Educational Software for Solving Electric Circuits with *Combinatorica*

Dejan V. Tošić

***Abstract*—An original educational software *CircuitSolver* is presented, which is a set of carefully designed *Mathematica* functions that perform a step-by-step solving of electrical circuits from the student's point of view. The circuits are assumed to be finite, lumped, linear, and time invariant. The full symbolic analysis and formulation of the circuit equations are considered. A circuit is represented by the *Combinatorica* graph structure, which is used for the graph-theoretic formulation of the circuit equations. The new software is intended for educators and students who teach, explore, or evaluate electric circuits.**

***Index Terms*—CircuitSolver, Combinatorica, computer algebra system, electric circuit, Mathematica, symbolic analysis**

I. Introduction

Electric Circuit Theory (ECT) is an important subject in engineering education because grasping concepts and phenomena from circuit theory is important in understanding electric power systems, electronics, telecommunication and control theory, and microwave engineering. One of the important aims of a circuit theory educator is to help students acquire a "functional understanding" of the subject . [1]

If educators wish to generate interest in ECT among the incoming engineering students, they should do a better job of promoting the profession by providing better teaching tools, and delivery methods, combining circuit theory courses with laboratory exercise. [2,3]

Mastering circuit analysis requires some degree of practice. One must grasp the idea of what, how, and even why something needs to be calculated, and yet be adept in algebraic manipulation. Often, the burden of algebraic manipulation causes the student to lose sight of the wood from the trees. [4]

New software tools that are available nowadays have dramatically influenced the possibilities of educational technology and play an important role in teaching Electric Circuit Theory. However, this does not simply suggest an improvement of hitherto used teaching methods: in many instances it is necessary to rebuild these methods from the ground up. [5]

Traditionally, computers were developed and used to expedite numerical calculations. A newer, and in the long run, very fruitful field is the manipulation of symbolic expressions. When these symbolic expressions represent mathematical entities, this field is generally called computer algebra. Today, with computer algebra systems (CAS), such as *Mathematica* [6,7], it is possible to calculate in minutes or hours the results that would (and did) years to accomplish by paper and pencil [8].

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 [9-12]. Therefore, CAS and symbolic techniques can help students acquire a "functional understanding" of Electric Circuit Theory and foster mastery of basic methodological skills in the study of electrical circuits.

This paper presents new software developed in *Mathematica* for automated computer-aided solving of finite, lumped, linear, and time invariant electric circuits. This software, referred to as *CircuitSolver*, makes use of *Combinatorica* [13] to represent and visualize circuits. *CircuitSolver* is targeted at students and educators having in mind contemporary trends in teaching ECT [14,15].

II. CircuitSolver

CircuitSolver is designed to be used as an instructional tool that is intended to help engineering students: (a) understand the conventions used for defining the arbitrary signs of algebraic currents and voltages, (b) learn how to proceed methodically in order to write a correct, complete and non-redundant set of circuit equations, and (c) acquire efficient skills in identifying circuit variables, equations, branches, nodes and loops.

From the educational viewpoint, *CircuitSolver* is a set of functions that guide the student step-by-step in the process of formulating and solving the circuit equations. A brief description of each function follows.

A circuit is specified and pictorially represented by the *Combinatorica* graph data structure as a named symbolic object, which contains all information required to analyze, solve and draw the circuit. The *Combinatorica* function **`ShowGraph`** draws the circuit.

Circuit netlist is extracted from the graph data structure by the *CircuitSolver* function **`CircuitNetlist`**.

The corresponding circuit variables, associated to the element ports, are obtained by the *CircuitSolver* function

This work was supported in part by the Serbian Ministry of Science and Technological Development under Grant TR-11021.

Dejan V. Tošić is with the University of Belgrade, School of Electrical Engineering, Bulevar kralja Aleksandra 73, PO Box 35-54, 11120 Belgrade, SERBIA (phone: ++381.11.337.007.4; fax: ++381.11.324.86.81; e-mail: tosic@etf.rs; http://kondor.etf.rs/~tosic/).

