Circuit Analyses with Nullors

Mihai Iordache, Lucia Dumitriu, Dragos Niculae,
Marilena Stanculescu, Victor Bucata and Georgiana Rezmerita

Abstract This chapter describes the nullor-based modelling of active devices from the circuit level of abstraction. After a brief overview on the nullor concept and its properties, the modelling of active devices not only at the voltage-mode but also at the current-mode and the mixed-mode of operation from two-port and four-terminal network point of view are described in details. The circuit analysis with nullors and the topological approach for transfer function generation by two-graph tree enumeration as well are presented. The generalized topological formula with homogeneous parameters is proved for all the circuit functions, and a simple representation of the four types of controlled sources by admittances is proposed, that allows a uniform treatment of the entire circuit in terms of admittances. In order to implement the procedure, the rules to automatically generate the two graphs and to enumerate the common spanning trees are presented. Some simplifications in the circuit and in the two graph structure before tree generation and a graph representation on levels, improve the efficiency of the tree enumeration procedure. The original approach, in which each edge is labelled with an admittance term, could handle only one type of active element, namely VCCS (voltage controlled current source), but the method was further developed by many researchers for general linear circuits to include virtually all active elements. Some techniques to convert the CCVSs (current controlled voltage sources), VCVSs (voltage controlled voltage sources) and CCCSs (current controlled current sources) in equivalent schemes containing only VCCSs together with admittances and the inductance modelling proposed in the literature are discussed.

1 Introduction

According to the symbolic analysis principles, the Nodal Analysis Method (NAM) is restrictive because the admittance matrix must contain only the elements compatible with the Nodal Analysis (NA). The problem can be easily resolved

M. Iordache (✉) · L. Dumitriu · D. Niculae · M. Stanculescu · V. Bucata · G. Rezmerita
POLITEHNICA University of Bucharest, Bucharest, Romania
e-mail: mihai.iordache.44@gmail.com; mihai.iordache@upb.ro

M. Fakhfakh and M. Pierzchala (eds.), *Pathological Elements in Analog Circuit Design*, Lecture Notes in Electrical Engineering 479,
https://doi.org/10.1007/978-3-319-75157-3_4

through the Modified Nodal Analysis Method (MNAM), adding a row and a column for each element which is not compatible with the classic Nodal Analysis Method [1–7]. One of the problems generated by this kind of approach is related to the size of the admittance matrix. This matrix will become bigger, according to the structure of the circuit and types of its elements.

Regarding the models to be used in the analogue circuit analysis, the requirement of a high accuracy could lead to complicated calculations and then compact models are preferred mainly for the use of much more simple equations [3–7]. These models are more effective for the optimization of design and simulation time during the analysis process. From this point of view, the nullors proved already their efficiency in the active devices modelling. In the models based on nullors, the parasitic elements can be included to analyze their contribution to the analogue circuit response. All the four controlled sources can also be represented with equivalent circuits using nullor elements. Consequently, the nullors are very useful for the analogue circuits modelling because the circuit topology can be described using only two-terminal components like resistors, capacitors, nullators, norators, independent and controlled sources. Considering that the model should be developed in the simplest manner and the accuracy of the circuit behaviour simulation must be in acceptable limits, this chapter will show the problems related to the small-signal models of the active devices modelled with nullors.

The nullator is an ideal circuit with two terminals (Table 2.1a), which is characterized by null values for the current and voltage at the terminals. It has two equations: $i = 0$, $v = 0$.

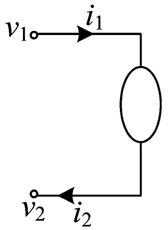
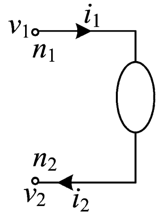
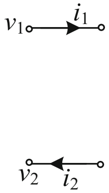
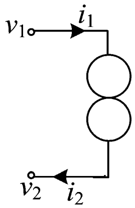
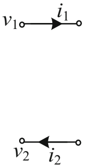
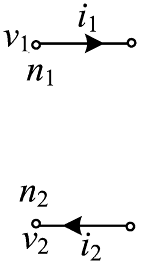
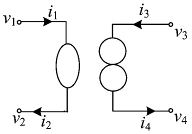
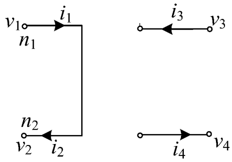
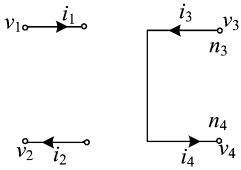
The norator is an ideal circuit with two terminals (Table 1b), which is characterized by random values for the current (i) and voltage (v) at the terminals. In other words, the norator does not have any equation. The current and the voltage values of this element are affected only by the external circuit connected to its terminals.

These two circuit elements can be used only in norator-nullator pairs called nullors (Table 1c), which has the number of equations equal to the number of gates. The nullor can be considered as an idealized operational amplifier, which has at the input gate null voltage and current and at the output gate an undetermined voltage and current (obtained by multiplying the null inputs by an infinite factor gain). In Fig. 1d, e is presented the symbol for the current (voltage) mirror.

Techniques for the analysis of linear/linearized circuit have been performed using the nullator and norator as theoretical active devices, [6–14]. Tellegen was the first who presented the ideal operational amplifier theory and later, in 1964, Carlin considered nullators and norators as single active devices in the circuit analysis—called nullor [5]. He thought that these active devices cannot be built physically. Tellegen also took in consideration that these devices must be seen only as mathematical models without any physical support. Table 1 presents the behaviour of the nullators, norators and nullors from the point of view of the voltage, respectively of the current, in G^v—the voltage graph and, respectively G^i—the current graph, [1–9].

The input port of the nullor is modelled by the nullator which is characterized by two equations:

Table 1 The behaviour of the nullators, norators and nullors

Symbol	Definitions	Voltage graph G^v	Current graph G^i
 Nullator (a)	$v_1 = v_2$ $i_1 = i_2 = 0$	 $v_1 = v_2 \Rightarrow n_1 \equiv n_2$ any $i_1 = i_2$	 any v_1, v_2 $i_1 = i_2 = 0$
 Norator (b)	any v_1, v_2 any $i_1 = i_2$	 any v_1, v_2 $i_1 = i_2 = 0$	 $v_1 = v_2 \Rightarrow n_1 \equiv n_2$ any $i_1 = i_2$
 v_1 i_1 i_3 v_3 v_2 i_2 i_4 v_4 Nullor (c)	$i_1 = i_2 = 0$ $v_1 = v_2$ any v_3, v_4 any $i_3 = i_4$	 v_1 i_1 n_1 i_3 v_3 n_2 v_2 i_2 i_4 v_4 $v_1 = v_2 = \text{arbitrary}$ $\Rightarrow n_1 \equiv n_2$, any $i_1 = i_2$, $i_3 = i_4 = 0$, any $v_3 \neq v_4$	 v_1 i_1 i_3 v_3 n_3 n_4 v_2 i_2 i_4 v_4 any $v_1 \neq v_2$ v_1, $i_1 = i_2 = 0$; $v_3 = v_4 \Rightarrow n_3 \equiv n_4$ any $i_3 = i_4$

$$v_1 = v_2 = \text{arbitrary},\ i_1 = i_2 = 0. \tag{1}$$

So, the nullator is simultaneously an open-circuit in G^i current graph and a short-circuit in G^v voltage graph. The output port of the nullor is modelled by the norator where both, the voltage and the current have arbitrary values:

$$v_1 \neq v_2 = \text{arbitrary},\ \ i_1 = i_2 = \text{arbitrary} \tag{2}$$

With these properties the nullor is a two-port element accepted as a universal active element [1–16, 30–34]. This concept means that the nullor along with capacitors and resistors can be used to design a maximum number of functions with the minimum number of active devices. If a suitable set of linear and nonlinear

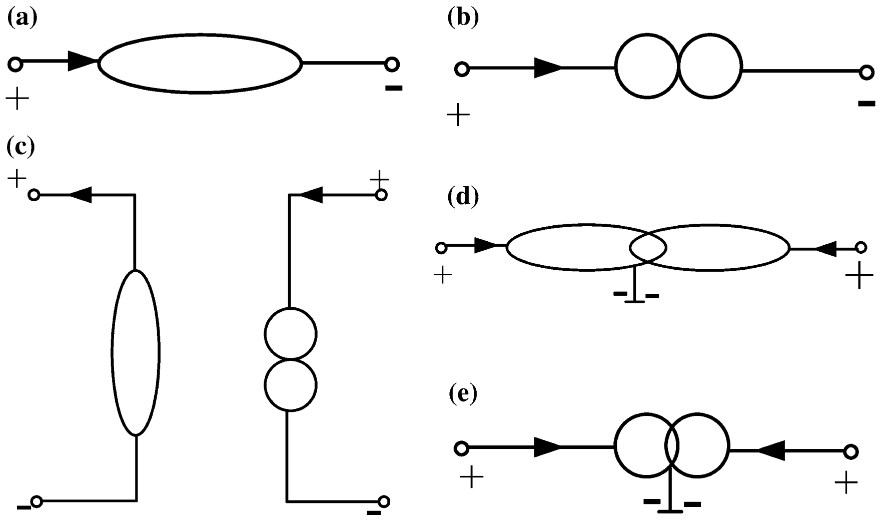


Fig. 1 **a** Nullator symbol; **b** Norator symbol; **c** Nullor symbol; **d** Current mirror; **e** Voltage mirror

passive elements is available, then no active element other than nullors are needed to implement any linear or nonlinear circuit function. So nullators, norators, resistances, along with capacitances can synthesize a complete set of linear or linearized equations.

2 Nullor Equivalences

From the beginning, the nullor circuit has been considered very efficient for the analog circuit analysis, modelling and synthesis. Therefore, there are many records regarding methods and algorithms based on nullor circuits, used for the active devices analysis and modelling [19–34]. Because any analog network can be modelled with nullators, norators and impedances, it is useful to mention the equivalence between some connections. These are shown in Fig. 2. For instance, in Fig. 2a, a current cannot flow from a to b since the current through the nullator is zero, so a series connection of the nullator and norator is equivalent to an open-circuit. In Fig. 2b, the current can flow from a to b through the norator, also the voltage across a and b becomes zero according to the property of the nullator, so a parallel connection of the nullator and norator is equivalent to a short-circuit. The remaining connections have equivalences according to the nullator and norator i–v characteristics.

In another approach, the nullors along with grounded resistors can be manipulated in order to obtain inverting properties, features that the nullator and the norator

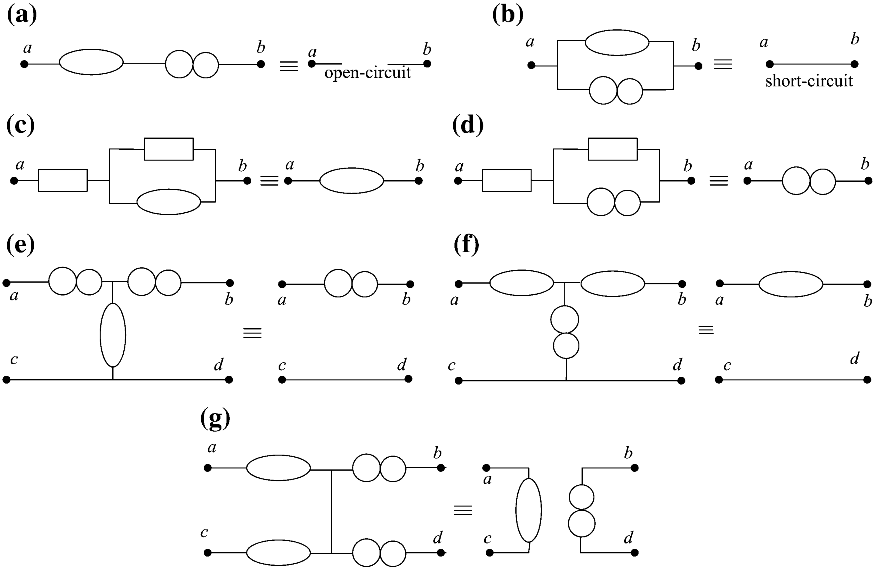


Fig. 2 Nullator and norator equivalences

cannot model by themselves [30, 31]. The main purpose of the introduction of the inverting properties is that the behaviour of some active devices involves inverting the voltage and current input-signals. In this sense, the Current-Mirror (CM) and the Voltage-Mirror (VM), both as active devices, can perform this task and their behaviour also should be modelled with nullors, [30–34]. Thus, by manipulating the nullor along with grounded resistors, the behaviour of a CM or of a VM, both with ideal unity-gain can easily be obtained, as shown in Fig. 3, [1].

Therefore, by analyzing the equivalent circuits, one can see that the VM, shown in Fig. 3a, is characterized by:

$$v_2 = -v_1 = \text{ arbitrary}, \; i_1 = i_2 = 0. \tag{3}$$

and the CM, shown in Fig. 2b, is characterized by:

$$v_2 \neq v_1 = \text{arbitrary}, \; i_1 = i_2 = \text{arbitrary} \tag{4}$$

At the end, the inverting behaviour of the nullator and norator is achieved. In [24, 31, 32], the nullor—based models of the VM and CM include parasitic elements. In the same manner as for the nullor, equivalences between the combinations of nullators, norators, CMs, VMs and impedances can be obtained. Note, however, that if v_1 or v_2 terminal from Fig. 3a is grounded and by applying the equivalences shown in Fig. 1, the VM is reduced to a nullator. In the same manner, if any terminal in Fig. 3b is grounded by applying the equivalences shown in Fig. 2, then a norator is obtained.

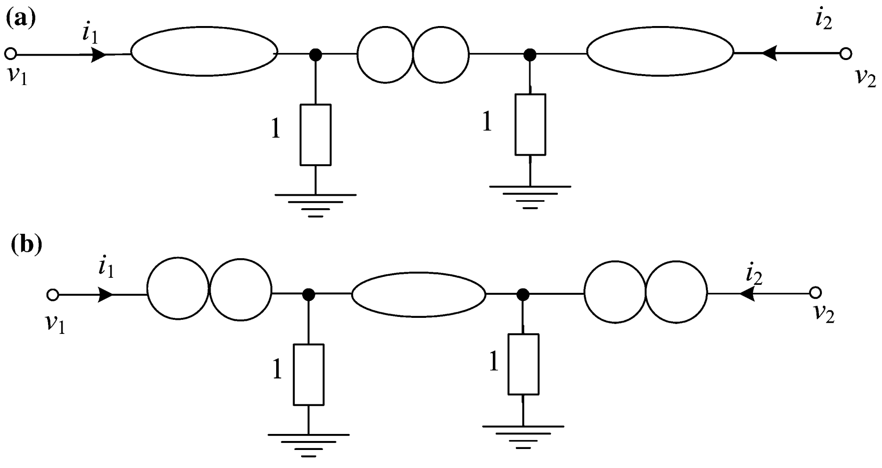


Fig. 3 Nullor and grounded resistor-based VM (**a**) and CM (**b**)

3 Loop Current Method for Circuits with Nullors

As it is well-known, the loop current method is based on introducing the loop currents as intermediary quantities which satisfy the first Kirchhoff's current law (KCL) and which can be determined by applying the Kirchhoff's voltage law (KVL) on the independent loops of the electric circuit.

Taking into account the definition of the nullator as a circuit element through which the current does not flow, it is useful and recommended to choose loop currents such that they do not flow through the branches that contain nullators. In order to respect such a condition, the branches containing the nullators should be eliminated by introducing an open-circuit between the terminals at which a nullator is connected. This leads to a decrease of the number of independent loops (l_{i}) with the nullator number (n_n)

$$l_i = b - n + 1 - n_n, \tag{5}$$

where: b—is the number of the circuit branches and n—is the number of the circuit nodes.

Applying KVL on the independent loops l_{i} a system of independent equations results from which we further can determine the loop currents.

The branch currents are expressed as an algebraic sum of the loop currents that flow through the respective branch.

If the electric circuit contains current sources, the branches which contain such sources cannot belong to a tree; a single loop current will be chosen to flow through such branch. The loop current value will be given by the source current.

In order that the system of l_{i} equations does not contain as unknowns the norator voltages, the l_{i} independent loops must not contain branches with norators.

The norator branches are replaced by open-circuits while the branches with nullators are kept.

The loop current equations corresponding to a number of l_{i} loops become:

$$\sum_{j=1}^{l_i}\left(\sum_{h\in\left[l_j\right]\cap\left[l_k\right]} R_h\right) I_{l_j} = \sum_{h\in\left[l_k\right]} E_h, \tag{6}$$

where: I_{l_j}—is the loop current corresponding to the l_j loop and E_h—is the *e.m.f* of b_h branch.

If we consider the current and voltage graphs with their loop-branch incidence matrices $\mathbf{B}^i$ and $\mathbf{B}^v$(see Table 1), then the matrix form of the loop current equations, [7, 9–14], can be written as follows:

$$\left(B^v R_b \left(B^i\right)^t\right) I_b^i = B^v\left(E_b + R_b J_b\right), \tag{7}$$

where, for example, $\mathbf{I}_b^i$ ($\boldsymbol{R}_b$) is the loop current vector in the current graph G^i (the diagonal matrix of the branch resistances).

Example 1 See (Figs. 4 and 5).

