Topology Reduction for Approximate Symbolic Analysis

Zdeněk KOLKA, Martin VLK, Martin HORÁK

Dept. of Radio Electronics, Brno University of Technology, Purkynova 118, 612 00 Brno, Czech Republic

kolka@feec.vutbr.cz

Abstract. *The paper deals with a procedure for approximate symbolic analysis of linear circuits based on simplifying the circuit model. The procedure consists of two main steps. First, network elements whose influence on the circuit function is negligible are completely removed, i.e. their parameters are removed from the resulting symbolic formula. The second step consists in modifying the voltage and current graphs in order to decrease the number of common spanning trees. The influence of each modification of the circuit model is ranked numerically. A fast method based on the use of cofactors is presented. It allows evaluating all the prospective simplifications using at most two matrix inversions per one frequency point.*

Keywords

Symbolic analysis, two-graph method, circuit reduction, linear circuits, frequency domain.

1. Introduction

Within the last several years, we have seen a growing interest in the symbolic analysis of large multi-physics systems (see [1] and its references). Symbolic analysis is used both for obtaining a qualitative description of the analyzed system and for the generation of behavioral models. Originally developed in the field of electrical circuits, symbolic analysis can be used in other physical domains on the basis of analogies [1], [2].

The applicability of exact symbolic analysis in the frequency domain is constrained to relatively small systems, as the size of the resulting expression grows exponentially with the number of components. If we appropriately restrict the range of frequency and network parameters, the majority of symbolic terms can be removed from large expressions without any significant numerical error [3]. Negligible symbolic terms are identified numerically, based on the known parameters of circuit components.

The simplification methods can be divided into three classes according to the stage of analysis at which the simplification is performed: *Simplification Before Generation* (SBG), *Simplification During Generation* (SDG), and *Simplification After Generation* (SAG) [5]. The SAG methods are simple, but very expensive in terms of computation and storage. Pure mathematical methods of the SDG type have problems with the interpretability of resulting expressions [4]. The SBG methods simplifying the circuit equations or graphs are the most effective ones, as they work with a relatively small number of circuit equations [4].

The only commercially available symbolic simplification tool – *Analog Insydes* – implements a matrix-based method [5] where individual matrix elements are removed to obtain a simplified solution. However, in some cases the equation simplification may surprisingly add symbolic terms that were not present in the original expression. The SBG method [6] is based on a heuristic approach consisting in modifying the graphs of the numerator and the denominator separately, which does not have a clear physical interpretation.

This paper presents an SBG procedure whose basic principle can be explained on a simple circuit in Fig. 1. Let the network parameters be: $R_B = 36$ kΩ, $r_\pi = 4$ kΩ, $g_m = 35$ mS, $r_o = 100$ kΩ, $R_L = 4$ kΩ.

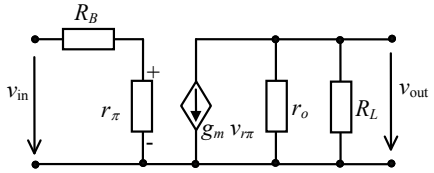


Fig. 1. AC model of simple amplifier with bipolar transistor.

The exact formula for the voltage transfer ratio is

$$K_V = \frac{v_{out}}{v_{in}} = -\frac{r_\pi}{R_B + r_\pi} g_m (R_L \parallel r_o) \,. \tag{1}$$

It can be easily seen that, with respect to the parameter values, the expression can be simplified. However, in the case of analysis of large systems, an exact formula cannot be generated at all. The first step, parametric simplification (PSBG), consists in removing the negligible circuit elements by setting their parameters to zero or infinity. In our example $r_o \gg R_L$, and thus it can be removed by setting $r_o \to \infty$. As $R_B \gg r_\pi$, the formula can be further simplified, but resistor r_π cannot be simply removed or its terminal shorted.