CircuitVariables.

The element equations are generated by the *CircuitSolver* function **CircuitElementEquations**.

The incidence matrix of the circuit is computed by the *CircuitSolver* function **CircuitIncidenceMatrix**.

The KCL equations (Kirchhoff's current law) are formulated by the *CircuitSolver* function **CircuitKCL**.

The circuit tree is obtained by the *CircuitSolver* function **CircuitTree**.

CircuitFundametalCycles is a *CircuitSolver* function that constructs fundamental loops (also referred to as f-cycles).

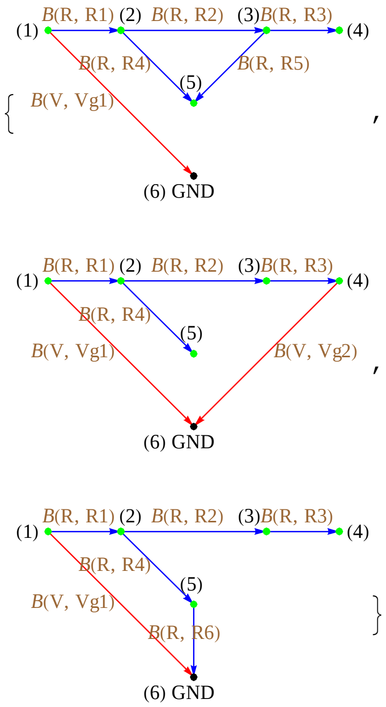
The loop matrix of the circuit can be computed by the function **CircuitFundamentalCycleMatrix**.

The KVL equations (Kirchhoff's voltage law) are formulated by the *CircuitSolver* function **CircuitKVL**.

The circuit response is computed by the *CircuitSolver* function **CircuitResponse**.

Here is an example of finding f-cycles (see Fig. 1):

```
ShowGraph /@
 First[CircuitFundamentalCycles[circuit, 1]]
```



III. Example of CircuitSolver Operation

Consider a model electric circuit shown in Fig. 1.

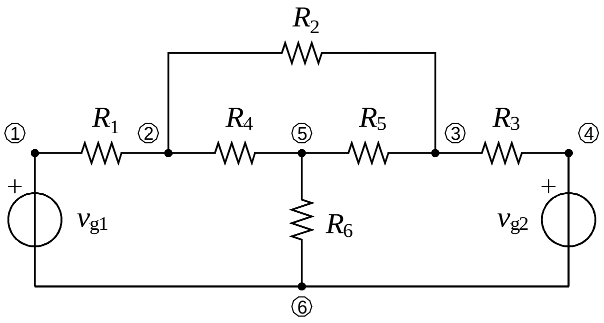


Fig. 1. Model electric circuit.

The simplest way to specify the model circuit is (a) to identify its nodes and (b) to specify each element by an ordered pair of nodes and a unique label; the label is required to distinguish elements connected in parallel. This set of elements is often referred to as the netlist.

