ELEKTRYKA 2013
Zeszyt 1 (225) Rok LIX

Marcin SOWA
Politechnika Śląska w Gliwicach

APPLICATION OF STAMPS IN THE AUTOMATIC FORMULATION OF CIRCUIT EQUATIONS – PART I. BASICS

Summary. The paper presents a general algorithm for the formulation of systems of equations for linear and nonlinear resistance circuits (with controlled sources). The subsequent steps of the algorithm are presented along with a general implementation idea for Matlab. The proposed algorithm can be used for educational and scientific purposes (in the future after its extension to a wider class of problems). This is the first part of the paper, in which the base ideas of the implementation are presented.

Keywords: automatic formulation, circuit equations, nonlinear circuits, stamps

ZASTOSOWANIE SZABLONÓW W AUTOMATYCZNYM FORMUŁOWANIU RÓWNAŃ OBWODU – CZĘŚĆ I: PODSTAWY

Streszczenie. W artykule przedstawiono algorytm do formułowania układów równań dla liniowych i nieliniowych obwodów rezystancyjnych (ze źródłami sterowanymi). Zaprezentowano kolejne etapy formułowania równań oraz ogólne wskazówki do implementacji w środowisku Matlab. Zaproponowany algorytm może służyć zarówno do celów dydaktycznych, jak i naukowych (w przyszłości, po jego rozbudowaniu do szerszej klasy problemów). Przedstawiona jest pierwsza część analizy, w której zaprezentowano podstawowe właściwości zaproponowanej implementacji.

Słowa kluczowe: automatyczne formułowanie, równania obwodu, obwody nieliniowe, szablony

1. INTRODUCTION

The formulation of circuit equations is a well known topic that was described thoroughly in Polish [2, 3, 7] and foreign literature [1, 6]. It is therefore not the author's intention to provide a solution for new problems in circuit theory but rather, for educational purposes, propose a general set of procedures, which together form an easily implementable and versatile algorithm for automatic formulation of circuit equations. Some of the presented procedures base on known methods and only vary in description (so that they may be then implemented in a chosen computer environment), while others are presented as simply implementable alternatives to existing methods.

This paper presents an algorithm for linear and nonlinear resistive circuits with controlled sources (because the mathematical relations are the same for complex numbers – it can be used also for linear AC circuits).

2. GENERAL CONSIDERATIONS

2.1. Circuit topology

A basic and probably most general presentation of a circuit topology is the one presented in [2], which is used e.g. in PSPICE [8]. It consists of information on:

- the type of an element,
- its connections (in most cases – two node identifiers),
- its parameters (e.g. applied voltage, value of the resistance, inductance or capacitance, element name – in case of a current controlled source).

Like in PSPICE, the presented algorithm also bases on a general assumption – the first letter of the element name defines its type. However, the author has chosen to use different notations in the paper, which for DC problems are:

- R – resistor,
- E – autonomous voltage source,
- e – voltage controlled voltage source,
- f – current controlled voltage source,
- J – autonomous current source,
- j – current controlled current source,
- i – voltage controlled current source,
- N – nonlinear element defined by $U(I)$ function,
- n – nonlinear element defined by $I(U)$ function.

2.2. Number of equations

Basing on a given topology the circuit equations are formulated. The following notations are used further on to describe certain integers:

- n – total number of unknown variables for which the system is solved for,
- n_{e} – number of two-terminal circuit elements,
- n_{n} – number of nodes, excluding the ground node,
- n_E – number of voltage source elements (autonomous or controlled),
- n_{N} – number of nonlinear elements.

The most basic way is to formulate equations basing on both KCL and KVL. However, this method yields a number of unknowns equal to twice the number of elements ($n = 2n_e$) as the unknowns are simply the circuit voltages and currents (this is known as STA – **S**parse **T**ableau **A**pproach [4]).

The nodal analysis allows to replace the voltages by potential differences – this yields $n = n_e + n_n$ equations – hence less redundant variables.

Currents of branches with a resistance representation (or in general – impedance representation) can be replaced by their respective node potential relations. Further on – the current source relations are also substituted if a constant or node potential dependency can be given. By applying this – one can obtain a system of $n = n_e + n_E + n_N$ unknowns. This is called MNA – the **M**odified **N**odal **A**nalysis [6].

