ANALOG CIRCUIT ANALYSIS BASED ON SEMI-STATE VARIABLE APPROACH VIA MATLAB

RODICA VOICULESCU, LAVINIA IORDACHE (BOBARU), M. IORDACHE
University Politehnica of Bucharest
E-mail: voiculescu.rodica@gmail.com, laviniabobaru@yahoo.com, mihai.iordache@upb.ro

Abstract. ***An improving efficient algorithm for generation of the semi-state equations in full symbolic, partly symbolic or numeric form is presented. This algorithm has been implemented in a program – called SSEMG – Semi-State Equation Matrix Generation. The program allows the formulation of the semi-state matrices in full symbolic, partly symbolic or numeric form. Using the semi-state equations in which the parameters associated to the nonlinear elements are considered as symbols, we obtain an improving of the accuracy in the numerical calculations. The semi-state equations can be integrated by a Matlab routine. Using the semi-state equations in operational form (for the linear and/or piecewise-linear nonlinear analog circuits) the transfer functions in full symbolic, partly symbolic or numeric form are generated. An illustrative example is given to prove that SSEMG program is a very useful tool for symbolic analysis and design of linear and/or nonlinear time-invariant analog circuits.***

Keywords: Semi - State Equations, Symbolic Analysis, Analog Circuit, Nonlinear Circuit, Computer - Aided Design.

1. INTRODUCTION

The nodal approach in circuit analysis has two variants, [1, 2, 7, 17]:
1. The first - using companion circuits (resistive discrete circuits associated with an integration implicit algorithm), that gets the dynamic circuit response by the analysis of a sequence of resistive circuits. In the modified nodal analysis (MNA) the circuit matrix is obtained by augmentation of the nodal conductance (admittance) matrix corresponding to the non-NA-compatible circuit elements with additional rows and columns;
2. The second one - using the circuit equations in dynamic behavior obtained by KCL and the constitutive equations of the circuit elements. The dynamic behavior description of the circuit by modified nodal equations is known as the semi-state variable method, and uses those variables which are most convenient from the point of view of the analysis results.

The independent variables of both methods above are: $n - 1$ node voltages that make up the voltage vector $\boldsymbol{v}_{n-1}$; m branch currents, which cannot be expressed in respect to the node voltages or to the first order derivatives of the node voltages and the circuit parameters, which make up the vector of controlling currents $\boldsymbol{i}_m$.

The vector $\boldsymbol{i}_m$ contains the currents of the independent voltage sources, the currents of voltage sources controlled both in current or in voltage, the controlling currents of the current-controlled voltage sources (CCVSs) and of the current-controlled current sources (CCCSs), the currents of the current-controlled nonlinear resistors, the currents of the current-controlled nonlinear inductors, and the linear inductor (magnetic coupled or not) currents.

Because numerical differentiation is a relatively inaccurate operation, we approximate the q_k - v_k(v_k - q_k) curve of each voltage-controlled (charge-controlled) nonlinear capacitor and the $\varphi_k - i_k$ $(i_k - \varphi_k)$ curve of each current-controlled (flux-controlled) nonlinear inductor by piecewise-linear segments, [1-15, 17]. In order to simplify the description of the nonlinear resistors, their v - i (v - i) curves will be approximated by piecewise-linear continuous curves, or by new curves in which the nonlinearities are transferred to the sources. [16].

Using the semi-state equations in which the parameters associated to the nonlinear elements are considered as symbols, we obtain an improving of the accuracy in the numerical calculations. In the case of the linear time-invariant analog circuits, the semi-state matrix in symbolic form can be farther used to obtain a simplified expression of any circuit function, preserving a "cluster" of poles and zeroes [1 - 11, 17].

In this paper we propose an improving algorithm to automatically formulate the semi-state equations in full symbolic, partly symbolic or numeric normal-form**,** for large time-invariant analog circuits. Starting from the circuit description in the netlist form, our method is based on Kirchhoff's laws, constitutive equations of the circuit elements, and it uses the facilities of symbolic simulator Maple. The state equations are integrated by a routine from MATLAB package.

