Transactions on Engineering Sciences vol 31, © 2001 WIT Press, www.witpress.com, ISSN 1743-3533

MATLAB based analog and digital filter design

David Báez-López[1], David Báez-Villegas[2], René Alcántara[3], Juan José Romero[1] & Tomás Escalante[1]
[1]*Departamento de Ingeniería Electrónica*
Universidad de las Américas-Puebla, México
[2]*Department of Classical and Modern Languages and Literatures*
Texas Tech University, Texas, USA
[3]*Bosch Corporation*
El Paso, Texas, USA

Abstract

Electric filters have a relevant importance in electronic systems because they are present in almost any electronic system. For example, communication systems, as many other electric systems, make intensive use of filtering to separate unwanted noise from the desired signal. Unfortunately, filter design is an intensive computational task requiring a significant amount of numerical calculations to obtain either the parameters of a filter transfer function or the element values for a filter circuit realization. This paper describes a software package whose purpose is to provide a tool to be used as a teaching aid in analog and digital filter design courses. The feature of this package is that it uses MATLAB [2] for the numerical computations. The main advantage of the filter design software package described in this paper is that makes uses of one of the MATLAB toolboxes, the signals toolbox (which is used for analog and digital filter design), but used with an interface that makes it possible even to the novice user to readily design filters, either analog or digital, without any previous knowledge of MATLAB or the signals toolbox.

1 Introduction

Electric filters have a relevant importance in electronic systems because they are present in almost any electronic system [1]. For example, communication systems make intensive use of filtering to separate unwanted noise from the desired signal. Power supplies use filters to reject ripple and improve the dc signal quality. Audio equalizers use filters to amplify or attenuate bands in the audio range to improve audio quality depending upon room acoustic characteristics. Digital video needs digital filters to reduce noise due to coding and transmission through a noisy channel, and so on.

Unfortunately, filter design is an intensive computational task requiring a significant amount of numerical calculations to obtain either the parameters of a filter transfer function or the element values for a filter circuit realization.

On the other hand, computer usage has reached every corner in everyday life. Thus, computer software development has become an important part of technological development. An area that has been most influenced by this development is education. Nowadays, there exists a large number of software packages especially dedicated to filter design, but they have several drawbacks. One of the most important is that those packages have a very high price.

This article describes a software package whose purpose is to provide a tool to be used as a teaching aid in analog and digital filter design courses. The feature of this package is that it uses MATLAB [2] for the numerical computations and thus it is called MFILTERS. MATLAB is now available in any university or industry, and it is used, among many other things, in the design of circuits and systems. One of the main characteristics of MATLAB is the availability of a set of toolboxes almost ready to be used in the design of filters. Unfortunately, use of these toolboxes requires a considerable length of time to master them, thus discouraging novice users to use them.

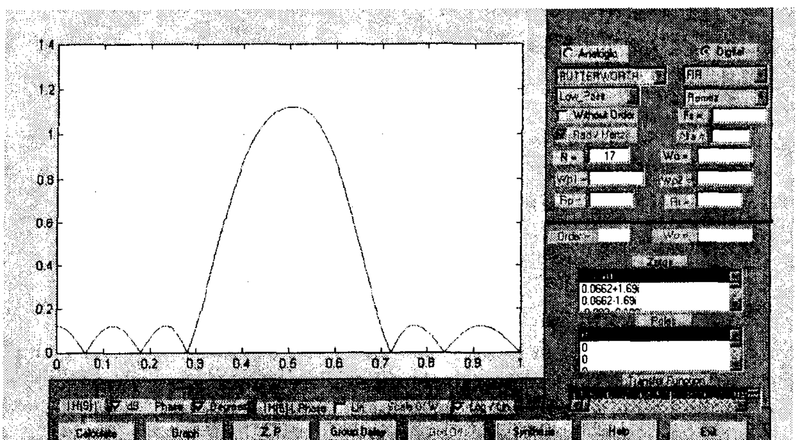

Figure 1: Main window for MFILTERS. Input and output data is given and seen at this window for an FIR digital filter.