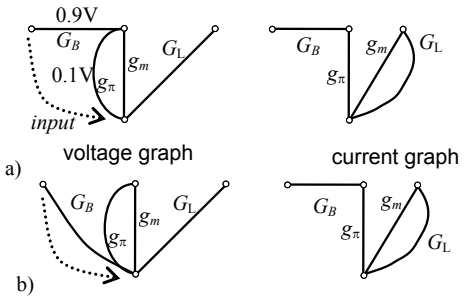


Fig. 2. a) Original; and b) modified graphs.

The second step, topological simplification (TSBG), consists in modifying the circuit topology without removing any element. Fig. 2a shows the voltage and current graphs of the circuit. Let the voltage across input be 1 V. Then the voltage across r_π is 0.1 V. The voltage can be neglected in loops {*input*-G_B-g_π} and {*input*-G_B-g_m} but not in loop {g_π-g_m}. A simple modification in Fig. 2b removes g_m and g_π from the "high-voltage" loop, but retains the "low-voltage" loop. The modification leads to the expected simplified formula

$$K_{V,approx} = -\frac{r_\pi}{R_B} g_m R_L . \tag{2}$$

An inspection of the voltage and current graphs shows that the voltage-controlled current source from Fig. 2a was replaced by a current-controlled current source in Fig. 2b. The basic principle of the topological method consists in a selective removal of the low-voltage edges from the high-voltage loops. Similar transformations can be found for cuts of the current graph.

The simplification of network equations is a sequence of individual steps. The control procedure searches for steps introducing the lowest error, which is numerically expensive for large systems [3]. Section 2 of the paper describes in detail the PSBG and TSBG algorithms, including an effective method for the evaluation of errors. Section 3 provides an example analysis.

2. Topology Simplification Procedure

2.1 Control Algorithm

The simplification introduces an error whose maximum value should be theoretically guaranteed on the interval $F \times D$, where $F = \langle f_1, f_2 \rangle$ is the frequency interval of interest, and $D \subset R^r$ is the interval of parameters of network elements. Due to computational complexity the error is only checked at several selected reference points of $F \times D$ with specified magnitude and phase tolerances ΔM_ω and $\Delta\varphi_\omega$ [5].

Assuming m reference points ω_i the error criterion is

$$\varepsilon_A = \max_{i=1..m} \left\{ \frac{20\log|E(\omega_i)|}{\Delta M_{\omega_i}} + \frac{\arg(E(\omega_i))}{\Delta\varphi_{\omega_i}} \right\} \tag{3}$$

where $E(\omega) = F_A(\omega)/F_R(\omega)$, and $F_A(\omega)$ and $F_R(\omega)$ are simplified and reference network functions, respectively.

Fig. 3 shows the main cycle of the procedure, which is essentially the same for PSBG and TSBG. First, all the prospective operations are ranked according to the error their application would cause. One or more operations with the lowest error are actually performed and the numerical solution is updated. The procedure is repeated until the maximum error is reached.

```
compute reference numerical solution;
while εA < εmax  {
  generate all possible operations;
  compute the error of each operation;
  perform operation(s) with the lowest error;
  update numerical solution and εA;
}
undo last operation;
```

Fig. 3. Main cycle of simplification method.

2.2 Parametric Simplification

The circuit being analyzed is represented by a set of linear equations obtained by using, for example, the Modified Nodal Analysis [9]. Without independent sources, we obtain

$$\mathbf{Hx} = \mathbf{0} \tag{4}$$

where **H** is a network matrix, and **x** is the vector of unknown voltages and currents.

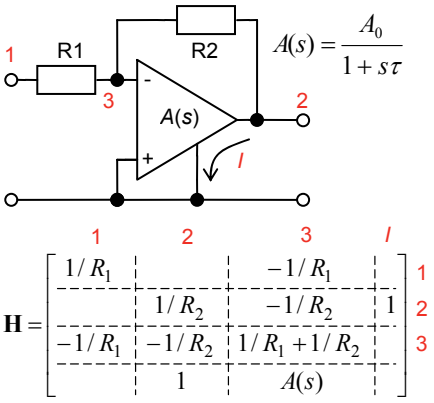


Fig. 4. Parametric simplification of an amplifier.

Fig. 4 shows a simple circuit where the operational amplifier is modeled as a voltage-controlled voltage source with a complex transfer function $A(s)$.