The loop current equations are obtained by applying the KVL on the independent loops from the voltage graph (Fig. 6) and taking the currents from attached to the loops from the current graph (Fig. 5). Proceeding in this manner, it results the following system of Eqs. (8a) and (8b)

$$\begin{cases} R_3 \cdot I_{l_1} + R_5 \cdot I_{l_1} + R_4 \cdot I_{l_1} - R_5 \cdot I_{l_2} = 0 \\ R_5 \cdot I_{l_2} + R_2 \cdot I_{l_2} - R_5 \cdot I_{l_1} = -E_2 \end{cases}. \tag{8a, b}$$

From Eq. 8b it results:

$$I_{l_2} = \frac{-E_2 + J_1 \cdot R_5}{R_2 + R_5}. \tag{9}$$

From Eq. 8a it results:

$$I_{l_3} = \frac{-J_1 \cdot \left(R_3 + R_5\right) + R_5 \cdot \left(\frac{-E_2 + J_1 \cdot R_5}{R_2 + R_5}\right)}{R_4}. \tag{10}$$

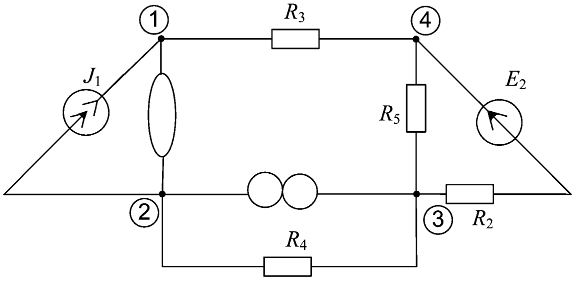


Fig. 4 Initial circuit to be analyzed using loop current method

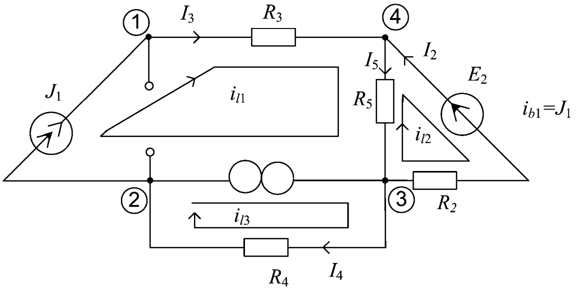


Fig. 5 Choosing of the loop currents

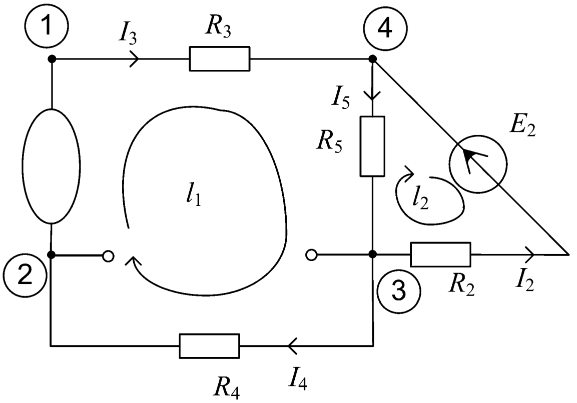


Fig. 6 Loops for KVL writing

4 Nodal Analysis Method for Circuits with Nullors

The unknown variables of this method are represented by the $n - 1$ electric potentials corresponding to the circuit nodes, excepting the nth node whose potential is the reference potential and it is considered to be zero. These unknowns satisfy KVL for any circuit loop. The computation of these unknowns is based on KCL written in $n - 1$ nodes and on the generalized Ohm's law to express each branch current depending on the node potentials.

The equations of the node potentials for the circuits containing nullators will have a different form taking into account that the voltage at the nullator terminals is equal to zero, which results in a decrease of the number of the unknown potentials (Fig. 7).

If the circuit contains norators, the norator currents should not be present in the system of equations. This is why we choose sections that do not include the branches with norators.

$$\sum_{j=1}^{N-1}\left(\sum_{h\in\left[l_j\right]\cap\left[l_k\right]} G_h\right)V_j = -\sum_{h\in[l_k]}\left(J_h+\frac{E_h}{R_h}\right). \tag{11}$$

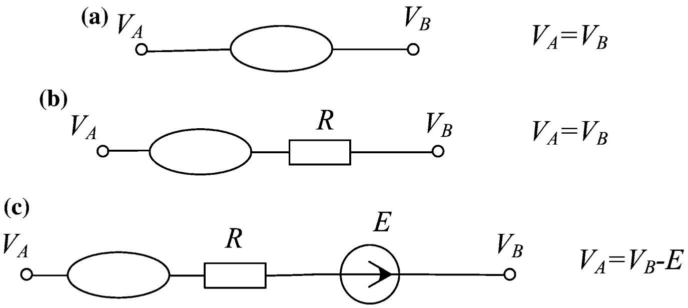


Fig. 7 The equations of the node potentials for the branches containing nullators

Considering the current and voltage graphs with their reduced node-branch incidence matrices $\mathbf{A}^i$ and $\mathbf{B}^v$(see Table 1) the matrix form of the nodal equations, [7, 9–14], is:

$$\left(\mathbf{A}^i\mathbf{G}_b(\mathbf{A}^v)^{\mathrm{t}}\right)\mathbf{V}^v_{n-1} = -\mathbf{A}^i(\mathbf{G}_b\mathbf{E}_b + \mathbf{J}_b), \tag{12}$$

where, for example, $\mathbf{V}^v_{n-1}$ ($\boldsymbol{G}_b$) is the potential vector of the $n-1$ independent nodes from the voltage graph G^v (the diagonal matrix of the branch conductances).

Example 2 The nodal analysis method for the circuits containing nullors can be applied as follows: KCT is written in the independent nodes of the current graph and there are used the potentials associated to the $n-1$ independent nodes from the voltage graph. Applying the Nodal analysis method, we obtain the following Eqs. (13)–(18) (Fig. 8):

Appling the KCL in the node (n_1) it results:

$$\frac{V_1}{R_3} = J_1. \tag{13}$$

According to the KCL on the cut-set (S_2) to obtain:

$$\frac{V_2}{R_5} + \frac{V_2}{R_2} = -J_1 - \frac{E_2}{R_2}. \tag{14}$$

From Eq. (13) it results:

$$V_1 = R_3 \cdot J_1. \tag{15}$$

From (14) we can obtain:

$$V_2 = \frac{R_5 \cdot R_2 \cdot \left(-J_1 - \frac{E_2}{R_2}\right)}{R_2 + R_5}, \tag{16}$$

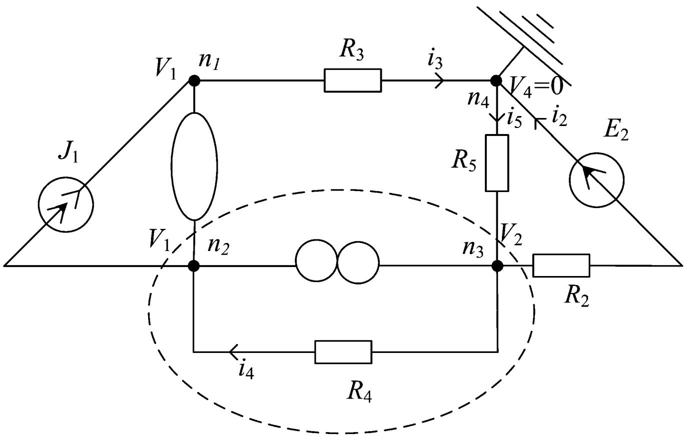


Fig. 8 Initial circuit to be analyzed using Nodal Analysis Method

$$I_4 = \frac{V_2 - V_1}{R_4} = \frac{\frac{R_5 \cdot R_2 \cdot \left(-J_1 - \frac{E_2}{R_2}\right)}{R_2 + R_5} - R_3 J_1}{R_4}, \tag{17}$$

$$I_2 = \frac{V_2 - V_4 + E_2}{R_2} = \frac{\frac{R_5 \cdot R_2 \cdot \left(-J_1 - \frac{E_2}{R_2}\right)}{R_2 + R_5} + E_2}{R_2}. \tag{18}$$

The values obtained for I_2 and I_4 are the same as those obtained by using the loop method.

5 The Generalized Topological Formula for Transfer Function Generation by Two-Graph Tree Enumeration

5.1 Introduction

One of the most important approaches for nonreciprocal circuit analysis is the two-graph tree enumeration method, mainly due to Mayeda and Seshu [16]. The original approach, in which each edge is labelled with an admittance term, could handle only one type of active element, namely VCCS, but the method was further developed by many researchers for general linear circuits to include virtually all active elements. In [17] some techniques to convert the CCVSs, VCVSs and CCCSs in equivalent schemes containing only VCCSs and admittances are introduced, and some techniques to model an inductance proposed in the literature are discussed. The resulted models have a bigger number of branches in the two graphs and some supplementary nodes are introduced in the original circuit. The method based on a two-graph representation using a unity gyrator to model the non-admittance components was implemented [18] in order to obtain symbolic network function expressions in other terms than admittances. The price paid by all these approaches consists in the increase of the circuit complexity leading to a bigger number of common spanning trees.

Topological formulas for transfer functions of active networks using tree enumeration method have been derived firstly by Mayeda and Seshu, but their procedure for determining the sign factor is tedious. In [15] the main results in this direction up to that date are presented, and similar formulas are obtained while modelling all the controlled sources by equivalent schemes containing only two terminal elements and VCCSs. A sorting scheme is preferred to obtain symbolic network functions from the node determinant of an augmented network.

Based on the original concepts of the two-graph tree enumeration method a modelling technique of the four types of controlled sources has been elaborated and a topological formula with homogeneous parameters for the transfer admittance has been proved using the nodal approach [7]. Some innovative approaches to symbolic generation of the transfer functions have been developed: an algorithm using systematic loop opening and closing, a diakoptic approach, and a procedure based on graph decomposition on levels [7–14].

In this chapter, a set of rules for generating and using the two graphs is stated, and the generalization of the topological formula to generate all network functions is proved. These rules are applicable to a linear circuit containing: all four types of linear controlled sources, resistors, inductors, capacitors, nullors (for ideal opamps operating in the linear mode), and any multi-terminal or multiport circuit element having an equivalent scheme made up only by two-terminal elements and controlled sources. The generalized topological formula with homogeneous parameters that we propose to generate the transfer functions, can handle our models for the four types of controlled sources in a very efficient manner. Performing some reductions in the structure of the two graphs and representing them on levels we obtain an important improvement of the common tree enumeration process.

In Sect. 5.2 of this chapter we obtain the equivalent schemas in admittances that model in the two graphs the four types of controlled sources starting from the functional schemas with nullors. This representation makes possible the proof of the generalized topological formula with homogeneous parameters, valid for any transfer function of a lumped, linear and time-invariant circuit. Section 5.3 is dedicated to this proof. It is shown that the numerators of all the four types of transfer functions are identical and the treatment of the input/output ports according to the transfer function to be generated is given.

Section 5.4 is dedicated to a very efficient algorithm for tree enumeration in a graph represented on levels, which was implemented for network function generation, and Sect. 5.5 describes an efficient algorithm for sign factor generation. In Sect. 5.6 the rules for automatic generation of the network functions are introduced, and some techniques to increase the efficiency of the common spanning tree enumeration are discussed. The entire procedure of network function generation including simplification after generation is illustrated in Sect. 5.7.

5.2 Controlled Source Modelling in the Two Graphs

Consider two-port containing only linear passive two-terminal elements (resistors, capacitors, and inductors). It is well known that any circuit function can be written as a ratio of admittance polynomials using Kirchhoff's topological formula. Each monomial in these polynomials corresponds to the admittance value of a tree. This property leads to a circuit graph whose edges have the admittances as their weights.

Kirchhoff's type topological formulas have been developed by Mayeda and Seshu [16] for circuits containing linear passive two-terminal elements and voltage controlled current sources (VCCS) only. In these formulas each monomial corresponds to a common tree in the current graph G^i and the voltage graph G^v in which each passive element is represented by an edge having the admittance as its weight; a VCCS is modelled by an edge with the same weight (the control admittance) but with different positions in the two graphs: the position of the controlling branch in G^i and the position of the controlled branch in G^v. G^i is used to write the Kirchhoff's current law while G^v is used for the Kirchhoff's voltage law. The constitutive equations of all circuit elements are written as relationships between the G^i currents and G^v voltages.

Consider now a circuit containing two terminal elements and control sources of any type. In order to extend the abovementioned formulas to circuits with passive two-terminal elements and any type of controlled sources we build equivalent schemes of these sources using nullators and norators (nullors). A nullor equivalent scheme of a controlled source leads to its G^v and G^i representations considering the following properties: from the current point of view the nullator is an open-circuit while the norator is a short-circuit, and from the voltage point of view the nullator is a short-circuit while the norator is an open-circuit.

Starting from the equivalent schemes with nullors in Fig. 9, the two graph models of the controlled sources using only two terminal admittances can be built as it is shown in Fig. 9. The parameters associated with the controlled and the controlling branches are presented in Table 3. The subscript C is used for the controlling branch and the subscript c for the controlled one.

As it is shown in Fig. 9 the four types of controlled sources are modelled in the two graphs as follows:

- CCVS is modelled by a branch having the transfer impedance subscript identical with the controlled branch $Z_c = Z_{cC}$, having as parameter $Y_c = 1/Z_{cC}$, and which takes distinct positions in the two graphs:
  - In G^i it is connected to the controlling port, and it is oriented like the controlling current, the controlled branch being short-circuited;
  - In G^v this branch is connected to the controlled port, having the same direction with the voltage across this branch, the controlling branch being short-circuited.

In this way, a CCVS leads to a node contraction in each graph: in G^i the nodes of the controlled branch coincide, while in G^v the nodes of the controlling branch coincide. In order to keep the numbering of nodes in natural order (that is especially

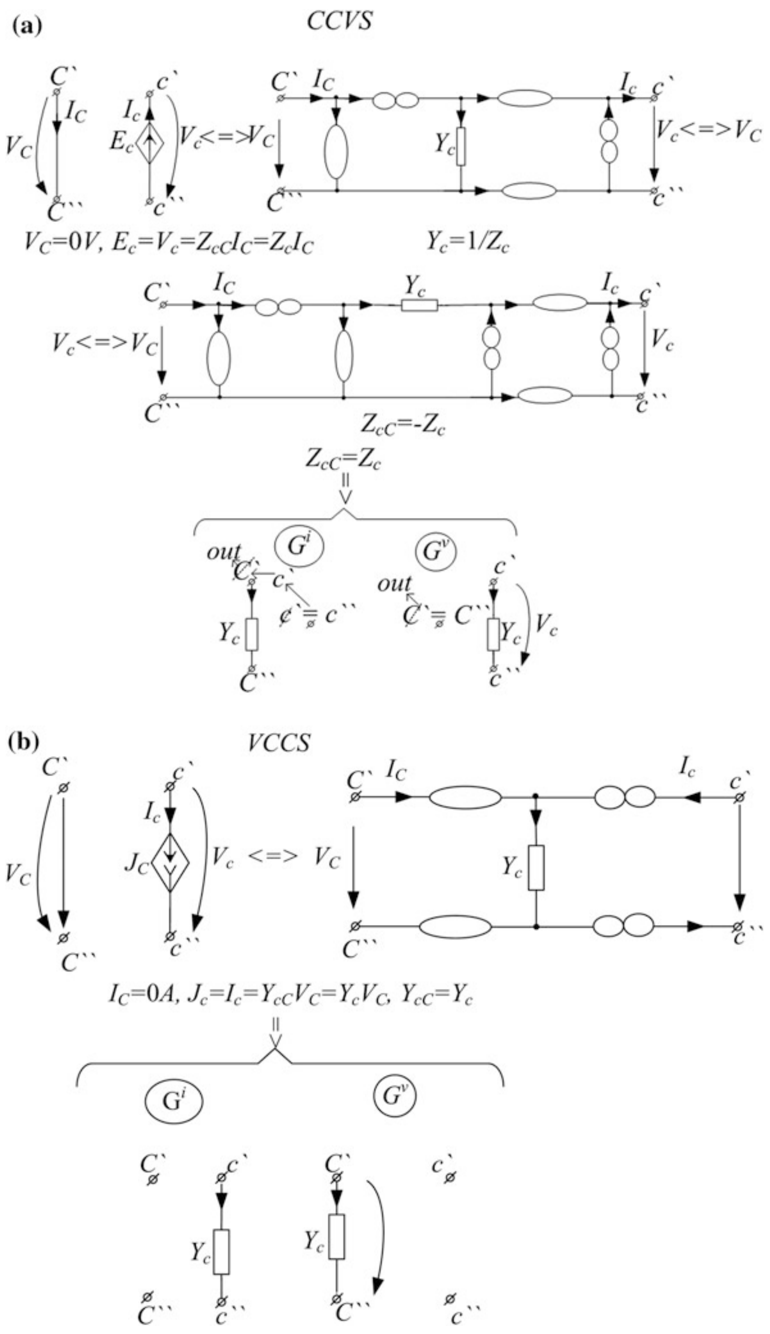


Fig. 9 Controlled source modelling in the two graphs

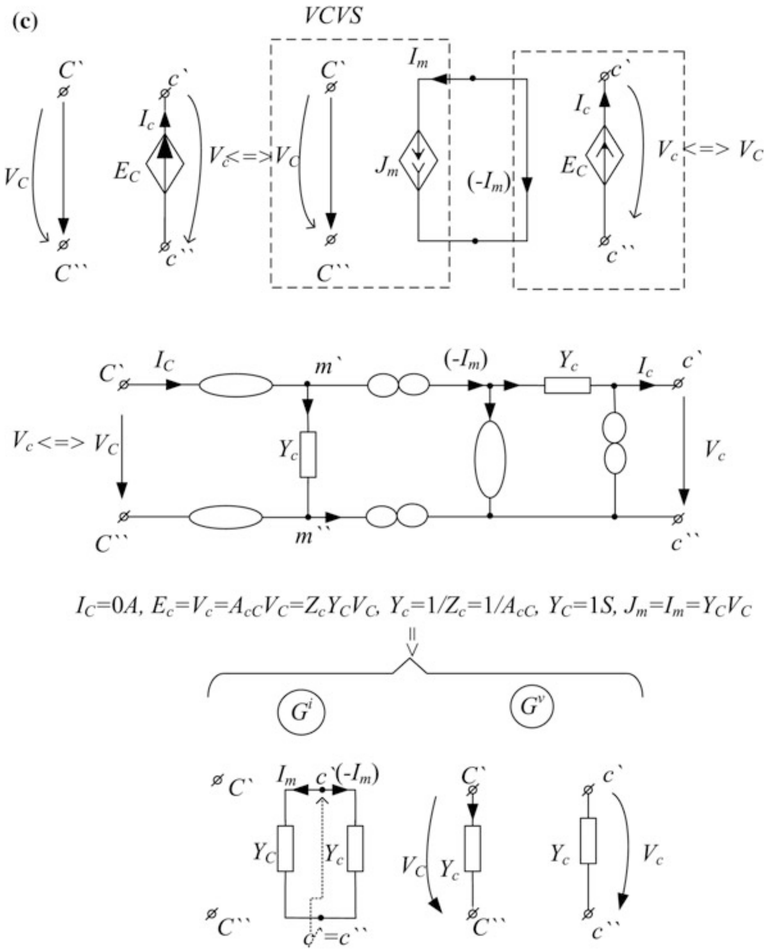


Fig. 9 (continued)

useful in tree enumeration and in the sign factor computation), we reduce by one all node numbers greater than the number of the eliminated node.

