebyshev analog filters. The grades of the designed filters range from n=1 to n=8. Using this application the proposed windows-based digital filter design worked better and faster.

Keywords- Analog filter, digital filter, filter design, PIC, C++

1. INTRODUCTION

Digital signal processing is concerned with the representation of signals by sequences of numbers or symbols, and the processing of these sequences. One of the purposes of digital signal processing is designed an algorithm or devices, which is called as a digital filter. Digital filters are classified such as linear or non-linear, continuous-time or discrete-time, and recursive or non-recursive. In a non-recursive filter the output depends only on present and previous inputs. So a non-recursive digital filter possesses only z-plane zero (apart from any poles at the origin). The output from recursive digital filter depends on one or more previous output values, as well as on inputs. In other words, it involves feedback. Thus, a recursive filter has one or more strategically placed z-plane poles. In general we may write its transfer function and difference equation as:

$$H\left[e^{j\Omega}\right]=\frac{\sum_{k=0}^{M} a_k e^{-j\Omega k}}{1+\sum_{k=1}^{N} b_k e^{-j\Omega k}} \tag{1}$$

and

$$y[n]=\sum_{k=0}^{M} a_k x[n-k]-\sum_{k=1}^{N} b_k y[n-k] \tag{2}$$

where N>0 and M≥0.

At the time-domain analysis, this is total of the convolution.

$$y[n] = \sum_{l=0}^{\infty} h[n]\ x[n - l] \tag{3}$$

where h[n] is the impulse response of digital filter.

In addition, properties of digital filter are generally given as frequency-domain. The frequency response of filter can be found by using transformation $z = e^{j\Omega}$ instead of z.

$$H\left[e^{j\Omega}\right] = \frac{\sum_{k=0}^{M} a_k e^{-j\Omega k}}{1 + \sum_{k=1}^{N} b_k e^{-j\Omega k}} \tag{4}$$

The frequency response of the digital filter obtained Eq. (4) is related to the frequency response of the continuous-time filter. In the impulse invariance design procedure, the digital filter specifications are transformed to continuous-time filter specifications by inverse Fourier transform of $H[e^{j\Omega}]$.

The design of digital filter involves three basic steps: 1) the specification of desired properties of the system; 2) the approximation of these specifications using a causal discrete-time system; and 3) the realizations of the system using finite-precision arithmetic [1]. In this study, it is confirmed the digital filters, which has infinite impulse response (IIR). Design of IIR filters implies approximation by a rational function of z. The transfer function of IIR filter can be written as;

$$H[z] = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-M}}{1 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}} \tag{5}$$

where $M \le N$, the parameters of b_k (k = 1,2,….,N) is not zero.

The aim of design is to find the parameters of a_k and b_k. The traditional approach to design of IIR digital filters involves the transformation of an analog filter into a digital filter, meeting prescribed specifications. This is reasonable approach because:

1) The art of analog filter design is highly advanced and, since useful results can be achieved, it is advantageous to utilize the design procedures already developed for analog filters.

2) Many useful analog design methods have relatively simple closed-form design formulas. Therefore, digital filter design methods based on such analog design formulas are rather simple to implement.

3) The standard approximation methods that work well for analog IIR filters do not lead to simple closed-form design formulas when these methods are applied directly to the digital IIR case [2].

The digital IIR filters are obtained from analog filter approximation methods such as:

1- Impulse invariance,
2- Matched z-transform, and
3- Bilinear transformation.

One of the most effective ways of converting an analog filter into a digital equivalent is by means of a bilinear transformation. Several programs are available for the design of digital filters by several different approaches [3,4]. The digital filters are designed by using the digital processors. For this purpose, a computer is very useful and faster. In this study we developed a computer program to specify the digital Butterworth and Chebyshev filters with low-pass, high-pass, band-pass, and band-stop characteristics.

2. AN EXAMPLE FOR BILINEAR TRANSFORMATION DESIGN

The major approximation methods for frequency-selective IIR analog filters are the Buttorworth, Chebyshev, and Elliptic function approximation methods. The details of these methods can be found in Daniels (1978), Weinberg (1975), and Lam (1979). These methods are generally explained and developed in terms of low-pass filter approximations [5-7].