Using the semi-state equations in operational form (for the linear and/or piecewise-linear nonlinear analog circuits) the transfer functions in full symbolic, partly symbolic or numeric form are generated. An illustrative example is given to prove that SSEMG program is a very useful tool for symbolic analysis and design of linear and/or nonlinear time-invariant analog circuits [3].

2. AUTOMATIC SETTIG-UP OF SEMI-STATE EQUATIONS FOR ANALOG CIRCUITS

In dynamic behavior, the inductor (capacitor) equations contain current (voltage) derivatives. Consequently, the modified nodal equations must contain derivatives of the independent variables. Their general form is, [1 – 11,17]:

$$\begin{cases} \boldsymbol{W}\dot{\boldsymbol{x}}(t)+\boldsymbol{G}\boldsymbol{x}(t)=\boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t)=\boldsymbol{L}^{\mathrm{t}}\boldsymbol{x}(t) \end{cases} \qquad (1)$$

where:

$$\boldsymbol{x}(t)=\begin{bmatrix} \boldsymbol{v}_{n-1} \\ \boldsymbol{i}_m \end{bmatrix} \qquad (2)$$

is the circuit independent variable vector, with initial condition $\boldsymbol{x}_0=\boldsymbol{x}(t_0)$;

$$\dot{\boldsymbol{x}}(t)=\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} \boldsymbol{v}_{n-1} \\ \boldsymbol{i}_m \end{bmatrix} \qquad (3)$$

is the derivative of the independent variable vector;
$\boldsymbol{W}$ and $\boldsymbol{G}$ are square matrices having $(n-1+m)\times(n-1+m)$ dimension;

$$\boldsymbol{u}=\begin{bmatrix} \boldsymbol{i}_{sc,n-1} \\ \boldsymbol{e}_m \end{bmatrix} \qquad (4)$$

is the input vector, with $\boldsymbol{i}_{sc,n-1}$ - the vector of the short-circuit currents, and $\boldsymbol{e}_m$ is the electromotive force vector of the ideal independent voltage sources, and of the sources resulted from the nonlinear characteristics of the nonlinear current-controlled elements which are piece-wise linearized.

$\boldsymbol{B}$ and $\boldsymbol{L}$ are selector matrices, with entries (–1, 0 or 1), and the superscript "t" denotes the transpose.

For the linear and/or piecewise-linear nonlinear analog circuits, the semi-state equations in frequency domain and with zero initial conditions have the following structure [3 – 11, 17]:

$$\begin{cases} (\boldsymbol{G}+s\boldsymbol{W})\boldsymbol{X}(s)=\boldsymbol{B}\boldsymbol{U}(s) \\ \boldsymbol{Y}(s)=\boldsymbol{L}^{\mathrm{t}}\boldsymbol{X}(s) \end{cases} \qquad (5)$$

For the case of nonlinear elements, the semi-state equations of the circuit become

$$\begin{cases} \boldsymbol{W}(\boldsymbol{x}(t))\cdot\dot{\boldsymbol{x}}(t)+\boldsymbol{G}\boldsymbol{x}(t)+\boldsymbol{F}(\boldsymbol{x}(t))=\boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t)=\boldsymbol{L}^{\mathrm{t}}\boldsymbol{x}(t) \end{cases} \qquad (6)$$

where $\boldsymbol{F}(\boldsymbol{x}(t))$ is a nonlinear function of $\boldsymbol{x}$.

If for the nonlinear inductors (nonlinear capacitors) the magnetic fluxes (electrical charges) are considered as independent variables, the matrix $\boldsymbol{W}$ is independent of $\boldsymbol{x}$.
Using the definition of the transfer function in matrix form, [4.]:

$$\boldsymbol{H}(s)=\frac{\boldsymbol{Y}(s)}{\boldsymbol{U}(s)}, \qquad (7)$$

and tacking into account the operational equations (6), we get the following expression of the transfer function matrix:

$$\boldsymbol{H}(s)=\frac{\boldsymbol{Y}(s)}{\boldsymbol{U}(s)}=\boldsymbol{L}^{\mathrm{t}}(\boldsymbol{G}+s\boldsymbol{W})^{-1}\boldsymbol{B}. \qquad (8)$$

