Design, Analysis, Validation, and Reporting of Continuous-Time Systems Using CAS

Miroslav Lutovac, Maja Lutovac-Banduka and
Aleksandra Pavlović

May 27, 2019

Design, Analysis, Validation, and Reporting of Continuous-Time Systems Using CAS

Miroslav Lutovac, *Senior Member, IEEE*, Maja Lutovac-Banduka, and Aleksandra Pavlović

***Abstract*—The automated symbolic manipulations are presented according to user preferences in such a way that all representations (mathematical, graphical, as net-list, software code, or time-domain and frequency domain responses) are obtained from the same visual system description using computer algebra system (CAS) as add-ons for extending software environment. The paper is devoted to researchers and scientist using basic electrical engineering tasks, so that time-consuming tasks are automated in software and the properties of the systems can be discovered, and optional conditions or discovered properties can be used for synthesis, verifications, simulations, and optimization with real parameters. All derived properties are available as closed-form relations, which can help the faster design of robust systems.**

***Index Terms* — Computer algebra systems, electrical engineering, graphical user interface, symbolic processing.**

I. Introduction

IN the process of designing contemporary electric systems usually we start with system specification and deal with system modeling, analysis, software and hardware synthesis and implementations, simulation, verification, tolerance centering, fault detections, design space optimization, and user interface (more details are available in [1] and [2]). We can use numeric or symbolic software environments to design and implement electronic systems. The symbolic software environment as CAS (Computer Algebra System) [3] is the best initial step because expressions from textbooks or published papers can be simply redrawn or rewritten in CAS environment. The programed functions appear in the same traditional forms as in textbooks. Also, the system represented by formulas can be transformed into realizable program or hardware chips, the hardware realization can be transformed into programs and expressions. It is easily to go between hardware descriptions, software implementations, and traditional mathematical expressions. Optimization, verification, and testing the accuracy are available at each design step between the system specification and implementation.

The unified hardware/software approach is using symbolic processing for more than a decade for the design, analysis, verification, and synthesis ([1], [4], and [5]). In spite of the superiority of the symbolic processing, the prevalent software for the design, analysis, verification, and synthesis is still numeric processing.

The purpose of this paper is to exemplify the benefits of symbolic approach. This paper is the third paper in series on this topic: the first part [6] is presenting the development process of GUI (Graphical User Interface) in Mathematica using modern trends in visual programing and some new functions of Mathematica, following the main features of Wolfram language; the second paper [7] is presenting derivation of mathematical representations from schematic description in a form of frequency responses, or as a set of equations and software for the system synthesis. Instead of using textual description of the system, the visual programming technique is used.

This paper starts with derived expressions in [7] and explains analysis and optimization steps.

It should be noticed that the usage of some of the popular graphical programming packages for teaching electrical engineering was presented in [8]. The importance of the visualization of programing in robotics was explained in [9].

II. GUI in CAS environment

GUI in CAS environment is used for visual programing in such a way that the system specification is generated as lists of element specifications by clicking the appropriate buttons. New element can be added into the lists as new function that has all drawing command for all possible node positions. One, two, and three port elements can be added into schematic by clicking only one or two points on the drawing table (see Figures 1, 2, and 3 in [6]). The distinctive feature of the GUI is that the third node is automatically added assuming the most frequently used element position. The default scale is automatically chosen so that element looks as in the most textbooks. The connection lines between the element and the circuit nodes are the minimal. Tooltip function displays label as a tooltip while the mouse pointer is in the area where the element is drawn; label contains element name and coordinates of all ports.

It is possible to draw smaller elements by specifying the element scale property, but not larger than that allowed by element node positions. For testing all positions of the multi-port elements, a special test-code was used with all possible positions in the drawing table.

For different irregular element positions, when mouse is

Miroslav Lutovac is with The School of Electrical and Computer Engineering of Applied Studies Belgrade, 283 Vojvode Stepe, 11000 Belgrade, Serbia, (e-mail: mlutovac@viser.edu.rs).

Maja Lutovac-Banduka is with RT-RK Institute for Computer Based Systems, Novi Sad, Serbia (e-mail: majalutovac@yahoo.com).

