Symbolic Analysis of Programmable Digital Filters

Miroslav D. Lutovac, Dejan V. Tošić and Brian L. Evans

Abstract— **This paper focuses on symbolic derivation of system and noise transfer functions and symbolic synthesis of programmable digital filters. A new symbolic approach is presented to obtain robust and cost-effective realizations not available by classical digital filter design. The advantages of the new approach are discussed.**

I. Introduction

The major goal of this paper is to introduce symbolic analysis of digital filters implemented by means of programmable logic devices. Symbolic analysis is intended to complement a more general hardware description language (HDL) and to shorten the design process.

If time-to-market is critical and a low volume custom applications are required, then, reconfigurable devices are preferred. The goal is to reduce the hardware complexity and to design robust cost-effective filters. This can be accomplished by using programmable logic devices and nonstandard lengths of multiplier coefficients.

For given filter specifications all possible realizations are analyzed and the optimal solution is selected to meet the design constraints. In order to find the optimal realization a great number of different structures must be analyzed and evaluated. Usually, the designer is discouraged to examine all the available solutions, because of the high computational cost. For example, for a fifth-order transfer function with two complex pole pairs and one real pole, there are altogether 2304 equivalent realizations, made out of only two types of allpass structures, all exhibiting very low passband sensitivity [1], [2]. The number of realizations dramatically increases (more than 10,000 realizations) if some other low-sensitive structures [3], [4], [5] are taken into account.

The symbolic approach to the digital filter design inherently keeps all existing symbolic relations between the specifications and the design variables. It provides a straightforward procedure for the optimal filter design [6] which is not possible with classical numerical procedures.

M. D. Lutovac is with IRITEL, Telecommunications and Electronics Institute, Batajnički put 23, 11080 Belgrade, Yugoslavia, E-mail: elutovac@ubbg.etf.bg.ac.yu.

D. V. Tošić is with School of Electrical and Computer Engineering, The University of Belgrade, Belgrade, Yugoslavia, E-mail: etosicde@ubbg.etf.bg.ac.yu.

B. L. Evans is with Department of Electrical and Computer Engineering, The University of Texas at Austin, E-mail: bevans@ece.utexas.edu.

In this paper we consider a part of a new general symbolic design methodology that provides a systematic automated procedure for classification and verification of different realizations.

Usually, filter realizations are presented in the form of the block-diagram or flow-chart that consists of the standard elements: adders, delays and multipliers. The existing procedures ("manual" or numeric) for deriving the noise transfer functions and transfer matrices require several modifications in the block-diagram (at the output of the multiplier an additional adder and a noise source are inserted). Such a concept unnecessary complicates the analysis and is not appropriate for the automated design and analysis. In order to find all the required transfer functions, without any modification of the block-diagram, a new multiplier model is introduced [7]. This model enables symbolic deriving of the required functions in only one analysis.

The proposed approach is used for testing published realizations in order to verify the reported results, and to safely identify and possibly correct typesetting errors, frequently encountered in open literature. An example from [3] is re-analyzed in this paper; it is shown that the realization of Fig. 7a does not match the reported system transfer function and the noise transfer functions. The symbolic analysis finds out the missing "-" sign at the input of the first adder of the block-diagram. Next, it derives the correct noise transfer functions and, finally, verifies the reported expressions for the variance of quantization noise due to rounding the outputs of multipliers.

II. Why symbolic analysis in the design of digital filters?

A digital filter design starts from a given filter specification, analyzes possible realizations, and selects the optimal solution meeting the design constraints. In order to find the optimal realization a great number of different structures must be analyzed and evaluated. Usually, the designer is discouraged to examine all the available solutions, because of the high computational cost. For example, for a fifth-order transfer function with two complex pole pairs and one real pole, there are altogether 14,400 equivalent realizations, made out of only three types of allpass structures, all exhibiting very low pass-band sensitivity [1]-[5].

The number of realizations is calculated as in [1]:

the fifth-order transfer function is decomposed into one first-order and two second-order sections. This filter can be realized as a parallel connection of one second-order allpass section and one cascaded connection of a first-order and a second-order allpass sections. The first-order allpass transfer function can be realized in 4 different ways, and are designated as type 1 [1]. The second-order function can be realized as 16 allpass filters of type 2, 8 allpass filters of type 3 [1] or 36 filters of type 4 [8]. The number of possible realization of a second-order allpass section is (16+8+36)=24. The second-order section in one parallel branch can be realized in 24 different ways, and the same holds for the other branch. Therefore, for the two second-order sections, 24^2 realizations exist. The overall number of possible realizations is $24^2 \times 4$ due to the 4 existing solutions of the first-order allpass sections.