3. MNA BASED ALGORITHM AND ITS IMPLEMENTATION

The chapter describes the subsequent procedures (in a proposed order) of an MNA based algorithm and general directions to its implementation in Matlab. First of, one needs to determine the presentation of the system. Generally, if considering a linear set of equations – they can be presented in a matrix form:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{1}$$

where $\boldsymbol{x}$ is the vector of unknown circuit variables, $\boldsymbol{A}$ is the system matrix representing the relations between the circuit variables, $\boldsymbol{b}$ is the vector of constants determined by autonomous sources. If the system of equations has nonlinear dependencies then it can be presented in the general form:

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}, \tag{2}$$

where $\boldsymbol{F}$ is a vector of functions dependent on the unknowns, $\boldsymbol{0}$ represents a vector of zeros, which has a length equal to n. A purely function-based representation is however not necessary – one can keep the conveniences that come from a matrix presentation of the linear part of the system by introducing the following general equation:

$$\boldsymbol{A}\boldsymbol{x} + \boldsymbol{F}_{\mathrm{nl}}(\boldsymbol{x}) = \boldsymbol{b}, \tag{3}$$

so that only the nonlinear dependencies are given in a form of a set of functions.

Preliminarily, whatever algorithm is chosen further on to reduce the number of unknowns, in most cases it is efficient for the matrix $\boldsymbol{A}$ and the function vector $\boldsymbol{F}_{\mathrm{nl}}$ to be implemented respectively as a sparse matrix and a sparse vector.

An efficient and transparent way of generating the system of equations is to define so called **stamps** [5] of individual elements i.e. the appropriate contributions of each element to the equation system.

It is a common choice for stamps to be presented by means of matrices [2, 5, 6] so that the complete matrix may be formulated by simply adding the stamp of each element. In this paper a different approach is presented, which allows for stamp representation through sparse vectors. The author also distinguishes two types of relations that contribute to a system matrix:

– current dependencies (which are later on added subject to KCL),
– potential dependencies (which e.g. define certain potential differences through voltage sources).

This, among many other advantages presented later on, allows to leave the element stamps for further processing. The system of equations is then formulated by adding up appropriate current stamps (called further on I-stamps) and considering voltage stamps (referred to later on as V-stamps).

3.1. Circuit topology definition through string cell

An algorithm is considered, where the circuit topology is presented through strings that define individual element connections. The considered implementation follows a certain set of patterns for each element:

– resistance R between nodes d and ϕ:

R<name> d ϕ R

– autonomous source with voltage increase from node ϕ to d:

E<name> d ϕ E

– voltage source from node ϕ to d controlled by voltage from node ϕ' to node d':

e<name> d ϕ β d' ϕ'

– voltage source from node ϕ to d controlled by current at the branch of element T<Tname> of type T:

f<name> d ϕ ρ T<Tname>

– autonomous current source enforcing a current from node ϕ to node d:

J<name> d ϕ J

– current source controlled by current at the branch of element T<Tname>:

j<name> d ϕ α T<Tname>

– current source controlled by voltage from node ϕ' to node d':

i<name> d ϕ γ d' ϕ'

– nonlinear element defined by the function $U(I)$, where the value of U is returned by the function `<funUI>`:

N<name> d ϕ <funUI>

– nonlinear element defined by the function $I(U)$, where the value of I is returned by the function `<funIU>`:

n<name> d ϕ <funIU>

Taking the above patterns into account, the exemplary circuit presented in Figure 1 (with the parameters: α=0.4, β=0.5, R_1 = 1kΩ, R_2 = 200 Ω, J = 0.02 A) can be given by the following string cell in Matlab:

```
topology={
       'R1  a    b    funcUI_1'
       'e1  b    c    0.5  a    d'
       'N1  c    d    @(x)[0.2*x^3,0.6*x^2]'
       'J1  d    b    2e-2'
       'n2  a    d    funcIU_1'
       'j0  e    a    0.4  n2'
       'R2  e    c    200'
};
```

The presented Matlab script contains nonlinear functions in the form of function names or directly given function handle definitions (the two returned values are: the function value and the time derivative). The application of a function handle is convenient as some function definitions can be simple and in many cases it would be tiresome to create additional function files for each separate nonlinear element with a different dependency.

The node names are associated later on with numbers by assigning indices from 0 to n_n in a sorted name list. Therefore, the node with the first name listed alphabetically is automatically selected to be the ground node.

Because of the actual direction of current flow, it is commonly assumed in circuit computation programs that the direction of the current opposes the direction of the voltage. It can be however more clear numerically that when defining an element – the directions of both current and voltage are assumed from node ϕ to d. If one wishes to keep the assumption is most commonly used, it is sufficient to change the sign all of the currents' values after the result had been obtained.

