Advances in symbolic simulation of systems

Tošić, V., Dejan; and Lutovac, D., Miroslav

Abstract—*A framework and recent advances in symbolic simulation of discrete-time and continuous-time systems are presented. The role and application of symbolic analysis in modern engineering are highlighted. A software realization of a symbolic system simulator is presented and exemplified. Real-life application examples are presented in which systems are symbolically solved and simulated with Mathematica. We introduce an original approach to algorithm development, system design and symbolic processing that successfully overcomes some problems encountered in the traditional approach. Benefits of symbolic methods and the role of computer algebra systems are highlighted from the viewpoint of both academia and industry.*

Index Terms — electronic circuits, SALEC, SchematicSolver, symbolic simulation, systems

1. INTRODUCTION

COMPUTER-AIDED simulation of electronic circuits and systems is a mature field, as evidenced by the wide usage of simulation software tools. The great majority of currently available programs belong to the "numeric" category in the sense that their outputs are numbers. Symbolic analysis and simulation, on the other hand, is aimed at producing outputs as closed-form expressions that contain variables and numbers [1]-[7].

Successes in automating the design of basic analog building blocks such as opamps and comparators have almost uniformly employed an "equation-based" approach that substitutes analysis equations for simulation in order to predict the performance of an analog circuit. Symbolic analysis can be used to automatically generate a significant fraction of analysis equations needed to characterize a new circuit topology. Therefore, symbolic analysis is an important step forward in the development of CAD tools that aid in analog circuit design [2].

Symbolic analysis can provide many results which are simply not available from numeric simulation methods. Most importantly, they can provide explicit insight into the dominant behavior and properties of a circuit or system. Important application of the insight obtained from symbolic analysis is the development of the equations which are required in the use of optimization techniques to provide solution to particular design specifications. In addition, symbolic analysis can also be used in compiled-code evaluation for statistical analysis, and automated synthesis or failure diagnostics of systems [3].

The enormous increase in computing power of the present computers, combined with the development of new and more efficient analysis algorithms, allows for symbolic analysis of larger and more complex systems in shorter time than was possible before.

In this paper we present our recent advances in symbolic simulation of continuous-time and discrete-time systems. Section 3 introduces the basic terminology and definitions. Section 4 focuses on general application aspects of symbolic simulation. Section 5 highlights practical application issues of symbolic techniques. Section 6 introduces multirate systems symbolic simulation. Sections 7, 8, and 9 exemplify symbolic simulation by illustrative practical systems and the corresponding symbolic analysis.

2. PROBLEM STATEMENT

We focus our research on creating a framework for the symbolic analysis of circuits and systems that is suitable for research as well as industrial and educational applications.

Existing solutions have been focused on numerical system analysis, and their drawbacks are the inherent inability to give insight and analytic expressions for system characteristics.

Suggested solutions to the system analysis are based on new algorithms and original software implementation in *Mathematica*, which generate closed-form expressions for characteristics of microwave circuits and multirate systems.

3. DEFINITION OF SYMBOLIC SIMULATION

Symbolic simulation is a formal technique to calculate the behavior or a characteristic of a system (e.g. digital system, electronic circuit, or continuous-time system) with an independent variable (sample index, time, or frequency), the dependant variables (sample values, signals, voltages, and currents), and (some or all) the system elements represented by symbols.

The symbolic technique is complementary to numerical techniques (where the variables and the elements are represented by numbers) and qualitative analysis (where only qualitative values

Manuscript received December 15, 2006.
D. V. Tošić is with the School of Electrical Engineering, University of Belgrade, Serbia (e-mail: tosic@etf.bg.ac.yu), the contact person.

M. D. Lutovac is with the School of Electrical Engineering, University of Belgrade, Serbia (e-mail: lutovac@etf.bg.ac.yu).

are used for signals, such as increase, decrease or no change) [4].

A *symbolic simulator* is a computer program that receives the system description as input and can automatically carry out the symbolic analysis and thus generate the symbolic expression for the desired system characteristic.

Majority of the symbolic simulation research has concerned the analysis of linear systems in the frequency domain. For lumped, linear, time-invariant (LTI) systems, the symbolic transfer functions obtained are rational functions in the complex frequency variable (*s* for continuous-time systems and *z* for discrete-time systems) and the system elements that are represented by a symbol (instead of numerical value).

Depending on whether the complex frequency is the only variable, and whether all of the system elements are characterized by symbolic parameters, we have three levels of symbolic representation: (1) rational transfer function of the complex frequency with numerical coefficients, (2) partially symbolic transfer function, and (3) fully symbolic transfer function [1].