For programming needs we keep in G^i the node towards the voltage across the controlled branch is oriented (*c*''), the other node number (*c*') being allocated to the new node introduced to identify the controlling branch.

- VCCS is modelled by a branch having the transfer admittance subscript identical with the controlled branch $Y_c = Y_{cC}$, and which takes distinct positions in the two graphs:
  - In G^i it is connected to the controlled port, and it is oriented like the controlled current, the controlling branch being open;

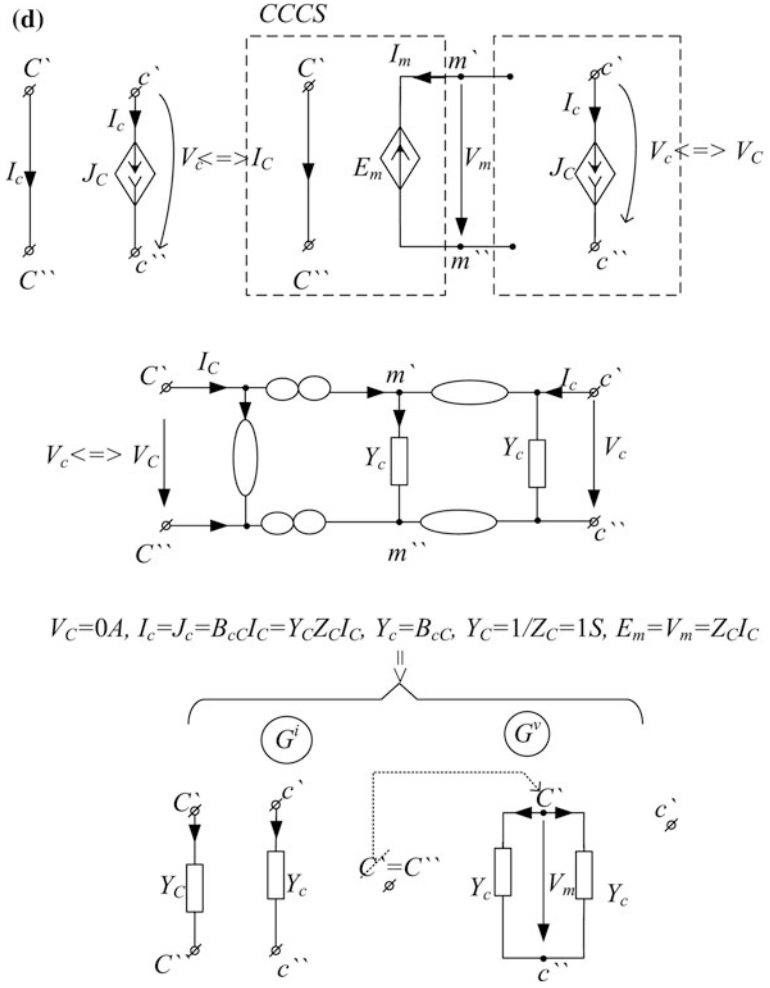


Fig. 9 (continued)

- In G^v this branch is connected to the controlling port, having the same direction with the controlling voltage, the controlled branch being open.

This controlled source does not modify the number of the two graph nodes.

- A VCVS is equivalent with a VCCS ($J_m = I_m = Y_C V_C$), in cascade with a CCVS with negative trans-impedance ($E_c = (-Z_c) \cdot (-I_m)$), and it is modelled
  - In G^v by two branches having the controlled branch number respectively that of the controlling branch, and the parameters presented in the Table 2; they are connected to the controlled branch, respectively to the controlling one,

having the direction of the voltage across the controlled source, and respectively of the controlling voltage;

- In G^i the two branches are connected in series, having the nodes *m'* and *m"* that are supplementary nodes. In order to keep the current graph node numbering, the number of the node *m"* will be that towards the voltage across the controlled branch is oriented (*c"*≡ *m"*); the number of the eliminated node (*c'*) will be attached to the other one (*c'*≡ *m'*). The controlled branch is oriented from *c"* to *c'*, and the controlling one from *c'* to *c"*.

- A CCCS is equivalent with a CCVS connected in cascade with a VCCS, being modelled
  - In G^i by two branches having the controlled branch number and respectively those of the controlling branch, and the parameters presented in the Table 2; these branches are connected to the controlled port, respectively to the controlling one, having the direction of the controlled current, respectively of the controlling one;
  - In G^v the two branches are connected in parallel and they have two common nodes, namely the node in which the controlled current goes in (*c"*) and the other one having the number of the node eliminated by short-circuiting of the controlling branch in G^v (*C'*). The two branches have the same direction in respect of their terminals.

Remarks

1. The other circuit elements (resistors, uncoupled inductors, capacitors) keep in the two graphs the same position as in the initial circuit, and are represented by their admittances.
2. The magnetic couplings are modelled by inductors and CCVSs [10].
3. The above modelling technique of the four controlled sources leads to two directed graphs having admittance branches only.
4. The two graphs have the same number of nodes, branches and loops. They differ only by the location of the controlling and controlled branches of the four types of controlled sources.
5. Because any branch contraction in the two graphs causes the elimination of one node, the number of nodes in G^i and G^v is smaller than in the initial circuit with the number of CCVSs: $n_{G^i} = n_{G^v} = n - n_{CCVS}$.

In Table 3 is given a comparison with some reported techniques taking into account the number of branches used to model the circuit elements in both graphs and the supplementary node number.

Table 2 Controlled source equations

Controlled	Equations	Associated	Parameter
Source		Controlled branch	Controlling branch
CCVS	$V_C = 0;$	$Y_c = 1/Z_c$	$Y_c = 1/Z_c$
VCCS	$I_C = 0;$	Y_c	Y_c
VCVS	$I_C = 0;\ E_c = V_c = A_{cC}V_C = Z_cY_CV_C$	$Y_c = 1/Z_c = 1/A_{cC}$	$Y_C = 1$ S
CCCS	$V_C = 0;$	$Y_c = B_{cC}$	$Y_C = 1/Z_C = 1$ S

Table 3 Comparison with some reported techniques

	Lin's	Models [17]	Rodanski's	Models [18]	Our	Models
Controlled source	Branches in the two graphs	Extra nodes	Branches in the two graphs	Extra nodes	Branches in the two graphs	Extra nodes
Resistor inductor capacitor	3	1	3	1	1	0
CCVS	4	1	5	2	1	−1
VCVS	3	1	3	1	2	0
CCCS	2 or 4	0 or 2	3	1	2	0

5.3 Generalized Topological Formula for Network Function Generation

Let us consider a linear nonreciprocal circuit (LNC) with null initial (i.c.) state and without independent sources and its associated model for operational calculus (Laplace). If we add to the input port an independent current source (Fig. 10), we can define the transfer impedance

$$Z_{oi} \overset{d}{=} \frac{V_o}{J_i}\bigg|_{I_o=0}. \tag{19}$$

The nodal equations of the circuit take the matrix form:

$$\mathbf{Y}_{n-1}\mathbf{V}_{n-1} = J_i, \tag{20}$$

where J_i can be expressed as

$$J_i = Y(V_{o'} - V_{o''}), \tag{21}$$

with

$$Y \overset{d}{=} \frac{1}{Z_{oi}}. \tag{22}$$

The Eq. (21) is equivalent to the substitution of J_i by a VCCS. Substituting Eq. (21) in (20) and rearranging we obtain:

$$\mathbf{Y}'_{n-1}\mathbf{V}_{n-1} = 0. \tag{23}$$

Consider the current and voltage graphs with their reduced node-branch incidence matrices $\mathbf{A}^i$ and $\mathbf{A}^v$.

Writing the Kirchhoff's current law in the current graph we obtain:

$$\mathbf{A}^i\mathbf{I}_b^i = 0, \tag{24}$$

where the branch currents can be expressed as:

$$\mathbf{I}_b^i = \mathbf{Y}_b\mathbf{V}_b^v. \tag{25}$$

The branch voltages in the voltage graph are:

$$\mathbf{V}_b^v = (\mathbf{A}^v)^{\mathrm{t}}\mathbf{V}_{n-1}^v. \tag{26}$$

Substituting (26) in (25) and the last one in (24) we obtain:

$$\mathbf{A}^i\mathbf{Y}_b(\mathbf{A}^v)^{\mathrm{t}}\mathbf{V}_{n-1}^v = 0. \tag{27}$$

If we denote

$$\mathbf{A}^i\mathbf{Y}_b(\mathbf{A}^v)^{\mathrm{t}} = \mathbf{Y}'_{n-1}, \tag{28}$$

we obtain (23).

Because the system (27) contains linear dependent equations it follows:

$$\det(\mathbf{A}^i\mathbf{Y}_b(\mathbf{A}^v)^{\mathrm{t}}) = 0. \tag{29}$$

$\boldsymbol{Y}_b$ being a symmetrical matrix, applying Binet-Cauchy theorem [6, 7] it results:

$$\det(\mathbf{A}^i\mathbf{Y}_b(\mathbf{A}^v)^{\mathrm{t}}) = \sum_{k=1}^{n_c} \Delta_k^i \Delta_k^v P_k, \tag{30}$$

where:Δ_k^i and Δ_k^v are determinants of order $n-1$, made up with elements of $\boldsymbol{A}^i$ and $(\boldsymbol{A}^v)^{\mathrm{t}}$ matrices, taking the k-th group of $n-1$ columns of $\boldsymbol{A}^i$ and respectively $n-1$ rows of $(\boldsymbol{A}^v)^{\mathrm{t}}$; P_k is the product of the operational branch admittances of $\boldsymbol{A}^i$ columns, respectively of $(\boldsymbol{A}^v)^{\mathrm{t}}$ rows that make up the k-th group; $n_c = C_b^{n-1}$.

Because Δ_k^i and Δ_k^v are nonzero if and only if the k-th groups of branches corresponding to the $n-1$ columns (rows) of $\boldsymbol{A}^i$ $((\boldsymbol{A}^v)^{\mathrm{t}})$ form trees in G^i (G^v) [7], (30) may be written as:

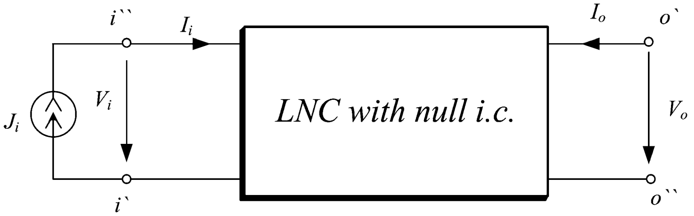


Fig. 10 The LNC transfer impedance definition

$$\det(\mathbf{A}^i\mathbf{Y}_b(\mathbf{A}^v)^{\mathrm{t}}) = \sum_{k=1}^{t_c} \Delta_k^i \Delta_k^v P_k = \sum_{k=1}^{t_c} \varepsilon_k P_k = 0, \tag{31}$$

where: t_c is the total number of common trees of G^i and G^v; P_k is the operational admittance product of the common tree T_k branches; ε_k represents the sign factor of the pair k of common trees.

In the expression (31) there are two kinds of terms: terms that contain the admittance Y, and the others that do not contain it, so that it follows:

$$\det(\mathbf{Y}'_{n-1}) = \sum_{k=1}^{t_c} \varepsilon_k P_k = YT_1(s) + T_p(s) = 0, \tag{32}$$

where:

$$T_1(s) = \sum_{k \in (T_{1c})} \varepsilon_k t_k, \tag{33}$$

$$T_p(s) = \sum_{k \in (T_{pc})} \varepsilon_k t_k, \tag{34}$$

and $\varepsilon_k = \pm 1$—is the sign factor for each common spanning tree of the pairs (G_1^i, G_1^v), respectively (G_p^i, G_p^v), where G_1^i (G_1^v) is the current (voltage) graph containing a unit weight branch at the input (output) port, and G_p^i (G_p^v) represents the current (voltage) graph in which the input/output ports are in short-circuit or open according to the generated network function (see Table 4); $T_{1c}(T_{pc})$ is the set of the common

Table 4 Treatment of the input/output ports

The circuit	The circuit port	
Function	Input	Output
Z_{oi}	Open	Open
Y_{oi}	Short-circuit	Short-circuit
A_{oi}	Short-circuit	Open
B_{oi}	Open	Short-circuit

spanning trees of G_1^i,G_1^v (G_p^i,G_p^v); t_k is the product term equal to the product of branch admittances of the common spanning tree k.

From Eq. (32) we obtain:

$$Y = -\frac{T_p(s)}{T_1(s)}. \tag{35}$$

According to (22) it results that

$$Z_{oi} = -\frac{T_1(s)}{T_p(s)}. \tag{36}$$

In the following we shall prove that, according to this approach, any transfer function of a lumped, linear, and time-invariant circuit, can be expressed in the form:

$$F_{oi} = -\frac{T_1(s)}{T_p(s)}, \tag{37}$$

all the four transfer functions having the same numerator, the denominator being different depending on the way the input and the output ports of the circuit are treated. From the above it results that the problem of generating all product terms in the irreducible expression of the transfer function is converted to the problem of finding all common spanning trees of the two graphs.

Let us consider a two-port circuit, containing any linear multi-terminal circuit elements that have an equivalent scheme made up only by two-terminal circuit elements and controlled sources. Modelling the controlled sources in the two graphs by two terminal circuit elements as in Fig. 9 allows a uniform treatment in admittances of the entire linear nonreciprocal circuit (*LNC*).

1. **Transfer impedance**

Using the circuit represented in Fig. 11, we define its transfer impedance as

$$Z_{oi}^{'} = \frac{V_o^{'}}{J_i}, \tag{38}$$

from which we can obtain the LNC transfer impedance:

$$Z_{oi} = \lim_{\substack{Y_i \to 0 \\ Yo \to 0}} Z_{oi}^{'} = -\frac{T_1(s)}{T_p(s)}, \tag{39}$$

where:

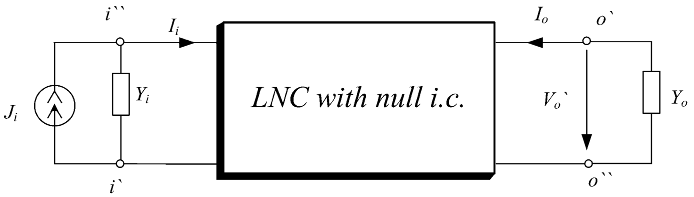


Fig. 11 The general scheme for transfer impedance definition

- T_1 is the sum of the algebraic values of the common trees in the graphs that contain the unity branch at the input (in G^i) respectively at the output (in G^v);
- T_p is the sum of the algebraic values of the common trees in G^i and G^v obtained by opening the input and output ports.

2. **Transfer admittance**

The transfer admittance of the circuit in Fig. 12 is:

$$Y'_{oi} = \frac{I'_o}{E_i} = \frac{Y_o V'_o}{J_i/Y_i} = Y_i Y_o Z'_{oi}, \tag{40}$$

and those of the LNC results as:

$$Y_{oi} = \lim_{\substack{Y_i \to \infty \\ Yo \to \infty}} Y'_{oi} = \lim_{\substack{Y_i \to \infty \\ Yo \to \infty}} Y_i Y_o Z'_{oi}. \tag{41}$$

Using the generalized Feussner formula for two branches we obtain:

$$\begin{aligned} Y_{oi} &= \lim_{\substack{Y_i \to \infty \\ Yo \to \infty}} Y_i Y_o Z'_{oi} = \\ &= -\lim_{\substack{Y_i \to \infty \\ Yo \to \infty}} Y_i Y_o \frac{T_1}{Y_i Y_o T'_{Y_{i,sc}Y_{o,sc}} + Y_i T'_{Y_{i,sc}Y_{o,op}} + Y_o T'_{Y_{i,op}Y_{o,sc}} + T'_{Y_{i,op}Y_{o,op}}} = \\ &= -\frac{T_1}{T'_{Y_{i,sc}Y_{o,sc}}} = -\frac{T_1}{T_p^Y}, \end{aligned} \tag{42}$$

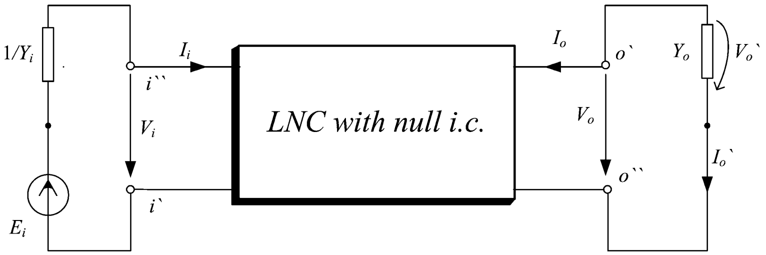


Fig. 12 The general scheme for transfer admittance definition

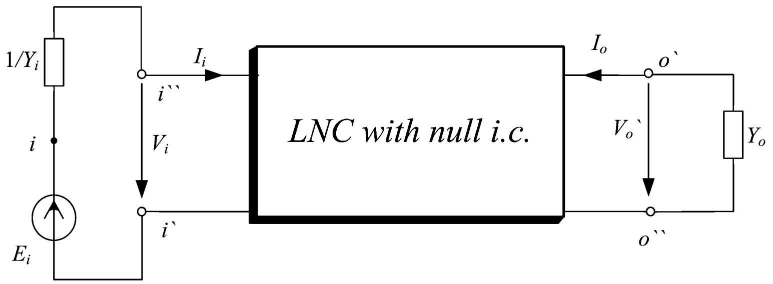


Fig. 13 The general scheme for voltage gain definition

where: T_1 is the same as in the case of Z_{oi}, and $T_p^Y = T'_{Y_{i,sc}Y_{o,sc}}$ is the sum of the algebraic values of the trees that are common to the G^i and G^v obtained by short-circuiting the input and output ports.