Let p be a parameter representing a single network element. Generally, it can appear on one (e.g. $A(s)$ in Fig. 4), two, or four (e.g. $1/R_1$) positions in **H**. First, the parametric simplification procedure tries to eliminate the parameter by setting $p \to 0$ or $p \to \infty$. For example, using $A(s) \to \infty$ replaces the controlled source by the ideal operational amplifier. When no eliminable parameter is left, the procedure continues inside parameters, which are given as

a formula. For example, in the case of $A(s)$, neglecting $s\tau$ may be acceptable on low frequencies whereas neglecting "1" in the denominator may be acceptable on high frequencies.

In all cases, the control algorithm has to compute how the change of p affects the numerical value of the network function for each control frequency.

Any network function F can be obtained as a ratio of two algebraic cofactors of $\mathbf{H}$

$$F = (-1)^{\alpha}\frac{\det(\mathbf{H}_1)}{\det(\mathbf{H}_2)} = (-1)^{\alpha}\frac{\left(\det(\mathbf{H}_1) - p_{nom}\Delta_1\right) + p\Delta_1}{\left(\det(\mathbf{H}_2) - p_{nom}\Delta_2\right) + p\Delta_2}. \quad (5)$$

Matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ are derived from $\mathbf{H}$ by means of adding and deleting some rows and columns, α depends on the indices of those rows and columns [9]. For parameter p appearing on four positions in $\mathbf{H}_1$ and $\mathbf{H}_2$, network function F can be expanded as shown in (5), where $\Delta_1 = \Delta_{\mathrm{H1}i:i} + \Delta_{\mathrm{H1}j:j} - \Delta_{\mathrm{H1}i:j} - \Delta_{\mathrm{H1}j:i}$ represents the algebraic cofactors of $\mathbf{H}_1$, Δ_2 represents the cofactors of $\mathbf{H}_2$, and p_{nom} is the nominal value of the parameter. The coordinates of p in $\mathbf{H}_1$ and $\mathbf{H}_2$ are generally different. If p appears on one or two positions, Δ_1 and Δ_2 are expressed just with one or two cofactors.

If we know the determinants and cofactors, it is easy to compute how F changes for any value of p, including infinity, by using (5). The algebraic cofactors can be obtained by a simple matrix inversion

$$\mathbf{H}_{1,\Delta} = \det(\mathbf{H}_1)\left(\mathbf{H}_1^{-1}\right)^T,\ \mathbf{H}_{2,\Delta} = \det(\mathbf{H}_2)\left(\mathbf{H}_2^{-1}\right)^T \quad (6)$$

where each element of $\mathbf{H}_{1,\Delta}$ and $\mathbf{H}_{2,\Delta}$ is the respective algebraic cofactor. The computation of one cofactor matrix requires approximately $O(n^3)$ operations, where n is the actual matrix size. The determinant is a byproduct of the matrix inversion.

After the topology change in each step, the determinants and cofactor matrices (6) should be updated. It can be done either by computing a new matrix inverse or by using the Sherman-Morrison formula, which allows expressing the matrix inverse in the case when only a few elements were changed [10]. To set a parameter $p \to 0$ a compensating term of $(-p)$ should be added to the matrix. Then, the inverse of the updated matrix will be

$$\left(\mathbf{A} - p\mathbf{u}\mathbf{v}^{\mathrm{T}}\right)^{-1} = \mathbf{A}^{-1} + \frac{p\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^{\mathrm{T}}\mathbf{A}^{-1}}{1 - p\mathbf{v}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{u}}. \quad (7)$$

Vectors $\mathbf{u}$ and $\mathbf{v}$ define the coordinates of the updated matrix elements. In the case of the four-position appearance of p the vectors are simply $\mathbf{u} = [0...0,1,0..0,-1,0....0]^T$ and $\mathbf{v} = [0...0,1,0..0,-1,0....0]^T$.