Majority of reported symbolic simulators [1]-[5] were implemented as compiled code generated with, for example, C or C++. Other simulators [8]-[27] were written as application packages (toolboxes) for computer algebra systems (CAS), such as *Mathematica* [28].

Implementing a symbolic simulator as a CAS toolbox has many advantages: (1) built in functions for the basic and advanced symbolic computations are highly optimized within the CAS, so the simulator relies on the most efficient code to be used for the analysis kernel, (2) the rich functionality of the CAS can be used for post-processing of the results generated by the simulator, (3) the supreme graphics/multimedia CAS support can be exploited for visualization, typesetting, or animation of the simulation results.

4. General application of symbolic simulation

The value of symbolic analysis is well recognized in both industry and academia. In industry it has been used as an aid in the design of systems and circuits. In academic institutions it has been found useful as an instructional aid.

There are many reasons why one may be interested in symbolic simulation. A few of the more important ones are as follows [1]:

Frequency response calculation. Suppose that an accurate magnitude response curve is desired over a frequency range with a specified frequency step. With a numerical program such as SPICE [29] or Matlab Simulink [30], the same network or system will have to be analyzed many times. On the other hand, if we obtain the symbolic transfer function first as a rational function with real coefficients, then we need only evaluate it at different values of frequency – obviously a much simpler task.

Parameter iteration. For solving piecewise linear resistive networks the exhaustive segmentation combination method is conceptually simple and can produce all solutions. In this case, the use of a symbolic technique is a natural choice. If we first obtain symbolic expressions for the response, then it is only necessary to substitute the parameters for each segment combination into the expressions and check whether the solutions lie within the ranges defining the segments. If not, that particular segment combination does not yield a solution and we also proceed to another segment combination.

Sensitivity analysis. In the design of any system, it is important to know the effect on the network performance due to the variation of some element values. A precise measure of the effect can be expressed in terms of the sensitivity functions. These functions contain partial derivatives of the system response with respect to element values (system parameters), so symbolic computation becomes a natural approach: find symbolic system function, compute (symbolically) required derivatives, and determine (a closed-form expression for) the desired sensitivity function.

Filter design by optimization. In this approach of filter design, a reasonable network configuration is first proposed, and initial element values are selected from an approximate analysis. The actual frequency response is then calculated and compared with the specified response. The process is repeated until the error is minimized. It is seen that in this approach of network design, the response of the network has to be calculated at many frequency points, and at many different sets of element values. Obviously, if a symbolic transfer function can be obtained first, repeated evaluation of the transfer function will be much simpler job than repeated analysis of the network.

Insight. Symbolic transfer function can provide better insight than numerical solutions. By inspection of the symbolic transfer function, it might be immediately clear how a parameter (or an element value) contributes to the performance and behavior of the system. Without a symbolic transfer function, these conclusions could only be reached after the analysis of many numerical cases, and even then some degree of uncertainty still exists [1]-[8], [26].

Instructional aid. Beginning linear circuit courses and signals and systems courses contain many exercises that ask students to derive the expressions for the transfer function, input impedance, voltage gain, current gain, etc. It is very easy for students (in fact, even for the instructors!) to make minor “math errors” that lead to incorrect answers. Therefore, it would be very helpful to the students to have a tool that (symbolically) checks their answers for correctness. Discovery of a wrong answer before handing in their work enables them to redo the problem and make corrections.

Symbolic analysis even has the potential to improve the training of young analog circuit designers and to guide more experienced through second-order phenomena such as distortion [2].

5. Practical application examples of symbolic techniques

Symbolic analysis is an intriguing topic in VLSI designs and it is crucial for the applications to the parasitic reduction and analog circuit evaluation [7].

For parasitic reduction, a huge amount of electrical parameters is approximated into a simplified RLC network. This reduction allows designers to handle very large integrated circuits. A symbolic analysis approach reduces the circuit according to the network topology. Thus, the designer can maintain the meaning of the original network and perform the analysis hierarchically.

For analog circuit designs, such as electrical filters or VLSI building blocks, symbolic analysis provides the relation between the tunable parameters and the characteristics of the circuit. Therefore, the analysis allows us to optimize the circuit behavior symbolically [6], [7].

Symbolic simulation of microwave linear circuits characterized by scattering parameters [14] has been found indispensable for generating analytic characterization of specific microwave networks (e.g., circuit models of microwave discontinuities) required by microwave software tools, such as *WIPL-D Microwave* [31].