The main advantage of the filter design software package described in this paper is the following. It makes uses of one of the MATLAB toolboxes, the signals toolbox (which is used for analog and digital filter design.) But used with an interface that makes possible, even to the novice user, to readily design filters, either analog or digital, without any previous knowledge of MATLAB or of the signals toolbox. By using this toolbox, Mfilters is less prone to errors as compared to the case of having to program each line of code in any given language. This is because the Signals toolbox has been widely used and tested. This fact makes MFILTERS truly unique. Several examples show the way the MFILTERS package works.

Transactions on Engineering Sciences vol 31, © 2001 WIT Press, www.witpress.com, ISSN 1743-3533

PSpice simulations of the circuits synthesized have shown that the output data is correct for each of a large number of filters designed with MFILTERS. Fig. 1 presents a typical window for filter design.

2 Structure of MFILTERS

MFILTERS is organized around the Matlab signals toolbox [3]. The idea behind MFILTERS is to take advantage of the expertise already available in this toolbox. Thus we developed an interface that asks users for input data in a friendly manner, passes the collected data to Matlab signals toolbox to perform the required calculations, and the output data is returned to MFILTERS for display. The advantage of designing filters in this way is twofold: first, we do not have to program each and every equation. That could take a considerable length of time to do and second, by using this well proven toolbox, we guarantee that there are no new errors introduced by programming everything ourselves. Thus, MFILTERS provides a very reliable and easy to use tool for filter design.

2.1 Theoretical background

MFILTERS is designed to solve the two fundamental problems in filter design, namely approximation and synthesis. It can be used for analog and digital filter design and in the case of analog filter is capable of addressing both problems. For the synthesis part it can perform the synthesis for passive and active filters.

2.1.1 Approximation techniques
The approximation techniques available in MFILTERS are the ones available in most text on both analog and digital signal processing. Namely, for analog and IIR (infinite impulse response) digital filters, we can obtain the transfer function for Butterworth, Chebyshev, inverse-Chebyshev, elliptic, and Thomson filters. These approximation characteristics were chosen because they are some of the most commonly used in textbooks dedicated to analog filter design and also in applications in industry. For digital filter design, the user is capable to design, in addition to IIR filters, FIR filters using either the window technique or Remez exchange algorithm.

2.2 Analog realizations

As we indicated above, passive and active realizations are available in MFILTERS. At this time we are only able to design passive realizations only for Butterworth and Chebyshev filter transfer functions.

2.2.1 Passive filters
Passive Filter design is available for Butterworth, Chebyshev, Elliptic and Thomson filters. The topology obtained is the ladder one [4].

2.2.2 Active filters

Transactions on Engineering Sciences vol 31, © 2001 WIT Press, www.witpress.com, ISSN 1743-3533

The cascade approach is used for active filter design. For active filters we have implemented circuit realizations for Sallen-Key, Multiple Feedback, and state variable realizations such as Tow-Thomas and the famous Kerwin-Huelsman-Newcomb state variable low-sensitivity realization as well as the GIC-based biquad.

2.3 Digital filters

It is possible to design IIR and FIR filters. The last ones using the windowing technique and the Remez exchange algorithm [5]. For the windowing technique typical windows used are Hamming, Hanning, rectangular, and Kaiser windows.

2.4 Transfer function output data

It is possible to obtain poles, zeros, and quadratic factors, and the numerator and denominator polynomials for the transfer function. Magnitude, phase, and group delay plots, as well as a pole and zero plots can be obtained. This data is obtained for either analog and/or digital filters.

3 Examples

In order to show how Mfilters works and the kind of data that it produces we present output for two examples.

Our first example shows a bandpass eighth order elliptic filter. Input data the set is the half-order filter (8/2=4), the bandpass cut-off frequencies ω_{p1}, and ω_{p2}, and the passband and stopband attenuations R_p and R_s.