2.3 Topological Simplification

Using the two-graph method [7], [11] the determinant of the admittance matrix can be computed as

$$\det \mathbf{Y} = \sum_{t \in T(G_{\mathrm{V}}) \cap T(G_{\mathrm{I}})} \varepsilon(t) Y^{(t)} \quad (8)$$

where $Y^{(t)}$ is the tree admittance product of tree t, and $T(G_{\mathrm{V}}) \cap T(G_{\mathrm{I}})$ represents the common spanning trees of current graph G_{I}, and voltage graph G_{V}. $\varepsilon(t) = \pm 1$ is the tree sign. The technique of augmented circuit allows using (8) to compute the cofactors for any network function [7].

If all edges represent a unique symbol, there are no two identical tree admittance products in (8) that would cancel each other. Thus, if there is a transformation that decreases the number of spanning trees, it automatically decreases the determinant complexity.

Let $V(G)$ be a set of vertices of a graph G, $E(G)$ a set of its edges, and $T(G)$ a set of its trees. The incidence of edge e in graph G, $\rho(e,G) = (i,j)$, assigns two vertices i, j to edge e. An edge with the incidence (v,v) is called self-loop. Graph G is said to be separable if there is a vertex whose removal splits the graph into two or more components.

Definition 1: *Separation* of a connected subgraph G_{S} from a graph G is an operation that transforms G into

$$G' = \widetilde{G} \cup G_{\mathrm{S}} \quad (9)$$

where $\widetilde{G}$ is a subgraph whose edge set is $E(\widetilde{G}) = E(G \setminus G_{\mathrm{S}})$. The incidence of any edge $e \in E(\widetilde{G})$ is transformed into $\rho(e,\widetilde{G}) = (f(v_i), f(v_j))$,

$$f(v) = \begin{cases} v_{\mathrm{c}} & \text{if } v \in V(G_{\mathrm{S}}) \\ v & \text{otherwise} \end{cases}. \quad (10)$$

v_{c} is an arbitrarily chosen vertex $v_{\mathrm{c}} \in V(G \setminus G_{\mathrm{S}}) \cap V(G_{\mathrm{S}})$. The operation, illustrated in Fig. 5, will be denoted $G' = G \triangleright G_{\mathrm{S}}$. Transforming the incidence by (10) may lead to the occurrence of selfloops, but $|E(G')| = |E(G)|$.

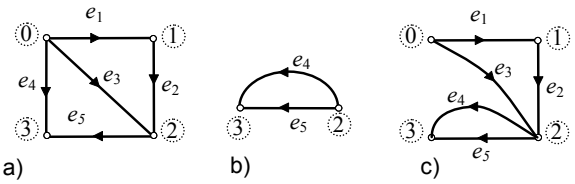


Fig. 5. Example of separation $G_{\mathrm{S}} = \{e_1,e_2,e_3\}$: a) original graph G; b) graph $\widetilde{G}$; c) $G' = G \triangleright G_S$.

Lemma 1: Let G and G' be two connected graphs for which it holds: $E(G') \subseteq E(G)$ and $|V(G')| = |V(G)|$. If for any loop $L \subseteq G$, $E(L) \subseteq E(G')$ there exists a loop $L' \subseteq G'$ such that $E(L') \subseteq E(L)$, then for any tree $t' \in T(G')$ there exists a tree $t \in T(G)$ such that $E(t') = E(t)$.

Proof: Let us assume that the conditions of Lemma 1 hold, yet there exists a tree $t' \in T(G')$ such that $t' \notin T(G)$. Then there must be a subgraph $G_{\mathrm{S}} \subseteq G$ induced by $E(G_{\mathrm{S}}) = E(t')$ containing at least one loop $L \subseteq G_{\mathrm{S}}$ because it is not a tree in G. However, if there exists a loop $L' \subseteq G'$ for L such that $E(L') \subseteq E(L)$, then $E(L') \subseteq E(t')$ because $E(L) \subseteq E(G_{\mathrm{S}})$

and $E(G_S) = E(t')$. This is a contradiction of the initial assumption that subgraph t' is a tree in G'. ■

Theorem 1: Let G be a connected non-separable graph consisting of at least two edges. The separation of any connected subgraph $G_S \subset G$ consisting of at least one edge decreases the number of trees of the transformed graph $G' = G \triangleright G_S$ without the occurrence of alien trees.

