Formula Simplification using Two-Graph Method

Zdenek Kolka, Martin Vlk, Dalibor Biolek, Viera Biolkova
Faculty of Electrical Engineering and Communication
Brno University of Technology
Brno, Czech Republic

Josef Dobes
Faculty of Electrical Engineering
Czech Technical University in Prague
Prague, Czech Republic

***Abstract*— The paper deals with a method of graph transformation and its application to approximate symbolic analysis of linearized circuits. Based on the two-graph circuit description the method changes the graph structure in order to decrease the number of common spanning trees, i.e. to decrease the formula complexity. Instead of the simple edge deletion or contraction voltage and current graphs are modified to remove negligible voltages from loops and negligible currents from cuts.**

Keywords-Symbolic analysis; graph theory, linear circuits

I. INTRODUCTION

The well-known property of symbolic analysis of linear circuits is the exponential growth of symbolic expression complexity with the circuit size. For example, the number of terms of voltage transfer function of type 741 operational amplifier is estimated to be 10^{19} [1]. If we restrict the range of frequency and circuit parameters appropriately, the majority of symbolic terms can be removed from large expressions without any significant numerical error [2].

The basic prerequisite is the knowledge of at least approximate values of network parameters. The process of simplification is a sequence of elementary steps (term removal, topology modification, etc.) controlled by a mechanism that executes these steps in a correct order, and checks for correctness of the approximate expression [3].

The simplification can be performed on the circuit equations, during symbolic formula generation, or on the resulting expression [4]. The third method is simple, but computationally and storage very expensive. The simplification of circuit equations or graphs based on physical principles is the most effective method. Additionally the resulting expressions are better interpretable in comparison with purely mathematical methods [5].

The only commercially available symbolic simplification tool – *Analog Insydes* – implements a matrix-based method [6]. Individual matrix elements are removed to obtain a simplified solution. The procedure is based on a solely empirical assumption that simplifying the network equations automatically leads to simplifying the expression. Generally, this does not hold true and alien symbolic terms can occur. Moreover the result depends on the matrix formulation method.

Provided that all network parameters are assigned a unique name, any network function F can be expanded by any parameter p_i into

$$F(p_i) = \frac{\tilde{q}_i\, p_i + \tilde{Q}_i}{\tilde{r}_i\, p_i + \tilde{R}_i} \ , \qquad (1)$$

where the polynomials $\tilde{q}_i, \tilde{Q}_i, \tilde{r}_i, \tilde{R}_i$ do not contain p_i. There are two possible cases where parameter p_i can be completely eliminated irrespective of the formulation method:

$$F(p_i) \approx \lim_{p_i \to 0} F(p_i) \quad \text{or} \quad F(p_i) \approx \lim_{p_i \to \infty} F(p_i) \qquad (2a,b)$$

As p_i represents a network element, (2a) and (2b) also simplify the circuit model. Generally, the numerator and the denominator of (1) can be simplified separately. However, such a simplification does not admit a straightforward circuit interpretation.

A method, published as the *Sifting Approach* [7], is based on a heuristic algorithm consisting of device parameter elimination from the numerator and denominator submatrices separately. Another method, called the *Two-graph Simplification* [8], modifies the voltage and current graphs constructed for the numerator and denominator separately. Despite its name, the method does not exploit the properties of the two-graph description.

The proposed method goes beyond the simple parametric simplification by introducing a graph transformation for decreasing the number of spanning trees. In Section 2 we formulate the problem. Section 3 introduces the graph-transformation concept. Section 4 deals with the simplification of circuit equations. An example illustrating the method is presented in Section 5.

II. PROBLEM FORMULATION

Let the first step of simplification consists in eliminating network elements that have negligible influence, using (2a) or (2b). Now, it is not possible to further remove any element without an unacceptable numerical error. The subsequent procedure will be illustrated on a simple example, Fig. 1a. Let the numerical values of network parameters be such that

$$i_{R_1} \approx g_m v_{R2} \ , \ R_1 >> R_2 \ . \qquad (3a,b)$$

This work was supported by the Czech Science Foundation under projects No. 102/10/1665, No. 102/08/0784, the Czech Ministry of Education under project No. MSM0021630513 and by the European Community's Seventh Framework Programme under grant agreement No. 230126.