Symbolic simulation has found successful application in the field of power engineering for computing the DC load flow in electric power systems [13].

Combinational networks were effectively analyzed symbolically by a simulator developed as a *Mathematica* toolbox [25].

Algorithm development can be greatly enhanced by symbolic techniques as reported in [26].

Analog Insydes is a *Mathematica* application package for modeling, analysis, and design of analog electronic circuits, tailored specifically for industrial applications [24].

SchematicSolver is a *Mathematica* application package that allows you to create symbolic representations of systems. It provides functionality for system drawing, solving, simulating, processing, and implementation [20]. The knowledge embedded in the representation can be used to generate implementation code or to analytically derive system properties, such as transfer functions or impulse responses.

SchematicSolver also automatically generates software implementations of linear and nonlinear discrete systems. This function can process symbolic samples: for a symbolic input sequence, you can compute the symbolic output sequence with both the system parameters and the states specified by symbols. Similarly, the transfer function of a complex multiple-input multiple-output (MIMO) system can be derived in terms of system parameters kept as symbols [23], [26].

Symbolic signal processing, an innovative feature of *SchematicSolver* not available in other software, brings you computation of transfer functions as closed-form expressions in terms of symbolic system parameters and can find the closed-form response of schematics. The derived result is the most general because all system parameters, inputs, and initial conditions (states) can be given by symbols [20].

The symbolic algorithm for the elliptic rational function was used to optimize the symbolic performance of analog and digital systems [21]. This optimization is not possible with traditional numeric algorithms. Specific application of this result is derivation of formulas for designing high-speed low-consumption systems known as quadrature mirror filter banks.

Moreover, we found a new function, known as *minimum-Q elliptic*, by symbolically optimizing the elliptic rational function [21]. Minimum-Q elliptic became a standard function in manufacturing integrated filters. In addition, again using symbolic optimization, we implemented a very efficient digital signal processing (DSP) system using programmable logic devices and very large-scale integrated circuits. By an efficient DSP system, we mean processing by multiplierless systems that consist of a small number of adders and binary shifters.

6. Multirate systems symbolic simulation

Multirate systems are important constituents of the modern information communication technology (ICT) infrastructure, such as Internet or multimedia systems. These systems play a very important role in digital signal processing (DSP), and their design is a highly specialized field within ICT/DSP engineering [32].

We are currently developing a new concept of multirate systems simulation as outlined in [27]. Our original software package, being developed in *Mathematica*, is proposed for implementation and symbolic analysis of multiple-input multiple-output systems consisting of upsamplers, downsamplers, adders, multipliers, delays, and shift registers. Some initial works on our multirate symbolic analysis were presented in [20] and [26].

7. Example circuit simulation

Electric circuits are fundamental examples of continuous-time systems. Symbolic simulation of a lumped linear time-invariant electric circuit will be exemplified by the analysis of an active RC differentiator implemented with second generation current conveyors (CCII) as shown in fig. 1.

Active RC differentiator and integrator networks are widely useful in the analog signal processing applications, such as signal generating, computing, process control, and many test instrumentation circuits.

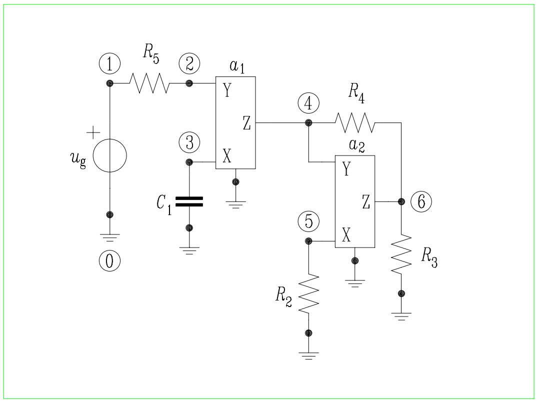

Figure 1. Differentiator with tunable time constant using current conveyors [33].

The schematic specification for the electric circuit of fig. 1 can be generated by DrawFilt [34] and is internally represented as a *Mathematica* symbolic object

numberOfNodes = 6
component[1] = {"V", "Ug", 1, 0, Ug}
component[2] = {"R", "R5", 1, 2, R5}
component[3] = {"C", "C1", 3, 0, C1}
component[4] = {"R", "R2", 5, 0, R2}
component[5] = {"R", "R3", 6, 0, R3}
component[6] = {"R", "R4", 4, 6, R4}
component[7] = {"CCII", "cc1", {4,0}, {2,3}, a1}
component[8] = {"CCII", "cc2", {6,0}, {4,5}, a2}
numberOfComponents = 8

which is stored in a file (e.g. Differentiator.m). The symbolic object contains all the details necessary for drawing, solving and simulating the circuit.

