Designing Filters Using the Digital Filter Design Toolkit

Rahman Jamal, Mike Cerna, John Hanks

Introduction

The importance of digital filters is well established. Digital filters, and more generally digital signal processing algorithms, are classified as discrete-time systems. They are commonly implemented on a general purpose computer or on a dedicated digital signal processing (DSP) chip. Due to their well-known advantages, digital filters are often replacing classical analog filters. In this application note, we introduce a new digital filter design and analysis tool implemented in LabVIEW with which developers can graphically design classical IIR and FIR filters, interactively review filter responses, and save filter coefficients. In addition, real-world filter testing can be performed within the digital filter design application using a plug-in data acquisition board.

Digital Filter Design Process

Digital filters are used in a wide variety of signal processing applications, such as spectrum analysis, digital image processing, and pattern recognition. Digital filters eliminate a number of problems associated with their classical analog counterparts and thus are preferably used in place of analog filters. Digital filters belong to the class of discrete-time LTI (linear time invariant) systems, which are characterized by the properties of causality, recursibility, and stability. They can be characterized in the time domain by their unit-impulse response, and in the transform domain by their transfer function. Obviously, the unit-impulse response sequence of a causal LTI system could be of either finite or infinite duration and this property determines their classification into either finite impulse response (FIR) or infinite impulse response (IIR) system. To illustrate this, we consider the most general case of a discrete-time LTI system with the input sequence denoted by *x(kT)* and the resulting output sequence *y(kT)*. As it can be seen from (1), if for at least one v, $a_v \neq 0$, the corresponding system is recursive; its impulse response is of infinite duration (IIR system). If $a_v = 0$, the corresponding system is nonrecursive (FIR system); its impulse response is of finite duration and the transfer function $H(z)$ is a polynomial in z^{-1}. Commonly, b_μ is called the μ^{th} forward filter coefficient and a_v the v^{th} feedback or reverse filter coefficient. For a detailed discussion refer to standard signal processing textbooks such as Reference 1.

$$y(kT) = \sum_{\mu=0}^{m} b_\mu x(kT - \mu T) - \sum_{v=1}^{n} a_v y(kT - vT) \iff H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{\mu=0}^{m} b_\mu z^{-\mu}}{1 + \sum_{v=1}^{n} a_v z^{-v}} \qquad (1)$$

The design of digital filters involves the following basic steps:

- Determine the desired response. The desired response is normally specified in the frequency domain in terms of the desired magnitude response and/or the desired phase response.
- Select a class of filters (for example, linear-phase FIR filters or IIR filters) to approximate the desired response.
- Select the best member in the filter class.

- Implement the best filter using a general purpose computer, a DSP, or a custom hardware chip.
- Analyze the filter performance to determine whether the filter satisfies all the given criteria.

Digital Filter Design Application

The Digital Filter Design (DFD) Toolkit is a complete filter design and analysis tool that you can use to design digital filters to meet required filter specifications. The underlying filter design software is implemented in LabVIEW (References 2 and 3), a complete graphical software development system. With the toolkit, users can graphically design IIR and FIR filters, interactively review filter responses, save filter design work, and load design work from previous sessions. In addition, the DFD Toolkit includes a digital filter DLL that can be accessed from other Windows applications such as Visual Basic or Microsoft C. Plus, the DFD Toolkit includes functions so that other applications such as LabVIEW and LabWindows/CVI can load the filter coefficient and execute the filter design within the environment. Lastly, interactive real-world filter testing can be performed within the DFD application using a standard National Instruments data acquisition device. You can view the time waveforms or the spectra of both the input signal and the filtered output signal to show how the present filter performs on real-world signals. This feature delivers the ability to test the filter design before it is implemented in a custom application in LabVIEW, LabWindows, Visual Basic, or Microsoft C.

Classical IIR Filter Design

The Classical IIR Filter Design (Fig. 1) panel is used to design classical IIR digital filters. These filters include the classic types such as lowpass, highpass, bandpass, and the classic designs such as Butterworth, Chebyshev, Inverse Chebyshev, and Elliptic. The IIR Design panel consists of a graphical interface with the Magnitude vs Frequency cursors and plot on the left side and a text-based interface with digital controls on the right side. You design classical IIR filters by adjusting the filter specifications on the panel. You design the passband and stopband requirements either by using the text entry or the cursors in the Magnitude vs Freqeuncy graph. As the cursors are moved, the text entries are updated accordingly.