Aleksandra Pavlović is with The School of Electrical and Computer Engineering of Applied Studies Belgrade, 283 Vojvode Stepe, 11000 Belgrade, Serbia, (e-mail: aleksandrap@viser.edu.rs).

over that element in the drawing table, the corresponding tooltip shows what is wrong.

III. TRANSFORMATION OF SCHEMATICS WITH MULTI-PORT ELEMENTS INTO SCHEMATIC WITH BASIC ELEMENTS

In [7], it is demonstrate the transformation of multi-port elements into a combination of connected two-port elements, and automated generation of system of equations that can be solved using the basic Ohm's law and Kirchhoff's current and voltage laws. As an example schematic is used the schematic that is successfully implemented in many applications as robust solution; that is the general-purpose second order KHN filter section (Kerwin-Huelsman-Newcomb). The well-known schematic is modified so that a new opamp is added for implementing transfer function zero. The new properties of the modified filter section are not documented in the textbooks, and the designer should solve the system. The GUI is used for visual programing (actually for drawing the schematic), the multi-port elements are replaced with standard two-port elements, and the system of equation is automatically derived for setting up circuit equations.

The derivation of the transfer function, of filter section presented in Figure 1, can be a serious problem even for experienced designers when the amplification *A* is of finite value or given as an expression. The whole procedure is explained in [7]. A part of the example schematic specification is as follows:

```
mySchematic = {
   {"VoltageSource",{{3,7},{3,9}},vin,""},...
   {"Resistor",{{14,11},{17,11}},r5,""}, ...
   {"Capacitor",{{17,13},{21,13}},c6,""},...
   {"OpAmplifier",{{17,9},{17,11},{21,11}},A,a3},...
   {"Node",{21,11},VB,"",TextOffset→{1,-1}},...
   {"Node",{17,11},V17p11,"",TextOffset→{1,1}},...
   {"Node",{17,9},V17p9,"",TextOffset→{1,-1}},...,
   {"GroundE",{17,9},0,"",ElementSize→{1.5,1.5}},...
   {"Output",{35,11},Yx,"",ElementSize→{2.5,2.5}},...
   {"Line",{{17,11},{17,13}}},
   {"Line",{{21,11},{21,13}}},...
};
```

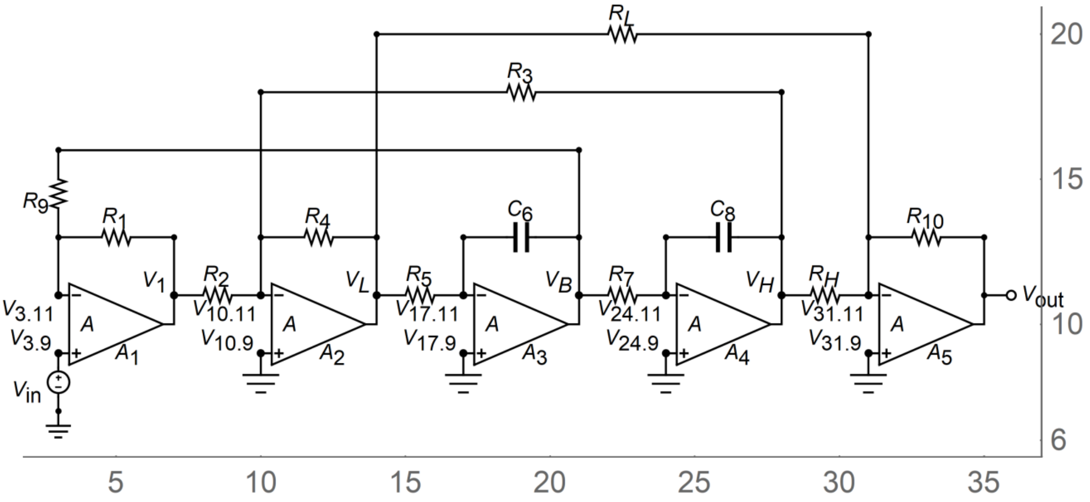

Fig. 1. Modification of KHN filters section.