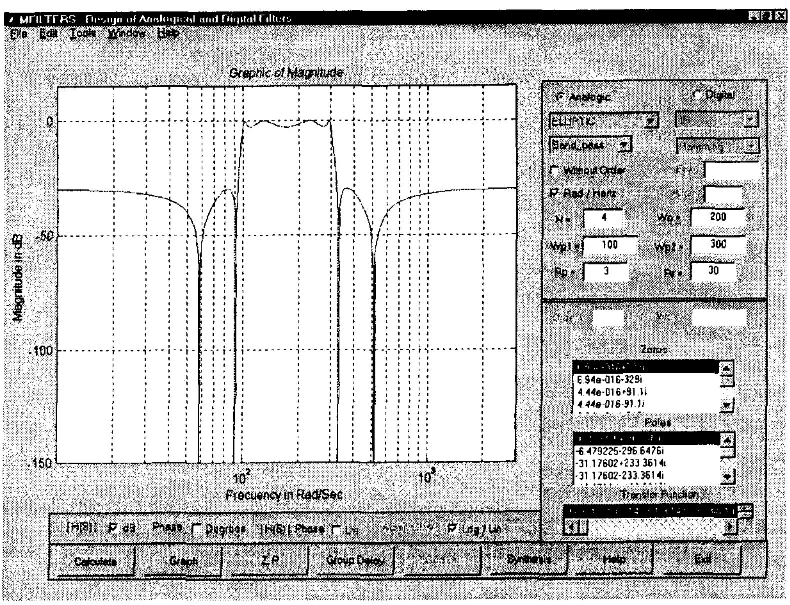

Figure 2: Window for an 8th order elliptic bandpass filter.

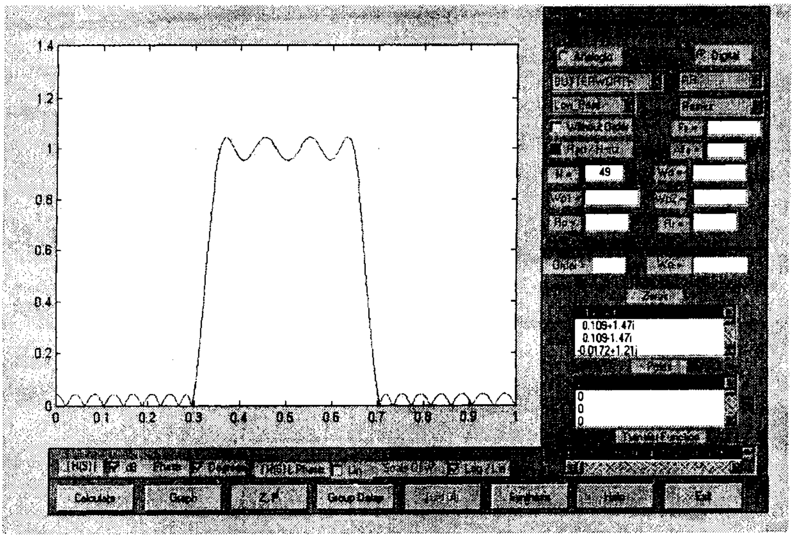

Figure 3: Magnitude for a 49th order FIR filter designed using the Remez algorithm.

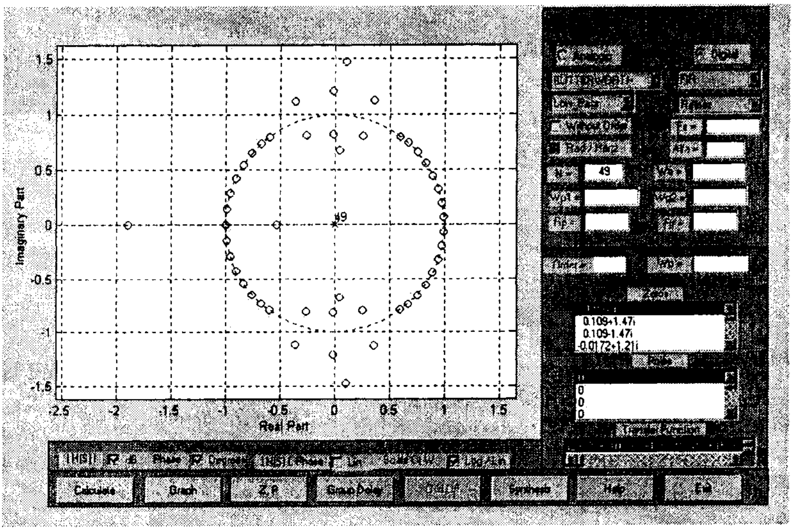

Figure 4: Pole-zero plot for the FIR filter.