The value of v_{R2} is negligible compared to v_{R1}, but resistor R_2 cannot be simply removed because its voltage v_{R2} is sensed by the voltage-controlled current source that influences the input current circuit significantly.

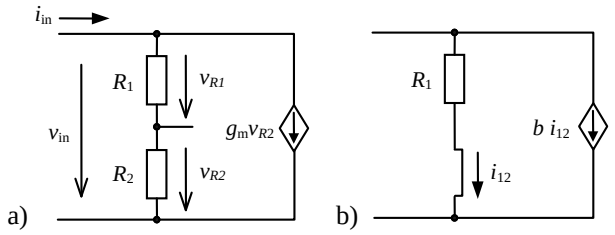

Figure 1. Small example circuit ($b = g_m R_2$).

With respect to (3b) the formula for input resistance can be directly simplified to

$$R_{in} = \frac{v_{in}}{i_{in}} = \frac{R_1 + R_2}{1 + R_2 g_m} \approx \frac{R_1}{1 + R_2 g_m} \quad . \tag{4}$$

If we replace resistor R_2 and the voltage-controlled current source by a current-controlled current source we get the same input resistance, Fig. 1b. This is equivalent to a voltage and current graph transformation as shown in Fig. 2.

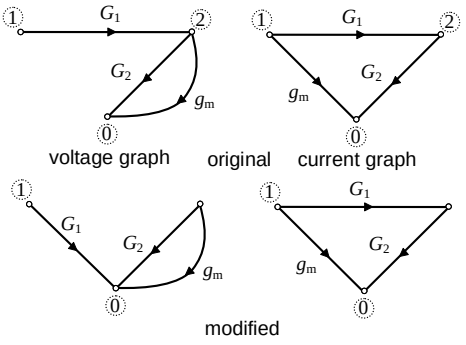

Figure 2. Graphs of circuits from Fig. 1a and Fig. 1b.

Using the two-graph method [9], [10] the determinant of the admittance matrix can be computed as

$$\det \mathbf{Y} = \sum_{t \in T(G_V) \cap T(G_I)} \varepsilon(t) Y^{(t)} \quad , \tag{5}$$

where $Y^{(t)}$ is the tree admittance product of tree t, and $T(G_V) \cap T(G_I)$ represents 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 to use (5) to compute cofactors of **Y** for network functions [10].

If all edges represent a unique symbol there are no two identical tree admittance products in (5) that would cancel each other. Thus, if there is a transformation that decreases the number of spanning trees, it automatically decreases the determinant complexity. The proposed transformation reduces the number of spanning trees in each graph separately, irrespective of whether they are common or not. It is likely that the reduction of spanning trees in one graph causes also the reduction of common spanning trees.

III. Graph transformation

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 selfloop. Graph G is said to be separable if there is a vertex whose removal splits the graph into two or more components. A block is a maximal non-separable subgraph.

Definition 1: *Separation* of a connected subgraph G_S from a graph G is an operation that transforms G into

$$G' = \tilde{G} \cup G_S \quad , \tag{6}$$

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

$$f(v) = \begin{cases} v_c & \text{if } v \in V(G_S) \\ v & \text{otherwise} \end{cases} \quad . \tag{7}$$

v_c is an arbitrary but fixed vertex $v_c \in V(G \setminus G_S) \cap V(G_S)$. The operation, illustrated in Fig. 3, will be denoted $G' = G \triangleright G_S$. Transforming the incidence by (7) may lead to the occurrence of selfloops, but $|E(G')| = |E(G)|$.

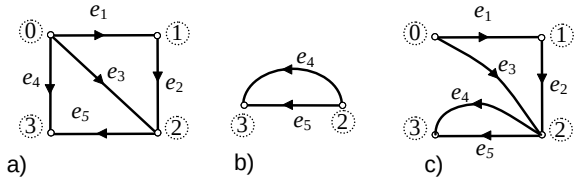

Figure 3. Example of separation $G_S = \{e_1, e_2, e_3\}$: a) original graph G; b) graph $\tilde{G}$; c) $G' = G \triangleright G_S$.