It was reported [6] that higher-order filters are more appropriate for cost-effective realizations (for example the 9th-order filter is realized with fewer multipliers than the 5th-order filter). In case of the 7th-order filter there are $24^3 \times 4 = 864{,}000$ realizations with three types of second-order sections and one first-order section. The number of possible 9th-order filter realizations is $24^4 \times 4 = 51{,}840{,}000$. Without a systematic approach it is practically impossible to find the optimal solution.

The above enumerating of possible realizations was made on the assumption that we know the numerical values of coefficients of the transfer function. On the contrary, there is an infinite number of transfer functions that satisfy required specifications. If we take into account other realizations (cascade, parallel, lattice, etc.) the number of filters to be examined increases. The infinite number of transfer functions and the large number of realizations are the main reasons for an extremely high computational cost.

So far, the traditional numerically-oriented design process could not find a robust and cost-effective solution in the straightforward manner. One possible alternative is to introduce symbolic computation.

The symbolic approach to the digital filter design inherently keeps all existing relations between the specifications and the design variables in a closed form. It provides a straightforward procedure for the optimal filter design [6] which is not possible with classical numerical procedures.

III. Programmable digital filters

We define the Programmable Digital Filter (PDF) as hardware or software implementation of a digital filter whose properties are programmable by external control signals.

Let us consider a digital filter realized as a parallel connection of two allpass sections [1]. The output of the two branches are the inputs to an adder. The adder can sum or subtract the input signals. We can obtain

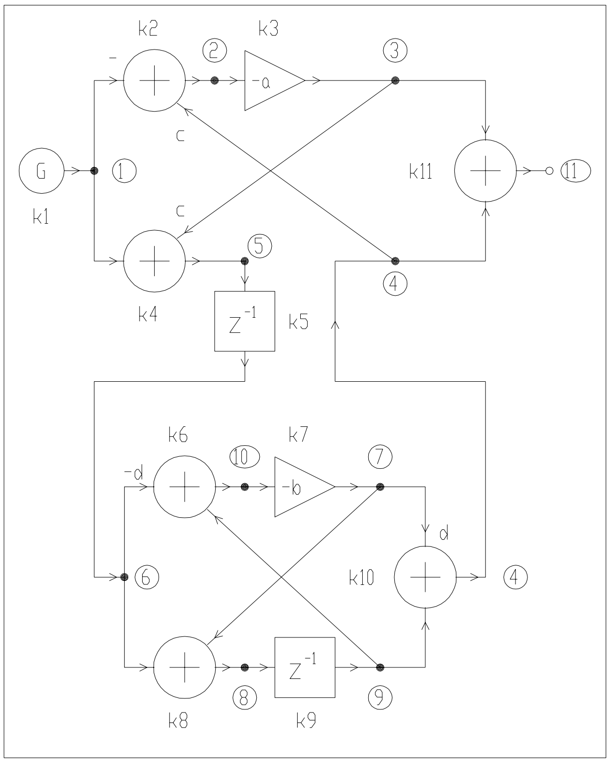

Fig. 1. Realization of allpass section

a low-pass filter by summing the input signals, or a high-pass filter by subtracting them. Therefore, the two quite different filters can be realized by single hardware and one control signal. The control signal selects the sum/subtract function of the adder.

Another kind of programmable filters has programmable sign at the adder inputs, but the system transfer function does not depend on the control signals. The adder sign pattern effects the noise transfer functions only. This means that without changing the configuration of the implemented filter we can select (by the control signals) among different noise transfer functions for the same system transfer function. The filter characteristics remain the same for all realizations, while the noise due to product quantization varies.

In hardware implementation, the proper selection of the noise transfer function means that the wordlength is shorter, and, therefore, the hardware is less complex.

If the goal is to minimize the product-quantization noise, then, we can use the same implementation, adjust the control signals, and optimize the filter without any hardware modification.

IV. Symbolic filter synthesis

In this paper, we define symbolic synthesis of a digital filter as a process of converting the transfer function into a discrete system or software, keeping some or all coefficients as symbols (not numbers). We introduce a new symbolic adder which takes into account the sum and the subtract functionality. The adder output (Y) is a weighted sum of its inputs (X_1 and X_2) of the form $Y = cX_1 + dX_2$, where the symbolic weights c and d represent the values $+1$ or -1.