Multiplying both sides of the first equation in the system (5) by $\boldsymbol{G}^{-1}$ (the matrix $\boldsymbol{G}$ being a nonsingular matrix), we can write:

$$\begin{cases} (\boldsymbol{I}-s\boldsymbol{P})\boldsymbol{X}(s)=\boldsymbol{R}\boldsymbol{U}(s) \\ \boldsymbol{Y}(s)=\boldsymbol{L}^{\mathrm{t}}\boldsymbol{X}(s) \end{cases} \qquad (9)$$

where

$$\boldsymbol{P}=-\boldsymbol{G}^{-1}\boldsymbol{W},\ \boldsymbol{R}=\boldsymbol{G}^{-1}\boldsymbol{B}. \qquad (10)$$

From (9), the network function $\boldsymbol{H}(s)$ takes the form:

$$\boldsymbol{H}(s)=\frac{\boldsymbol{Y}(s)}{\boldsymbol{U}(s)}=\boldsymbol{L}^{\mathrm{t}}(\boldsymbol{I}-s\boldsymbol{P})^{-1}\boldsymbol{R}. \qquad (11)$$

Because the expression (11) needs the inversion of two matrices, $\boldsymbol{G}$ and ($\boldsymbol{I}$ – s$\boldsymbol{P}$), the formula (8) is more advantageous.

A more efficient procedure to generate network functions in matrix form, based on the semi-state equations in frequency domain (5), consists in the following steps, [17]:
Step 1: Starting from the input file of the netlist type (in which the circuit can contain multiple inputs) the semi-state equations in time-domain are generated;
Step 2: Considering null initial conditions, the semi-state equations in time domain are transformed in the frequency domain (using Laplace transform);
Step 3: The algebraic equations obtained in the previous step are solved in respect of the Laplace transforms of the semi-state variables. In the case of large-scale analog circuits we can use the reduction algorithm of the semi-state equation number [41];
Step 4: The suitable output variables are expressed in respect of the inputs and of the complex variable *s*;
Step 5: Taking into account the definitions, all network functions are performed.

Remarks

- For MIMO systems the most efficient procedure to generate the transfer functions is the procedure based on the direct solving, either of the state equations or of the semi-state equations, in the complex frequency domain, because these techniques do not inverse any matrix. Because of this, they allow a large number of symbols that is, without doubt, a great advantage for the automatic design;

- In general, the dimension of (***sI*** - ***A***) matrix is much smaller than the dimensions of (***sW*** + G) and (***I*** - ***sP***) matrices, but it is less sparse.

The semi-state variable method can be applied to a large class of nonlinear circuits containing both linear and nonlinear non-monotonic resistors, inductors and capacitors, independent voltage and current sources, the four types of linear controlled sources, time-variable resistors (as models for the switches) and any type of excess elements. It has a lot of advantages, [3, 17]:

- the semi-state equations are simply formulated from the netlist;
- it does not impose restrictions to the nonlinear characteristics;
- it does not require an associated discrete equivalent circuit which needs an additional computing effort, especially when the numeric integration is made with changeable time step;
- the matrices ***W*** and ***G*** are sparse, consequently, they require just a little space of memory in a computer implementation; this compensates the disadvantage to operate with a large number of variables;
- by simply cancellation of the matrix ***W*** and by solving an algebraic system of equations, the DC analysis can be performed; this can be necessary for finding initial conditions of the dynamic behaviors;
- the method can be applied for nonlinear and/or switching circuits;
- for the nonlinear circuit analysis, the method has the great advantage that allows keeping as symbols only the parameters of the nonlinear elements; so the elements of ***W*** and ***G*** matrices, depending on the linear circuit elements, are computed only once, at the beginning of the iteration procedure.

The above method to systematically formulate the semi-state equations in symbolic, form for linear and/or nonlinear time-invariant analog circuits with excess elements was presented in detail in [1 - 11, 17].