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 $G_S \subseteq G$ induced by $E(G_S) = E(t')$ containing at least one loop $L \subseteq G_S$ 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 $E(L) \subseteq E(G_S)$ and $E(G_S) = E(t')$. This is in contradiction to the initial assumption that subgraph t' is a tree in G'. ■

Lemma 2: Let t be a tree of a connected graph G with reduced incidence matrix **A** and reference vertex v_{ref}. The determinant of the tree major submatrix $\mathbf{A}_t$ does not change if the incidence (u,v) of any edge $e \subset t$ is transformed by

$$(u,v) = \begin{cases} (u, v_{\mathrm{ref}}) & \text{if } d_t(u, v_{\mathrm{ref}}) > d_t(v, v_{\mathrm{ref}}) \\ (v_{\mathrm{ref}}, v) & \text{otherwise} \end{cases}, \qquad (8)$$

where $d_t(u,v)$ is the topological distance (the minimum number of tree edges traversed) between vertices u a v.

Proof. The reduced incidence matrix **A** is derived from the full incidence matrix by deleting of the row corresponding to the reference node. There thus exits at least one column containing the only nonzero element ±1 because the graph is connected. Such a column also exists in major submatrix $\mathbf{A}_t$ since tree t contains also an edge connected to the reference node. Let element $\mathbf{A}_{t(i,j)}$ be the only nonzero element in column j. If we expand the determinant along the ith row, it is obvious that all elements of the row except for $\mathbf{A}_{t(i,j)}$ can be zeroed without changing the determinant. Because of graph connectivity, a new column with a single nonzero element appears. Repeating the procedure leads to a transformed matrix $\mathbf{A}_t$', where each column contains just one nonzero element.

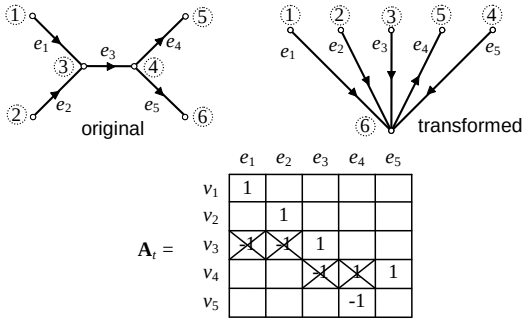

Figure 4. Transformation of tree and modification of its major (v_{ref}=6).

The tree corresponding to the new incidence matrix $\mathbf{A}_t$' can be simply obtained by changing the incidence of any edge such that the topologically closer vertex is replaced by v_{ref}, Fig. 4. ■

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

Proof. The proof strategy consists of verifying: (a) the absence of alien trees, (b) the preservation of tree signs, (c) 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 $G \setminus G_{\mathrm{S}}$ are edge-disjoint and $G = G_{\mathrm{S}} \cup (G \setminus G_{\mathrm{S}})$, only one of the following three cases may happen:

1. The whole loop L is contained in G_{S}, i.e. $E(L) \subseteq E(G_{\mathrm{S}})$. 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 both in G_{S} and $G \setminus G_{\mathrm{S}}$. Let $G_{\mathrm{C}} = L \cap (G \setminus G_{\mathrm{S}})$, then there exists at least one open trail with endpoints $v_i, v_j \in V(G_{\mathrm{S}})$ in G_{C}. Both these endpoints are transformed by (7) into v_{C}, which forms at least one loop $L' \subseteq G'$.

3. The whole loop L is contained in $G \setminus G_{\mathrm{S}}$, i.e. $E(L) \subseteq E(G \setminus G_{\mathrm{S}})$. Transformation (7) of $G \setminus G_{\mathrm{S}}$ 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) Let us consider the conditions of Lemma 2, where the reference node v_{ref} in (8) is the same as v_{c} used for the separation according to (7). Let $t \subseteq G$ be a tree of G for which there exists a tree $t' \subseteq G'$ such that $E(t') = E(t)$. Procedure (7) transforms some vertices of subgraph $t \cap (G \setminus G_{\mathrm{S}})$. Let us consider an arbitrary edge $e_{\mathrm{c}} \in t \cap (G \setminus G_{\mathrm{S}})$ having the incidence $\rho(e_{\mathrm{c}},G) = (u, v)$. Only one vertex of e_{c} can be the subject of transformation (7) since e_{c} is a member of tree t'. Otherwise a selfloop occurs, which is in contradiction to t being a tree. Let v be the transformed vertex. Vertex u is not transformed and thus $u \notin V(G_{\mathrm{S}})$. As $v_{\mathrm{c}} \equiv v_{\mathrm{ref}} \in V(G_{\mathrm{S}})$ the distance $d_t(u,v_{\mathrm{ref}}) > d_t(v,v_{\mathrm{ref}})$ since the only trail from u to v_{ref} in tree t goes through vertex v. The transformation of e_{c} always changes such a vertex in tree t that is topologically close to v_{ref}. Thus this is in accordance with (8) and the signs of t and t' are the same.