Consider a second-order allpass filter section with symbolic coefficients and symbolic adders, as shown on Fig. 1. The multiplier coefficients a and b are kept as symbols, as well as c and d. The actual value of b is within the range

$$-1 < b < 1 \tag{1}$$

while a must fall into the range

$$0 < a < 1 \tag{2}$$

The values of c and d can be $+1$ or -1

$$c = \pm 1 \tag{3}$$

$$d = \pm 1 \tag{4}$$

It is important to notice that the system transfer function does not depend on c and d

$$H(z) = \pm \frac{a + b(1+a)z^{-1} + z^{-2}}{1 + b(1+a)z^{-1} + az^{-2}} \tag{5}$$

The noise transfer functions for the two multipliers are

$$T_a(z) = \frac{1 + b(1+c)z^{-1} + cz^{-2}}{1 + b(1+a)z^{-1} + az^{-2}} \tag{6}$$

$$T_b(z) = \frac{(1-ac)(d+z^{-1})}{1 + b(1+a)z^{-1} + az^{-2}} \tag{7}$$

It could be shown [6] that the section symbolic parameters a and b might be explicitely related to the filter quantities like the selectivity factor, the passband edge frequency, the passband ripple, and so on. For instance, the multiplier coefficients can be expressed in terms of these quantities. By equating the multiplier coefficients with some preferred numerical values, we determine the selectivity factor, the passband edge frequency, and the passband ripple, as reported in [6].

When the coefficients a and b are known the noise transfer functions can be optimized to minimize the product-quantization noise. By varying c and d we search for the minimal noise variance.

In the computer-aided automated symbolic approach we can perform efficient multi-criteria optimization and eliminate the non-feasible solutions. On the contrary, in the traditional design the optimal solution, for some given criteria, can lead to realizations that are impossible to implement. For example, the type 3 section in [1] is reported to have the minimum product-quantization noise, but, in the particular case of half-band filters it could not be implemented.

V. Symbolic analysis of digital filters

Digital signal processing has always been tied closely to computer implementations, where the signals are viewed as a stream of numbers. On the other hand, the design of a signal processing system treats the signals as functions in the mathematical sense. In symbolic signal processing, the signal is represented in a computer as a formula, rather than as a sequence of numbers. Thus, the value of the signal might only be known in terms of a formula, instead of a number. In a similar manner, signal processing operators, the building blocks for systems, are maintained in symbolic form, as sets of symbolic transformation rules and definitions to transform signals from one symbolic form into another. This enables machine to simplify, rearrange, and rewrite symbolic expressions until they take a desired form [9].

This section describes a concept that provides a general mechanism for encoding knowledge about the fully symbolic analysis of digital filters, and its implementation, the program SALDTIS [7][10], in Mathematica [11].

A digital filters, that has to be analyzed, is specified by the block diagram which depicts the general system operation. It is determined by its topology, by the nodes and the building blocks defined in the standard SALDTIS library. The five elementary blocks are: the generator (GEN), the multiplier (AMP), the unit delay (DELAY), the adder (SUM) and the general functional block (BLOCK).

Nodes and blocks are labeled by consecutive integer numbers starting from 1. At least, one input source (called a generator or stimulus) must exist. All the library blocks are uniquely specified by a list of elements: {type, name, output, inputs, parameters}.

The block type is a code (keyword) to identify the class a block belongs to. The name is the block individual name to distinguish among blocks of the same type. The output is a node label of the block output terminal. The inputs is a list of one or more node labels designating the block inputs. All inputs must be connected. Outputs can be floating (no connection to other nodes). The parameters are used to define a block. It is a single symbol, a list of symbols or symbolic expressions. Also, the parameters can be given by arithmetic expressions or specific numeric values: integer, real or complex. The symbol z is a reserved symbol representing the z-transform variable. It can appear in the expressions describing block parameters. The library blocks are defined by equations: GEN $\to Y(z) = G$, AMP $\to Y(z) = aX(z) + Q$, DELAY $\to Y(z) = z^{-1}X(z)$, SUM $\to Y(z) = \sum_{i=1}^{n} c_i X_i(z)$, $c_i = \pm 1$, BLOCK $\to Y(z) = \sum_{i=1}^{n} H_i(z) X_i(z)$.

The block GEN defines a stimulus (source) to the system. Its output is a known excitation in z. Typically, when transfer functions are wanted, the excitation is set to $G = 1$. At least, one GEN must be connected to an input terminal of a block.

The AMP block, also referred to as the amplifier block or gain block, performs multiplication by a constant. Usually, this constant is a single symbol or a specific numeric value. The first, mandatory, parameter of the block specifies the multiplication constant. The second, optional, parameter (Q) specifies the noise due to rounding the output.

The DELAY block represents a unit delay (z^{-1}). No parameter is required.

The block SUM performs the summing operation. Its output is an algebraic sum of its inputs. A list of signs is provided to specify addition or subtraction.

The BLOCK is a general multi-input-single-output functional block. It is primarily targeted for substitution of complex systems of known transfer functions. Generally, it can stand for a combination of AMP, DELAY and SUM blocks.