To illustrate the properties of the basic IIR approximation methods, we present an example of realizations of a set of filter specifications. Suppose we wish to design a low-pass digital filter. To design a filter like this, we have to know some specifications such as:

- the cut-off frequency
- dB of the attenuation after the cut-off frequency
- the type of amplitude-frequency characteristic

The cut-off frequency depends on the frequency of our desired signal. The coefficient of filter (Q(s)) is determined by order of function of normalized filter. For example; Let's suppose cut-off frequency, f_c = 18 Hz, and the second order Butterworth filter. The analog filter coefficients of the Butterworth are found as:

$$b_0 = 1.0000 \ ; \qquad b_1 = 1.41421 \tag{6}$$

and it can be written as:

$$Q(s) = 1 + 1.41421\, s' + s'^{\,2} \tag{7}$$

Our transfer function is written as:

$$H(s) = \frac{1}{Q(s)} = \frac{1}{1 + 1.41421\,s' + s'^2} \tag{8}$$

From this normalized LPF, if it is transformed like $s' = s / \omega_c$, we obtain

$$H(s) = \frac{\omega_c^2}{\omega_c^2 + 1.41421\,s\,\omega_c + s^2} \tag{9}$$

The bilinear transformation corresponds to replacing s by

$$s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \tag{10}$$

that is;

$$H[z] = H_a(s)\Big|_{s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}} \tag{11}$$

As in the case of impulse invariance, a sampling parameter T is included in the definition of bilinear transformation. Let's apply this transformation for Eq. (10),

$$H[z] = \frac{\omega_c^2}{1 + 1.41421\omega_c \frac{2}{T}\frac{(1-z^{-1})}{(1+z^{-1})} + \left(\frac{2}{T}\frac{(1-z^{-1})}{(1+z^{-1})}\right)^2} \tag{12}$$

If A = 2/T is from Eq. (12), we obtain;

$$H[z] = \frac{\omega_c^2(1 + 2z^{-1} + z^{-2})}{\omega_c^2 + 1.41421A\omega_c + A^2 + z^{-1}(2\omega_c^2 - 2A^2) + z^{-2}(\omega_c^2 - 1.41421A\omega_c + A^2)} \tag{13}$$

where T: sampling period, ω_c: cut-off frequency.

If it is written such as A =2.360 =720 and $\omega_c = 2\pi f_c = 113.097355$, and is divided all values by the constant of denominator, it is obtained;

$$H[z] = \frac{0{,}01978958281341 + 0{,}03957916562682z^{-1} + 0{,}01978958281341z^{-2}}{1 - 1{,}5645039898093z^{-1} + 0{,}64366232106300z^{-2}} \tag{14}$$

The constants of Equation 14 can be found using MATLAB with these commands below;

[b, a]=butter(2,0.1,'z') (15)

b =0.0201 0.0402 0.0201; a =1.0000 -1.5610 0.6414

As it can be seen, the constants of calculated using Eq.14 and the constants of MATLAB are very close. The corresponding difference equation is;

$$H[z]=0{,}0198x[n]+0{,}0396x[n-1]+0{,}0198x[n-2]+1{,}5645y[n-1]-0{,}6436y[n-2] \quad (16)$$

where x[n] is input, y[n] is output.

Also x[n-1], x[n-2], and y[n-1], y[n-2] are previous inputs and outputs respectively. Thus, this digital filter depends on values of present and previous input, and values of previous output.

3. THE PROGRAM OF WINDOWS-BASED DIGITAL FILTER

Realized digital filter program runs harmoniously with designed I/O port, which is shown in Fig.1. The standard parallel port with EPP mode must be supported by a PC for recognizing I/O port by the program. This process is with PIC program, which is shown in Fig.2. The PIC software is written as a Borland C++ builder 4, using the object windows-library facilities. It consists of four main windows. These are: filter design and diagram, filtering and port test, foldering and data selection, and signal screen.

3.1. The window of filter design and diagram

In this section, it is illustrated filter design, which has desired properties, and to watch filter frequency as a dB. It can be seen in Fig.3, the cut-off frequency of designed filter of illustrative example is 250 Hz, and it is a 4^{th} order low-pass filter, which has 31 dB attenuation for 500 Hz. If it is desired to design any filter; filter type, filter selection, and oscillation selection (only for Chebyshev) are built, as it can be seen on upper corner of left-hand side. From this point of view; a various type of Chebyshev and Butterworth filters can be designed as a LPF, HPF, BPF, and BSF up 8^{th} order. If we want to record designed filter, we click record button as it is seen on lower left corner in Fig.3. The properties of designed filter are listed on the filtering and port test as a record.

3.2. The window of filtering and port test

As it can be seen in Fig.4, the designed filters are listed on filter design and diagram. During the filtering, only one of the filters and active selection from selection of Active/Passive must be selected. In this section of LPT port test checks the port, which is ready or not. If it is clicked on the button of port test, it checks two bit of port.