3. SSEMG SYSEG PROGRAM DESCRIPTION

SSEMG – **S**emi – **S**tate **E**quation **M**atrix **G**eneration program was developed (Figure 1) in order to formulate the semi-state equations in symbolic**,** numeric-symbolic or numeric normal-form for a large class of linear or nonlinear analog circuits, [3]. These circuits may contain both linear and nonlinear resistors, inductors, and capacitors, independent voltage and current sources, and the four types of linear controlled sources. SSEMG is written in C++ language and it is implemented on the compatible IBM PC. It is an improved software environment that allows the analysis and design of linear or nonlinear analog circuits, even when they have excess elements. The SSEMG generates the semi-state matrices (***W***, ***G*** and ***B***) in symbolic**,** numeric-symbolic or numeric form.

This is an interactive tool that combines symbolic and numeric computational techniques, and which uses the facilities of symbolic simulator Maple to manipulate the symbolic expressions.

The SSEMG principle window is presented in Figure 1.

The SSEMG program has two variants:
1. SSEMG_maple_e – generates the semi-state matrices in the symbolic and numeric form for the Maple symbolic simulator;
2. SSEMG_matlab_e – generates the semi-state matrices in the symbolic and numeric form for the Matlab simulator.

In order to understand easy the use of the SSEMG program we consider, as example, the circuit shown in Figure 2.

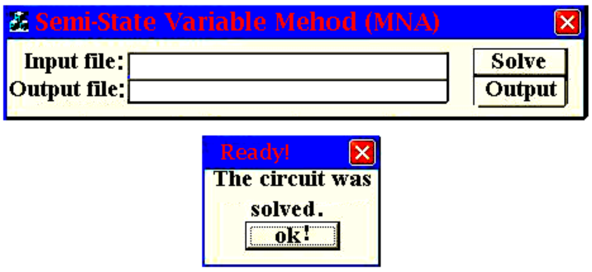

Figure 1. The SSEMG principle window.

In order to understand better the SSEMG program we consider, as example, the circuit shown in Figure 2.

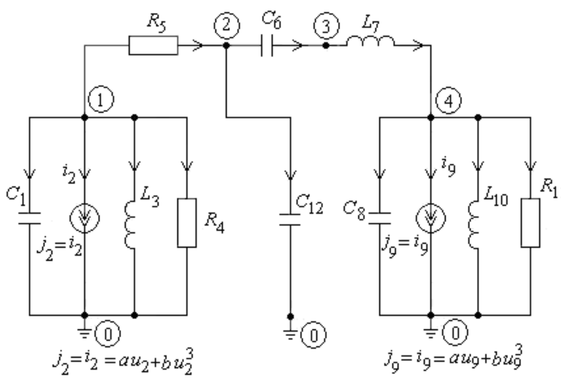

Figure 2. Two parallel resonant circuits coupled through a two-port passive network.

The two voltage-controlled nonlinear resistors R_{dvc2} and R_{dvc9}, describing by the two polynoms of the third degree, are substituted by two ideal independent current sources. After opening the main window, the input file ***ex7_1.txt*** which describes the topology of the circuit in Figure 2 is generated. The input ***ex7_1.txt*** has the following structure*:*

```
C1 1 0 1.0e-11          C8 4 0 2.0e-11
```

I2 1 0 1.0
L3 1 0 2.8095e-09
R4 1 0 3500.0
R5 1 2 113.5
C6 2 3 2.53e-12
L7 3 4 10.60786e-09
I9 4 0 1.0
L10 4 0 1.2678e-09
R11 4 0 3500.0
C12 2 0 6.5e-12
.END

On the next step, call the ***SSEMG_maple_e.exe*** program and introduce the input file ***ex7_1.txt*** and the output file ***ex7_1_maple.txt*** (see Figure 3);

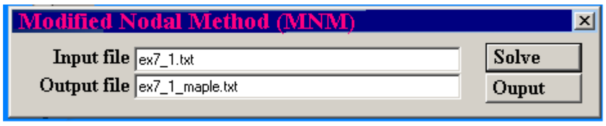

Figure 3. Introduction of the input file *ex7_1.txt* and the output file *ex7_1_maple.txt.*

By pressing "**Solve**" button, after a small time, the message shown in Figure 4 appears;

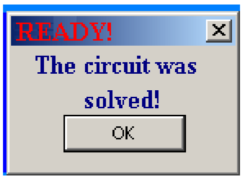

Figure 4. The message which appears when we press on the "Solve" button.