	Coefficients of transfer function denominator
s^0	1
s^1	$\frac{C_8 R_7 \left(R_1 R_4 \left(A^3 R_3 + R_2 + R_3\right) + (A+1) R_1 R_2 R_3 + (A+1) R_9 \left((A+1) R_2 R_3 + (R_2 + R_3) R_4\right)\right) + C_6 R_5 (R_2 (A R_3 + R_3 + R_4) + R_3 R_4)(A R_9 + R_1 + R_9)}{R_9 \left(R_2 \left(A^3 R_4 + A R_3 + R_3 + R_4\right) + R_3 R_4\right) + ((A-1) A + 1) R_1 (R_2 + R_3) R_4 + R_1 R_2 R_3}$
s^2	$\frac{(A+1) C_6 C_8 R_5 R_7 (R_2 (A R_3 + R_3 + R_4) + R_3 R_4)(A R_9 + R_1 + R_9)}{R_9 (R_2 (A^3 R_4 + A R_3 + R_3 + R_4) + R_3 R_4) + ((A-1) A + 1) R_1 (R_2 + R_3) R_4 + R_1 R_2 R_3}$

	Coefficients of transfer function numerator
s^0	1
s^1	0
s^2	$\frac{C_6 C_8 R_5 R_7 R_L}{R_H}$

Fig. 2. Derived coefficients of modified KHN filters section.

After transformation of the schematic elements into two-port elements, we can use the basic Ohm's law and Kirchhoff's current and voltage laws for setting up circuit equations. For each resistor or capacitor element we can use generalized Ohm's law. For each node that is a connection to at least two elements, we can setup equations using Kirchhoff's current law. For the voltage sources between two nodes we can use Kirchhoff's voltage law. The system of equations consists of 27 equations with 27 variables. The final results cannot be presented in viewable format. The transfer function coefficients are presented in Figure 2.

Assuming that amplification of the amplifiers is infinitive, A→∞, the simplified transfer function of the second-order function is obtained using limiting value of the expression when *A* approaches ∞. The simplified transfer function can be used for the filter synthesis. In Figure 3 we have used the traditional type of displaying the derived result.

$$\frac{R_3\, R_4\, (R_1 + R_9)\, R_{10}\, \left(C_6\, C_8\, R_5\, R_7\, s^2\, R_L + R_H\right)}{R_H\, R_L\, \left(C_6\, C_8\, R_2\, R_3\, R_5\, R_7\, R_9\, s^2 + C_8\, R_1\, R_3\, R_4\, R_7\, s + R_2\, R_4\, R_9\right)}$$

Fig. 3. The derived transfer function of KHN filter section for A→∞.

IV. Determining design constrains

In this section we have demonstrated how the initial schematic and the automated deriving of the transfer functions help us to choose the most preferred values for implementation.

For known transfer function and the circuit that has to implement it, we can prepare synthesis procedure. In this case the transfer function that we have to implement can be low-pass H_{LP} or low-pass-notch H_{LPN} with four symbolic parameters K_0, ω_z, ω_p, Q_p:

$$H_{LP} = K_0 \frac{s^2}{s^2 + \frac{\omega_p}{Q_p} + + \omega_p^2} \qquad (1)$$

$$H_{LPN} = K_0 \frac{s^2 + \omega_z^2}{s^2 + \frac{\omega_p}{Q_p} + + \omega_p^2} \qquad (2)$$

For the 6th order filter, on disposal are 12 values that we have to determine to implement the filter section with the transfer function. Most values can be arbitrary chosen. As the first attempt, let us choose R_3, R_4, R_9, and R_H as variables that we can determine from the filter transfer function. All other values we would like to specify arbitrary. From the schematic we can determine relations between four parameters K_0, ω_z, ω_p, Q_p and four resistances. For example,

$$R_3 = R_2 \frac{-1 + C_8 Q_p R_7 \omega_p}{C_8 Q_p R_7 \omega_p} \qquad (3)$$

The value of the resistance R_3 can be negative in the case when *Q* factor is too low:

$$C_8 Q_p R_7 \omega_p < 1 \qquad (4)$$

The lowest *Q* factor for elliptic-type filters is 0.5. Therefore we can determine constrains for the resistor R_7:

$$R_7 > \frac{2}{C_8 \omega_p} \qquad (5)$$

The design procedure is as follows assuming that R_x and C_x are from the set of the most preferred values for the implementation:

$$\begin{aligned} &R_x > \frac{2}{C_x \omega_p} \\ &R_1 = R_2 = R_5 = R_7 = R_{10} = R_L = R_x \\ &C_6 = C_8 = C_x \end{aligned} \qquad (6)$$

The next step is to determine the design procedure. The design procedure means that we have to automatically derive a code for computing all filter values. Again, comparing the transfer function in terms of filter parameters (K_0, ω_z, ω_p, Q_p) and the transfer function in terms of resistances and capacitance, we can obtain all implementation values as a code in the Wolfram language [1].