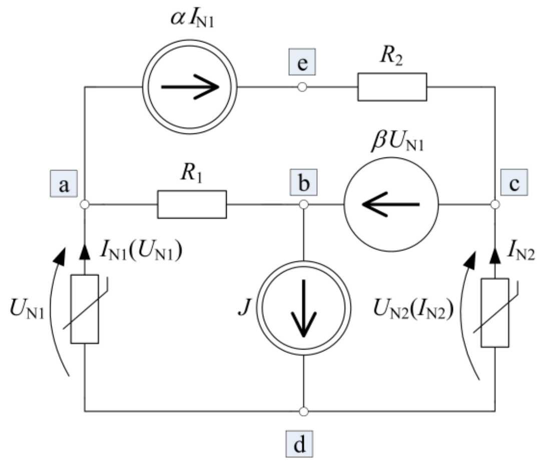


Fig. 1. Exemplary circuit with nonlinear elements and controlled sources
Rys. 1. Przykładowy obwód z elementami nieliniowymi i źródłami sterowanymi

3.2. Representation of stamp vectors

In the considered implementation an individual element's dependencies are divided into V-stamps and I-stamps:

– I-stamps are dependencies which are added or subtracted subject to KCL, in general they just define the current at the element's branch,
– V-stamps present dependencies that result from basic voltage relations – each of them is considered as a separate equation.

The V-stamps and appropriate sums of I-stamps form the base set of linear equations. Therefore, these are placed into the rows of the $\boldsymbol{A}$ matrix and their respective $\boldsymbol{b}$ vector components.

The following vectors are introduced into I-stamps and V-stamps (each presenting different types of dependencies):

- $\boldsymbol{v}$ – dependencies on potential variables,
- $\boldsymbol{i}$ – dependencies on current variables (given in V-stamps only in cases of current controlled voltage sources),
- b – contribution in the form of a constant,
- $\boldsymbol{a}$ – dependence on auxiliary variables.

With the above taken into account, a stamp structure can be given in the form:

$$\boldsymbol{S} = [\boldsymbol{v} \quad \boldsymbol{i} \quad b \quad \boldsymbol{a}]^{\mathrm{T}}, \tag{4}$$

which in Matlab can be given as a cell structure of length 4. The appropriate cells for $\boldsymbol{v}$, $\boldsymbol{i}$ and $\boldsymbol{a}$ are filled with sparse vectors. The third cell contains the value of b.

If a circuit contains nonlinear elements then additional equations should be formulated in the form either:

$$U_{\mathrm{N}} = U_{\mathrm{N}}(I_{\mathrm{N}}), \tag{5}$$

or:

$$I_{\mathrm{N}} = I_{\mathrm{N}}(U_{\mathrm{N}}), \tag{6}$$

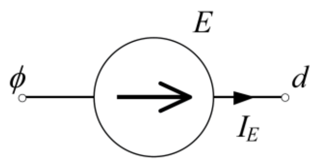
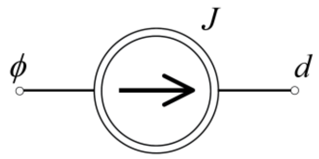
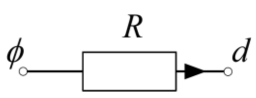
in order to provide *n* linearly independent equations of *n* unknowns.

3.3. Stamps for basic source and resistance elements

Table 1 presents the contributions of autonomous voltage and current sources and resistances. The contributions are presented with a general explanation and a stamp vector representation. The designation $(\Sigma I)_c$ means the balance of currents in node c subject to KCL. The symbol ι in the table denotes an element's unique index.

Table 1

Contributions of autonomous sources and resistance elements to the system of equations