After running ***GMESS_maple_e*** program, the following ouput files are generated:

1. lit_ex7_1_out.txt;
2. num_ex7_1_out.txt;

To generate the ***W, G*** and ***B*** semi-state matrices in a form according to the Maple program, we call the ***Maple_lit.exe*** program (Figure 5).

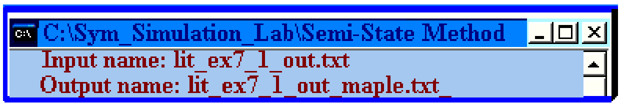

Figure 5. Calling the *Maple_lit.exe* program.

This program gives the ***lit_ex7_1_out_maple.txt*** output file, which has the structure:

W:=matrix([
[C1, 0, 0, 0, 0, 0, 0],
[0, C6+C12, -C6, 0, 0, 0, 0],
[0, -C6, C6, 0, 0, 0, 0],
[0, 0, 0, C8, 0, 0, 0],
[0, 0, 0, 0, -L3, 0, 0],
[0, 0, 0, 0, 0, -L7, 0],
[0, 0, 0, 0, 0, 0, -L10]]);

G:=matrix([
[G4+G5, -G5, 0, 0, 1, 0, 0],
[-G5, G5, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 1, 0],
[0, 0, 0, G11, 0, -1, 1],
[1, 0, 0, 0, 0, 0, 0],
[0, 0, 1, -1, 0, 0, 0],
[0, 0, 0, 1, 0, 0, 0]]);

B:=matrix([[-I2],[0],[0],[-I9],[0],[0],[0]]);

If we call the ***GMESS_matlab_e*** program and we introduce the ***ex7_1.txt*** input file and the ***ex7_1_matlab.txt*** output file, then it is called the ***GMESS_matlab_e*** program with the ***lit_ex7_1_matlab.txt*** input file. Finally, the ***Matlab_lit.exe*** program gives the ***lit_ex7_1_matlab_out.txt*** output file**.** This output file has the following structure:

W=[C1, 0, 0, 0, 0, 0, 0;
0, C6+C12,-C6, 0, 0, 0, 0;
0, -C6, C6, 0, 0, 0, 0;
0, 0, 0, C8, 0, 0, 0;
0, 0, 0, 0, -L3, 0, 0;
0, 0, 0, 0, 0, -L7, 0;
0, 0, 0, 0, 0, 0, -L10];

G=[G4+G5, -G5, 0, 0, 1, 0, 0;
-G5, G5, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 1, 0;
0, 0, 0, G11, 0, -1, 1;
1, 0, 0, 0, 0, 0, 0;
0, 0, 1, -1, 0, 0, 0;
0, 0, 0, 1, 0, 0, 0];

B=[-I2; 0; 0; -I9; 0; 0; 0]

Because the ***G*** matrix is a non-singular matrix, the semi-state equations of the circuit in Figure 2 can be written in the following matrix form:

$$\dot{x} = \boldsymbol{P}^{-1}(\boldsymbol{x}+\boldsymbol{R})\boldsymbol{P} = -\boldsymbol{G}^{-1}\boldsymbol{W},\quad \boldsymbol{R} = \boldsymbol{G}^{-1}\boldsymbol{B} \qquad (12)$$

For the numeric values of the circuit parameters (see Figure 2): C_1=10 nF, L_3=2.8095nH, R_4=3500 Ω, R_5=113.5 Ω, L_7=10.60786 nH, C_8=20 nF, L_{10}=1.2678 nH, R_{11}=3500 Ω, C_{12}=6.5 nF, a=–0.0085 S, and b=0.00071 S/V$_2$, and the initial conditions $u_{C1}(0)$=2 V and $u_{C8}(0)$=2 V, to integrate the equations (12), by a Matlab integration routine, to obtain the results presented in Figures 5 and 6.