V. Design of higher-order filters using known schematic

Once the circuit is well analyzed and design procedure is available, we can use as embedded solution in higher order systems.

Suppose we have to implement low-pass filter that satisfies the following conditions: the passband edge frequency F_p=3000 Hz, the stopband edge frequency F_s=4500 Hz, the maximal passband variation A_p=0.2 dB, the minimal stopband attenuation A_s=40 dB.

The minimal filter order of the most efficient elliptic filter is 5. Since we are planning to use the same second-order filter section, we will choose the sixth order approximation. Thera are many solutions from the design space [1]. We choose the minimal Q factor elliptic filter [1] with the selectivity factor 1.635, and the frequency normalization 1.1 (move the passband edge to the transition region). This approximation was robust as digital filters [1], and we are expecting that the analog filter can be also robust to many parasitic effects and imperfections.

The filter parameters are for three sections:

	H_3	H_2	H_1
ω_p^2	4.8016 10^8	4.8016 10^8	4.8016 10^8
Q_p	4.24831	1.13453	0.557171
ω_z^2	9.48906 10^9	1.40637 10^9	8.29491 10^8

Since all ω_p are the same, we can choose the referent capacitance C_x=10 nF. The minimal value from design constrains is $R_{x,min}$= 9127 Ω: we chose the large value R_x=10 kΩ. All three sections have the same value for resistances R_1=R_2 =R_5=R_7=R_{10}=R_L =10 kΩ and capacitances C_6= C_8=10 nF. Other resistances are computed using design procedure for known filter parameters:

	H_3	H_2	H_1
R_3 [Ω]	8930	5980	1810
R_4 [Ω]	42900	28700	8690
R_9 [Ω]	83500	14900	2210
R_H [Ω]	82900	141000	949000

In this section we have demonstrated how the initial schematic, the automated deriving of the transfer functions from the schematic description, and derived design procedure, can be used to design higher order filter as cascaded connection of initial second-order filter sections.

VI. Verification of the Magnitude Response of Designed Circuit

In the verification step, we would like to obtain the same magnitude response with computed resistances and capacitances as the taken approximation. In the derived transfer function in terms of resistances and capacitances we are replacing computed values that are within 0.1% tolerances. The magnitude responses of the theoretical and implemented filter are presented in the Figures 4 and 5.

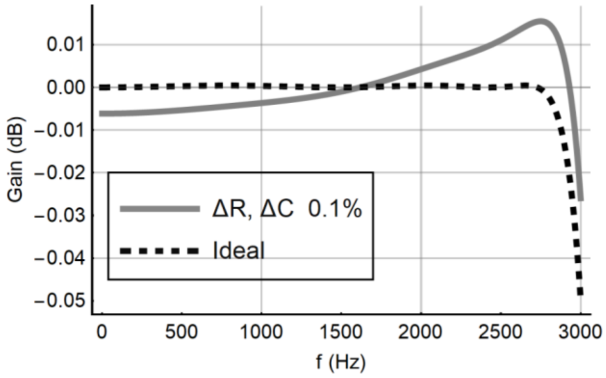

Fig. 4. Pass-band magnitude response of theoretical and implemented filter.

The pass-band variation is within required specification and significantly smaller than 0.2 dB. The minimum stopband attenuation is approximately 40 dB.

VII. Study of Imperfection

The purpose of the study of imperfection is to identify the critical components that can harm the specification and regular operation. In this section we will consider the finite gain of the amplifier as an example. The magnitude responses of the theoretical and implemented filter are presented in the Figures 6 and 7.