If it has positive result, the words that "please, check to the port" transform to "port available", and start button is Active. Furthermore, from PC to the port and from the port to PC bi-direction data is transferred by the port. At the section of port speed and transformation of direction, there are three different tests such as: read only, write only (one direction data transfer), and read/write only (bi-direction data transfer).

Sampling frequency (f_s), as seen in Fig.4, is used to filter. User enters this value using the keyboard. Default value is defined as 360. If a real-time filtering is built, speed test is made for bi-direction transfer, and sampling frequency per second f_s is written instead of previous sampling frequency.

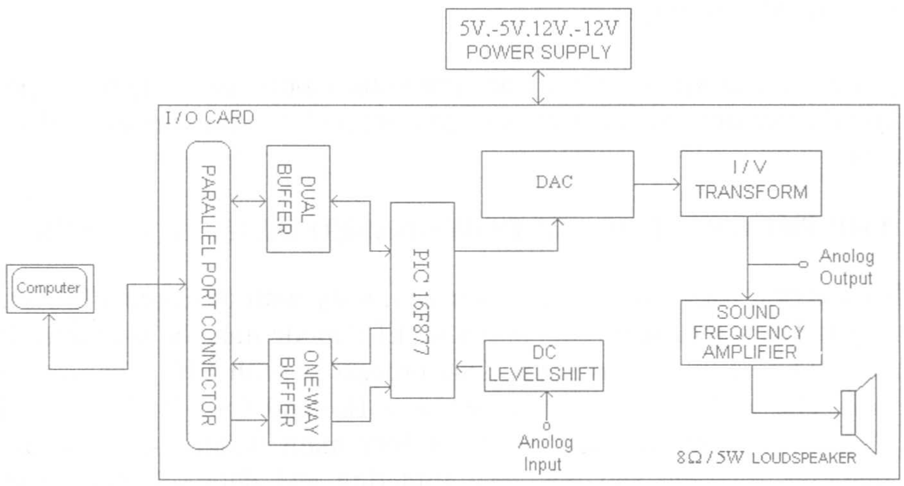

Fig. 1. The structure of I/O Card

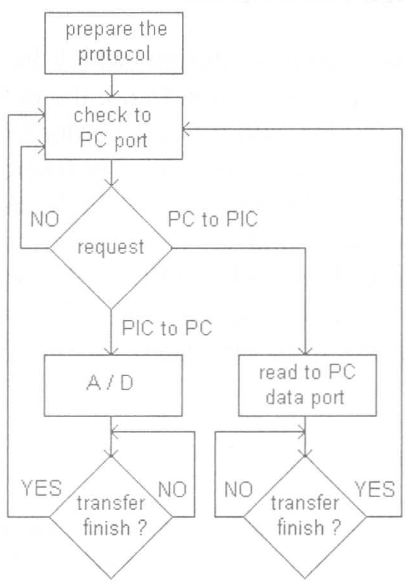

Fig. 2. The Algorithm of PIC Program

3.3. The window of foldering and data selection

As you can see in Fig.5, it is created a file belong to a person when a signal is recorded from I/O. Then, it is saved as a private data file. To filter this data: first the data is chosen, and, then the function of filter. The filtered file is recorded again after adding capital F in front of the previous file name.