Proof: The proof strategy consists of: (a) verifying the absence of alien trees, (b) decreasing the number of trees.

a) The absence of alien trees will be proved by verifying the conditions of Lemma 1. Since $|V(G')| = |V(G)|$ and $E(G') = E(G)$, then it is sufficient to show that for any loop $L \subseteq G$ there exists a loop $L' \subseteq G'$ such that $E(L') \subseteq E(L)$. Let us consider a loop L in graph G. As subgraphs G_S and $G \setminus G_S$ are edge-disjoint and $G = G_S \cup (G \setminus G_S)$, only one of the following three cases may happen:

1. The whole loop L is contained in G_S, i.e. $E(L) \subseteq E(G_S)$. Then there exists a loop $L' \subseteq G'$ such that $E(L') = E(L)$, because G_S is also a subgraph of G'.

2. Loop L is contained in both G_S and $G \setminus G_S$. Let $G_C = L \cap (G \setminus G_S)$, then there exists at least one open trail with endpoints $v_i, v_j \in V(G_S)$ in G_C. Both these endpoints are transformed by (10) into v_C, which forms at least one loop $L' \subseteq G'$.

3. The whole loop L is contained in $G \setminus G_S$, i.e. $E(L) \subseteq E(G \setminus G_S)$. Transformation (10) of $G \setminus G_S$ causes the occurrence of one or several loops or selfloops L_i' such that $E(L_i') \subseteq E(L)$ holds for each of them.

b) Subgraph $G \setminus G_S$ may be disconnected. Therefore, let us consider an arbitrary component $C \subseteq G \setminus G_S$ and its tree $t_C \in T(C)$. Evidently, there exists a tree $t \in T(G)$ such that $t_C \subset t$. Since G is non-separable, subgraph C and its complement $G \setminus C$ have at least two vertices u and v in common such that $u, v \in E(G_S)$. Vertices u and v are transformed by (10) into a single vertex. This creates a loop $L' \subset G'$ such that $E(L') \subset E(t)$. Thus the edges of t cannot be a tree in G'. Therefore, the separation decreases the number of trees. ■

Let the circuit be represented by current graph G_I and voltage graph G_V with edges $e_1, e_2, \ldots, e_b$, whose weights are the magnitudes of branch currents and voltages obtained numerically for a particular frequency.

Let $L_1, L_2, \ldots, L_B \subseteq G_V$ be all the loops of the voltage graph G_V. The voltage $v(e_j)$ of an edge $e_j \in E(L_i)$ will be considered numerically negligible in loop L_i if

$$\left|v(e_j)\right| < \varepsilon_V \max_{h \in E(L_i)} \left|v(h)\right| \tag{11}$$

where $\varepsilon_V \in (0, 1)$ is the threshold value. Provided that it causes an acceptable numerical error it is possible to remove all negligible edges from L_i.

Let us assume that voltage graph G_V can be decomposed into two edge-disjoint subgraphs G_H^V and G_L^V such that the condition

$$\max_{e \in G_L^V \cap L} \left|v(e)\right| < \varepsilon_V \max_{e \in L} \left|v(e)\right| \tag{12}$$

holds for any loop $L \subseteq G_V$ that is contained in both subgraphs. Then it is possible to remove the low-voltage edges by

$$G'_V = G_V \triangleright G_L^V \ . \tag{13}$$

Let $C_1, C_2, \ldots, C_Q \subseteq G_I$ be all the cuts of the current graph G_I. The current $i(e_j)$ of a cut edge $e_j \in E(C_i)$ will be considered numerically negligible in C_i if

$$\left|i(e_j)\right| < \varepsilon_I \max_{h \in E(C_i)} \left|i(h)\right| \tag{14}$$

where $\varepsilon_I \in (0, 1)$ is the threshold value.