CircuitSolver representation of an electric circuit is the *Combinatorica* graph structure as follows:

```
circuit =
 Graph[
  {{{1, 6}, EdgeColor → Red,
     EdgeLabel → B["V", "Vg1"], EdgeWeight → Vg1,
     EdgeLabelPosition → UpperLeft},
    {{4, 6}, EdgeColor → Red,
     EdgeLabel → B["V", "Vg2"], EdgeWeight → Vg2,
     EdgeLabelPosition → UpperRight},
    {{1, 2}, EdgeColor → Blue,
     EdgeLabel → B["R", "R1"], EdgeWeight → R1,
     EdgeLabelPosition → {0.1, 0.1}},
    {{2, 3}, EdgeColor → Blue,
     EdgeLabel → B["R", "R2"], EdgeWeight → R2,
     EdgeLabelPosition → {0.1, 0.1}},
    {{3, 4}, EdgeColor → Blue,
     EdgeLabel → B["R", "R3"], EdgeWeight → R3,
     EdgeLabelPosition → {0.1, 0.1}},
    {{2, 5}, EdgeColor → Blue,
     EdgeLabel → B["R", "R4"],
     EdgeLabelPosition → UpperLeft,
     EdgeWeight → R4},
    {{3, 5}, EdgeColor → Blue,
     EdgeLabel → B["R", "R5"],
     EdgeLabelPosition → UpperRight,
     EdgeWeight → R5},
    {{5, 6}, EdgeColor → Blue,
     EdgeLabel → B["R", "R6"], EdgeWeight → R6,
     EdgeLabelPosition → {0.09, 0.05}}},
  {{{-2, 1}, VertexLabel → "(1)",
     VertexLabelPosition → {-0.03, 0}},
    {{-1, 1}, VertexLabel → "(2)",
     VertexLabelPosition → {0.07, 0.1}},
    {{1, 1}, VertexLabel → "(3)",
     VertexLabelPosition → {-0.02, 0.1}},
    {{2, 1}, VertexLabel → "(4)",
     VertexLabelPosition → {0.1, 0}},
    {{0, 0}, VertexLabel → "(5)",
     VertexLabelPosition → {0.03, 0.13}},
    {{0, -1}, VertexLabel → "(6) GND",
     VertexColor → Black,
     VertexLabelPosition → {0.07, -0.1}}},
  EdgeLabelColor → Brown, VertexColor → Green]
```

Figure 2 shows another pictorial representation of the model circuit, which captures all structural characteristics of the circuit. Circuit element symbols are replaced by colored arrows and the element labels are represented as *B*(*T*,*L*) where *T* represents the type (R-Resitor, V-Voltage source, etc.) and *L* denotes a unique label. The arrows represent the assumed references for voltage and current, the positive voltage at the "upstream" end of the arrow and the current reference in the direction of the arrow.

In general, if current is indicated as flowing from the "high" (+) voltage terminal to the "low" (–) voltage terminal of an element, the current conducted by the element and the voltage developed across the element to cause this flow of current are said to be in associated reference polarity.

```
circuit // ShowGraph[#, BaseStyle → {14, Bold}] &
```

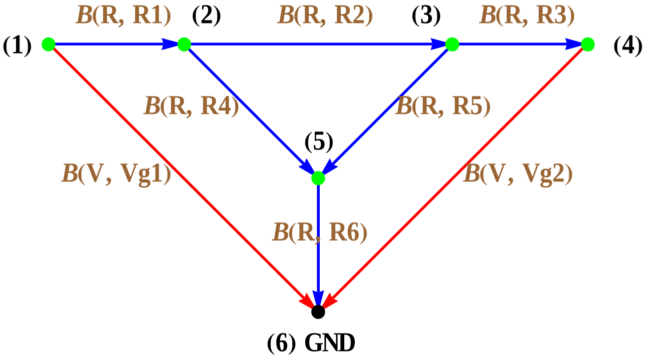


Fig. 2. *Combinatorica* pictorial representation of the model circuit.

CircuitResponse is the key *CircuitSolver* function that performs symbolic circuit analysis, which is the process of producing an expression that describes a certain behavioral aspect of the circuit with one, some, or all the circuit elements represented as symbols. In this particular example, all circuit parameters are kept as symbols or symbolic expressions.

Assuming that (a) node 4 is the reference node, (b) branch 1 is in the circuit tree, (c) **v** is the voltage variable symbol, and (d) **i** is the current variable symbol, the circuit response is computed as shown in Fig. 3.