3. **Voltage Gain**

   Using the circuit in Fig. 13 we define the voltage gain as:

$$A'_{oi} = \frac{V'_o}{E_i} = \frac{V'_o}{J_i/Y_i} = \frac{Y_i V'_o}{J_i} = Y_i Z'_{oi}, \tag{43}$$

   from which we obtain the LNC transfer function

$$A_{oi} = \lim_{\substack{Y_i \to \infty \\ Yo \to 0}} A'_{oi} = \lim_{\substack{Y_i \to \infty \\ Yo \to 0}} Y_i Z'_{oi}. \tag{44}$$

   Applying the generalized Feussner formula we obtain:

$$\begin{aligned} A_{oi} &= \lim_{\substack{Y_i \to \infty \\ Yo \to 0}} Y_i Z'_{oi} = \\ &= - \lim_{\substack{Y_i \to \infty \\ Yo \to 0}} Y_i \frac{T_1}{Y_i Y_o T'_{Y_{i,sc}Y_{o,sc}} + Y_i T'_{Y_{i,sc}Y_{o,op}} + Y_o T'_{Y_{i,op}Y_{o,sc}} + T'_{Y_{i,op}Y_{o,op}}} = \\ &= - \frac{T_1}{T'_{Y_{i,sc}Y_{o,op}}} = - \frac{T_1}{T_p^A}, \end{aligned} \tag{45}$$

   where: T_1 is the same as in the case of Z_{oi} and Y_{oi}, and T_p^A is the sum of the algebraic values of the trees that are common to the G^i and G^v obtained by short-circuiting the input port and opening the output port.

4. **Current gain**

   For the circuit in Fig. 14 the current gain is:

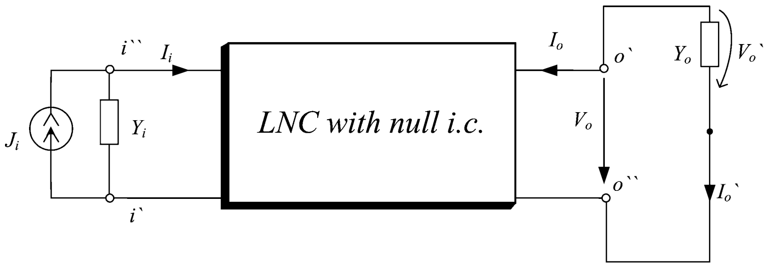


Fig. 14 The general scheme for current gain definition

$$B_{oi}^{'} = \frac{I_o^{'}}{J_i} = \frac{Y_o V_o^{'}}{J_i} = Y_o Z_{oi}^{'}, \tag{46}$$

and for the LNC we obtain

$$B_{oi} = \lim_{\substack{Y_i \to 0 \\ Y_o \to \infty}} B_{oi}^{'} = \lim_{\substack{Y_i \to 0 \\ Y_o \to \infty}} Y_o Z_{oi}^{'}, \tag{47}$$

that means

$$\begin{aligned} B_{oi} &= \lim_{\substack{Y_i \to 0 \\ Yo \to \infty}} Y_o Z_{oi}^{'} = \\ &= -\lim_{\substack{Y_i \to 0 \\ Yo \to \infty}} Y_o \frac{T_1}{Y_i Y_o T_{Y_{i,sc}Y_{o,sc}}^{'} + Y_i T_{Y_{i,sc}Y_{o,op}}^{'} + Y_o T_{Y_{i,op}Y_{o,sc}}^{'} + T_{Y_{i,op}Y_{o,op}}^{'}} = \\ &= -\frac{T_1}{T_{Y_{i,op}Yo,sc}^{'}} = -\frac{T_1}{T_p^B}, \end{aligned} \tag{48}$$

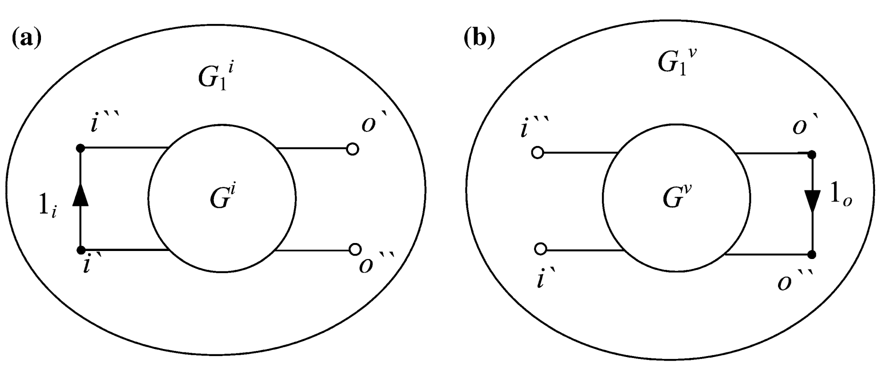


Fig. 15 Current and voltage graphs that contain the unit branch

where: T_1 is the same as in the above three cases, and T_p^B is the sum of the algebraic values of the trees that are common to the G^i and G^v obtained by opening the input port and short-circuiting the output port.

From the above analysis, we can conclude that to obtain all the transfer functions the input/output ports must be treated as in Table 4.

For the automatic generation of the graphs G_1^i and G_1^v the input gate of the analyzed circuit is connected to a current source that is controlled by the output gate voltage, which has the transfer admittance equal to the unit [27]. According to the Sect. 5.2 it will be represented in the two graphs as in Fig. 15.

5.4 Algorithm for Tree Enumeration in a Graph Represented on Levels

We have proved that the network functions generation by the topological method of tree enumeration, in the case of nonreciprocal circuits, means the generation of all common spanning trees. Since the number of graph trees increases rapidly with the graph size, a highly efficient algorithm is needed. This problem was widely studied and several algorithms of varying efficiency have been proposed in the literature. Ones of the well-known are Minty's algorithm which has the complexity $O(b + n + bt)$, and the algorithm due to Gabow and Myers having the complexity $O(b + n + nt)$, where b is the number of branches, n is the number of graph nodes, and t is the number of spanning trees.

The most used is, however, Char's algorithm, that some studies [23] show it to be superior to the other ones. This algorithm generates for the beginning an initial spanning tree which needs $O(b + n)$ operations, and starting from this one it enumerates all the spanning trees of the graph. During this enumeration, the algorithm generates also certain sequences which are not trees, called *non-tree sequences*. The original algorithm has the complexity $O(b + n + n(t + t_0))$, where t_0 is the number of non-tree sequences.

An implementation, called MOD-CHAR, of Char's spanning tree enumeration algorithm, introduces several heuristics for the selection of the initial spanning tree and for decreasing the number of the non-tree sequences. With these improvements for almost all graphs, the complexity of MOD-CHAR is $O(nt)$ [21–23]. It seems that for large dense graphs the complexity of MOD-CHAR algorithm is $O(t)$, being superior to Char's original algorithm, while for sparse graphs, it seems that Char's original implementation is superior to MOD-CHAR [22].

In the following we present an efficient algorithm for the enumeration of all the common spanning trees based on a representation on levels of the two graphs, and on a sequential computation (by substituting a branch in the previous common spanning tree), which has the complexity $O(t)$ for all kinds of graphs [13].

Let us consider the connected graph represented in Fig. 16a, and described on levels as in Fig. 16b, where:

n, b, l—represent the number of nodes, branches, and levels, respectively;

b[i] = (xi, y_i)—is the branch i, connected between the nodes x_i and y_i; the node set is ordered so that

- level $(x_i) \leq$ level (y_i);
- $\forall\, 0 \leq i \leq b - 1$, level $(x_i) \leq$ level $(x_i + 1)$ and level $(y_i) \leq$ level $(y_i + 1)$;

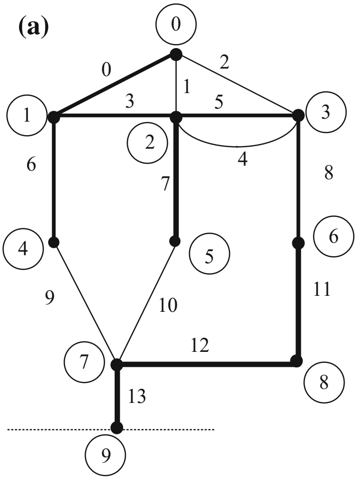
$niv[j]$ is the first node of level j, with $0 \leq j \leq l$;

$bet[k]$ is the first branch which connects the levels k and $k + 1$, where $0 \leq k \leq l - 1$;

$inter[m]$ is the first branch which connects two nodes from the level m, where $1 \leq m \leq l - 1$;

The algorithm for tree enumeration is the following:

(a)



(b)

$n = 10$
$b = 14$
$l = 5$
$niv = (0,1,4,7,9)$
$bet = (0,6,9,13)$
$inter = (3,12)$

b[0]=[0,1]	b[1]=[0,2]
b[2]=[0,3]	b[3]=[1,2]
b[4]=[2,3]	b[5]=[2,3]
b[6]=[1,4]	b[7]=[2,5]
b[8]=[3,6]	b[9]=[4,7]
b[10]=[5,7]	
b[11]=[6,8]	
b[12]=[7,8]	
b[13]=[7,9]	

Fig. 16 Graph representation on levels

```
place inside (level, k)
        if (level=0) {write tree; output of procedure}
        place between (level, niv[level])
        for i=k to bet[level]-1 do
                    {
                    save the colors on the stack
                    put in the tree (i)
                    place inside (level, i+1)
                    take off from the tree the last registration
                    restore the colors from the stack
                    }
place between (level, col)
        OK=1
        for C=col to niv[level+1]-1 do
             if exist the color C
               for i=bet[level-1] to inv[level]-1 do
                   if color[b[i].y]=C
                      {
                      OK=0
                      save the colors on the stack
                      put in the tree (i)
                      place rest (level, C, i+1)
                      take off from the tree the last registration
                      restore the colors from the stack
                      }
        if (OK=1) place inside (level-1, inv[level-1]);

      place rest (level,col,k)
         place between (level, col+1)
         for i=k to inv[level]-1 do
              if color[b[i].y]=col
                if do not make a cycle ⇔ if color [b[i].x]≠col
                  {
                  save the colors on the stack
                  put in the tree (i)
                  place rest (level, col, i+1)
                  take off from the tree the last registration
                  restore the colors from the stack
                  }
```

Remarks

1. The representation of any graph in level form is equivalent with node sorting. If a heap-sort procedure is used, the time complexity is $O(b\lg b + n\lg n)$ while using an array technique it will be $O(2b + 2n)$ [12].
2. The algorithm does not generate any non-tree sequence.
3. Any branch that obeys the algorithm rules, when is introduced in the sequence, leads to a tree.
4. The time complexity of the tree enumeration algorithm is proportional to the number of trees, $O(kt)$, where k is, statistically, about 1, when the time allocated for the level decomposition of the graph is neglected.
5. The space complexity is $O(n + n^2)$, when the necessary of memory for preserving the graph structure (which is insignificant) is neglected.

Testing the algorithm for many graphs, to make a comparison with Char's algorithm, we obtained the results presented in Table 5, and in Fig. 17.

We can see that the efficiency of our algorithm rises rapidly with the number of trees.

5.5 Algorithm for Sequential Generation of the Sign Factor

For all the terms of the numerator and of the denominator the sign factor must be computed. The sign of the tree admittance product can be found using Mayeda and Seshu's algorithm [8] or performing a depth-first or breadth-first traversal on both the G^i and G^v trees [2]. In [10, 11] an original method for the sign factor determination is presented. The tree admittance product sign ε_k is defined as:

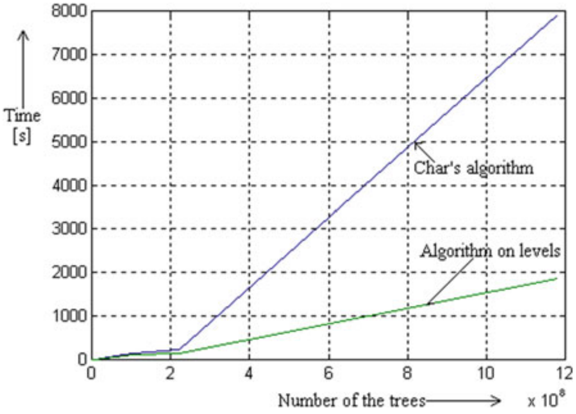
$$\varepsilon_k = M_{T_k^i} \cdot M_{T_k^v}, \tag{49}$$

where $M_{T_k^i}, M_{T_k^v}$ are major determinants from the branch-node incidence matrices $\mathbf{A}^i$ and $\mathbf{A}^v$ corresponding to the common spanning trees T_k^i and T_k^v, respectively. To describe the current (voltage) spanning tree T_k^i (T_k^v) we use a matrix with two rows

Table 5 Comparison between the proposed algorithm and Char's algorithm

Number of the trees	Time [s]	
	Char's algorithm	Algorithm on levels
107512	0.53	0.56
9877412	14.1	8.66
61314527	82	55
103472385	154	103
220581744	235	156
1182369421	7920	1860

Fig. 17 Comparison with Char's algorithm



and $n\,-\,1$ columns (n being the node number of the graph)—called the *current* (*voltage*) *tree description matrix* CTDM (VTDM). Each column of this matrix contains the initial node and the final node of the tree branch corresponding to this column. For example, the tree description matrix (TDM) corresponding to the spanning tree shown in Fig. 18 has the following form:

$$\mathrm{TDM} = \begin{bmatrix} 1 & 2 & 4 & 4 & 6 & 7 & 5 & 6 \\ 2 & 4 & 3 & 6 & 7 & 5 & 8 & 9 \end{bmatrix}. \tag{50}$$

The determinants $M_{T_k^i}$ and $M_{T_k^v}$ are computed by performing simple operations on the rows of the tree description matrices. In Fig. 18 is described the computing algorithm of the determinant M_T corresponding to the spanning tree $T = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8\}$.

Let B, N be two finite sets B, $N \subset \mathbf{N}$. The directed graph is, by definition, the triplet $G = (B, N, f)$, in which $f\colon B \to N \times N$. Let $x = (x_1, x_2)$ be an element of the set $N \times N$. We define:

$$p_1\colon N \times N \to N, \quad p_1(x) = x_1; \quad p_2\colon N \times N \to N, \quad p_2(x) = x_2, \tag{51}$$

the projections of an element in $N \times N$. Let $T = \{b_1, b_2, \ldots, b_{n-1}\}$, with $b_j \in B,\ 1 \le j \le n-1$, be a spanning tree and let

$$\mathbf{P} = \begin{bmatrix} p_1(f(b_1)) & p_1(f(b_2)) & \ldots & p_1(f(b_{n-1})) \\ p_2(f(b_1)) & p_2(f(b_2)) & \ldots & p_2(f(b_{n-1})) \end{bmatrix}, \tag{52}$$

be the matrix built with the rows 1 and 2 of TDM corresponding to the spanning tree T.

The algorithm for the determinant M_T calculation has the following steps:

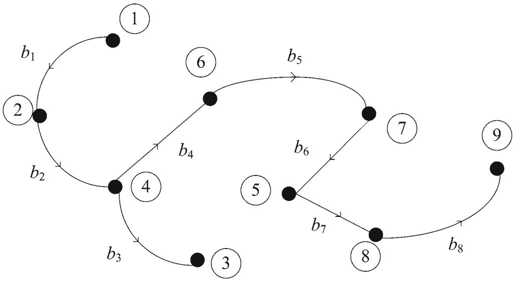


Fig. 18 Algorithm of the sign factor determination

$$\begin{array}{c} b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 \\ \begin{bmatrix} 1&2&4&4&6&7&5&8 \\ 2&4&3&6&7&5&8&9 \end{bmatrix} \end{array} = P \;\; for \;\; T = \{b_1, b_2, \cancel{b_3}, b_4, b_5, b_6, b_7, b_8\}$$

$$\begin{bmatrix} 1&2&4&4&6&7&5&8 \\ 2&4&3&6&7&5&8&9 \end{bmatrix} = P_0 \Rightarrow (-1)^{1+1}$$

$$\begin{bmatrix} 1&3&3&5&6&4&7 \\ 3&2&5&6&4&7&0 \end{bmatrix} = P_1 \Rightarrow (-1)^{1+1}$$

$$\begin{bmatrix} 2&2&4&5&3&6 \\ 1&4&5&3&6&0 \end{bmatrix} = P_2 \Rightarrow (-1)^{1+1+1}$$

$$\begin{bmatrix} 1&3&4&2&5 \\ 3&4&2&5&0 \end{bmatrix} = P_3 \Rightarrow (-1)^{1+1}$$

$$\begin{bmatrix} 2&3&1&4 \\ 3&1&4&0 \end{bmatrix} = P_4 \Rightarrow (-1)^{2+1}$$

$$\begin{bmatrix} 2&1&3 \\ 1&3&0 \end{bmatrix} = P_5 \Rightarrow (-1)^{2+1}$$

$$\begin{bmatrix} 1&2 \\ 2&0 \end{bmatrix} = P_6 \Rightarrow (-1)^{1+1}$$

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} = P_7 \Rightarrow (-1)^{1+1}$$

$$M_T = (-1)^{2+2+3+2+3+3+2+2} = -1$$

1. In the matrix $\boldsymbol{P}$, we assign zero value to the node with the greatest index (e.g. n_9 in Fig. 18). In this way, we obtain a matrix which is denoted by $\boldsymbol{P}_0$;
2. We are looking for the node which exists only once in the matrix $\boldsymbol{P}_0$, beginning with the node having the smallest index. Let this be $n_j = p_k(b_j)$, with $k = 1$ or $k = 2$ (e.g. n_1 in Fig. 18);
3. We develop the determinant M_T on the row corresponding to the node n_j, namely

$$M_T = (-1)^{n_j + j} M_T^j, \tag{53}$$

if $k = 1$, and

$$M_T = (-1)^{n_j + j + 1} M_T^j, \tag{54}$$

if $k = 2$, where j is the column of the matrix $\boldsymbol{P}_0$ corresponding to the node n_j, and M_T^j represents the determinant obtained from M_T after the elimination of the row n_j and of the column j;
4. If the node number n_j, found in step 2, is less than the greatest node number in $\boldsymbol{P}_0$ (if $n_j < n - 1$), then all elements of $\boldsymbol{P}_0$ having the values greater than n_j are reduced by a unit, and all columns of the matrix $\boldsymbol{P}_0$, which are on the right side of the column j, change the places with a column to the left side. Thus, we obtain a matrix $\boldsymbol{P}_m$, $m \leftarrow m + 1$ (initially $m \leftarrow 0$), having the column number less than $\boldsymbol{P}_0$ with a unit;
5. If $n_j \geq 1$ and if the column number of the matrix $\boldsymbol{P}_m$ is greater than one, go to step 2, where the matrix $\boldsymbol{P}_m$ takes the place of the matrix $\boldsymbol{P}_0$. If the $n_j = 1$ and if the matrix $\boldsymbol{P}_m$ has a single column, the determinant M_T^j is developed on the row corresponding to the node n_1 and go to step 6;
6. Check up if the exponent of (-1) is an even or odd number.