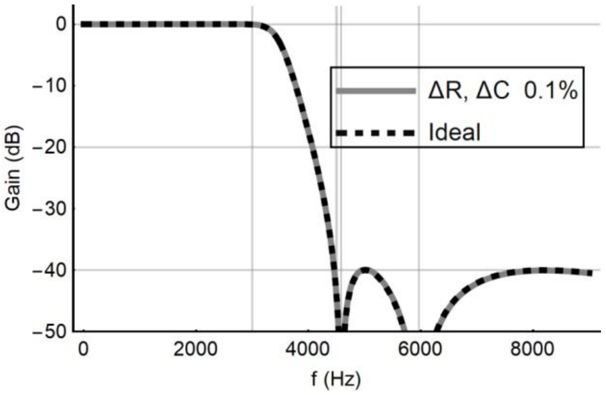

Fig. 5. Stop-band magnitude response of theoretical and implemented filter

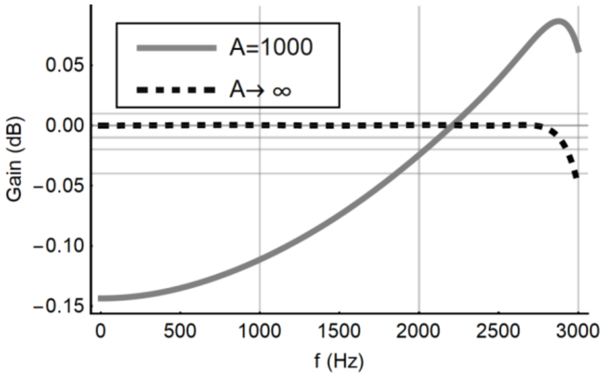

Fig. 6. Pass-band magnitude response of theoretical and implemented filter for A=1000.

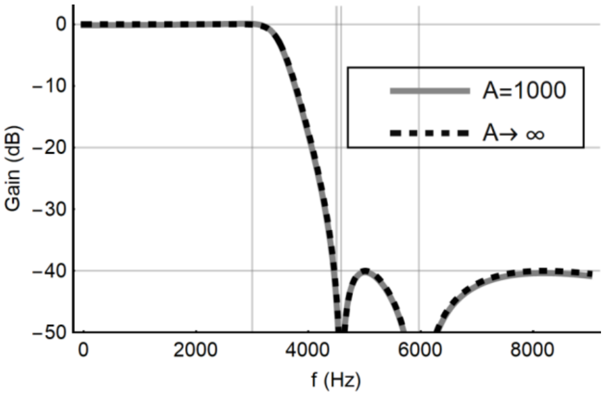

Fig. 7. Stop-band magnitude response of theoretical and implemented filter for A=1000.

The pass-band variation is slightly larger than the required specification 0.2 dB. The minimum stopband attenuation is approximately 40 dB. This implies that for implementation is required the opamp with larger gain then 1000.

VIII. Optimization and tuning procedure

The purpose of the optimization step is to optimize some values for the optimal values using different approaches. One possibility is to find referent resistance R_x for minimal gain-sensitivity product. Since all expressions are available as closed-form relations, it is simple to derive all sensitivity functions and to find the minimal value using procedures from [1]. In practice, many elements can be modeled using manufacturer instructions, and that models can be used to replace ideal values. For example, the gain of the amplifier can be modeled as single pole function that is frequency dependence. Replacing this model into derived transfer function will lead to increasing the order of the transfer function. This is not serious problem when we are using CAS, because all substitutions are in closed form and in the overall response we can identify the expression of the second-order section. From that expression we can discover the influence of R_3, R_4, R_9, and R_H as variables on four parameters K_0, ω_z, ω_p, Q_p.

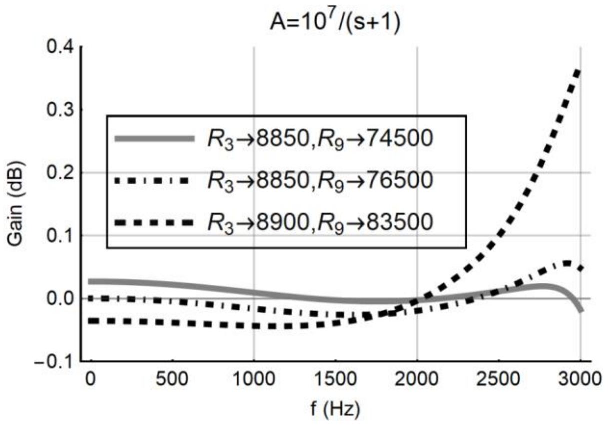

Fig. 8. Pass-band magnitude response of implemented filter for different values of R_3 and R_9.