The circuit response can be typeset in the traditional form by the *CircuitSolver* function **CircuitTypeset**, which is the last function applied to the response (Fig. 3).

```
r = CircuitResponse[circuit, 1, 6, v, i]; r // Simplify // CircuitTypeset
```

$$i_{\{1,2\}\mathrm{R1}} \to \frac{R_2 \left(R_3 V_{\mathrm{g1}}+R_5 V_{\mathrm{g1}}+R_6 V_{\mathrm{g1}}-R_6 V_{\mathrm{g2}}\right)+R_3 V_{\mathrm{g1}} (R_4+R_5)+\left(V_{\mathrm{g1}}-V_{\mathrm{g2}}\right) (R_4 (R_5+R_6)+R_5 R_6)}{R_1 (R_2 (R_3+R_5+R_6)+R_3 (R_4+R_5)+R_4 R_5+R_4 R_6+R_5 R_6)+R_2 (R_3 (R_4+R_6)+R_4 (R_5+R_6)+R_5 R_6)+R_3 (R_4 (R_5+R_6)+R_5 R_6)}$$

$$i_{\{1,6\}\mathrm{Vg1}} \to -\frac{R_2 \left(R_3 V_{\mathrm{g1}}+R_5 V_{\mathrm{g1}}+R_6 V_{\mathrm{g1}}-R_6 V_{\mathrm{g2}}\right)+R_3 V_{\mathrm{g1}} (R_4+R_5)+\left(V_{\mathrm{g1}}-V_{\mathrm{g2}}\right) (R_4 (R_5+R_6)+R_5 R_6)}{R_1 (R_2 (R_3+R_5+R_6)+R_3 (R_4+R_5)+R_4 R_5+R_4 R_6+R_5 R_6)+R_2 (R_3 (R_4+R_6)+R_4 (R_5+R_6)+R_5 R_6)+R_3 (R_4 (R_5+R_6)+R_5 R_6)}$$

$$i_{\{2,3\}\mathrm{R2}} \to \frac{R_5 \left(-R_1 V_{\mathrm{g2}}+R_6 V_{\mathrm{g1}}-R_6 V_{\mathrm{g2}}\right)+R_3 R_4 V_{\mathrm{g1}}+R_4 (R_5+R_6) \left(V_{\mathrm{g1}}-V_{\mathrm{g2}}\right)}{R_1 (R_2 (R_3+R_5+R_6)+R_3 (R_4+R_5)+R_4 R_5+R_4 R_6+R_5 R_6)+R_2 (R_3 (R_4+R_6)+R_4 (R_5+R_6)+R_5 R_6)+R_3 (R_4 (R_5+R_6)+R_5 R_6)}$$

$$i_{\{2,5\}\mathrm{R4}} \to \frac{R_5 \left(R_1 V_{\mathrm{g2}}+R_3 V_{\mathrm{g1}}\right)+R_2 \left(R_3 V_{\mathrm{g1}}+R_5 V_{\mathrm{g1}}+R_6 V_{\mathrm{g1}}-R_6 V_{\mathrm{g2}}\right)}{R_1 (R_2 (R_3+R_5+R_6)+R_3 (R_4+R_5)+R_4 R_5+R_4 R_6+R_5 R_6)+R_2 (R_3 (R_4+R_6)+R_4 (R_5+R_6)+R_5 R_6)+R_3 (R_4 (R_5+R_6)+R_5 R_6)}$$

$$i_{\{3,4\}\mathrm{R3}} \to \frac{R_4 \left(-V_{\mathrm{g2}} (R_1+R_2)+R_5 \left(V_{\mathrm{g1}}-V_{\mathrm{g2}}\right)+R_6 \left(V_{\mathrm{g1}}-V_{\mathrm{g2}}\right)\right)+(R_2+R_5) \left(R_6 \left(V_{\mathrm{g1}}-V_{\mathrm{g2}}\right)-R_1 V_{\mathrm{g2}}\right)}{R_1 (R_2 (R_3+R_5+R_6)+R_3 (R_4+R_5)+R_4 R_5+R_4 R_6+R_5 R_6)+R_2 (R_3 (R_4+R_6)+R_4 (R_5+R_6)+R_5 R_6)+R_3 (R_4 (R_5+R_6)+R_5 R_6)}$$