In order to reduce the time needed to generate the circuit functions, a very fast algorithm for calculating the sign factor was developed and implemented. It is based on sequential computation, because knowing the sign of a term we can find the sign of the following by performing simple elementary operations (permutations) in a vector with n elements, representing the number of the graph nodes. These permutations aim to preserve the summations between lines of the reduced node-branch incidence matrix, without having to store it in the memory.

The algorithm pseudocode has the following structure:

```
ordine (x)
j=0
for I=1 to x do if rel[I]=I then inc(j)
if odd[j] return 1
          else return −1

add edge
while x≠rel[x] x=rel[x]
if (x=$nv$)
          while y≠rel[y] y=rel[y];
          rel[y]=$nv$;
          s= −s·ordine(y)
else s= s·ordine(y)

compute sign
$s$=1, for I=1 to $nv$ do rel[I]=I
for I=1 to $nv$−1 do add edge($I$)
return(j)
```

5.6 Automatic Generation of the Transfer Functions

As it has been shown in Sect. 5.6, in order to compute a transfer function, we have to use two pairs of graphs: $\left(G_1^i, G_1^v\right)$, for the numerator product terms, and $\left(G_p^i, G_p^v\right)$ (in accordance with Table 4), for the denominator product terms.

For the automatic generation of $\left(G_1^i, G_1^v\right)$, we must connect at the input port of the circuit a VCCS having as controlling variable the output voltage, the transfer admittance being 1. For this source, the controlled branch number is 1, while the controlling branch number is 2 (Fig. 19, *LNC*—*L*inear *N*onreciprocal *C*ircuit).

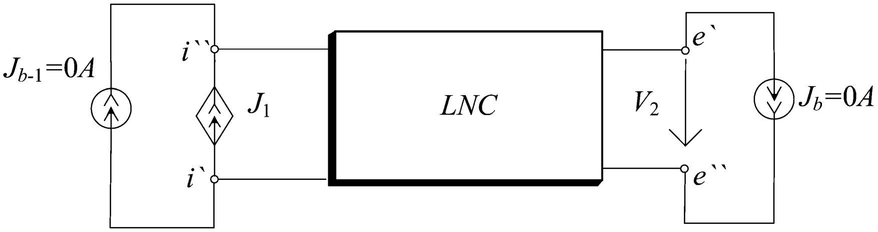


Fig. 19 Automatic generation of $\left(G_1^i, G_1^v\right)$ and $\left(G_p^i, G_p^v\right)$

For the automatic generation of $\left(G_p^i, G_p^v\right)$ we must connect at the input/output ports some ideal independent current sources having null currents and the last branch numbers: $b-1$, respectively b (Fig. 19).

The algorithm for transfer impedance Z_{oi} generation involves the following steps:

1. Gyrator circuits, voltage or current inverters, magnetic couplings, operational amplifiers and, in general, the multipole or multiport circuit elements contained in the analyzed circuit are replaced by equivalent schemes consisting of bipolar circuit elements and controlled sources only;
2. Controlled sources are simulated by passive two terminals elements that have distinct positions in G^i and G^v graphs (see Table 4);
3. Graphs G_1^i and G_1^v are generated (Figs. 15a and b). In G_1^i the branch with the unit weight connects the input port terminals, having the same sense as the sense of the input variable corresponding to the transfer function to be generated and the output port is open. In G_1^v, the branch with unit weight connects the output port terminals in the same sense as the sense of the output variable corresponding to the transfer function to be generated, the input port being open;
4. Graphs G_p^i and G_p^v are generated. In these graphs, the entry-exit ports are treated as in Table 4;
5. Determine the array of trees common to the graphs G_1^i and G_1^v that contains the branch with the unit weight
$$A_{1c} = A_1^i \cap A_1^v \tag{55}$$
where A_1^i $\left(A_1^v\right)$ is the array of the trees that contain the branch with the unit weight in the graph G_1^i $\left(G_1^v\right)$;
6. Determine the array of trees common to the graphs G_p^i and G_p^v
$$A_{pc} = A_p^i \cap A_p^v, \tag{56}$$
with A_p^i $\left(A_p^v\right)$, the array of the trees from the graph G_p^i $\left(G_p^v\right)$;
7. For each pair of common trees k, generated at steps P5 or P6, the sign factor ε_k is calculated with one of the algorithms described in Sect. 5;
8. Calculate the algebraic sum of tree values $\varepsilon_k P_k$ for the A_{1c} set and then for the set A_{pc}, P_k being the product of the weights (of operational admittances) of the common tree k branches;
9. With formula (37) calculate the transfer impedance Z_{oi};

If the numerator and the denominator of the relation (37) are multiplied by the product of the operational impedances of all branches of the circuit, it results:

$$Z_{ei} = -\frac{C_1}{C_p}, \tag{57}$$

where:

$$C_1 = \sum_{k \in C_{1c}} \varepsilon_k P_{ck}, \tag{58}$$

is the algebraic sum of the values (in impedances) of the co-trees common to the graphs G_1^i and G_1^v corresponding to the common trees that contain the branch with the unit weight and:

$$C_p = \sum_{k \in C_{pc}} \varepsilon_k P_{ck}, \tag{59}$$

is the algebraic sum of the values (in impedances) of the co-trees common to the graphs G_p^i and G_p^v.

It is easily to show [14, 21] that formula (55) can be used to generate any circuit function corresponding to the input-output ports treated as in Table 4.

Therefore

$$F_{oi} = -\frac{C_1}{C_p}, \tag{60}$$

where the function F_{oi} may be: the transfer impedance (either input or output), the transfer admittance (either input or output), the voltage transfer (gain) factor or the current transfer (gain) factor.

The algorithm for generating any of the above mentioned circuit functions is identical to the one presented for the transfer impedance Z_{oi}, the only difference being the treatment of the input-output ports (Table 4). To define input impedance (admittance), the input-output structure of the port is defined in Fig. 20a, b) by using a passive linear circuit (***PLC***). Analog is defined also the input-output structure of the two-port circuit for the calculation of the output impedance (the output admittance).

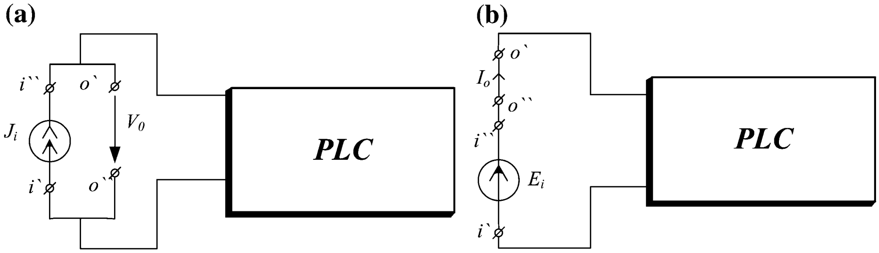


Fig. 20 The input-output structure of the two-port circuit for the calculation of the input impedance (**a**) and input admittance (**b**)

Before generating the transfer function in symbolic form, by tree enumeration in the two graphs, we must do some simplifications either in the circuit or in the structure of the two graphs called approximation-before-computation (ABC).

Firstly, for each parameter x, we perform a numerical computation of the transfer function sensitivity in the frequency range of interest.

This information could give us the reason to eliminate some branches either by element removal or by contraction of its terminal nodes that simplify the circuit structure. In order to control the accuracy of the computational process, we have to evaluate the errors in the transfer function magnitude and in the transfer function argument due to these operations in the frequency range of interest. Once the circuit structure was simplified, we can generate the two pairs of graphs: $\left(G_1^i, G_1^v\right)$ and $\left(G_p^i, G_p^v\right)$. In order to simplify the generation of all their common spanning trees, we perform some operations in the structure of these graphs, namely:

- Contraction of the unity weight branches;
- Substitution of the parallel branches in these graphs by an equivalent branch having the admittance equal to the sum of the parallel admittances;
- Contraction of all branches having a node of degree one.

After the generation of the spanning trees in the reduced graphs, we must add successively all the branches eliminated in the first step. This procedure increases the enumeration efficiency of the common spanning tree in the two pairs of graphs —$\left(G_1^i, G_1^v\right)$, and $\left(G_p^i, G_p^v\right)$.

To obtain the symbolic transfer function in a form to be easily interpreted, two approximation strategies are possible: approximation-during-computation (ADC) that produces the approximate expression without knowledge of the exact symbolic expression, and approximation-after-computation (AAC) that firstly generates the exact symbolic expression and operating on it produces an approximated one. The simplified form can be obtained because only a small number of the terms in the irreducible expanded expression of the transfer function have an important contribution in the numerator or in the denominator value. Of course, the most efficient method is to generate only the significant common trees (whose tree admittance value is not negligible) in an ADC process. To this end the common spanning trees must be generated in decreasing order of magnitude until the generated set is a good approximation of the exact network function value. Also, the generation of the common spanning trees in decreasing order of magnitude must be performed for each frequency of interest. Some techniques for ADC were reported [16, 23–26], based on a sensitivity simplification scheme, a 2-, respectively 3-matroid-intersection algorithm and on the determinant decision diagram (DDD) representation of the system determinant. Although it is not easy to compare the implementations of these algorithms because of the different simplification before generation performed, and because of the different error criteria, it seems [24] that ADC based on DDD yields better results concerning the time needed to generate a term in comparison with the other techniques.

In this chapter, an AAC procedure to obtain a network function in reduced symbolic form is adopted. To this end the numerator and the denominator expressions must be ordered in the decreasing order of the complex frequency powers. The coefficients of each complex frequency power must be ordered in the decreasing order of their values as well, and then the terms with the smallest value will be eliminated one by one if the magnitude and phase errors are kept within imposed limits in the frequency range of interest.

A very fast program for the network function generation in reduced symbolic form has been obtained by implementing the modelling technique of the controlled sources associated with the generalized topological formula and with the algorithm for common tree enumeration and sign factor computation.

5.7 Description of the Software Application SATE—Symbolic Analisys by Trees Enumeration

The symbolic generation algorithm of circuit functions for analogue linear and/or nonlinear (piecewise-linear approximation) circuits described in Sect. 5.6 have been implemented in a program called SATE—*S*imbolic *A*nalysis by *T*ree *E*numeration [28]. Starting from the description of the circuit through a netlist input file (*cir.* file extension), SATE generates symbolically, partial symbolically or numerically form any circuit function with respect to the user-specified input/output ports for the linear and/or non-linear (piecewise-linear approximation around a point of operation) electrical circuits.

The input data for the software application are:

nnode, *nb*, *pulsation*

where: *nnode*—is the number of circuit nodes, *nb*—is the branch number, and *pulsation (angular frequency)* is the pulsation value.

Follows a set of *nb* lines describing the branches of the circuit. The circuit elements are assigned as type numbers: **1**—for resistors; **2**—for capacitors; **3**—for inductors; **8**—for controlled sources $e_c(i_C)$; **9**—for controlled sources $j_c(v_C)$; **10**—for controlled sources $e_c(v_C)$; **11**—for controlled sources $j_c(i_C)$ and **12**—for the description of input-output ports.

For RLC circuit passive elements, the description statement has the form:

element_type parameter_real_value initial_node final_node

For a controlled source, the description statement has the following structure:

source_type parameter_real_value parameter_imaginary_value initial_node_c final_node_c initial_node_C final_node_C

where

initial_node_c final_node_c (initial_node_C final_node_C) represent the initial and final nodes for the controlled branch (controlling branch).

The last line of the input file describes the input/output ports and it has the following format:

12 initial_node_i final_node_i initial_node_o final_node_o

Remarks

1. The program gives to the branches numbers from 0 to b;
2. The last numbered branch, corresponding to the last line in the input file list, represents the branch weighting 1 in the current graph G_1^i and in the voltage graph G_1^v;
3. In the case of the current-controlled sources, $e_c(i_C)$ and $j_c(i_C)$, the controlling ports are simulated by resistors with a very low resistance value ($<10^{-8}\Omega$);
4. In the case of the homogeneous controlled sources ($e(v)$ and $j(i)$), the program assigns two branches to each source (in the following sequence: the controlled branch, the controlling branch), taking into account the modelling of these sources in the current graph or the voltage graph [20, 27]. Parameters corresponding to the two branches are assigned as follows:
   $A_{j\_k} = \frac{Y_k}{Y_j}$, where $Y_k = 1$ S and $Y_j = \frac{1}{A_{j\_k}}$, for the source e_c (v_C),
   $B_{j\_k} = \frac{Y_j}{Y_k}$, where $Y_k = 1$ S and $Y_j = B_{j\_k}$, for the source $j_c(i_C)$;
5. Magnetically coupled inductors are simulated by current-controlled voltage sources [20, 27, 28].

The main program **compute.bat** coordinates the entire process of generating the circuit function by successively calling the following subprograms:

- **cv_graph.exe**—it determines the current and voltage graphs;
- **tree.exe**—it generates the trees common to the two graphs;
- **comp_fix.exe**—it calculates the numerator and denominator terms of the circuit function;
- **getfunc.exe**—it factories the numerator and denominator expressions according to the chosen parameter;
- **draw.exe**—it draws the amplitude–frequency and phase–frequency characteristics of the generated circuit function.

The SATE program command line is:
compute input_file_name x
where:

- **input_file_name**—is the input file name with the extension *cir* (on the call the file extension is not written)
- and **x** represents the type of the circuit function that will be generated, as follows:

- **1**—the transfer impedance $Z_{ei}(s)$;
- **2**—the transfer admittance $Y_{ei}(s)$;
- **3**—the voltage transfer (gain) factor $A_{ei}(s)$;
- **4**—the current transfer (gain) factor $B_{ei}(s)$.

The SATE program generates the following output files:

- **file_name.gr1**—it contains the required information about the current graph;
- **file_name.gr(x + 1)—**it contains the required information about the voltage graph;
- **file_name.ar1**—it symbolically displays the numerator of the circuit function (A_1 from formula (19));
- **file_name.ar(x + 1)**—it symbolically displays the denominator of the circuit function (A_p from formula (19));
- a file containing numeric information about the value of the circuit function: the real part, the imaginary part, the module and the argument.

6 Examples

Example 3 Let us consider the linear circuit with lumped parameters represented in Fig. 21a. We want to determine the operational transfer admittance Y_{oi} from the input port *i'–i"* to the output port *o'–o"*, assuming that all the other parameters of the circuit are known.

In Fig. 21, additional sources were also represented $J_1 = 1.V_2$, $J_{13} = 0\text{A}$ and $J_{14} = 0\text{A}$, which aim to facilitate the automatic generation of graphs G_1^i, G_1^v (source $J_1 = 1.V_2$) and G_p^i, G_p^v (sources $J_{13} = 0\text{A}$ and $J_{14} = 0\text{A}$). The numbering of additional sources was done as indicated above.

In Fig. 21b–e the graphs G_1^i and G_1^v (G_p^i and G_p^v) are represented. The loops resulting by connecting in short-circuit of certain pairs of nodes in the graphs G_1^i, G_1^v, G_p^i and G_p^v have not been drawn in Fig. 21b–e (the branches of these loops cannot belong to the trees of these graphs).

The set of trees common to the graphs G_1^i and G_1^v (Figs. 21b and c), which contains the branch 1_i in G_1^i and, respectively, branch 1_o in G_1^v is:

$$A_{1c} = \{(1_i, 6, 9, 12; 1_o, 6, 9, 12), (1_i, 4, 10, 11; 1_o, 4, 10, 11)\}. \tag{61}$$

The set of trees common to the graphs G_p^i and G_p^v (Figs. 21d and e) has the following structure:

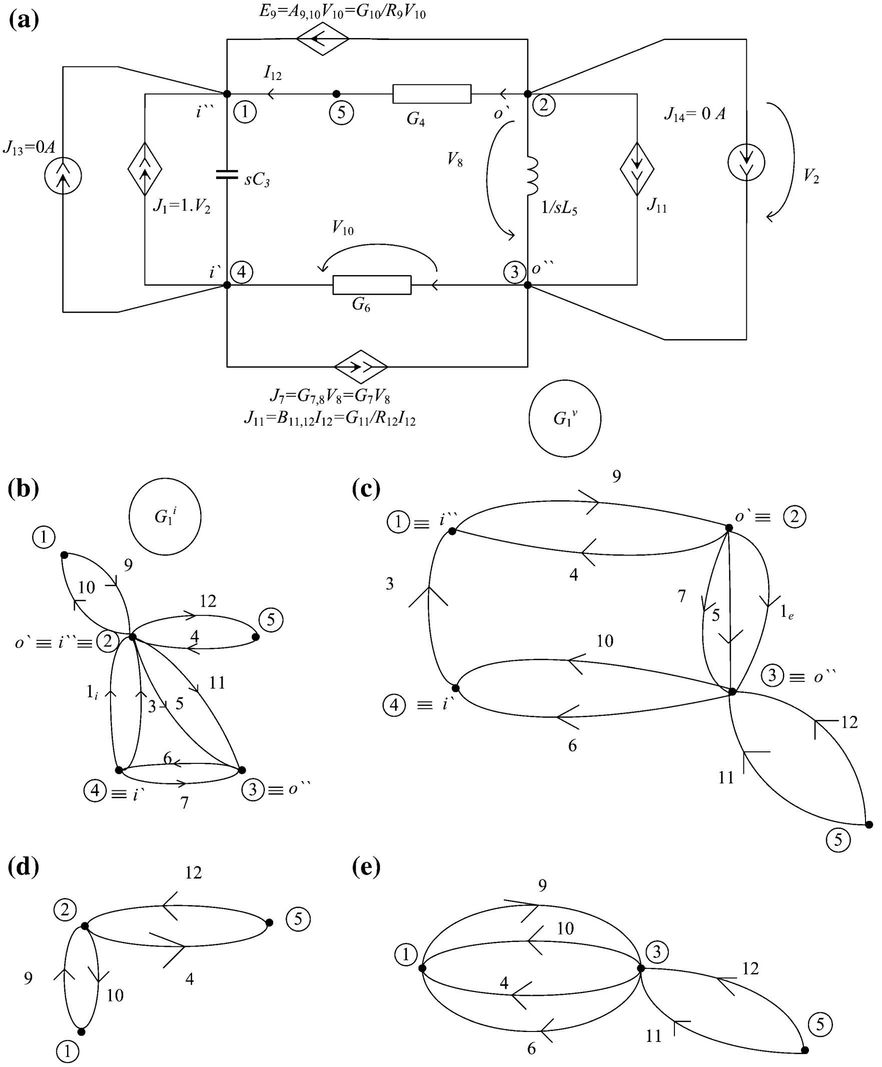


Fig. 21 A linear circuit and its pairs of graphs G_1^i and G_1^v, respectively G_p^i and $G_p^{v`}$

$$A_{pc} = \{(9, 12); (10, 12)\}, \tag{62}$$

Applying the above algorithm we obtain:

$$Y_{oi} = \frac{G_6 G_9 G_{12} + G_4 G_{10} G_{11}}{(G_9 + G_{10}) G_{12}} \tag{63}$$

or

$$Y_{oi} = \frac{R_4R_{10}R_{11} + R_6R_9R_{12}}{R_4R_6R_{11}(R_9 + R_{10})}. \tag{64}$$

Expressions (61) and (62) have been compared with those obtained with the programs *TFSYG*—*T*ransfer *F*unction *SY*mbolic *G*eneration and *CSAP*—*C*ircuit *S*ymbolic *A*nalysis *P*rogram, [28, 29], and it has been observed that these are identical.