Simulation is carried out in the complex domain by SALEC, a *Mathematica* package for **S**ymbolic **A**nalysis of **L**inear **E**lectric **C**ircuits, which yields the analytic response as shown in fig. 2.

The circuit response can be post-processed in *Mathematica* to obtain a more suitable form of the symbolic expression that reveals the differentiator behavior, as shown in fig. 3.

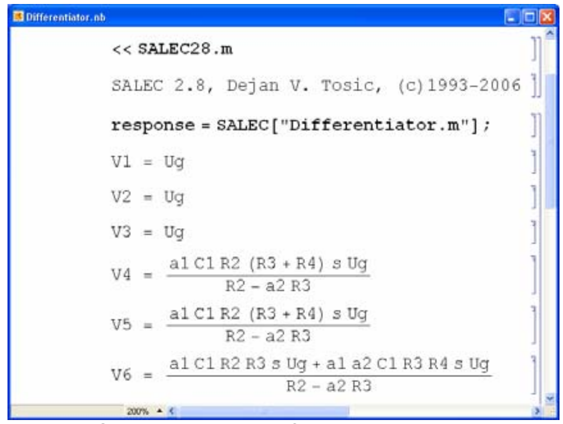

Figure 2. Symbolic response of the electric circuit shown in fig. 1, which is generated by SALEC. The node voltages are closed-form expressions in terms of element values (circuit parameters) and the Laplace variable (complex frequency).

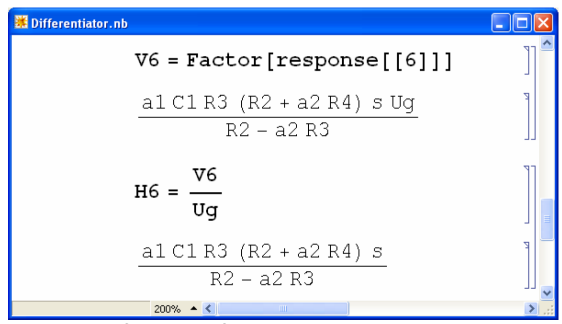

Figure 3. Refinement of the symbolic response that reveals the differentiator behavior: the transfer function is of the form *K s*, where *K* is a constant dependent on the element values, and *s* is the Laplace variable (complex frequency).

8. EXAMPLE CONTINUOUS-TIME SYSTEM SIMULATION

The following example describes a typical use of CAS in control system analysis. Suppose that the published block-diagram of a system and the corresponding transfer function should be proved. Manual derivation can be tedious and error prone even for a motivated researcher. Alternatively, *SchematicSolver* [20] can be used to draw the system schematic, such as that shown in fig. 4, and to find the transfer function. The transfer function matrix of this three-input four-output MIMO system is shown in fig. 5.

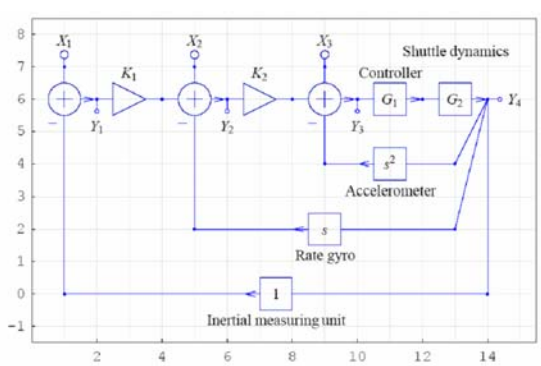

Figure 4. Simplified model of the pitch controller for the space shuttle.