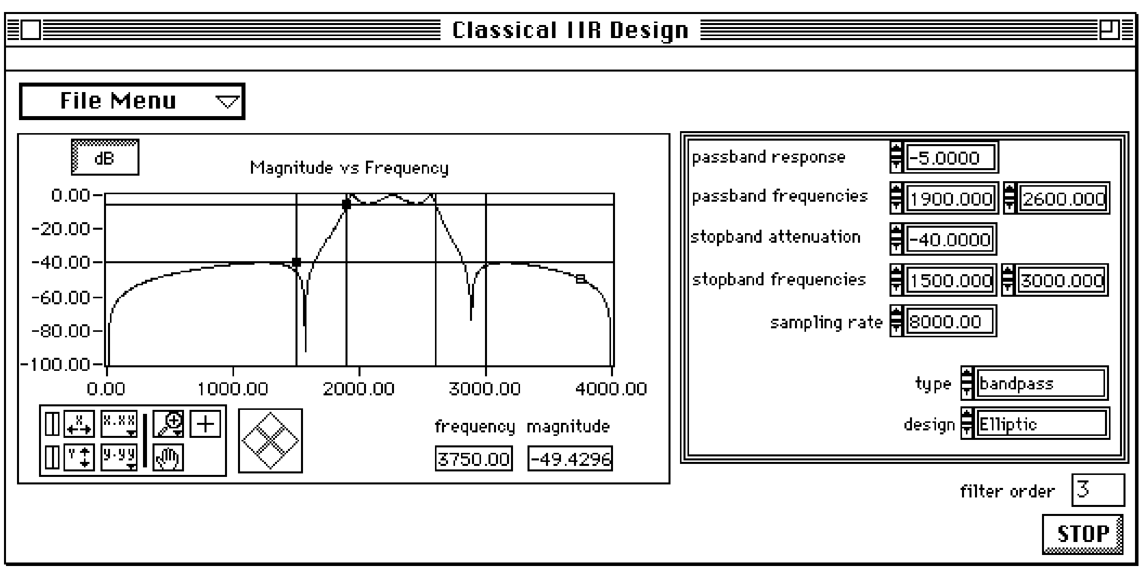

Figure 1. Classical IIR Design Panel

Classical FIR Filter Design

The classical FIR Design panel (Fig. 2) is used to design FIR filters. These filters include the classic types such as lowpass, highpass, bandpass, and bandstop and employ the Parks-Mclellan equiripple FIR filter design algorithm. This panel is very similar to the Classical IIR Design panel and operates in much the same way. The panel includes a graphical interface with the Magnitude vs Frequency cursors and plot on the left side and a text-based interface with digital controls on the right side. You design classical FIR filters adjusting the desired filter specifications. The desired passband and stopband requirements define a filter specification. You define the filter requirements by using either text entry or the cursors in the Magnitude vs Frequency graph. As the cursors are moved, the text entries are updated accordingly.