Remarks

1. The trees common to the graphs G_1^i and G_1^v, that contain the common branch of weight 1, are identical with the trees common to the graphs $G^i_{1,1_{i,sc}}$ and $G^v_{1,1_{o,sc}}$, obtained from the graphs G_1^i and G_1^v, in which the branches 1_i and 1_o are short-circuited.
2. In the case of the nonlinear circuits, any of the four transfer functions of the circuit can be calculated with formula (35) or (58), at every time moment $t_{n+1} = t_n + 1$, by making the circuit passive and by linearization around the operating point at this time moment.

Example 4 Let be the small signal equivalent circuit of a three-stage CMOS transistor amplifier, represented in Fig. 22. The voltage transfer (gain) factor has to be generated symbolically, in relation to the input-output ports, 1–5 and 4–5, respectively.

Using the algorithm based on the graph decomposition on levels, respectively of the SATE (Symbolic Analysis by Tree Enumeration) software [14, 21], we can proceed as follows:

1. The input file, ex2.cir, has to be edited with the following structure:

   5 13 314.00000000000000E + 0001 *(nodes number, branches number, pulsation)*

   1 1000.0 1 5 *(branch type, parameter value, initial node, final node)*

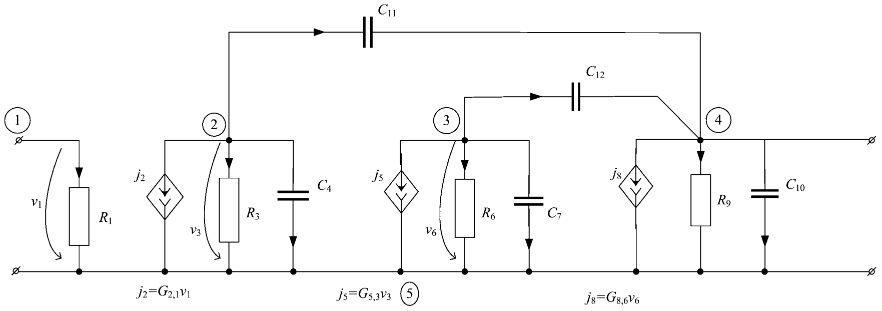


Fig. 22 Equivalent scheme of a small signal amplifier

9 0.001 0.0 2 5 1 5 *(9-source j(u), real value, imaginary value, initial node, final node for the controlled variable, initial node, final node for the controlling variable)*
1 1000.0 2 5
2 0.1e−08 2 5
9 0.001 0.0 3 5 2 5
1 1000.0 3 5
2 0.1e−08 3 5
9 0.001 0.0 4 5 3 5
1 20000.0 4 5
2 0.1e−08 4 5
2 0.1e−08 2 4
2 0.1e−08 3 4
12 1 5 4 5 (input and output ports)

1. After SATE program running, the following results are obtained:

Terms of the circuit function counter	
Sign factor	The value of common trees (in admittances)
1	$C_7C_{11}G_{2\_1}s^2$
1	$G_{2\_1}G_{5\_3}G_{8\_6}$
−1	$C_{12}G_{5\_3}G_{2\_1}s$
1	$C_{11}C_{12}G_{2\_1}s^2$
1	$C_{11}G_6G_{2\_1}s$

Terms of the circuit function denominator			
Sign factor	The value of common trees (in admittances)	Sign factor	The value of common trees (in admittances)
1	$C_{11}G_3G_6s$	1	$C_7C_{10}C_{11}s^3$
1	$C_4C_{11}G_6s^2$	1	$C_7C_{12}G_3s^2$
1	$G_3G_6G_9$	1	$C_4C_7C_{12}s^3$
1	$C_4G_6G_9s$	1	$C_7C_{11}C_{12}s^3$
1	$C_{11}G_6G_9s$	−1	$G_{5\_3}G_{8\_6}C_{11}s$
1	$C_{10}G_3G_6s$	1	$C_{11}C_{12}G_{5\_3}s^2$
1	$C_4C_{10}G_6s^2$	1	$C_{11}C_{12}G_3s^2$
1	$C_{10}C_{11}G_6s^2$	1	$C_4C_{11}C_{12}s^3$
1	$C_{12}G_3G_6s$	1	$C_{12}G_3G_{8\_6}s$
1	$C_4C_{12}G_6s^2$	1	$C_4C_{12}G_{8\_6}s^2$
1	$C_{11}C_{11}G_6s^2$	1	$C_{11}C_{12}G_{8\_6}s^2$
1	$C_4C_7G_3s^2$	1	$C_{12}G_3G_9s$
1	$C_4C_7C_{11}s^3$	1	$C_4C_{12}G_9s^2$
1	$C_7G_3G_9s$	1	$C_{11}C_{12}G_9s^2$
1	$C_4C_7G_9s^2$	1	$C_{10}C_{12}G_3s^2$

(continued)

(continued)

1	$C_7C_{11}G_9s^2$	1	$C_4C_{10}C_{12}s^3$
1	$C_7C_{10}G_3s^2$	1	$C_{10}C_{11}C_{12}s^3$
1	$C_4C_7C_{10}s^3$		

If

$$G_1 = G_3 = G_6 = G, C_4 = C_7 = C_{10} = C_{11} = C_{12} = C, G_{2\_1} = G_{5\_3} = G_{8\_6} = G_m$$

then the voltage transfer gain becomes:

$$A_{oi} = -\frac{G_m\left[C^2s^2 + C(G - G_m)s + G_m^2\right]}{8C^3s^3 + C^2(3G_m + 10G + 4G_9)s^2 + C\left(GG_m + 3G^2 + 4G_mG_9 - G_m^2\right)s + G^2G_9}$$

The sensitivity of the voltage transfer gain in relation to the parameter G_m, $S_{G_m}^{A_{0i}} = \frac{\partial A_{0i}}{\partial G_m} \cdot \frac{G_m}{A_{oi}}$, has the expression:

Sei =(-16*C^4*s^4*Gm+28*C^4*s^4*G+8*C^4*s^4*G9+16*C^3*s^3*G^2+
+3*C^2*s^2*G^3+21*C^3*s^3*Gm^2+6*Gm^3*C^2*s^2+3*Gm^2*G^2*G9+
+16*C^5*s^5-20*C^3*s^3*G*Gm-8*C^3*s^3*Gm*G9+2*C^2*s^2*G^2*G9+
+4*C^3*s^3*G*G9-6*C^2*s^2*G^2*Gm+C*s*G^3*G9+
+29*C^2*s^2*Gm^2*G+8*C^2*s^2*Gm^2*G9+
+2*Gm^3*C*s*G+9*Gm^2*C*s*G^2+8*Gm^3*C*s*G9-
-2*C*s*Gm*G^2*G9)/((2*C^2*s^2+C*s*G-C*s*Gm+Gm^2)*(8*C^3*s^3+
+3*C^2*s^2*Gm+10*C^2*s^2*G+4*C^2*s^2*G9+C*s*G*Gm+3*C*s*G^2+
+4*C*s*Gm*G9+G^2*G9)).

For the numeric values $C = 1$ nF, $G = 0.001$ S, $G_m = 0.001$ S, $G_9 = 0.00005$ S and replacing s with $j\omega$, the voltage gain expression $A_{ei}(j\omega)$ becomes:

$$A_{oi}(j\omega) = -\frac{125000 \cdot (\omega^2 - 0.1 \cdot 10^{13})}{j\omega^3 + 0.165 \cdot 10^7\omega^2 - 4 \cdot 10^{11}j\omega - 625.10^{13}}.$$

Figure 23 shows the Bode diagram, and Fig. 24 presents the distribution of poles and zeros in the complex plane.

Fig. 23 Bode diagram for Aoi

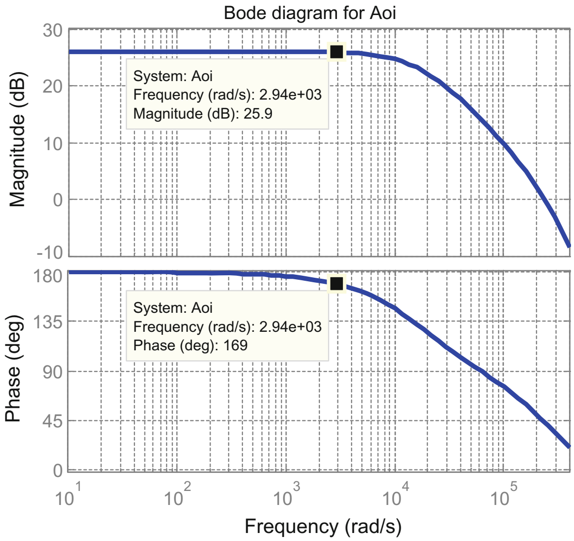
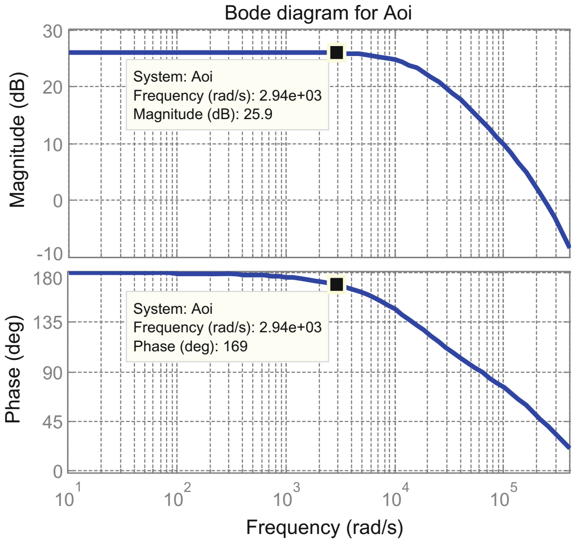


Fig. 24 Pole and zero locations of small signal amplifier



For the above numeric values, the output file provides the following data about the required circuit function:

Real part: +1.999243e + 01	Imaginary part: 4.022459e + 01
Module: 1.999648e + 01	Argument: 3.121475e + 00

Example 5 The circuit in Fig. 25a, contains two operational amplifiers and passive circuit elements *R*, *C*. This circuit operates in a permanent harmonic regime as a capacitance multiplier with respect to the input terminals (4−5). By replacing the operational amplifiers with the equivalent scheme of Fig. 25b, the equivalent circuit represented in Fig. 25c, is obtained. The complex input impedance $\underline{Z}_{ii}(\omega)$ (with respect to the input terminals 4−7) has to be calculated with SATE.

For the case when the resistances $R_5 = R_7 = 0\,\Omega; R_4 = R_6 = 1$ Meg; C_3 = 10 pF (node 6 becomes 1, node 5 becomes 3, node 7 becomes 5, and the resistance R_6 becomes R_5) and the voltage gains $a_{8\_4} = a_{9\_6} = 2 \cdot 10^5$, the input file required by the SATE software, *ex3.cir*, has the following structure:

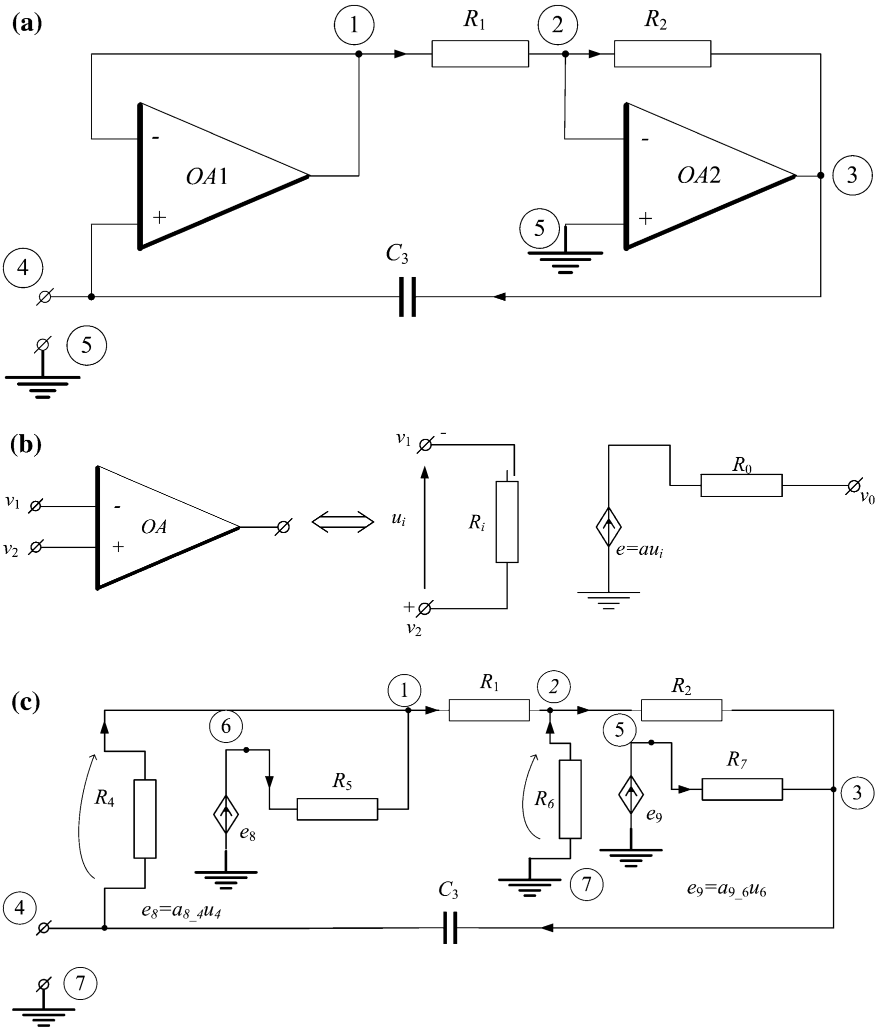


Fig. 25 Capacitance multiplier

```
5 8 314.0
1 100.0 1 2
1 1e + 05 2 3
2 1e−11 3 4
1 1e + 06 4 1
1 1e + 06 5 2
10 2e + 05 0.0 5 1 4 1
10 2e + 05 0.0 5 3 5 2
12 4 5 4 5
```

For homogeneous controlled sources ($e_c(v_C)$ and $j_c(i_C)$), the program assigns two branches to each source (in this sequence: the controlled branch, the controlling branch). For the considered circuit, in the above simplified situation, where e_8 becomes e_6 with the controlling branch l_7 and e_9 becomes e_8 with the controlling branch l_9, the corresponding voltage gains have the expressions:

$$a_{6\_7} = \frac{G_7}{G_6} \text{ şi } a_{8\_9} = \frac{G_9}{G_8},$$

where: $G_7 = 1$ S and $G_9 = 1$ S, and $G_6 = G_8 = 1/2 \cdot 10^5$ S.

Results from the output file are as follows:

Terms of the circuit function numerator		Terms of the circuit function denominator	
Sign factor	The value of common trees (in admittances)	Sign factor	The value of common trees (in admittances)
1	$G_1G_6G_8$	−1	$j\omega C_3G_5G_6G_8$
1	$G_1G_7G_8$	−1	$j\omega C_3G_5G_7G_8$
1	$G_2G_6G_8$	−1	$G_4G_5G_6G_8$
1	$G_2G_6G_9$	−1	$G_1j\omega C_3G_6G_8$
1	$G_2G_7G_8$	−1	$G_1j\omega C_3G_7G_8$
1	$G_2G_7G_9$	−1	$G_1j\omega C_3G_7G_9$
1	$G_5G_6G_8$	−1	$G_1G_4G_6G_8$
1	$G_5G_7G_8$	−1	$G_2j\omega C_3G_6G_8$
		−1	$G_2j\omega C_3G_6G_9$
		−1	$G_2j\omega C_3G_7G_8$
		−1	$G_2j\omega C_3G_7G_9$
		−1	$G_2G_4G_6G_8$
		−1	$G_2G_4G_6G_9$

For the numeric values, the output file of the program provides the following data about the required circuit function:

Real part: 5.010559e−001	Imaginary part: −3.165607e + 005
Module: 3.165607e + 005	Argument: −1.570795e + 000

If the two operational amplifiers are considered identical ($a_{6\_7} = a_{8\_9} = A$) and the resistances $R_4 = R_6 \rightarrow \infty$, while R_5, $R_7 = 0\,\Omega$, then the input complex impedance expression becomes

$$Z_{ii}(\omega) = \frac{(1 + A) \cdot [A(G_1 + G_2) + G_2](G_2)}{j\omega C_3[A^2(G_1 + G_2) + A(G_1 + 2G_2) + G_1 + G_2]}$$

Assuming that the operational amplifiers are ideal $(A \rightarrow \infty)$ we obtain:

$$\underline{Z}_{ii}(\omega) = \frac{G_2}{j\omega C_3(G_1 + G_2)} = \frac{1}{j\omega C_3\left(1 + \frac{R_2}{R_1}\right)}.$$

The input impedance sensitivity, in respect of the conductance G_1, has the expression:

$$S_{G_1}^{\underline{Z}_{ii}(\omega)} = -\frac{G_1}{G_1 + G_2}.$$

From the last expression of the complex input impedance an equivalent capacity results as:

$$C_e = C_3\left(1 + \frac{R_2}{R_1}\right) = 10.10^{-12}\left(1 + \frac{10^5}{10^2}\right) = 10.01.10^{-9}\ \text{F} = 10.01\ \text{nF}.$$

This capacitance is about a thousand times greater than capacity C_3. This circuit is used in integrated circuits technology to achieve high capacities. Due to miniaturization, integrated circuit technology usually produces capacitors with low capacities. The multiplication effect of the capacity is called *the Miller effect* for capacities [20, 27].