$$
\begin{pmatrix}
\frac{G_1 G_2 s^2+G_1 G_2 K_2 s+1}{G_1 G_2 s^2+G_1 G_2 K_2 s+G_1 G_2 K_1 K_2+1} & -\frac{G_1 G_2 K_2}{G_1 G_2 s^2+G_1 G_2 K_2 s+G_1 G_2 K_1 K_2+1} & -\frac{G_1 G_2}{G_1 G_2 s^2+G_1 G_2 K_2 s+G_1 G_2 K_1 K_2+1} \\
\frac{G_1 G_2 K_1 s^2+K_1}{G_1 G_2 s^2+G_1 G_2 K_2 s+G_1 G_2 K_1 K_2+1} & \frac{G_1 G_2 s^2+1}{G_1 G_2 s^2+G_1 G_2 K_2 s+G_1 G_2 K_1 K_2+1} & \frac{-s G_1 G_2-G_1 K_1 G_2}{G_1 G_2 s^2+G_1 G_2 K_2 s+G_1 G_2 K_1 K_2+1} \\
\frac{K_1 K_2}{G_1 G_2 s^2+G_1 G_2 K_2 s+G_1 G_2 K_1 K_2+1} & \frac{K_2}{G_1 G_2 s^2+G_1 G_2 K_2 s+G_1 G_2 K_1 K_2+1} & \frac{1}{G_1 G_2 s^2+G_1 G_2 K_2 s+G_1 G_2 K_1 K_2+1} \\
\frac{G_1 G_2 K_1 K_2}{G_1 G_2 s^2+G_1 G_2 K_2 s+G_1 G_2 K_1 K_2+1} & \frac{G_1 G_2 K_2}{G_1 G_2 s^2+G_1 G_2 K_2 s+G_1 G_2 K_1 K_2+1} & \frac{G_1 G_2}{G_1 G_2 s^2+G_1 G_2 K_2 s+G_1 G_2 K_1 K_2+1}
\end{pmatrix}
$$

Figure 5. Transfer function matrix of the example multiple-input multiple-output (MIMO) continuous-time system shown in fig. 4.

9. EXAMPLE DISCRETE-TIME SYSTEM SIMULATION

SchematicSolver can be used to (a) draw the schematic of a discrete multiple-input multiple-output system, (b) compute the system transfer function directly from the schematic, (c) find the response for given input sequences, or (d) derive some properties of the system.

Consider a high-speed filter that can be used in multirate systems for interpolation and decimation with a factor of two. The automatically generated schematic of the filter is composed of all-pass filters and extra multipliers, and is shown in fig. 6 [23].

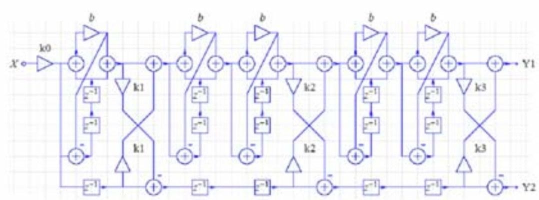

Figure 6. Power complementary high-speed filter suitable for multirate systems for interpolation and decimation with a factor of two.

Symbolic simulation by *SchematicSolver* [20] finds the partial transfer function for the second output as shown in fig. 7.