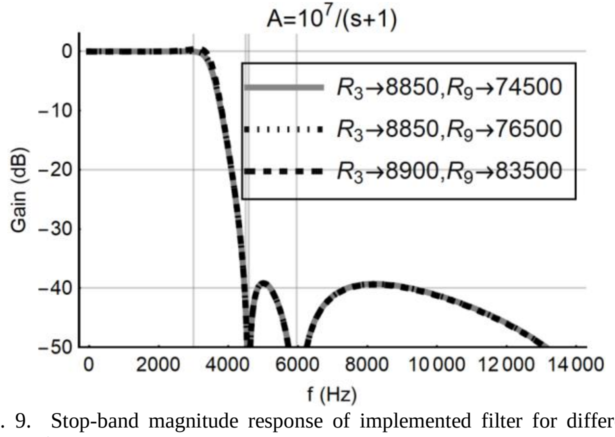

Fig. 9. Stop-band magnitude response of implemented filter for different values of R_3 and R_9.

Those resistances can be used for discovering tuning procedure. As an example, the magnitude responses of the implemented filters for several values of R_3 and R_9 are presented in the Figures 8 and 9.

After tuning R_3 and R_9, the pass-band variation is within required specification 0.2 dB. The minimum stopband attenuation is slightly smaller than 40 dB. This implies that the optimization or tuning procedure is possible by computing the new values of several resistances (for example, R_3 and R_9).

IX. Conclusion

In this paper, we have developed an environment and graphical user interface (GUI) so that the system description can be generated with the small number of tasks. The software is not like usual canned applications and it is possible to combine system description with many other specific user targets such as manipulate with symbolic expressions and analyze complex non-ideal element and system models.

Acknowledgment

This work is supported in part by the Ministry of Education and Science of Serbia under Grant TR 32023

References

[1] M. Lutovac, D. Tosic, and B. Evans, *Filter Design for Signal Processing Using MATLAB and Mathematica*, Upper Saddle River, Prentice Hall, 2001.

[2] B. Kleinjohann, G. R. Gao, H. Kopetz, L. Kleinjohann, and A. Rettberg, *Design Methods Applications Distributed Embedded Systems*, IFIP 18th World Computer Congress, Springer Publishing Company, 2013.

[3] S. Wolfram, *An Elementary Introduction to the Wolfram Language*, Champaign, Wolfram Media, IL 2015.

[4] K. Strehl, *Symbolic Methods Applied to Formal Verification and Synthesis in Embedded Systems Design*, Swiss Federal Institute of Technology Zurich, 2000.

[5] B. L. Evans and J. H. McClellan, “Symbolic Analysis of Signals and Systems,” in *Symbolic and Knowledge-Based Signal Processing*, (eds. A. V. Oppenheim and S. H. Nawab), Prentice Hall, pp. 88-141, 1992.

[6] M. D. Lutovac, V. Mladenović, and M. Lutovac-Banduka , “Graphical User Interface for Electrical Engineering Systems using Wolfram Language,“ TELFOR, Belgrade, Serbia, pp. 1-4, 22-23 Nov. 2016.

[7] M. D. Lutovac, M. Lutovac-Banduka, and V. Mladenović, “Environment and Graphical User Interface for Design of Continuous-Time Systems,“ IcETRAN, Palić, Serbia, pp. 802-805, 11-14 Jun. 2018.

[8] E. Lunca, S. Ursache, and O. Neacsu, “Graphical Programming Tools for Electrical Engineering Higher Education,” *International Journal of Online Engineering*, vol. 7, no. 1, pp. 19-24, iJOE, 2011.

[9] M. Lutovac-Banduka, “Robotics First – A Mobile Environment for Robotics Education,” *International Journal of Engineering Education*, vol. 32, no. 2A, pp. 818–829, 2016.