VI. Example of symbolic analysis of PDF

Symbolic analysis will be illustrated by a sample analysis of an allpass digital filter shown on Fig. 7a. in [3]. By using SALDTIS we derived the transfer function of this filter as

$$H(z) = \frac{-a + b(1-3a)z^{-1} + (1-2a)z^{-2}}{1 + b(1+a)z^{-1} + az^{-2}} \qquad (8)$$

which is different from the reported expresion, and it is not an allpass filter characteristic (5). Next, we derived the noise transfer function of the filter shown on Fig. 7b. Our result was different from that presented in the paper. The symbolic analysis, that is evaluated in few seconds on a PC, is correct and it eliminates any doubt usually existing in extensive manual derivations. The next step is to try to find the correct realization. Let's start from the general realization of Fig. 1. Regardless the values of c and d, the transfer function is of allpass type as required in [3]. If $c = 1$, $d = 1$, the noise transfer functions are

$$T_a(z) = \frac{1 + 2bz^{-1} + z^{-2}}{1 + b(1+a)z^{-1} + az^{-2}} \qquad (9)$$

$$T_b(z) = \frac{(1-a)(1+z^{-1})}{1 + b(1+a)z^{-1} + az^{-2}} \qquad (10)$$

If $c = 1$, $d = -1$, the noise transfer functions are

$$T_a(z) = \frac{1 + 2bz^{-1} + z^{-2}}{1 + b(1+a)z^{-1} + az^{-2}} \qquad (11)$$

$$T_b(z) = \frac{(1-a)(-1+z^{-1})}{1 + b(1+a)z^{-1} + az^{-2}} \qquad (12)$$

The second case ($c = 1$, $d = -1$) is in fact realization shown in Fig. 7b; therefore the correct noise transfer function is given by (12) that is formerly [3] assigned to Fig. 7a. The first case ($c = 1$, $d = 1$) result in the noise transfer function formerly assigned to Fig. 7b. [3]. If those two transfer functions are mixed, than this case is in fact the target realization of Fig. 7a. The only difference between Fig. 7a. (that is not allpass) and the first case ($c = 1$, $d = 1$) is in the sign of the first adder; thus our symbolic analysis finds out the missing "$-$" sign at the input of the first adder of the block-diagram.

VII. Conclusion

Automated, computer-aided symbolic derivation of various transfer functions of discrete-time systems is important in analysis and design of digital filters. A new symbolic approach has been proposed for finding transfer functions as symbolic expressions. A new program has been developed to carry out symbolic analysis, and help symbolic synthesis, of digital filters. Its operation is illustrated by an example of programmable digital filter design. It is targeted at analysis and design of digital filters, and in education. It can serve as a reliable tool to verify the existing solutions, to test and evaluate new realizations, and to explore design alternatives in filter synthesis to obtain simpler and more cost-effective industry solutions.

References

[1] S. K. Mitra, K. Hirano, "Digital all-pass networks", *IEEE Trans. Circuits Syst.*, vol. CAS-21, pp. 688-700, Sept. 1974.

[2] P. P. Vaidyanathan, S. K. Mitra and Y. Neuvo, "A new approach to the realization of low-sensitivity IIR digital filters", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 350-361, April 1986.

[3] R. Ansari and B. Liu, "A class of low-noise computationally efficient recursive digital filters with applications to sampling rate alterations", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 90-97, Febr. 1985.

[4] A. Fettweiss, H. Levin and A. Sedlemeyer, "Wave digital lattice filters", *Int. J. Circuit Theory Appl.*, pp. 203-211, June 1974.

[5] L. Gazsi, "Explicit formulas for lattice wave digital filters", *IEEE Trans. Circuits Syst.*, vol. CAS-32, , pp. 68-88, 1985.

[6] M. Lutovac, D. Tošić and B. Evans, "Algorithm for symbolic design of elliptic filters", *4th International conference on Symbolic Methods and Application in Circuit Design SMACD'96*, Leuven, Belgium, October 1996.

[7] D. V. Tošić, "A contribution to algorithms in computer-aided symbolic analysis of linear electric circuits and systems", Doctoral Dissertation, University of Belgrade, School of Electrical Engineering, 1996 (in Serbian).

[8] A. Simić and M. Lutovac, "Symbolic synthesis of all-pass transfer function", *13th Int. Conf. DSP97*, Santorini, Greece, July 1997.

[9] B. L. Evans, et *al.*, "User's Guide to the Signal Processing Packages and Notebooks for Implementing Linear System Theory in Mathematica 1.2 and 2.0", Georgia Institute of Technology, Atlanta, GA, 1994.

[10] D. V. Tošić, M. D. Lutovac, and I. M. Markoski, "Symbolic derivation of transfer functions of discrete-time systems," *Proc. 9th Int. Symp. Theoretical Electrical Engineering ISTET97*, Palermo, Italia, June 1997, pp. 311-314.

[11] S. Wolfram, "Mathematica - A System for Doing Mathematics by Computer", 2nd Edition, *Addison - Wesley*, 1991.