c) Subgraph $G \setminus G_{\mathrm{S}}$ may be disconnected. Therefore, let us consider an arbitrary component $C \subseteq G \setminus G_{\mathrm{S}}$ and its tree $t_{\mathrm{C}} \in T(C)$. Evidently, there exists a tree $t \in T(G)$ such that $t_{\mathrm{C}} \subset t$. 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 $u, v \in E(G_{\mathrm{S}})$. Vertices u and v are transformed by (7) 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. ■

IV. Circuit Equations

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 $L_1, L_2, \ldots, L_B \subseteq G_{\mathrm{V}}$ 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

$$\left|v(e_j)\right| < \varepsilon_{\mathrm{V}} \max_{h \in E(L_i)} \left|v(h)\right| \quad , \qquad (9)$$

where $\varepsilon_{\mathrm{V}} \in (0, 1)$ 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 $G_{\mathrm{H}}^{\mathrm{V}}$ and $G_{\mathrm{L}}^{\mathrm{V}}$ such that the condition

$$\max_{e\in G_{\mathrm{L}}^{\mathrm{V}}\cap L}\left|v(e)\right| < \varepsilon_{\mathrm{V}} \max_{e\in L}\left|v(e)\right| \qquad (10)$$

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

$$G'_{\mathrm{V}} = G_{\mathrm{V}} \triangleright G_{\mathrm{L}}^{\mathrm{V}} \ . \qquad (11)$$

Let $C_1, C_2, \ldots, C_Q \subseteq G_{\mathrm{I}}$ 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

$$\left|i(e_j)\right| < \varepsilon_{\mathrm{I}} \max_{h\in E(C_i)}\left|i(h)\right| \ , \qquad (12)$$

where $\varepsilon_{\mathrm{I}} \in (0, 1)$ is the threshold value.

It is convenient to reformulate criterion (12) for cut edges into another criterion for loops of the current graph. Let us consider current graph G_{I} and its arbitrary loop $L \subseteq G_{\mathrm{I}}$ containing an edge $e_{\min}$ with the minimum weight, Fig. 5. Let t be a tree of G_{I} such that $L \setminus \{e_{\min}\} \subseteq t$, Fig. 5a. In accordance with (12) it is possible to neglect $i(e_{\min})$ in the maximum-current cut, i.e. edge $e_{\max}$ can be removed from loop L.

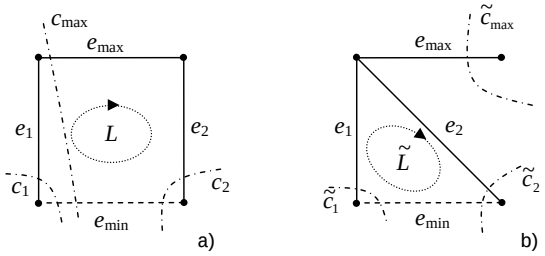

Figure 5. a) Loop in G_{I} ; b) Modified graph.

Let G_{I} be decomposed with respect to a threshold value $\varepsilon_{\mathrm{I}} \in (0, 1)$ into two edge disjoint subgraphs $G_{\mathrm{H}}^{\mathrm{I}}$ and $G_{\mathrm{L}}^{\mathrm{I}}$, and for any loop $L \subseteq G_{\mathrm{I}}$ contained in both subgraphs the condition

$$\min_{e\in E(L)}\left|i(e)\right| < \varepsilon_{\mathrm{I}} \min_{e\in E(G_{\mathrm{H}}^{\mathrm{I}}\cap L)}\left|i(e)\right| \qquad (13)$$

holds. Then it is possible to remove all edges of $G_{\mathrm{H}}^{\mathrm{I}}$ from any loop contained in both $G_{\mathrm{H}}^{\mathrm{I}}$ and $G_{\mathrm{L}}^{\mathrm{I}}$ by means of

$$G'_{\mathrm{I}} = G_{\mathrm{I}} \triangleright G_{\mathrm{H}}^{\mathrm{I}} \ . \qquad (14)$$

V. Example Analysis

The method is demonstrated on an analysis of a current-feedback amplifier in Matlab. Parametric preprocessing reduced the original 100 network parameters to 14. The topological algorithm separated two subgraphs from the voltage graph and one subgraph from the current graph. The controlling procedure checked for magnitude and phase errors ($\Delta M_{\max}$, $\Delta\varphi_{\max}$) at the specified frequencies. The formula for voltage transfer was further simplified by means of the sensitivity-based SAG postprocessor of [1].