The set of output data includes lists of poles and zeros and the transfer function coefficients. Also, we have options to plot magnitude, phase, group delay, and a pole-zero plot. Fig. 2 shows the window produced by MFILTERS. We see the convenience of having all the output data available in a single window.

Our second example shows an FIR filter designed using the Remez algorithm. Fig. 3 shows the window with the magnitude response and the transfer function coefficients and zeros. (All the poles are located at the origin.) The vectors for this

Transactions on Engineering Sciences vol 31, © 2001 WIT Press, www.witpress.com, ISSN 1743-3533

49th order filter are: for frequency points [0 .3 .35 .65 .7 1] and for magnitude at these frequency points [0 0 1 1 0 0].

Our last example shows in Fig. 5 the output window for an 8th order analog Butterworth bandpass filter passive realization. There we see a general topology and the element values listed to the right of the schematic circuit. In case of an active realization the element values are listed above the schematic circuit.

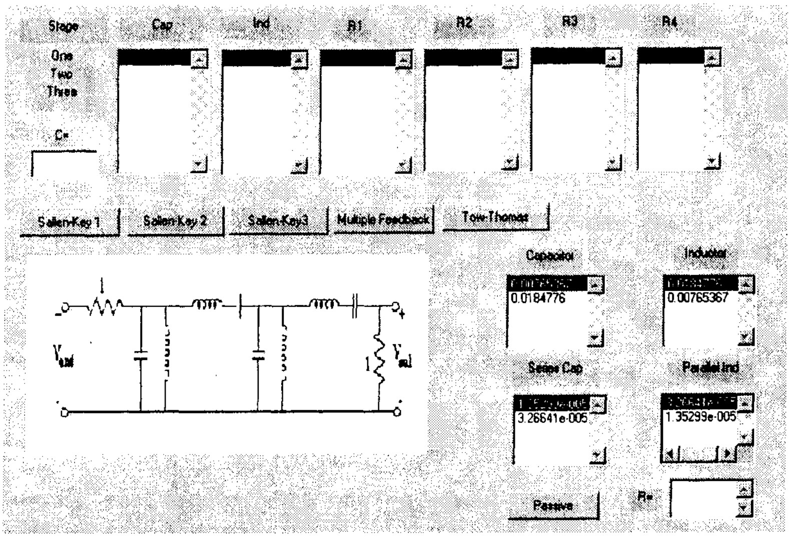

Figure 5: Element values for an 8th order Butterworth bandpass passive filter.

4 Conclusions

We have presented a software package for analog and digital filter design. Although many packages exist for either analog and/or digital filter design, the package presented here is different in that the computational engine that designs the filters is the Signals toolbox available in MATLAB. This precludes the presence of errors arising mainly by having to program a great deal of code and also avoids the need for debugging long programs. Several examples show the main characteristics of MFILTERS among which we can mention that it is very easy to use and that most of the input/output data is available in the same window.

References

[1] Huelsman, L. P., *Active and Passive Analog Filter Design*; Mc Graw Hill, 1993.
[2] MATLAB *The Language of Technical Computing*, The MathWorks Inc., Natick, MA., 1997.
[3] Alcántara, René, *MFILTERS Design of analog and digital filters*, Master's thesis, (in Spanish), Universidad de las Américas, Puebla, 2000.
[4] Escalante, T., *Passive and active filter realizations in MFILTERS 2.0*, B.S. thesis, (in Spanish), Universidad de las Américas, Puebla, 2000.

[5] Romero, J. J., *MFILTERS 2.0 FIR filter design using Remez algorithm*, B.S. thesis, (in Spanish), Universidad de las Américas, Puebla, 2000.