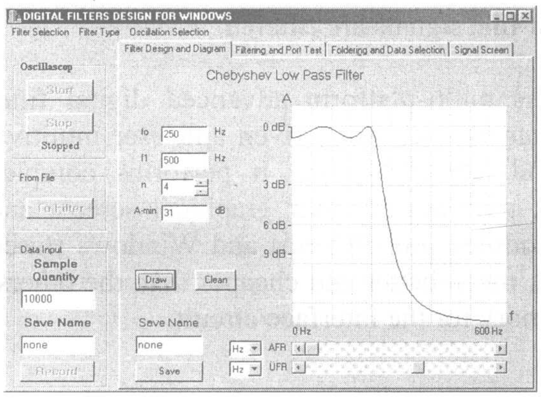

Fig. 3. Illustrative example for the filter design and diagram

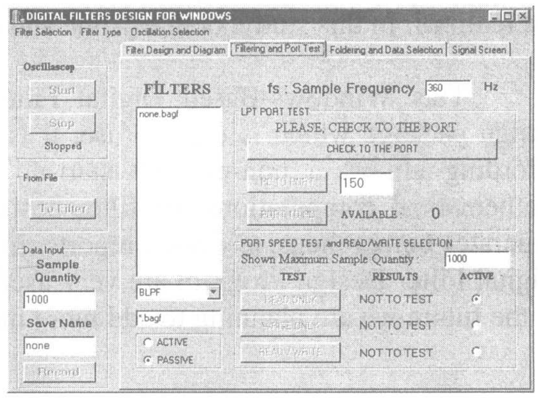

Fig. 4. Illustrative example of filtering and port test

3.4. The window of signal screen

The Fig.6 shows graphics of the filtered amplitude and the non-filtered amplitude. In the Fig.4, if read only is active, and then it is clicked on the start button, the non-filtered amplitude works as a real-time oscilloscope. Input signal is obtained on the screen. If write only is active in the Fig.4, and then it is clicked on the start button, as it can be seen in Fig.6, foldering and data selection is transferred to the output of I/O port. Then the graphics of the filtered amplitude is obtained on the screen as in figure. If the foldering doesn't finish, and it is clicked stop button, data transfer is interrupted. If read/write is active (see Fig.4), and then it is clicked on start button (see Fig.6), input signal is accepted while the filter is active at the same time, and then filtered signal is transferred to the output as a real-time.

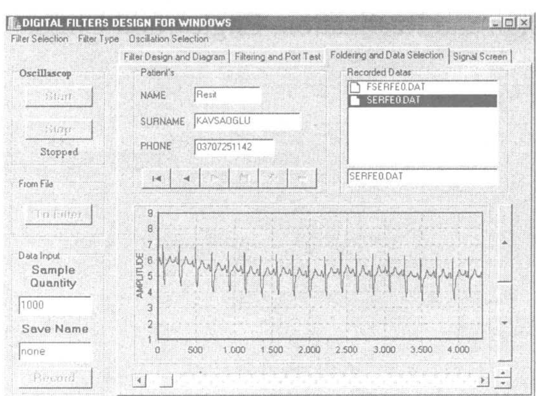

Fig. 5. Illustrative example for foldering and data selection

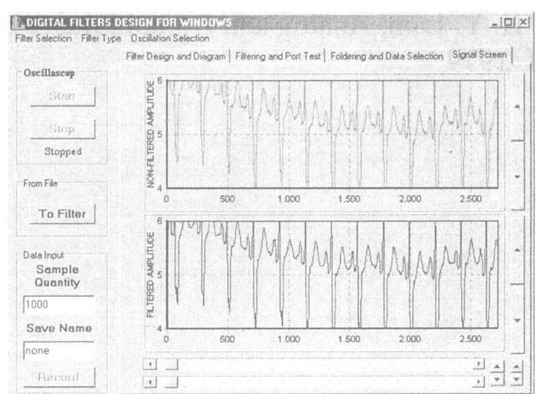

Fig. 6. Illustrative example for the signal screen

4. DISCUSSION

A computer-aided digital filter, which is obtained from Chebyshev and Butterworth's analog approaches, was presented in this study. First the desired filter function, which is taken from Chebyshev and Butterworth normalized functions, is transformed to z-domain using Bilinear-transformation. Then the difference equation is obtained. This difference equation is calculated by using a PC, and then desired filters are realized. In this study, both off-line and on-line signals are filtered.

This Windows-Based Digital Filter is multi-platform advanced digital filter design package which is easy to use. It is completely menu-driven and user intuitive, affording almost a non-existent start-up and learning curve. It performs complex mathematical computations for filter design, provides superior graphical screen, and generates comprehensive design reports. In addition, our I/O port and Windows-Based Digital Filter Design Program are very useful, much better and cheaper than the others. In the future we are planning to add more memory for the interface circuit.

REFERENCES

[1] A.V. Oppenheim and R.W. Schafer, Digital Signal Processing, Prentice Hall, New Jersey, 1975

[2] A.V. Oppenheim and R.W. Schafer, Discrete Time Signal Processing, Prentice Hall, New Jersey, 1989

[3] P.A. Lynn and W. Fuerst, Introductory Digital Signal Processing with Computer Applications, John Wiley&Sons, New York, 1996

[4] M. S. Filho, *et all.* Graphical User Interactive Interface for Digital Filter Design and DSP Implementation, CD-ROM Proceeding of ICSPAT'2000, pp. 885-889, 2000

[5] R. W. Daniels, Approximation Methods for Electronic Filter Design, McGraw-Hill, New York, 1975

[6] L. Weinberg, Network Analysis and Synthesis, R.E. Krieger-Huntington, N.Y. 1975

[7] H. Y. Lam, Analog and Digital Filters: Design and Realization, Englewood Cliffs, N.J. 1979