$$
\begin{aligned}
&\frac{1}{z^5\,(b+z^2)^5}\\
&(b^5\,k0\,k3 - b^4\,k0\,k1\,k3\,z + b^3\,k0\,k2\,z^2 + 5\,b^4\,k0\,k3\,z^2 - b^3\,k0\,k1\,k2\,k3\,z^2 -\\
&\quad b^2\,k0\,k1\,k2\,z^3 - 4\,b^3\,k0\,k1\,k3\,z^3 - b^5\,k0\,k1\,k3\,z^3 - b^2\,k0\,k2\,k3\,z^3 +\\
&\quad b\,k0\,k1\,z^4 + 3\,b^2\,k0\,k2\,z^4 + 2\,b^4\,k0\,k2\,z^4 + 10\,b^3\,k0\,k3\,z^4 -\\
&\quad 3\,b^2\,k0\,k1\,k2\,k3\,z^4 - 2\,b^4\,k0\,k1\,k2\,k3\,z^4 + k0\,z^5 - 2\,b\,k0\,k1\,k2\,z^5 -\\
&\quad 3\,b^3\,k0\,k1\,k2\,z^5 - 6\,b^2\,k0\,k1\,k3\,z^5 - 4\,b^4\,k0\,k1\,k3\,z^5 - 2\,b\,k0\,k2\,k3\,z^5 -\\
&\quad 3\,b^3\,k0\,k2\,k3\,z^5 + k0\,k1\,z^6 + 4\,b^2\,k0\,k1\,z^6 + 3\,b\,k0\,k2\,z^6 + 6\,b^3\,k0\,k2\,z^6 +\\
&\quad b^5\,k0\,k2\,z^6 + 10\,b^2\,k0\,k3\,z^6 - 3\,b\,k0\,k1\,k2\,k3\,z^6 - 6\,b^3\,k0\,k1\,k2\,k3\,z^6 -\\
&\quad b^5\,k0\,k1\,k2\,k3\,z^6 + 5\,b\,k0\,z^7 - k0\,k1\,k2\,z^7 - 6\,b^2\,k0\,k1\,k2\,z^7 -\\
&\quad 3\,b^4\,k0\,k1\,k2\,z^7 - 4\,b\,k0\,k1\,k3\,z^7 - 6\,b^3\,k0\,k1\,k3\,z^7 - k0\,k2\,k3\,z^7 -\\
&\quad 6\,b^2\,k0\,k2\,k3\,z^7 - 3\,b^4\,k0\,k2\,k3\,z^7 + 4\,b\,k0\,k1\,z^8 + 6\,b^3\,k0\,k1\,z^8 +\\
&\quad k0\,k2\,z^8 + 6\,b^2\,k0\,k2\,z^8 + 3\,b^4\,k0\,k2\,z^8 + 5\,b\,k0\,k3\,z^8 - k0\,k1\,k2\,k3\,z^8 -\\
&\quad 6\,b^2\,k0\,k1\,k2\,k3\,z^8 - 3\,b^4\,k0\,k1\,k2\,k3\,z^8 + 10\,b^2\,k0\,z^9 - 3\,b\,k0\,k1\,k2\,z^9 -\\
&\quad 6\,b^3\,k0\,k1\,k2\,z^9 - b^5\,k0\,k1\,k2\,z^9 - k0\,k1\,k3\,z^9 - 4\,b^2\,k0\,k1\,k3\,z^9 -\\
&\quad 3\,b\,k0\,k2\,k3\,z^9 - 6\,b^3\,k0\,k2\,k3\,z^9 - b^5\,k0\,k2\,k3\,z^9 + 6\,b^2\,k0\,k1\,z^{10} +\\
&\quad 4\,b^4\,k0\,k1\,z^{10} + 2\,b\,k0\,k2\,z^{10} + 3\,b^3\,k0\,k2\,z^{10} + k0\,k3\,z^{10} -\\
&\quad 2\,b\,k0\,k1\,k2\,k3\,z^{10} - 3\,b^3\,k0\,k1\,k2\,k3\,z^{10} + 10\,b^3\,k0\,z^{11} - 3\,b^2\,k0\,k1\,k2\,z^{11} -\\
&\quad 2\,b^4\,k0\,k1\,k2\,z^{11} - b\,k0\,k1\,k3\,z^{11} - 3\,b^2\,k0\,k2\,k3\,z^{11} - 2\,b^4\,k0\,k2\,k3\,z^{11} +\\
&\quad 4\,b^3\,k0\,k1\,z^{12} + b^5\,k0\,k1\,z^{12} + b^2\,k0\,k2\,z^{12} - b^2\,k0\,k1\,k2\,k3\,z^{12} +\\
&\quad 5\,b^4\,k0\,z^{13} - b^3\,k0\,k1\,k2\,z^{13} - b^3\,k0\,k2\,k3\,z^{13} + b^4\,k0\,k1\,z^{14} + b^5\,k0\,z^{15})
\end{aligned}
$$

Figure 7. Fully symbolic partial transfer function of the power complementary high-speed filter shown in fig. 6.

Needless to say, derivation of this transfer function by hand is very time consuming and difficult (if possible at all) for more complex high-speed filters.

The filter of fig. 6 is a power complementary filter, that is, the partial transfer functions satisfy the power complimentary equation (condition):

$$H_1(z)H_1(1/z)+H_2(z)H_2(1/z)=1$$

This condition cannot be simply proved without a powerful computer algebra system (CAS), e.g., *Mathematica* [28], and an appropriate symbolic simulator, such as *SchematicSolver* [20]. Furthermore, *SchematicSolver* and *Mathematica* derive the exact analytic condition to be met by the filter coefficients [26]:

$$k0 \to \frac{1}{\sqrt{2}\,\sqrt{1+k1^2}\,\sqrt{1+k2^2}\,\sqrt{1+k3^2}}$$

Definitely, manual derivation of such a formula would be a fatiguing labor.

10. CONCLUSION

Contemporary trends to use very sophisticated algorithms combine expertise in many areas, such as multirate system design, control engineering, analog electronic circuit VLSI design, and signal processing. This trend caused programming to become a task of knowledge accumulation and an efficient human-machine interface.