TABLE I. Results of Simplification

accuracy		parametric	graph-based	SAG
$\Delta M_{\max}$	allowed	± 1dB @ 1kHz and 600kHz		
	actual	0.47dB@1kHz	0.56dB@600kHz	0.21dB@1kHz
$\Delta\varphi_{\max}$	allowed	± 3° @ 1kHz and 600kHz		
	actual	2.4°@600kHz	2.4°@600kHz	3.0°@600kHz
numerator terms		44	4	3
denominator terms		232	65	46
parameters		14	14	14
runtime		4.9s	1.3s	0.9s

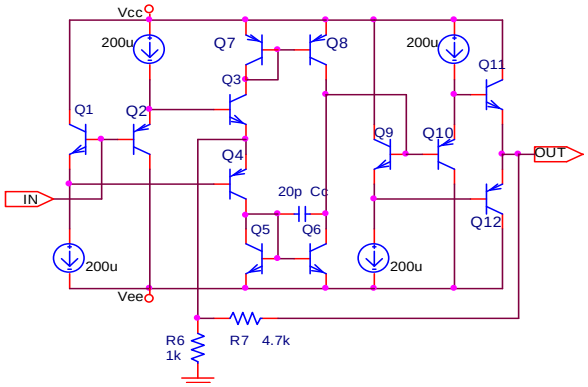

Figure 6. Current-feedback amplifier.

VI. Conclusion

The method of topological simplification consists in such transformations of the voltage and current graphs that decrease the number of common spanning trees. The occurrence of alien trees is avoided by the operations.

References

[1] Hennig E. *Symbolic Approximation and Modeling Techniques for Analysis and Design of Analog Circuits*. Ph.D. Dissertation, Univ. of Kaiserslautern, 2000.

[2] Fernández FV, Rodríguez-Vázquez A, Martín JD, Huertas JL. Formula Approximation for Flat and Hierarchical Symbolic Analysis. Int *J Analog Integrated Circuits and Signal Processing*, vo1. 3, no. 1, 1993, pp. 43–85.

[3] Daems W et al. Evaluation of Error-Control Strategies for the Linear Symbolic Analysis of Analog Integrated Circuits. *IEEE Trans Circ Syst.-I*, vol. 46, no. 5, 1999, pp. 594-606.

[4] Fernández FV, Rodrígues-Vázques A, Huertas JL, Gielen G. *Symbolic Analysis Techniques–Applications to Analog Design Automation*. IEEE Press, 1998.

[5] Daems W, Gielen G, Sansen W. Circuit Simplification for the Symbolic Analysis of Analog Integrated Circuits. *IEEE Trans Comp Aided Design In. Circ Syst*, vol. 21, no. 4, 2002, pp. 395-407.

[6] Sommer R, Hennig E, Dröge G. Equation-Based Symbolic Approximation by Matrix Reduction with Quantitative Error Prediction. *Alta-Frequenza – Rivista di Elettronica*, vol. 5, no. 6, 1993, pp. 29-37.

[7] Hsu JJ, Sechen C. DC small signal symbolic analysis of large analog integrated circuits. *IEEE Trans Circ Syst-I*, vol. 41, no. 12, 1994, pp. 817-828.

[8] Yu Q, Sechen C. A Unified Approach to the Approximate Symbolic Analysis of Large Analog Integrated Circuits. *IEEE Trans Circ Syst-I*, vo. 43, no. 8, 1996, pp. 656-69.

[9] Mayeda W, Seshu S. *Topological Formulas for Network Functions*. Eng. Exp. Station, University of Illinois, Bulletin 446, Urbana, 1957.

[10] Lin PM. *Symbolic Network Analysis*. Amsterdam: Elsevier, 1991.