It is convenient to reformulate criterion (14) for cut edges into another criterion for loops of the current graph. Let us consider current graph G_I and its arbitrary loop $L \subseteq G_I$ containing an edge e_{min} with the minimum weight, Fig. 6. Let t be a tree of G_I such that $L \setminus \{e_{min}\} \subseteq t$, Fig. 6a. In accordance with (14) it is possible to neglect $i(e_{min})$ in the maximum-current cut, i.e. edge e_{max} can be removed from loop L.

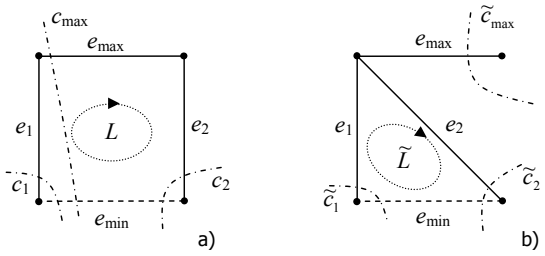


Fig. 6. a) Loop in G_I; b) Modified graph.

Let G_I be decomposed with respect to a threshold value $\varepsilon_I \in (0, 1)$ into two edge disjoint subgraphs G_H^I and G_L^I, and for any loop $L \subseteq G_I$ contained in both subgraphs the condition

$$\min_{e \in E(L)} \left|i(e)\right| < \varepsilon_I \min_{e \in E(G_H^I \cap L)} \left|i(e)\right| \tag{15}$$

holds. Then it is possible to remove all edges of G_H^I from any loop contained in both G_H^I and G_L^I by means of

$$G'_I = G_I \triangleright G_H^I \ . \tag{16}$$

The topology transformations can be interpreted as a graph-edge changeover between vertices. Fig. 7 shows the effect on the nodal matrix if the edge represents a conductance, originally between nodes m and n.

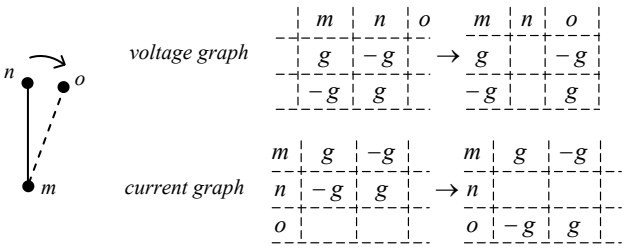


Fig. 7. Edge transformation and its influence on the nodal matrix.

The changeover to node *o* can be modeled by adding four compensating elements to the original matrix. In the case of the voltage graph modification we obtain

$$\mathbf{H}_{1,tr} = \mathbf{H}_1 + \begin{array}{c|c|c|c} & m & n & o \\ \hline m & & g & -g \\ \hline n & & -g & g \end{array}, \qquad (17a)$$

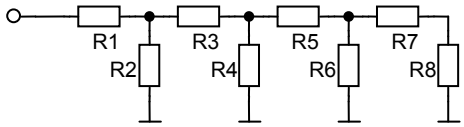
which leads to

$$\det(\mathbf{H}_{1,tr}) = \det(\mathbf{H}_1) + g(\Delta_{H1,mn} + \Delta_{H1,no} - \Delta_{H1,mo} - \Delta_{H1,nn}), \qquad (17b)$$

and similarly for $\mathbf{H}_2$ and the current graph. Thus the effect of graph transformation can be evaluated similarly to (7).

2.4 Lossless Topological Simplification

This step consists in identifying the series or parallel connections of similar two-ports, and transforming them into a single element. This step does not introduce any error, but can greatly simplify the symbolic expression.




```
Input impedance:
(R1+R2||(R3+R4||(R5+R6||(R7+R8))))
```

Fig. 8. Losless simplification of ladder network.

Fig. 8 shows a simple ladder network and the SNAP output for the input impedance [2], [8]. The same formula in the expanded plain format contains 34 terms in the numerator and 21 terms in the denominator.