$$i_{\{3,5\}\mathrm{R5}} \to \frac{R_2 \left(V_{\mathrm{g2}} (R_1+R_4)+R_6 \left(V_{\mathrm{g2}}-V_{\mathrm{g1}}\right)\right)+R_1 R_4 V_{\mathrm{g2}}+R_3 R_4 V_{\mathrm{g1}}}{R_1 (R_2 (R_3+R_5+R_6)+R_3 (R_4+R_5)+R_4 R_5+R_4 R_6+R_5 R_6)+R_2 (R_3 (R_4+R_6)+R_4 (R_5+R_6)+R_5 R_6)+R_3 (R_4 (R_5+R_6)+R_5 R_6)}$$

$$i_{\{4,6\}\mathrm{Vg2}} \to \frac{R_4 \left(-V_{\mathrm{g2}} (R_1+R_2)+R_5 \left(V_{\mathrm{g1}}-V_{\mathrm{g2}}\right)+R_6 \left(V_{\mathrm{g1}}-V_{\mathrm{g2}}\right)\right)+(R_2+R_5) \left(R_6 \left(V_{\mathrm{g1}}-V_{\mathrm{g2}}\right)-R_1 V_{\mathrm{g2}}\right)}{R_1 (R_2 (R_3+R_5+R_6)+R_3 (R_4+R_5)+R_4 R_5+R_4 R_6+R_5 R_6)+R_2 (R_3 (R_4+R_6)+R_4 (R_5+R_6)+R_5 R_6)+R_3 (R_4 (R_5+R_6)+R_5 R_6)}$$

$$i_{\{5,6\}\mathrm{R6}} \to \frac{R_2 \left(V_{\mathrm{g2}} (R_1+R_4)+R_3 V_{\mathrm{g1}}+R_5 V_{\mathrm{g1}}\right)+(R_4+R_5) \left(R_1 V_{\mathrm{g2}}+R_3 V_{\mathrm{g1}}\right)}{R_1 (R_2 (R_3+R_5+R_6)+R_3 (R_4+R_5)+R_4 R_5+R_4 R_6+R_5 R_6)+R_2 (R_3 (R_4+R_6)+R_4 (R_5+R_6)+R_5 R_6)+R_3 (R_4 (R_5+R_6)+R_5 R_6)}$$

$$v_{\{1,2\}\mathrm{R1}} \to \frac{R_1 \left(R_2 \left(R_3 V_{\mathrm{g1}}+R_5 V_{\mathrm{g1}}+R_6 V_{\mathrm{g1}}-R_6 V_{\mathrm{g2}}\right)+R_3 V_{\mathrm{g1}} (R_4+R_5)+\left(V_{\mathrm{g1}}-V_{\mathrm{g2}}\right) (R_4 (R_5+R_6)+R_5 R_6)\right)}{R_1 (R_2 (R_3+R_5+R_6)+R_3 (R_4+R_5)+R_4 R_5+R_4 R_6+R_5 R_6)+R_2 (R_3 (R_4+R_6)+R_4 (R_5+R_6)+R_5 R_6)+R_3 (R_4 (R_5+R_6)+R_5 R_6)}$$

$$v_{\{1,6\}\mathrm{Vg1}} \to V_{\mathrm{g1}}$$

$$v_{\{2,3\}\mathrm{R2}} \to \frac{R_2 \left(R_5 \left(-R_1 V_{\mathrm{g2}}+R_6 V_{\mathrm{g1}}-R_6 V_{\mathrm{g2}}\right)+R_3 R_4 V_{\mathrm{g1}}+R_4 (R_5+R_6) \left(V_{\mathrm{g1}}-V_{\mathrm{g2}}\right)\right)}{R_1 (R_2 (R_3+R_5+R_6)+R_3 (R_4+R_5)+R_4 R_5+R_4 R_6+R_5 R_6)+R_2 (R_3 (R_4+R_6)+R_4 (R_5+R_6)+R_5 R_6)+R_3 (R_4 (R_5+R_6)+R_5 R_6)}$$