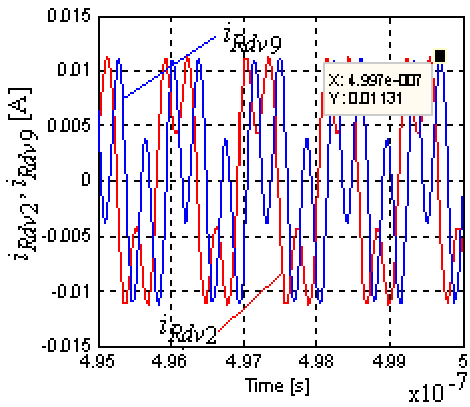

Figure 6. The currents i_2 - i_{Rdv2} and i_9 - i_{Rdv9} vs time.

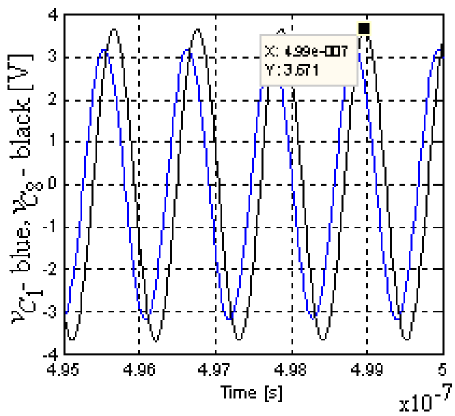

Figure 7. The voltages v_{C1} and v_{C8} vs time.

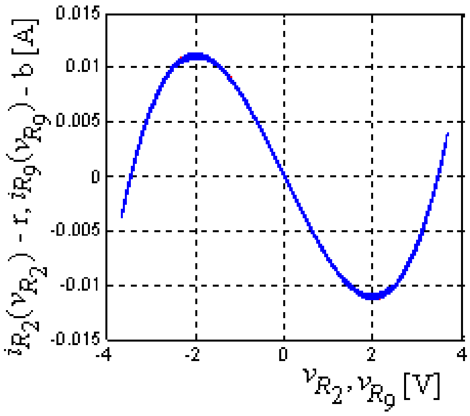

Figure 8. The characteristics $i_{R2} = f(v_{R2})$ and $i_{R9} = f(v_{R9})$.

4. CONCLUSIONS

The paper presents an improving efficient algorithm for generation of the semi-state equations in full symbolic, partly symbolic or numeric form. This algorithm has been implemented in a program – called ***SSEMG –*** *S*emi-*S*tate *E*quation *M*atrix *G*eneration.

The program allows the formulation of the semi-state matrices in full symbolic, partly symbolic or numeric form**.** Using the semi-state equations in which the parameters associated to the nonlinear elements are considered as symbols, we obtain an improving of the accuracy in the numerical calculations. The semi-state equations can be integrated by a Matlab routine.

Using the semi-state equations in operational form (for the linear and/or piecewise-linear nonlinear analog circuits) the transfer functions in full symbolic, partly symbolic or numeric form are generated.

Circuits may contain both linear and nonlinear elements such as: resistors, inductors or capacitors, independent voltage and current sources, linear controlled voltage and current sources.

Starting from the circuit description in the netlist form, our method is based on Kirchhoff's laws, constitutive equations of the circuit elements, and it uses the facilities of symbolic simulator Maple. The state equations are integrated by a routine from MATLAB package.

5. REFERENCES

[1] L. O. Chua, and P. M. Lin, Computer-aided analysis of electronic circuits, Englewood Cliffs, N. J.: Prentice Hall, Inc., 1975.

[2] A. E. Schwarz, Computer-aided design of microelectronic circuits and systems, London: Academic Press, 1987.

[3] M. Iordache, Lucia Dumitriu, I. Matei, SSEMG – Semi - State Equation Matrix Generation, User Guide, Electrical Department Library, Politehnica University of Bucharest, 2009.

[4] L. O. Chua, and R. A. Rohrer, "On the Dynamic Equations of a Class of Nonlinear RLC Networks", IEEE Trans. Circuit Theory, Vol. CT-12, N0. 4, Dec. 1965, pp. 475-489.

[5] T. E. Stern, "On the Equations of Nonlinear Networks", ", IEEE Trans. Circuit Theory, Vol. CT-13, N0. 1, Mar. 1961, pp. 74-81.