3. Example Analysis

The capabilities of the topological transformation are demonstrated on an analysis of the μA741 operational amplifier [5], Fig. 9. All transistors were modeled using the Standard Gummel-Poon Spice model. The aim was to simplify the circuit model for open-loop gain in the neighborhood of the dominant pole frequency of 3 Hz. The required accuracy was 1.5 dB for magnitude and 5° for phase checked at frequencies of 0.1 Hz and 5 Hz. The original number of symbolic terms in the denominator is estimated to be 10^{19} [5].

The parametric preprocessing reduced the original 196 network parameters to 14. The topological procedure was able to further reduce the number of terms in the numerator and the denominator. Candidates for separation were generated for $\varepsilon_V = \varepsilon_I = 0.2$. The topological algorithm separated two subgraphs from the voltage graph:

$\{ g_{\pi 16}, g_{m16} \}$, $\{ g_{o4}, g_{o6}, g_{\pi 17}, g_{m17}, G_8, G_9 \}$

and two subgraphs from the current graph:

$\{ g_{o131}, g_{m17}, g_{\pi 17}, g_{m16}, G_8, G_9 \}$,

$\{ g_{o131}, g_{m17}, G_8 \}$.

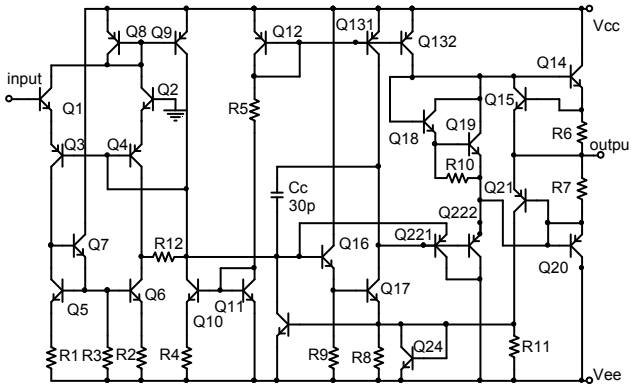


Fig. 9. Internal structure of μA741.

		PSBG	TSBG	SAG
ΔM_{max}	allowed	± 1.5dB		
	real	0.81dB @0.1Hz	1.08dB @0.1Hz	1.39dB @0.1Hz
$\Delta\varphi_{max}$	allowed	± 5°		
	real	0.53°@5Hz	1.30°@5Hz	2.17°@5Hz
k_N		12	1	1
k_D		120	16	14
n_p		14	14	14
runtime		1.8s	1.2s	0.7s

Tab. 1. Results of simplification steps for 741 opamp (ΔM_{max} – magnitude error, $\Delta\varphi_{max}$ – phase error, k_N, k_D – number of symbolic terms in numerator and denominator, n_p – number of circuit parameters).

After TSBG, the formula for voltage transfer was further simplified by means of the SAG method, which reduced only the denominator terms from 16 to 14.

Fig. 10 shows a similar analysis for circuit with a current-feedback amplifier [12]. PSBG reduced the original 100 network parameters to 14. TSBG separated two subgraphs from the voltage graph and one subgraph from the current graph.

4. Conclusions

The paper presents an effective procedure for simplifying the circuit model, which is independent of method used for subsequent symbolic analysis. The utilization of cofactor matrices allows evaluating all the prospective simplifications using at most two matrix inversions per one reference frequency point.

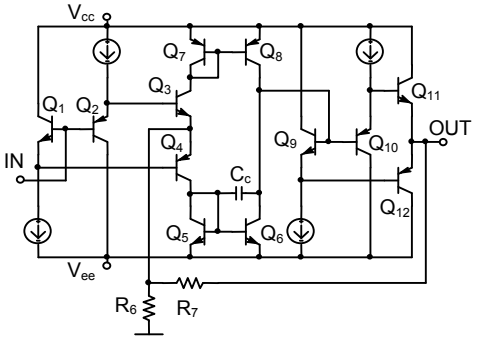


Fig. 10. Circuit with current-feedback amplifier (CFA).