$$v_{\{2,5\}\mathrm{R4}} \to \frac{R_4 \left(R_5 \left(R_1 V_{\mathrm{g2}}+R_3 V_{\mathrm{g1}}\right)+R_2 \left(R_3 V_{\mathrm{g1}}+R_5 V_{\mathrm{g1}}+R_6 V_{\mathrm{g1}}-R_6 V_{\mathrm{g2}}\right)\right)}{R_1 (R_2 (R_3+R_5+R_6)+R_3 (R_4+R_5)+R_4 R_5+R_4 R_6+R_5 R_6)+R_2 (R_3 (R_4+R_6)+R_4 (R_5+R_6)+R_5 R_6)+R_3 (R_4 (R_5+R_6)+R_5 R_6)}$$

$$v_{\{3,4\}\mathrm{R3}} \to \frac{R_3 \left(R_4 \left(-V_{\mathrm{g2}} (R_1+R_2)+R_5 \left(V_{\mathrm{g1}}-V_{\mathrm{g2}}\right)+R_6 \left(V_{\mathrm{g1}}-V_{\mathrm{g2}}\right)\right)+(R_2+R_5) \left(R_6 \left(V_{\mathrm{g1}}-V_{\mathrm{g2}}\right)-R_1 V_{\mathrm{g2}}\right)\right)}{R_1 (R_2 (R_3+R_5+R_6)+R_3 (R_4+R_5)+R_4 R_5+R_4 R_6+R_5 R_6)+R_2 (R_3 (R_4+R_6)+R_4 (R_5+R_6)+R_5 R_6)+R_3 (R_4 (R_5+R_6)+R_5 R_6)}$$

$$v_{\{3,5\}\mathrm{R5}} \to \frac{R_5 \left(R_2 \left(V_{\mathrm{g2}} (R_1+R_4)+R_6 \left(V_{\mathrm{g2}}-V_{\mathrm{g1}}\right)\right)+R_1 R_4 V_{\mathrm{g2}}+R_3 R_4 V_{\mathrm{g1}}\right)}{R_1 (R_2 (R_3+R_5+R_6)+R_3 (R_4+R_5)+R_4 R_5+R_4 R_6+R_5 R_6)+R_2 (R_3 (R_4+R_6)+R_4 (R_5+R_6)+R_5 R_6)+R_3 (R_4 (R_5+R_6)+R_5 R_6)}$$

$$v_{\{4,6\}\mathrm{Vg2}} \to V_{\mathrm{g2}}$$

$$v_{\{5,6\}\mathrm{R6}} \to \frac{R_6 \left(R_2 \left(V_{\mathrm{g2}} (R_1+R_4)+R_3 V_{\mathrm{g1}}+R_5 V_{\mathrm{g1}}\right)+(R_4+R_5) \left(R_1 V_{\mathrm{g2}}+R_3 V_{\mathrm{g1}}\right)\right)}{R_1 (R_2 (R_3+R_5+R_6)+R_3 (R_4+R_5)+R_4 R_5+R_4 R_6+R_5 R_6)+R_2 (R_3 (R_4+R_6)+R_4 (R_5+R_6)+R_5 R_6)+R_3 (R_4 (R_5+R_6)+R_5 R_6)}$$

Fig. 3. Response of the model electric circuit generated by *CircuitSolver*.