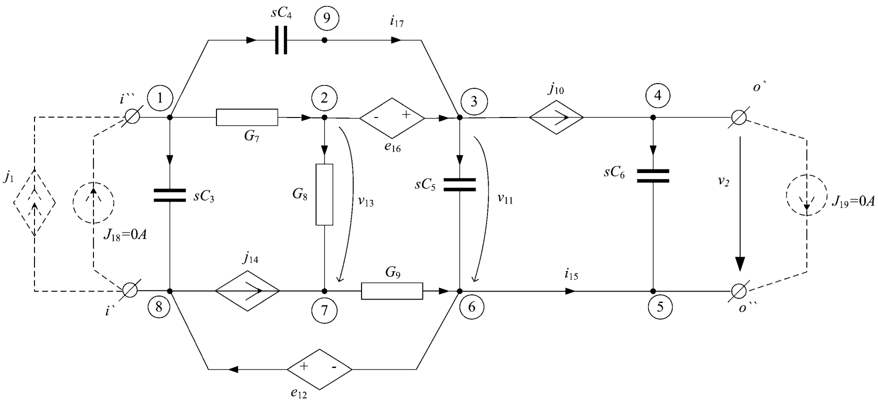


Fig. 26 A circuit containing all types of controlled sources

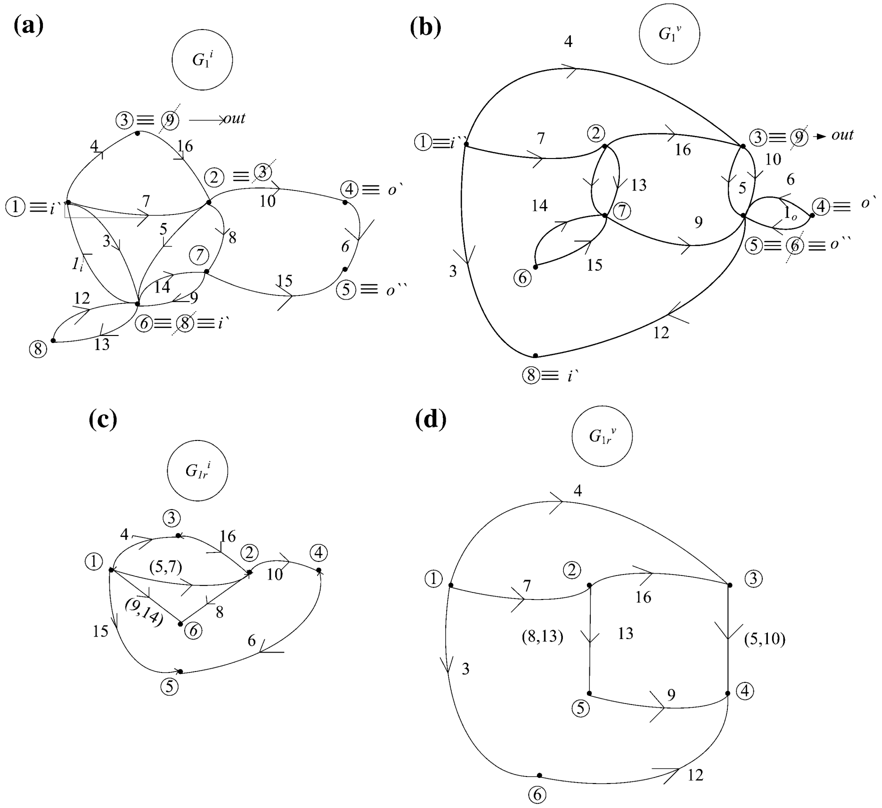


Fig. 27 The complete graphs $\left(G_1^i, G_1^v\right)$ and the reduced ones $\left(G_{1r}^i, G_{1r}^v\right)$

Example 6 The circuit containing all the types of controlled sources shown in Fig. 26, has the graphs $\left(G_1^i, G_1^v\right)$ shown in Fig. 27a and b. By performing the contractions presented in paragraph 6, we obtain the reduced graphs $\left(G_{1r}^i, G_{1r}^v\right)$ shown in Fig. 27c and d.

After operating the simplifications, the number of spanning trees in the two reduced graphs becomes much smaller. If we make similar simplifications in the graphs $\left(G_p^i, G_p^v\right)$, shown in Fig. 28a and b, we get the reduced graphs $\left(G_{pr}^i, G_{pr}^v\right)$ from Fig. 28c and d. The results of these simplifications are shown in Table 6. The number of trees in a graph has been calculated as it is presented in [20, 27].

We can observe a significant reduction in the number of trees in this simplification phase.

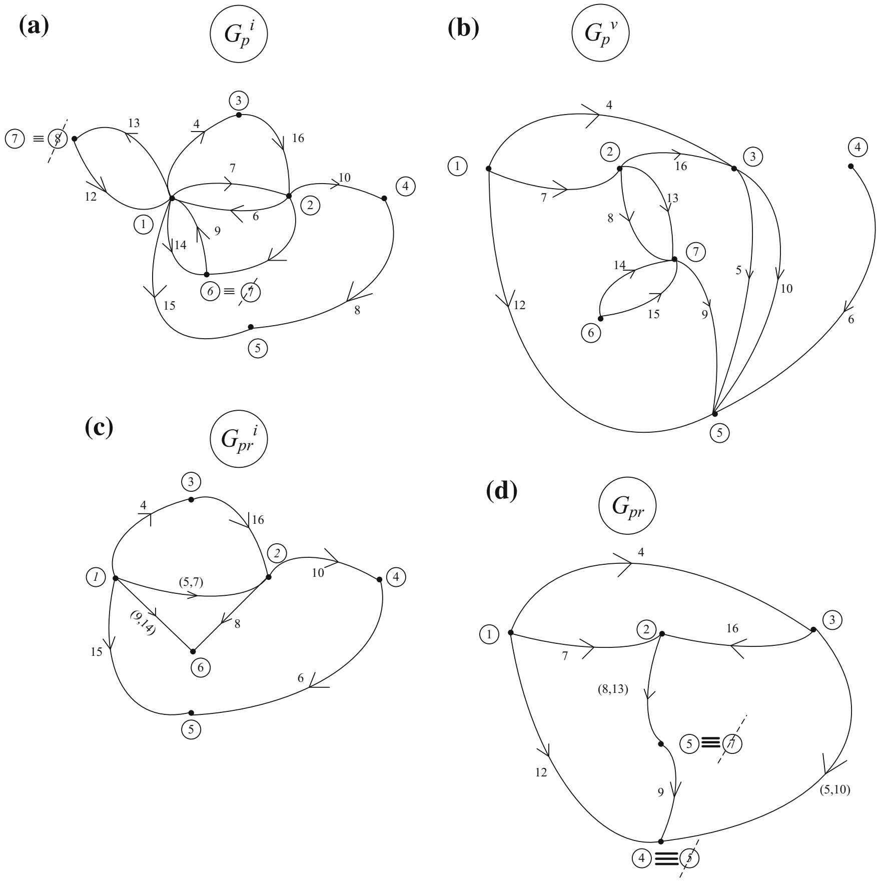


Fig. 28 The complete graphs $\left(G_p^i, G_p^v\right)$ and the reduced ones $\left(G_{pr}^i, G_{pr}^v\right)$

Table 6 The results of the simplification procedure

$n_{t,G_1^i} = 360 \Rightarrow n_{t,G_{1r}^i} = 28$	$n_{t,G_p^i} = 126 \Rightarrow n_{t,G_{pr}^i} = 28$
$n_{t,G_1^v} = 360 \Rightarrow n_{t,G_{1r}^v} = 28$	$n_{t,G_p^v} = 122 \Rightarrow n_{t,G_{pr}^v} = 24$

The command equations of the controlled sources are as follows:

$$j_1 = 1 \cdot v_2, e_{12} = a_{12,13} v_{13} = R_{12} G_{13} v_{13} = \frac{G_{13}}{G_{12}} v_{13},$$

$$j_{14} = b_{14,15} i_{15} = G_{14} R_{15} i_{15} = \frac{G_{14}}{G_{15}} \mathrm{i}_{15}, e_{16} = R_{16} i_{17} = \frac{1}{G_{16}} i_{15}.$$

The above simplifications do not affect the accuracy of the calculation, because they are operated in the graphs structure, which simplifies it, maintaining their equivalence.

If we consider $C_3 = C_4 = C_5 = C_6 = C$; $G_7 = G_8 = G_9 = G$, then the voltage gain factor Aoi has the following expression:

$$
\begin{aligned}
&Aoi := G12\ G15\ G10\ ((2.\ G16\ C\ G + 3.\ C\ G^2)\ s + 2.\ G16\ G^2)\ /\ (\\
&\quad (3.\ G15\ C^2\ G12\ G^2 - 3.\ G15\ C^2\ G13\ G^2 + 4.\ C^2\ G16\ G12\ G15\ G - 2.\ C^2\ G16\ G13\ G15\ G)\ s^2 + (2.\ C\ G10\ G15\ G12\ G16\ G\\
&\quad + 3.\ C\ G12\ G15\ G16\ G^2 + 3.\ G15\ C\ G10\ G13\ G^2 - 2.\ C\ G16\ G13\ G15\ G^2 + C\ G10\ G14\ G12\ G16\ G\\
&\quad + 2.\ G16\ C\ G13\ G10\ G15\ G)\ s + 2.\ G16\ G13\ G10\ G15\ G^2)
\end{aligned}
$$

For the numeric values of the parameters

$$
\begin{gathered}
C = 1.0\text{e}04\ \text{F}; G = 0.0001\ \text{S}; G_{10} = 0.0002\ \text{S}; G_{16} = 0.0001\ \text{S};\\
G_{14} = 2.0\ \text{S}; G_{15} = 1.0\ \text{S}; G_{13} = 4.0\ \text{S and}\ \ \text{G}_{12} = 1.0\ \text{S}.
\end{gathered}
$$

Figure 29 shows the Bode diagram, and Fig. 30 presents the distribution of poles and zeros in the complex plane.

Example 7 The analog circuit shown in Fig. 31 contains all four types of linear controlled sources. The graph pairs $\left(G_1^i, G_1^v\right)$, and $\left(G_p^i, G_p^v\right)$, generated according to the rules presented in Sect. 5, are given in Figs. 32 and 33, respectively.

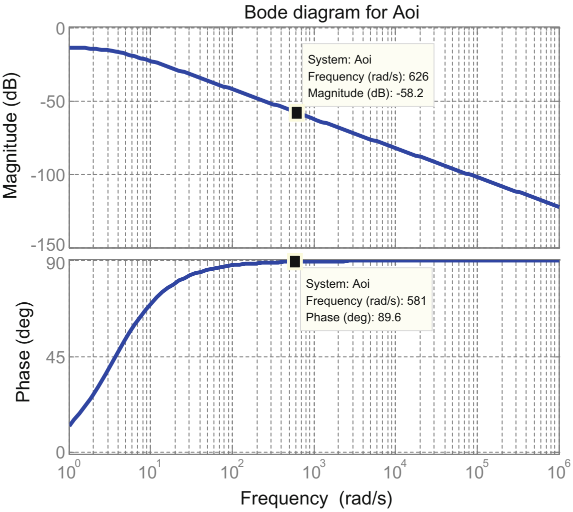


Fig. 29 Bode diagram for A_{oi}

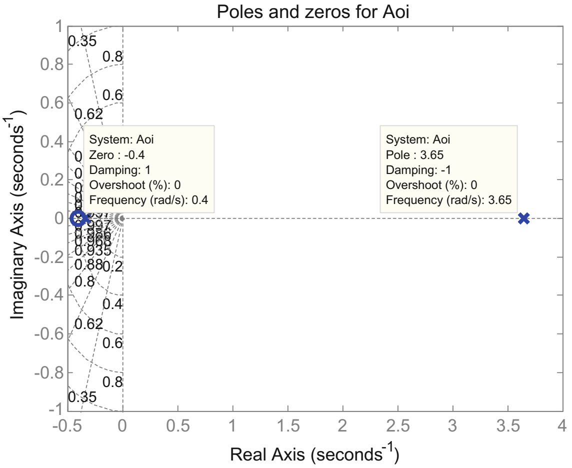


Fig. 30 Pole and zero locations of small signal amplifier

At the beginning, we generate the voltage gain $A_{oi\_ex}(s)$, and we evaluate it at the nominal parameter values, keeping only the complex frequency s as a symbol. Then we compute the transfer function sensitivity A_{oi_ex} $(s,\ x)$ in respect of each parameter x. The analysis is performed considering an initial sampling in the frequency range of interest and checking the error in some intermediate points. The circuit elements that have a small value of the relative sensitivity in this frequency range can be eliminated. To this end both zero-admittance (element removal) and a zero-impedance (contraction of the terminal nodes) can be used. The value of the voltage gain in which some nodes/branches have been contracted/eliminated, A_{oi} $_{ap}(s)$, is computed.

The magnitude and phase errors are given by:

$$\varepsilon_{|A_{oi}|} = \frac{|A_{oi\_ex}(j\omega)| - |A_{oi\_ap}(j\omega)|}{|A_{oi\_ex}(j\omega)|}, \tag{65}$$

$$\Delta\phi_{A_{oi}} = \frac{\arg(A_{oi\_ex}(j\omega)) - \arg(A_{oi\_ap}(j\omega))}{\arg(A_{oi\_ex}(j\omega))}. \tag{66}$$

For the circuit in Fig. 31 we find that only the capacitor C_{22} can be eliminated by contraction of its nodes, because the voltage gain sensitivity is small in the frequency range of interest, as it is shown in Fig. 34. In Fig. 35 the error variations in the same frequency range of the transfer function magnitude and of the transfer function phase are represented.

After the capacitor C_{22} removal, and applying the above procedure we obtain the reduced graphs $\left(G^i_{1r}, G^v_{1r}\right)$ and $\left(G^i_p, G^v_p\right)$. The tree number reduction of these graphs is shown in Table 7.

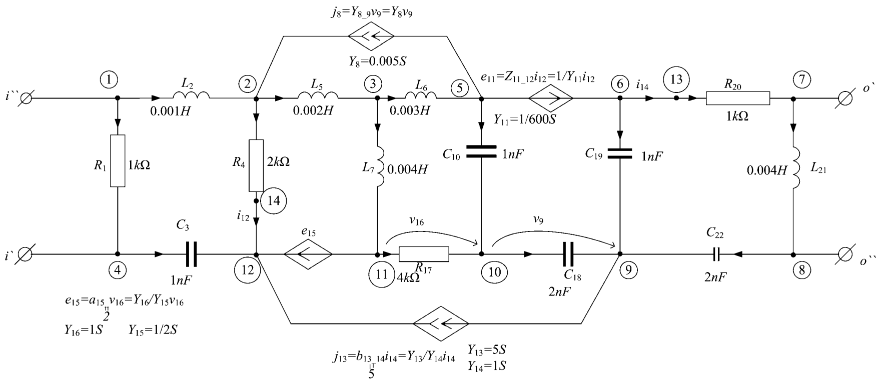


Fig. 31 Circuit diagram

If the representation of the inductors and controlled sources based on the unity gyrator model [3, 4], is used, the number of trees in the current graph increases at least at 2821968 (612 times bigger than with our models). The running time to enumerate these trees on an AMD XP 2700, 2.16 GHz, 512 MB of RAM is presented by comparison in Table 8.

The next step is to generate the numerator and the denominator expressions of the transfer function in the decreasing order of the complex frequency powers, and the coefficients of each complex frequency power in the decreasing order of their values. In this way we can eliminate one by one, the terms with the smallest values, if an error criteria for the magnitude and phase is verified over the frequency range.

The numerator of the voltage gain for the analog circuit in Fig. 31 has the following full symbolic expression:

$$\begin{aligned} numerator := {} & C3\;C18\;L21\;Y14\;(L6\;C10\;L7\;Y15 + L5\;L6\;C10\;Y15 + L5\;C10\;L7\;Y15)\,s^4 \\ & + C3\;C18\;L21\;Y14\;(L7\;C10\;R17\;Y15 + C10\;L7\;R17\;Y16 + L5\;C10\;Y15\;R17)\,s^3 \\ & + C3\;C18\;L21\;Y14\;(L5\;Y15 + L7\;Y15 + Y11\;R4\;L7\;Y15)\,s^2 \end{aligned},$$

and the denominator contains 387 terms.

According to the above procedure of elimination we obtain finally a reduction in the transfer function denominator from 387 to 31 terms.