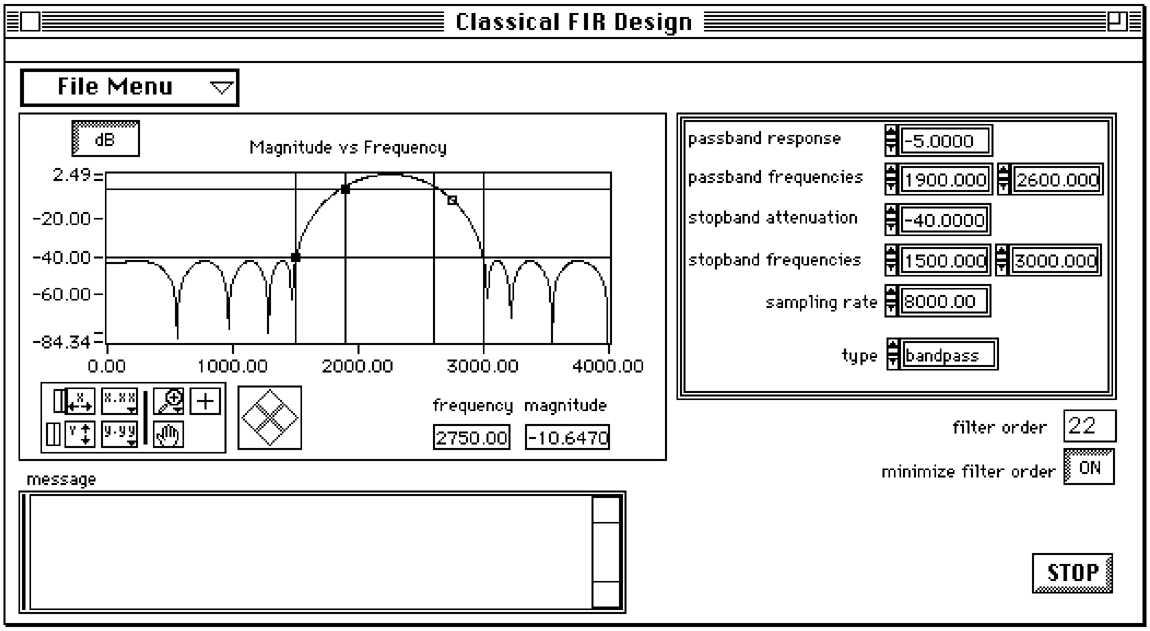

Figure 2. Classical FIR Design Panel

Pole-Zero Placement Filter Design

Figure 3 shows the Pole-Zero Placement filter design panel. The panel includes a graphical interface with the z-plane pole and zero cursors on the left side, and a text-based interface with digital controls on the right side. This panel can be used to design IIR digital filters by manipulating the filter poles and zeros in the z plane. The poles and zeros initially may have come from classical IIR designs. You can also use this panel to move or delete existing poles and zeros directly on the z plane plot for accurate control of their important characteristics. You can describe the poles and zeros by using either the text entry or the cursors in the z-plane plot. As the cursors are changed, the text entries update automatically. Likewise, as the text entries are modified, the pole/zero cursors update automatically. The pole and zero locations in the z-plane, the characteristics of each pole and zero, the gain, and the sampling rate fully describe pole-zero filter designs. Any change in these parameters corresponds to a change in the filter coefficients. The DFD Toolkit matches the poles and zeros and creates stable second-order stages for IIR filter coefficients. It then uses these coefficients to compute the filter magnitude response. The Magnitude vs Frequency plot updates automatically whenever the poles or zeros are changed giving an immediate graphical feedback to the users pole-zero filter designs.