IV. Conclusion

Mastering concepts and phenomena from electrical circuit theory is very important in understanding electrical power systems, electronics, telecommunication and control theory, and microwave engineering. If educators wish to generate interest in this field among the incoming engineering students, they should do a better job of promoting the profession by providing better teaching tools, and delivery methods. New software tools that are available nowadays have dramatically influenced the possibilities of educational technology and play an important role in teaching Electric Circuit Theory.

It has been observed that engineering students are usually somewhat confused by the circuit theory conventions used for defining the arbitrary signs of algebraic currents and voltages. In addition, they lack any methodology and do not have rigorous understanding of the concepts of node and loop, so they often come up with wrong equations or an insufficient or redundant set of equations when they try to solve an electrical circuit.

In this paper a new teaching concept has been presented, based on *Mathematica* – a computer algebra system, in which a circuit is solved pedagogically from the student's viewpoint. Original software *CircuitSolver* has been presented. *CircuitSolver* is targeted at students or educators seeking a computer assisted learning in mastering Electric Circuit Theory and clarifying its concepts and procedures.

References

[1] A.-K. Carstensenb and J. Bernharda, "Critical aspects for learning in an electric circuit theory course – an example of applying learning theory and design-based educational research in developing engineering education," In *Proc. 1st International Conference on Research in Engineering Education (ICREE)*, Honolulu, USA, June 22–24, 2007, pp. CD-ROM.

[2] Lj. Trajković, "Teaching Circuit Courses to Engineering Students," in *Proc. CAS Education Workshop*, Seattle, USA, May 22, 2008, pp. CD-ROM.

[3] Y. Tsividis, "Turning students on to circuits," *IEEE Solid-State Circuits Newsletter*, pp. 6–9, Jan. 2008.

[4] L. R. J. Costa, M. Honkala, and A. Lehtovuori, "Applying the Problem-Based Learning Approach to Teach Elementary Circuit Analysis," *IEEE Transactions on Education*, vol. 50, no. 1, pp. 41–48, Feb. 2007.

[5] D. Mayer and J. Kus, "Teaching Electric Circuit Theory with the Help of Tutorial Software," *Global Journal of Engineering Education*, vol. 2, no. 1, pp. 77–85, 1998.

[6] S. Wolfram, *The Mathematica Book*. Cambridge, MA: Cambridge University Press, Wolfram Media, 2003.

[7] *Mathematica*, http://www.wolfram.com/, Wolfram Research, Inc., USA, Version 7.0.1 released Feb. 18, 2009.

[8] M. Trott, *The Mathematica GuideBook for Symbolics*. New York, NY: Springer Science+Business Media, 2006.

[9] M. D. Lutovac and D. V. Tošić, *SchematicSolver*, [online] http://www.wolfram.com/products/applications/schematicsolver/ Version 2.2 (for Mathematica 7) released August 18, 2009.

[10] D. V. Tošić and M. D. Lutovac, "Symbolic Simulation of Engineering Systems," In *Proc. 4th IEEE European Conference on Circuits and Systems for Communications* (ECCSC'08), Politehnica University, Bucharest, Romania, July 10-11, 2008, plenary lecture, pp. XIV-XXIII.

[11] M. D. Lutovac and D. V. Tošić, "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, vol. 79, no. 11, pp. 1368–1381, Nov. 2006.

[12] M. D. Lutovac, D. V. Tošić, and B. L. Evans, *Filter Design for Signal Processing using MATLAB and Mathematica*. Upper Saddle River, NJ: Prentice Hall, 2001. Translated in Chinese, Publishing House of Electronics Industry, PHEI, Beijing, P. R. China 2004.

[13] S. Skiena and S. Pemmaraju, *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. New York: NY, Cambridge University Press, 2003.

[14] A. Agarwal and J. H. Lang, *Foundations of Analog and Digital Electronic Circuits*. San Francisco: CA, Morgan Kaufmann, Elsevier, 2005.

[15] C. K. Alexander and M. N. O. Sadiku, *Fundamentals of Electric Circuits*. New York: NY, McGraw-Hill, 2001.