[6] R. A. Rohrer, Circuit Theory: An Introduction to the State Variable Approach, McGraw-Hill, New York, 1970.

[7] W. J. McCalla, Fundamentals of computer-aided circuit simulation, Kluwer Academic Publishers, Boston, 1988.

[8] M. Iordache, Lucia Dumitriu, "Efficient Decomposition Techniques for Symbolic Analysis of Large – Scale Analog Circuits by State Variable Method", Analog Integrated Circuits and Signal Processing, Kluwer, Vol. 40, No. 3, September 2004, Kluwer Academic Publishers, pp. 235-253.

[9] M. Iordache, Lucia Dumitriu, „Time Domain Diakoptic Analysis Based on Reduced-Order State Equations", International Journal of Bifurcation and Chaos (IJBC), Volume: 17, Issue: 10, October 2007, pp. 3625 – 3631

[10] M. Iordache, Lucia Dumitriu, Computer aided simulation of analog circuits, Politehnica Press, Bucharest, 2002 (in romanian).

[11] M. Iordache, Lucian Mandache, Computer-aided analysis of analogic nonlinear circuits, Politehnica Press, Bucharest, 2004 (in romanian).

[12] J. Vlach, J.M. Wojciechowski, and A. Opal, "Analysis of nonlinear networks with inconsistent initial conditions", Circuits and Systems I, IEEE Transactions on 42, 5, pp: 195–200, 1995.

[13] Mihai Iordache, Lucia Dumitriu, "A new Tearing Approach for Symbolic Computation of the Circuit Functions and of Large-Change Multiparameter Sensitivities", Proceedings of the 9th International Conference on Optimization of Electrical and Electronic Equipment, OPTIM'04, Brasov, May 20-21, 2004, Vol. 1, pp. 57-62

[14] H.G. Brachtendorf, "On consistent initial conditions for circuit's DAEs with higher index", Circuits and Systems I, IEEE Transactions on 48, 5, pp: 606–612, 2001.

[15] G. Reißig, H. Boche, and P. I. Barton, "On Inconsistent Initial Conditions for Linear Time-Invariant Differential-Algebraic Equations", IEEE Transaction on Circuits an Systems—I, 49, 11, pp: 1646–1648, 2002.

[16] Fl. Hantila, "A Method for Solving Nonlinear Resistive Networks", Rev. Roum. Sci. Techn. Électrotechn. et Énerg., 24, 2, 1979, p. 217-226.

[17] Mihai Iordache, Lucia Dumitriu, Generation of the Transfer Functions for MIMO Systems, Chapter 4; in book "Design of analog circuits through symbolic analysis, eneration of the transfer functions for Mimo system", Mourad Fakhfakh, Esteban Tlelo-Cuautle and Francisco V Fernandez (Eds), Bentham Science Publishers, 2012, pp. 83-114, pp. 228-262.

PSpice and Matlab for Elecrtonics, an Integrated Approach, 2002, Florida, USA. CRC Press. ISBN 0-8493-1263-9 (alk. paper).Â Long recognized and widely acclaimed as the classic introductory text in circuit analysis, this tenth edition represents over three decades of leadership in its field. Its clear and precise explanations, practical examples, and comprehensive, up-to-date coverage deliver a solid and complete foundation in a style that is both engaging and easy to understand. This book is ideally s...Â IEEE Transactions on Circuits and Systems â€" II: Analog and Digital Signal Processing, Vol. 47, No. 4, April 2000. â€" pp. 274-286. The design of the radio frequency (RF) section in a communication integrated circuit (IC) is a challenging problem. 5.6 State Variable Approach to RC Circuit. 5.7 State Variable Approach to an RLC Circuit Analysis. 5.8 State Variable Analysis of a Network. Â© 1999 CRC Press LLC. Chapter six AC analysis and network functions.Â variable = expression Expressions typed by the user are interpreted and immediately evaluated by the MATLAB system. If a MATLAB statement ends with a semicolon, MATLAB evaluates the statement but suppresses the display of the results. MATLAB is also capable of executing a number of commands that are stored in a file.