This paper presented recent advances and the role of symbolic computations in modern engineering and signal processing. It provided illustrative application examples as appropriate to linear systems, modeling, and simulation.

System models were highlighted as visualized algorithms by means of block-diagram representations—schematics. The schematic was established as a symbolic object that contained all details for drawing, symbolic solving, simulating, and implementing the system. It was not seen as a static picture.

It was shown how computer algebra systems (CAS) analyzed the schematic as the symbolic object. The knowledge embedded in the schematic object was used according to the required task, such as, to generate the electric circuit response or to derive the transfer function.

A typical use of CAS in system analysis was illustrated by solving a continuous-time linear MIMO system; the transfer function matrix was determined directly from the schematic.

Transfer function matrix of a complex MIMO discrete system was derived in terms of system parameters kept as symbols. The important power complimentary property was proved, for a class of high-speed filters, for arbitrary symbolic system parameters. The proof involved manipulation of complex expressions that was practically impossible to perform by hand.

Benefits of symbolic methods and the role of CAS were highlighted from the viewpoint of both academia and industry.

REFERENCES

[1] Lin, Pen-Min, "Symbolic network analysis," *Elsevier*, Amsterdam, The Netherlands, EU, 1991.
[2] Gielen, G., Sansen, W., "Symbolic analysis of automated design of analog integrated circuits," *Kluwer Academic Publishers*, Norwell, MA, USA, 1991.
[3] Huelsman, L., Gielen, G., "Symbolic Analysis of analog circuits: Techniques and applications," *Kluwer Academic Publishers*, Norwell, MA, USA, 1993.
[4] Gielen, G., Wambacq, P., Sansen, W., "Symbolic analysis methods and applications for analog circuits: A tutorial overview," *Proceedings of the IEEE*, 1994, pp. 286-304.
[5] Fernández, F., Rodríguez-Vázquez, A., Huertas, J., Gielen, G., "Symbolic analysis techniques: Applications to analog design automation," *Wiley-IEEE Press*, 1997.
[6] Lutovac, D., Tošić, D., Evans, B., "Filter Design for Signal Processing using MATLAB and *Mathematica*," *Prentice Hall*, Upper Saddle River, NJ, USA, 2001.
[7] Qin, Z., Tan, S., Cheng, C., "Symbolic analysis and reduction of VLSI circuits," *Springer*, 2004.
[8] Riddle, A., Dick, S., "Applied electronic engineering with *Mathematica*," Addison-Wesley, Reading, MA, USA, 1995.
[9] Tošić, D. V., "SALECAS - a package for symbolic analysis of linear circuits and systems," in *Proc*. 4th International Workshop on Symbolic Methods and Applications to Circuit Design, 1996, pp. 227–230.
[10] Tošić, D. V., Hribšek, M. F., Reljin, B. D., "Generation and design of new continuous-time second order gain equalizers using program SALEC", *International Journal of Electronics and Communications* (AEÜ - Archiv für Elektronik und Übertragungstechnik), 1996, pp. 226-229.
[11] Tošić, D. V., Kovačević, B. D., Reljin, B. D., "Symbolic Analysis of Linear Dynamic Systems", *Control and Computers*, 1996, pp. 54-59.
[12] Tošić, D. V., "A contribution to algorithms in computer-aided symbolic analysis of linear electric circuits and systems," *Doctoral dissertation*, University of Belgrade, School of Electrical Engineering, Belgrade, Serbia, 1996.
[13] Škokljev, I. A., Tošić, D. V., "A new symbolic analysis approach to the DC load flow method", *Electric Power System Research Journal*, 1997, pp. 127-135.
[14] Tošić, D. V., Djordjević, A. R., Reljin, B. D., "Symbolic Analysis of Microwave Circuits", *Journal of Applied Electromagnetism*, 1997, pp. 37-45.
[15] Tošić, D. V., Lutovac, M. D., "Symbolic analysis of digital filters", *Académie Roumaine*, Revue Roumaine des Sciences Techniques, Série Électrotechnique et Énergétique, Bucarest, 1997, pp. 29-38.
[16] Lutovac, M. D., Tošić, D. V., "Symbolic computation of digital transfer function using MATLAB", in *Proc*. 23rd Int. Conf. Microelectronics, MIEL, 2002, pp. 651-654.