		PSBG	TSBG	SAG
ΔM_{max}	allowed	± 1.5dB @ 1kHz and 600kHz		
	real	0.47dB @1kHz	0.56dB @600kHz	0.21dB @1kHz
$\Delta\varphi_{max}$	allowed	± 3°@ 1kHz and 600kHz		
	real	2.4° @600kHz	2.4° @600kHz	3.0° @600kHz
k_N		44	4	3
k_D		232	65	46
n_p		14	14	14
runtime		4.9s	1.3s	0.9s

Tab. 2. Results of simplification steps for CFA (see Tab. 1 for parameter description).

Acknowledgements

This work has been supported by the Czech Science Foundation under grant No. P102/10/1665. The research has been a part of the COST Action IC 0803, which is financially supported by the Czech Ministry of Education under grant No. OC09016. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 230126.

References

[1] SOMMER, R., HALFMANN, T., BROZ, J. Automated behavioral modeling and analytical model-order reduction by application of symbolic circuit analysis for multi-physical systems. *Simulation Modelling Practice and Theory*, 2008, no. 16, p. 1024–1039.

[2] KOLKA, Z., BIOLEK, D., KALOUS, J., BIOLKOVÁ, V. Program for Multi- Domain Symbolic Analysis. In *Proc. of 11th International Workshop on Symbolic and Numerical Methods, Modeling and Application to Circuit Design SM2ACD 2010*. Tunis (Tunisia), 2010, p. 120-123.

[3] DAEMS, W. et al. Evaluation of error-control strategies for the linear symbolic analysis of analog integrated circuits. *IEEE Trans Circ Syst.-I*, 1999, vol. 46, no. 5, p. 594-606.

[4] DAEMS, W., GIELEN, G., SANSEN, W. Circuit simplification for the symbolic analysis of analog integrated circuits. *IEEE Trans Comp Aided Design*, 2002, vol. 21, no. 4, p. 395-407.

[5] HENNIG, E. Symbolic approximation and modeling techniques for analysis and design of analog circuits. *Ph.D. Dissertation*, Univ. of Kaiserslautern (Germany), 2000.

[6] YU, Q., SECHEN, C. A unified approach to the approximate symbolic analysis of large analog integrated circuits. *IEEE Trans Circ Syst-I*, 1996, vol. 43, no. 8, p. 656-69.

[7] LIN, P. M. *Symbolic Network Analysis*. Amsterdam: Elsevier, 1991.

[8] KOLKA, Z., BIOLEK, D., BIOLKOVÁ, V. Symbolic analysis of linear circuits with modern active elements. *WSEAS Transactions on Electronics*, 2008, vol. 5, no. 6, p. 88-96.

[9] CHEN, W. K. *Active Network Analysis*. World Scientific Publishing, UK, 1991.

[10] HOUSEHOLDER, A. S. *The Theory of Matrices in Numerical Analysis*. New York, 1964.

[11] MAYEDA, W, SESHU, S. *Topological Formulas for Network Functions*. Eng. Exp. Station, University of Illinois, Bulletin 446, Urbana, 1957.

[12] PALUMBO, G. Bipolar current feedback amplifiers: Compensation guidelines. *Int. J. Analog Integrated Circuits and Signal Processing*, May 1999, vol. 19, no. 2, p. 107–114.

About Authors ...

Zdeněk KOLKA received the M.Sc. and Ph. D. degrees in 1992 and 1997 from the Brno University of Technology, Czech Republic. Currently he is a professor at the Department of Radio Electronics, Brno University of Technology. His research interests are in computer-aided design of electronic circuits, modeling, and numerical algorithms. He is the principal author of symbolic simulator SNAP.

Martin VLK received the M.Sc. degree in 1998 and the Ph.D. degree in 2005, both in Electrical Engineering, from the Brno University of Technology. In 1998 he joined the Department of Radio Electronics, Brno University of Technology, where he worked on the development of algorithms for the simplification of symbolic expressions. Currently he works as a system application engineer at the Czech System Center of Freescale Semiconductor.

Martin HORÁK received the M.Sc. degree in 2002 and the Ph.D. degree in 2006, both in Electrical Engineering, from the Brno University of Technology. In 2005 he joined the Department of Radio Electronics, Brno University of Technology, where he has been working on the development of algorithms and software for circuit analysis.