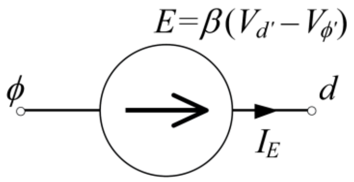
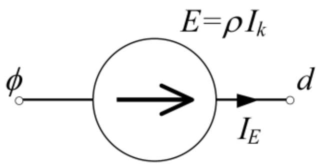
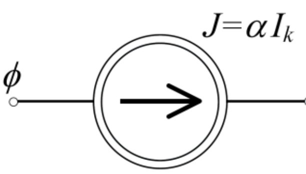
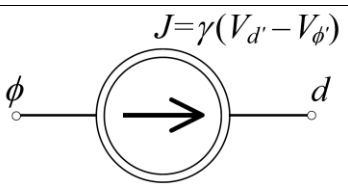
<table>
<tr><th>Circuit representation</th><th>Contributions</th><th>Stamp</th></tr>
<tr><td rowspan="3">
</td><td>$V_d - V_\phi = E$</td><td>V-stamp:
$\underset{d\neq 0}{\forall} \boldsymbol{v}_d = 1$
$\underset{\phi\neq 0}{\forall} \boldsymbol{v}_\phi = -1$
$b = E$</td></tr>
<tr><td>$+I_E$ to $(\Sigma I)_d$</td><td rowspan="2">I-stamp: $\boldsymbol{i}_\iota = 1$</td></tr>
<tr><td>$-I_E$ to $(\Sigma I)_\phi$</td></tr>
<tr><td rowspan="2">
</td><td>$+J$ to $(\Sigma I)_d$</td><td rowspan="2">I-stamp: $b = -J$</td></tr>
<tr><td>$-J$ to $(\Sigma I)_\phi$</td></tr>
<tr><td rowspan="2">
</td><td>$-\dfrac{V_d - V_\phi}{R}$ to $(\Sigma I)_d$</td><td rowspan="2">I-stamp:
$\underset{d\neq 0}{\forall} \boldsymbol{v}_d = -\dfrac{1}{R}$
$\underset{\phi\neq 0}{\forall} \boldsymbol{v}_\phi = \dfrac{1}{R}$</td></tr>
<tr><td>$+\dfrac{V_d - V_\phi}{R}$ to $(\Sigma I)_\phi$</td></tr>
</table>

3.4. Stamps for controlled sources

Table 2 presents the dependencies that result from placing controlled sources in the circuit. As before – the circuit representation is shown along with contributions and their respective stamp representations. Current controlled sources are presented as dependent on I_k, where k is the unique index of the element through which the controlling current passes.

Table 2

Contributions of controlled voltage and current sources to the system of equations

<table>
<tr><th>Circuit representation</th><th>Contributions</th><th>Stamp</th></tr>
<tr><td rowspan="3"></td><td>$V_d - V_\phi = \beta(V_{d'} - V_{\phi'})$</td><td>V-stamp:
$\underset{d\neq 0}{\forall} \boldsymbol{v}_d = 1$
$\underset{\phi\neq 0}{\forall} \boldsymbol{v}_\phi = -1$
$\underset{d'\neq 0}{\forall} \rightarrow$ add $(-\beta)$ to $\boldsymbol{v}_{d'}$
$\underset{\phi'\neq 0}{\forall} \rightarrow$ add β to $\boldsymbol{v}_{\phi'}$</td></tr>
<tr><td>$+I_E$ to $(\Sigma I)_d$</td><td rowspan="2">I-stamp: $\boldsymbol{i}_\iota = 1$</td></tr>
<tr><td>$-I_E$ to $(\Sigma I)_\phi$</td></tr>
<tr><td rowspan="3"></td><td>$V_d - V_\phi = \rho I_k$</td><td>V-stamp:
$\underset{d\neq 0}{\forall} \boldsymbol{v}_d = 1$
$\underset{\phi\neq 0}{\forall} \boldsymbol{v}_\phi = -1$
$\boldsymbol{i}_k = -\rho$</td></tr>
<tr><td>$+I_E$ to $(\Sigma I)_d$</td><td rowspan="2">I-stamp: $\boldsymbol{i}_\iota = 1$</td></tr>
<tr><td>$-I_E$ to $(\Sigma I)_\phi$</td></tr>
<tr><td rowspan="2"></td><td>$+\alpha I_k$ to $(\Sigma I)_d$</td><td rowspan="2">I-stamp: $\boldsymbol{i}_k = \alpha$</td></tr>
<tr><td>$-\alpha I_k$ to $(\Sigma I)_\phi$</td></tr>
<tr><td rowspan="2"></td><td>$+\gamma(V_{d'} - V_{\phi'})$ to $(\Sigma I)_d$</td><td rowspan="2">I-stamp:
$\boldsymbol{v}_{d'} = \gamma$
$\boldsymbol{v}_{\phi'} = -\gamma$</td></tr>
<tr><td>$-\gamma(V_{d'} - V_{\phi'})$ to $(\Sigma I)_\phi$</td></tr>
</table>

Many of the preliminary stamps can be changed, which allows a further reduction of n. In two cases the stamps need to be changed unconditionally:

– the V-stamp of a current controlled voltage source if the controlling current is not one that passes through a voltage source or a nonlinear $I(U)$ element (because the stamp cannot define a dependence on a variable that does not exist or was initially substituted by a dependence on other variables),
– the I-stamp of a current controlled current source under the same conditions as above for the mentioned V-stamp.