[17] Lutovac, M. D., Tošić, D. V., "Symbolic Signal Processing and System Analysis", *Facta Universitatis*, Series: Electronics and Energetics, 2003, pp. 423-431.
[18] Bakshee, I., "Control System Professional," *Wolfram Research*, Champaign, USA, 2003.
[19] DLR, Institut fur Robotik und Mechatronik, "PARADISE, Parametric Robustness Analysis and Design Interactive Software Environment," [online] http://www.dlr.de/rm/en/desktopdefault.aspx/tabid-482/admin-1/, 2004.
[20] Lutovac, M.D., Tošić, D.V., "*SchematicSolver* version 2," [online] http://www.schematicsolver.com, 2004.
[21] Lutovac, M.D., Tošić, D.V., "Elliptic Rational Functions," *The Mathematica Journal*, 2005, pp. 598-608.
[22] Palancz, B., Benyo, Z., Kovacs, L., ''Control System Professional Suite'', *IEEE Control Systems Magazine*, 2005, pp. 67–75.
[23] Lutovac, M. D., Tošić, D. V., "High-Speed Filter Design using *Mathematica*", in *Proc*. IEEE EUROCON 2005 – The International Conference on "Computer as a Tool", 2005, pp. 1626-1629.
[24] Fraunhofer ITWM, "*Analog Insydes* version 2.1," [online] http://www.analog-insydes.de, 2005.
[25] Tošić, D. V., Simić, S. K., "Analysis of Combinational Networks with *Mathematica*", *Univ. Beograd. Publ. Elektrotehn. Fak. Ser. Mat.*, 2005, pp. 76–87.
[26] Lutovac, M. D., Tošić, D. V., "Symbolic analysis and design of control systems using *Mathematica*", *International Journal of Control*, Special issue on the use of computer algebra systems for computer aided control system design, 2006, pp. 1368–1381.
[27] Tošić, D. V., Lutovac, M. D., "Multirate systems simulation with *Mathematica*", in *Proc*. 14th Telecomm. forum TELFOR, 2006, pp. 588-591.
[28] Wolfram, S., The *Mathematica* Book, 5th Edition, *Cambridge University Press*, Cambridge, UK, 2003.
[29] Nagel, L. "SPICE2: A computer program to simulate semiconductor circuits," *Memorandum No. M520, University of California*, Berkeley, CA, USA, 1975.
[30] MATLAB Version 7, *MathWorks*, Inc., Natick, MA, USA, 2005.
[31] Kolundžija, B. M., Ognjanović, J. S., Sarkar, T. K., Šumić, D.S., Paramentić, M.M., Janić, B. B., Olćan, D.I., Tošić, D. V., Tasić, M. S., "WIPL-D Microwave: Circuit and 3D EM Simulation for RF & Microwave Applications," *Artech House*, Norwood, MA, USA, 2005.
[32] Mitra, S., "Digital Signal processing, A Computer Based Approach," *McGraw-Hill*, New York, USA, 2006.
[33] Liu, S., Hwang, Y., "Dual-Input Differentiators and Integrators with Tunable Time Constants Using Current Conveyors," *IEEE Trans. Instrumentation and Measurement*, 1994, pp. 650-654.
[34] Lutovac, M. D., Tošić, D. V., Huelsman, L. P. (editor), "DRAWFILT - Drawing Filter Realizations in MATLAB," *IEEE Circuits & Devices*, 2001, pp. 3-4.
[35] Tošić, D. V., "SALEC version 2.8," a *Mathematica* package for symbolic analysis of linear electric circuits, *University of Belgrade*, School of Electrical Engineering, Belgrade, Serbia, 2006.

Dejan V. Tošić is a professor in the School of Electrical Engineering at the University of Belgrade, Serbia. He has focused his research on creating a framework for the symbolic analysis of circuits and systems that is suitable for research as well as industrial and educational applications. He is developing automation tools for optimizing the design and synthesis of analog and digital systems. He is coauthor of the book "Filter Design for Signal Processing Using MATLAB and *Mathematica*," published by Prentice-Hall in 2001, and *SchematicSolver*, a *Mathematica* application for mouse-driven interactive drawing of systems and for solving and implementing systems.

Miroslav D. Lutovac (SM'99) is a professor at the University of Belgrade, Serbia. His research interests include the theory and implementation of analog and digital signal processing, and the symbolic analysis and synthesis of multiplierless and multirate digital systems. He is coauthor of the book "Filter Design for Signal Processing Using MATLAB and *Mathematica*," published by Prentice-Hall in 2001, and *SchematicSolver*, a *Mathematica* application for mouse-driven interactive drawing of systems and for solving and implementing systems.