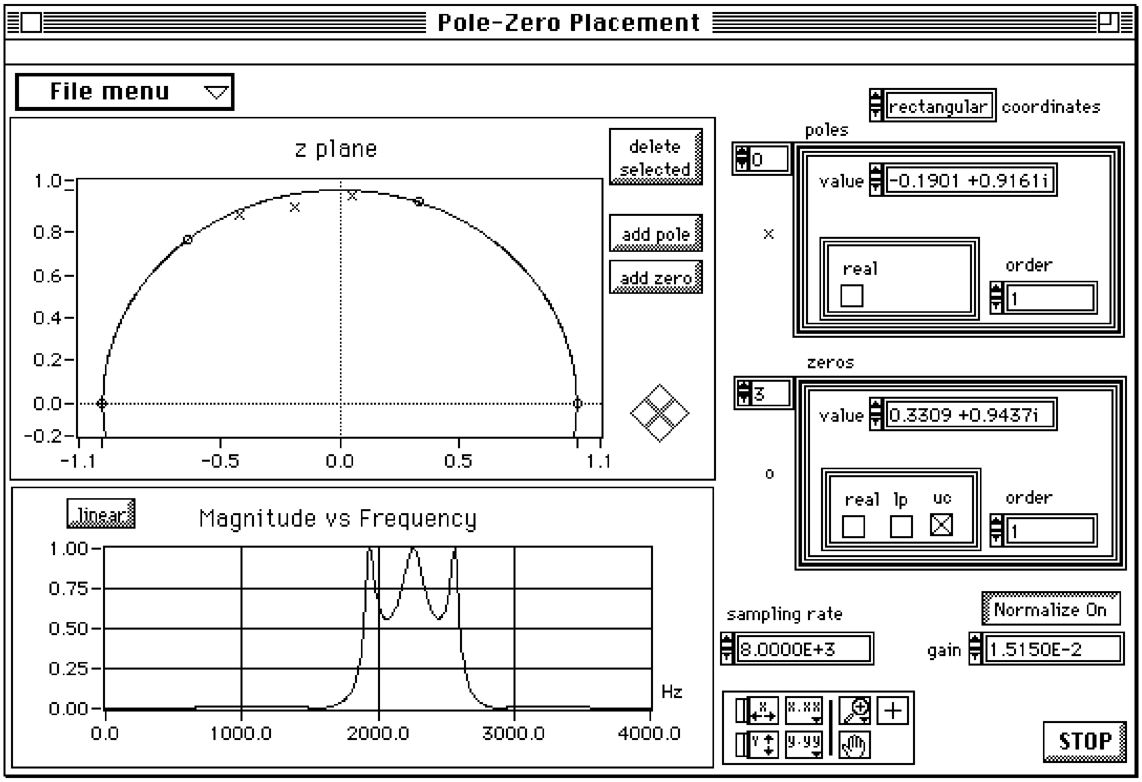

Figure 3. Pole-Zero Placement Panel

A Design Example

As an example, we will consider the design of an IIR lowpass filter with the following specifications – minimum passband magnitude response of 0.9 at frequencies at and below 400 Hz, and a maximum allowed stopband magnitude response of 0.1 at and above 500 Hz, with the sampling rate at 2000 Hz. With the DFD Toolkit, these values are simply entered into the text entries in the Classical IIR Design panel shown in Figure 4.

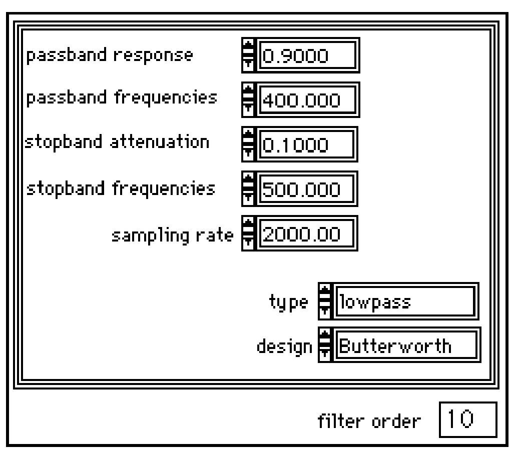

Figure 4. Filter Specification

We have selected a Butterworth lowpass filter. The filter order and internal design frequencies are computed , and the magnitude response graph is automatically updated:

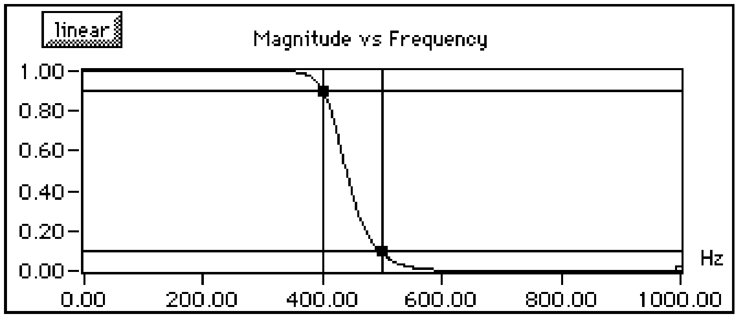

Figure 5. Maginitude Response

We can now select other designs and get an immediate feel for the required filter order as well as the actual filter shape. With the previous specifications, the Butterworth filter requires 10th order, the Chebyshev and Inverse Chebyshev require 5th order, and the Elliptic filter requires only a 3rd-order IIR filter. Figure 6 shows the plot of the 3rd-order Elliptic filter that meets our filter specifications; Figure 7 is its pole-zero plot:

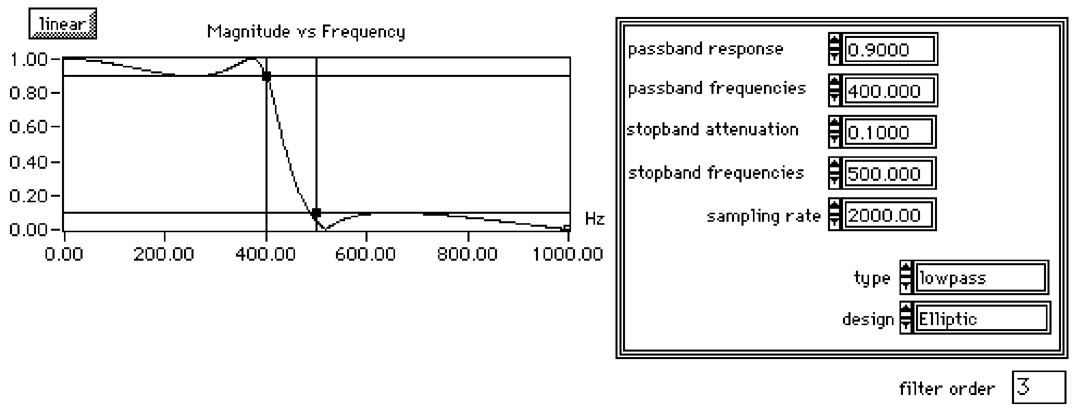

Figure 6. 3rd order Elliptic filter

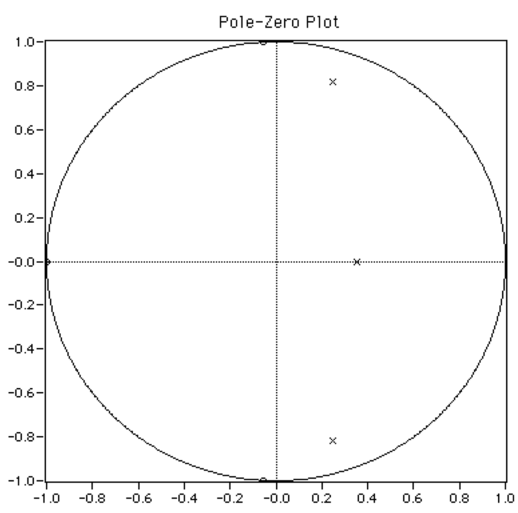

Figure 7. Pole-zero plot of the 3rd order Elliptic filter

We can save the coefficients to a text file (Table 1) containing all necessary information for implementing the designed digital filter.

Sampling Rate:	2kHz
Stage Order:	2
Number of Stages:	2
4 a Coefficients:	-3.503960E-1; 0; -4.991879E-1; 7.280681E-1
6 b Coefficients:	1.888360E-1; 1.888360E-1; 0; 1; 1.137000E-1; 1

Table 1. IIR Filter Cluster

The corresponding transfer function *H(z)* is given by:

$$H(z) = \frac{0{,}1888360(1+z^{-1})(1+0{,}1137z^{-1}+z^{-2})}{(1-0{,}3504z^{-1})(1-0{,}4992z^{-1}+0{,}7281z^{-2})} \quad (2)$$

In this example, the IIR filter is composed of two second-order stages. Because the underlying filter is really only third order, the second *a* coefficient and the third *b* coefficient are identically zero, and can be ignored for efficient implementation. To see how the present filter design performs on real-world signals, you can set up a National Instruments data acquisition device within the DFD Toolkit environment to acquire real signals. The acquired data then passes through the currently designed filter, and the DFD Toolkit plots the input and output waveforms and spectra.

Summary

Digital filter design and implementation consist of several interacting steps and call for efficient design and simulation tools. To facilitate this, National Instruments has developed a new graphical filter design and analysis tool in LabVIEW called the Digital Filter Design Toolkit. The graphical user interface greatly eases the burden associated with the specification-design-test cycle at the heart of digital filter design. The result of repeated interactive graphical design sessions is that the designer can acquire a feel for how design parameters affect filter performance.

References

1. S.K.Mitra, J. Kaiser, *Handbook for Digital Signal Processing*, 1993 John Wiley and Sons, Inc.
2. J. Kodosky, J. McCrisken, G. Rymar, *Visual Programming Using Structured Dataflow*, Proceedings of 1991 IEEE Workshop on Visual Languages, October 8-11, 1991/Kobe, Japan.
3. R. Jamal, L. Wenzel, *The Applicability of the Visual Programming Language LabVIEW to Large Real-World Applications*, Proceedings of 1995 IEEE Symposium on Visual Languages, September 4-8, 1995/Darmstadt, Germany.
4. J. Kodosky, E. Perez, *Linear Systems in LabVIEW*, National Instruments Application Note 08, January 1991.

341234A-01 Dec96