3.5. Stamps for nonlinear elements

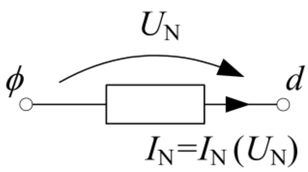
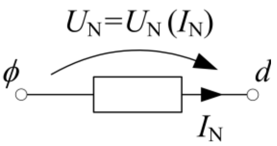
Table 3 shows the linear contributions provided by nonlinear elements. The nonlinear contributions are added separately as equations (3.5) or (3.6). η is an index which is unique to every nonlinear element.

Therefore, an I(U) defined element yields two additional circuit variables, while in the case of an U(I) nonlinear element it is only 1. This is because the voltage in the second case is replaced by its equivalent difference of node potentials.

There are ways of reducing the numbers of unknowns in a system of circuit equations, however they are not covered in this paper.

Table 3

Contributions of nonlinear elements to the system of equations

<table>
<tr><th>Circuit representation</th><th>Contributions</th><th>Stamp</th></tr>
<tr><td rowspan="4">
</td><td>$V_d - V_\phi = U_N$</td><td>V-stamp: $\underset{d\neq 0}{\forall}\boldsymbol{v}_d = 1$
$\underset{\phi\neq 0}{\forall}\boldsymbol{v}_\phi = -1$
$\boldsymbol{a}_\eta = -1$</td></tr>
<tr><td>$+I_N$ to $(\Sigma I)_d$</td><td rowspan="2">I-stamp: $\boldsymbol{i}_\iota = 1$</td></tr>
<tr><td>$-I_N$ to $(\Sigma I)_d$</td></tr>
<tr><td colspan="2">Nonlinear equation: $I_N = I_N(U_N)$</td></tr>
<tr><td rowspan="3">
</td><td colspan="2">Nonlinear equation: $V_d - V_\phi = U_N(I_N)$</td></tr>
<tr><td>$+I_N$ to $(\Sigma I)_d$</td><td rowspan="2">I-stamp: $\boldsymbol{a}_\eta = 1$</td></tr>
<tr><td>$-I_N$ to $(\Sigma I)_d$</td></tr>
</table>

4. SUMMARY

The paper is the first part of a guide to the implementation of automatic formulation for circuit equations. In many ways the approach is different from the one frequently encountered in literature [2, 5, 6]. This part of the paper provides the basic notations for the circuit topology representation and the general form of the structures that are used to describe the contribution of each type of supported element to the whole system of equations describing the distribution of potentials and currents in a circuit.

So far the presented approach can be mainly used for educational purposes. In future papers however, generalizations are planned to be made for dynamic problems (hence – an algorithm for the automatic formulation of a circuit's state equations) including nonlinear problems.

BIBLIOGRAPHY

1. Chua L.O., Lin P. M.: Computer-aided analysis of electronic circuits. Algorithms and Computational Techniques. Wydawnictwo Naukowo-Techniczne, Warszawa 1981 (in Polish).
2. Dobrowolski A.: Under the hood of SPICE. Methods and algorithms for the analysis of electronic circuits (in Polish). BTC Publishing House, Warszawa 2004.
3. Garczarczyk Z., Lasicz A., Paszek S., Walczak J.: Numerical methods in theoretical electrical engineering. Part I: Analysis of linear and nonlinear circuits. Wydawnictwo Politechniki Śląskiej, Gliwice 1988 (in Polish).
4. Hachtel G.D., Brayton R.K., Gustavson F.G.: The Sparse Tableau Approach to Network Analysis and Design. „IEEE Transactions on Circuit Theory" 1971, Vol. 18, Iss. 1, p. 101-113.
5. Litovski V., Zwolinski M.: VLSI Circuit Simulation and Optimization. Chapman & Hall, London 1997.
6. Vlach J., Singhal K.: Computer methods for circuit analysis and design – Van Nostrand Reinhold Electrical/Computer Science and Engineering Series, 1983.
7. Walczak J., Pasko M.: Components of the dynamics of linear electric circuits (in Polish). Silesian University of Technology Publishing House, Gliwice 2011.
8. http://www.electronics-lab.com/downloads/schematic/013/tutorial/PSPICE.pdf

Dr inż. Marcin SOWA
Politechnika Śląska, Wydział Elektryczny
Instytut Elektrotechniki i Informatyki
ul. Akademicka 10
44-100 Gliwice
e-mail: Marcin.Sowa@polsl.pl