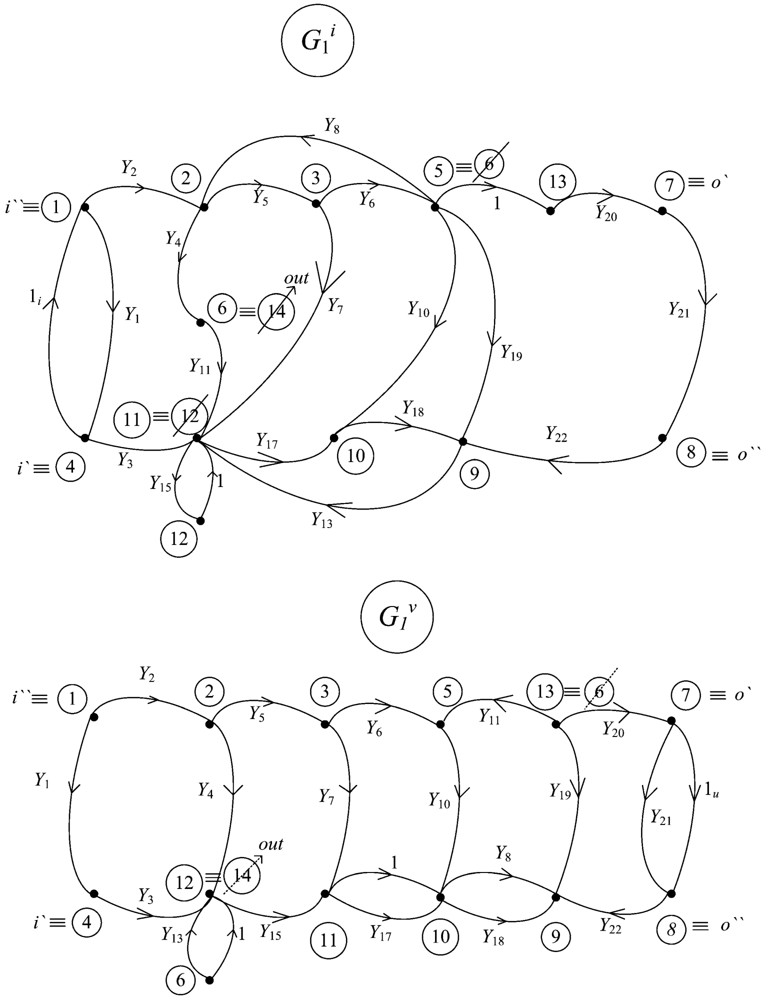


Fig. 32 $\left(G_1^i, G_1^v\right)$ graphs

In Fig. 36 the exact magnitude curve (502 terms in the denominator), that without C_{22} (387 terms in the denominator), and the approximated magnitude (31 terms in the denominator) are represented, and in Fig. 37 we can see the phase variation in the three cases. The maximum error of the transfer function magnitude is 1.6%.

A new method to formulate the system of equations in order to compute fully-symbolic small-signal characteristics of analog circuits by applying standard NA and/or loop current method has been presented.

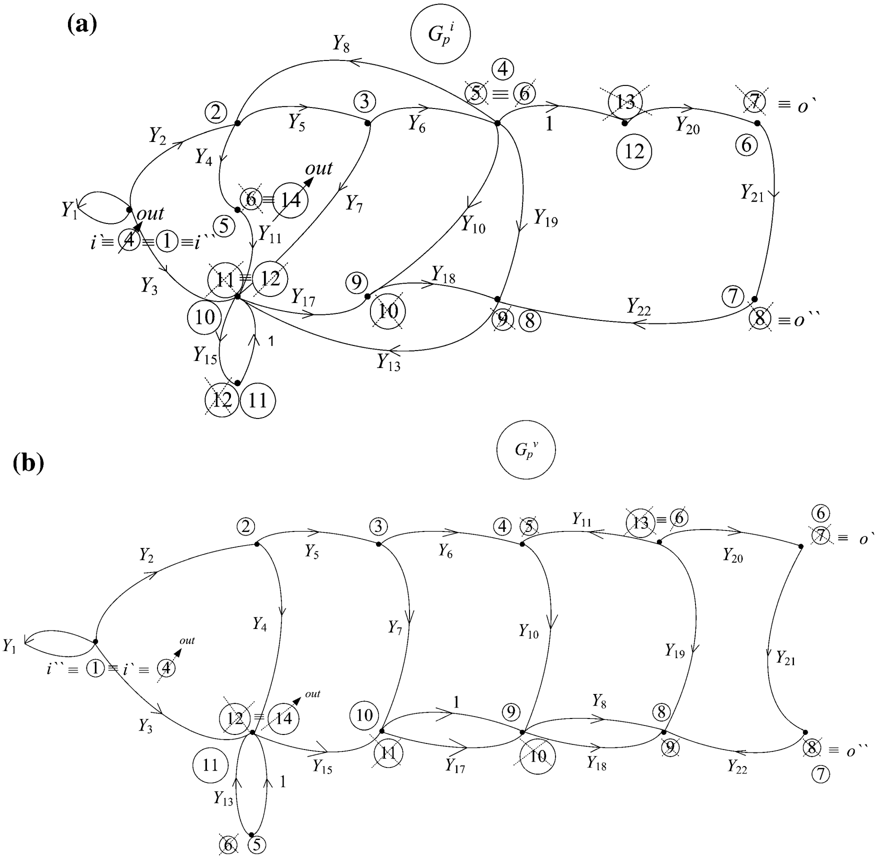
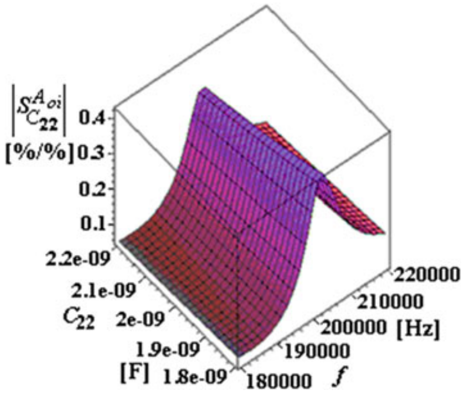


Fig. 33 $\left(G_p^i, G_p^v\right)$ graphs

Fig. 34 Sensitivity magnitude function of C_{22} and frequency f



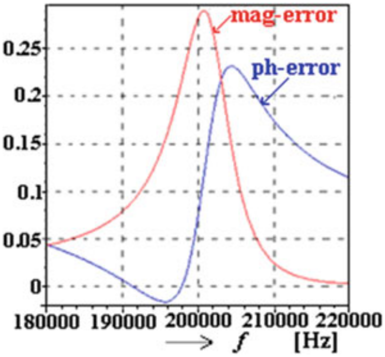


Fig. 35 Errors in voltage gain magnitude and in voltage gain phase

Table 7 Tree number reduction of the graphs from Figs. 32 and 33

$t_{G_1^i} = 4608$	$t_{G_1^v} = 3326$	$t_{G_p^i} = 4608$	$t_{G_p^v} = 3340$
$t_{G_{1r}^i} = 1792$	$t_{G_{1r}^v} = 209$	$t_{G_{pr}^i} = 1792$	$t_{G_{pr}^v} = 358$

Table 8 Comparison of running time for enumerating the trees corresponding to the graphs from Figs. 32 and 33

Number of trees	Time for tree	Enumeration [seconds]
	Algorithm on level	Char's algorithm
4608	≤1 μs	≤1 μs
2821968	1,781	4

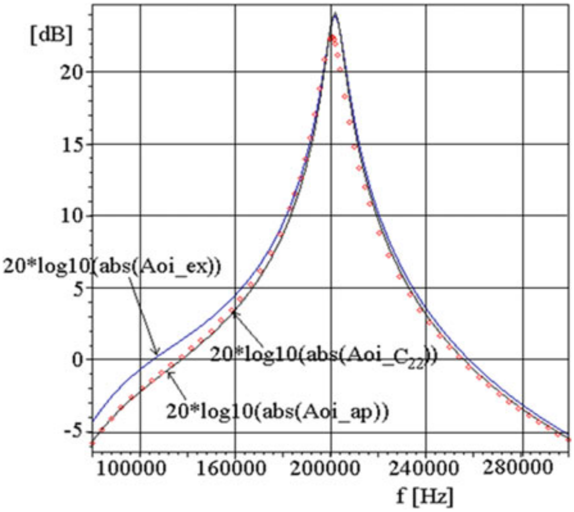
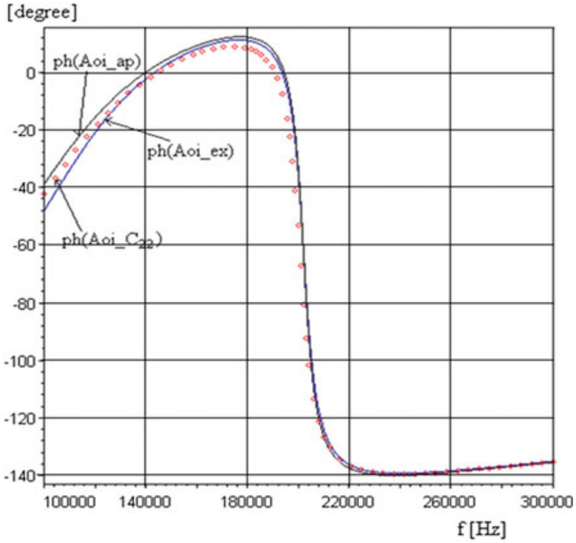


Fig. 36 Variation of the voltage gain magnitude in the frequency range

Fig. 37 Phase variation in the frequency range



7 Conclusions

By modelling electronic devices with equivalent circuits containing nullors and by associating to the analyzed circuit two graphs: one corresponding to the current one —G^i, necessary to formulate the KCL, and one corresponding to the voltage one—G^v, necessary to formulate the KVL, the nodal equations and the loop current equations can be formulated very simple for any non-reciprocal circuit. The two graphs have the same number of branches, nodes and independent loops, but they differ by their different positions they occupy in the two graphs, by the branches used to simulate the controlled sources and, in general, by the branches corresponding to the equivalent circuits containing nullors used to model the electronic devices. The characteristics of the branches are written using the voltages from the voltage graph and the currents from the current graph.

In this chapter, we propose a simple modelling procedure of the controlled sources in the two graphs. The equivalent circuits based on the functional schemes with nullors model both the controlling port and the controlled one by admittances placed in different positions in the two graphs. The two graphs obtained in this way have the same number of branches, nodes, and loops. A new method to formulate the system of equations in order to compute fully-symbolic small-signal characteristics of analog circuits by applying standard NA and/or loop current method has been presented.

A set of rules for generating and using the two graphs is stated, and the generalization of the topological formula to generate all network functions is proved. These rules are applicable to a linear circuit containing: all four types of linear controlled sources, resistors, inductors, capacitors, nullors (for ideal opamps operating in the linear mode), and any multi-terminal or multiport circuit element having an equivalent scheme made up only by two-terminal elements and controlled sources.

The models with nullors for all active electronic devices are more effective for the optimization of design and simulation time during the analysis process. From this point of view, the nullors proved already their efficiency in the active devices modelling. In the models based on nullors, the parasitic elements can be included to analyze their contribution to the analog circuit response. All the four controlled sources can also be represented with equivalent circuits using nullor elements. Consequently, the nullors are very useful for the analog circuits modelling because the circuit topology can be described using only two-terminal components like resistors, capacitors, nullators, norators, independent and controlled sources. Considering that the model should be developed in the simplest manner and the accuracy of the circuit behaviour simulation must be in acceptable limits, this chapter will show the problems related to the small-signal models of the active devices modelled with nullors.

Unlike other similar approaches our approach does not introduce supplementary branches and nodes with respect to initial circuit. Moreover, the number of nodes in the two graphs is smaller than in the initial circuit with the number of CCVS. Modelling the controlled sources by admittances allows an efficient generation of the network functions via the generalized topological formula with homogeneous parameters. This formula works for linear nonreciprocal networks containing any type of controlled sources. The rules for the automated generation of the two graph pairs using the controlled source models proposed in this chapter and a representation on levels of the graphs were implemented in a very fast program for the symbolic transfer function computation.

The generalized topological formula can generate any network function in a full symbolic form for very large-scale analog circuits because the numerator and the denominator terms are generated one by one and stored as lists. This gives the superiority of the topological approach in contrast to the determinant method that cannot provide a full symbolic form because of the symbolic manipulator that cannot solve huge systems of algebraic equations.

The list form in which the numerator and the denominator are obtained also allows performing the simplification after generation in a simple manner.

Examples have been introduced to show the usefulness of the nullor-based models and the potential of the proposed approach for the analysis and design of the analog linear/nonlinear circuits.

From two-port and four-terminal network point of view, all the proposed models have been generated by taking into account the impedance levels associated to the input-output terminals along with the gain-equations of the active devices. As one can see throughout the chapter, the nullor-based models are not complex and they can quickly be included into symbolic analyzers. Further, nullor-based active device models by including parasitic elements, has also been introduced. Furthermore, a novel method to formulate the system of equations in order to compute fully-symbolic small-signal characteristics of analog circuits by applying only standard NA has been presented. Thus, by using the relationships of nullators and norators and by manipulating their data-structures, the admittance matrix can quickly be constructed, avoiding waste of CPU-time and memory in the formulation process. Examples have been introduced to show the usefulness of the nullor-based models and the potentiality of the proposed formulation method

References

1. Fakhfakh M, Tlelo-Cuautle E, Fernandez FV (eds) (2012) Design of analog circuits through symbolic analysis. Bentham Science Publishers, pp. 83–114, 228–262 (Respectively e-book). https://doi.org/10.2174/97816080509561120101, ISBN: 9781-60805-095-6
2. Vlach J, Singhal K (1993) Computer methods for circuit analysis and design. Kluwer, Norwell, Massachusetts
3. Gielen G, Sansen W (1991) Symbolic analysis for automated design of analog integrated circuits. Kluwer Academic Publishers, USA
4. Fernández FV, Rodriguez-Vázquez A, Huertas JL, Gielen G (1998) Symbolic analysis techniques: applications to analog design automation. IEEE Press, Piscataway, NJ
5. Carlin HJ (1964) Singular networks elements. IEEE Trans Circuit Theory 11:67–72
6. Dumitriu L, Lordache M (1998) Teoria modernă a circuitelor electrice—Fundamentare teoretică, Aplicaţii, Algoritmi şi Programe de calcul, vol 1, Editura All Educational S.A., Bucureşti, ISBN 973–9337–99– 6
7. Lordache M, Dumitriu L (2006) The generalized topological formula for transfer functions' generation by two-graph tree enumeration. Analog Integr Circ Sig Process **47**(1): 85–100. Kluwer Academic Publishers
8. Mayeda W (1972) Graph theory. Wiley, New York
9. Lordache M (1980) Generalization of the topological formulas with homogeneous parameters. Rev Roum Sci Techn Électrotechn et Énerg 4:501–513
10. Cristea P, Lordache M, Dumitriu L, Spinei F (1995) On tree generation diakoptic method used in circuit symbolic analysis. In: Proceedings of European Conference on Circuit Theory and Design, ECCTD'95, Istanbul, Turkey, Vol II, 27–31 Aug 1995, pp 625–635
11. Lordache M, Dumitriu L (1997) Symbolic analysis of large analog integrated circuits using a two-graph tree enumeration method. In: Proceedings of European Conference on Circuit Theory and Design, ECCTD'97, Budapest, Hungary, pp 468–473
12. Lordache M, Dumitriu L (1997) An approximation symbolic analysis of large analog integrated circuits. Rev Roum Sci Techn Electrotechn et Energ 42(4):445–458, Bucharest
13. Lordache M, Dumitriu L, Muntean R, Botinant R (1998) An algorithm for finding all spanning trees in increasing weight order. In: Proceedings of Symbolic Methods and Applications in Circuit Design, SMACD'98, Kaiserslautern, Germany, 8–9 Oct 1998, pp 99–105
14. Dumitriu L, Lordache M, Muntean R, Botinant R (1998) Efficient generation of symbolic network functions using two-graph decomposition on levels. In: Proceedings of Symbolic Methods and Applications in Circuit Design, SMACD'98, Kaiserslautern, Germany, 8–9 Oct 1998, pp 191–198
15. Dumitriu L, Lordache M (1999) Techniques for fast symbolic analysis of large analogue integrated circuits. In: Proceedings of Signals, Circuits and Systems, SCS' 99, Iasi, Romania, 6–8 July 1999, pp 57–60
16. Mayeda W, Seshu S (1965) Generation of trees without duplications. IEEE Trans Circuit Theory CT-12(12):181–185
17. Yu Q, Sechen C (1996) A unified approach to the approximation symbolic analysis of large analog integrated circuits. IEEE Trans Circuits Syst I Fundam Theory Appli 43(8):656–669

18. Lin PM (1991) Symbolic network analysis. Elsevier, Amsterdam, Oxford, New York, Tokyo
19. Rodanski B (2002) Extension of the two-graph method for symbolic analysis of circuits with non-admittance elements. In: Proceedings of Symbolic Methods and Applications in Circuit Design, SMACD'02, Sinaia, Romania, 10–11 Oct 2002, pp 17–20
20. Gielen G, Sansen W (1991) Symbolic analysis for automated design of analog integrated circuits. Kluwer Academic, Boston, MA
21. Dumitriu L, Lordache M (2003) Efficient procedures for the automatic generation of transfer functions in symbolic form by two-graph tree enumeration. In: Proceedings of the European Conference on Circuit Theory and Design, ECCTD'03, Cracow, Poland, 1–4 Sept 2003, pp II-398–II-401
22. Jayakumar R, Thulasiraman K, Swamy MNS (1984) Complexity of computation of a spanning tree enumeration algorithm. IEEE Trans Circuits Syst CAS-31(10):853–860
23. Jayakumar R, Thulasiraman K, Swamy MNS (1989) MOD-CHAR: an implementation of char's spanning tree enumeration algorithm and its complexity analysis. IEEE Trans Circuits Syst 36(2):219–228
24. Galan M, Fernandez FV, Vazquez AR (1997) A New 3-Matroid Intersection Algorithm for Simplification During Generation in Symbolic Analysis of Large Analog Circuits. In: Proceedings of European Conference on Circuit Theory and Design, ECCTD'97, Budapest, Hungary, pp 1310–1315
25. Verhagen W, Gielen G (1998) An efficient evaluation scheme for linear transfer functions using the determinant decision diagram representation of the system determinant. In: Proceedings of the Fifth International Workshop on Symbolic Methods and applications in Circuit Design, SMACD'98, Kaiserslautern, Germany, 8–9 Oct 1998, pp 99–105
26. Rodriguez-Garcia JD, Guerra O,Roca E, Fernandez FV, Vazquez AR (1998) A new simplification before and during generation algorithm. In: Proceedings of the Fifth International Workshop on Symbolic Methods and Applications in Circuit Design, SMACD'98, Kaiserslautern, Germany, 8–9 Oct 1998, pp 110–124
27. Galan M, Fernandez FV, Vazquez AR (1997) Comparison of matroid intersection algorithms for large circuit analysis. In: Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS'97, Hong Kong, 9–12 June 1997, pp 1784–1787
28. Lordache M, Dumitriu L (2014) Computer-aided simulation of analogue circuits: algorithms and computational techniques. POLITEHNICA Press Publishing, Bucharest
29. Iordache M (2015) Symbolic, numeric—symbolic and numeric simulation of analog circuits —user guides. MATRIX ROM Publishing, Bucharest in ROM
30. Tellegen BDH (1966) On nullators and norators. IEEE Trans Circuit Theory CT-13:466–469
31. Carlosena A, Moschytz GS (1993) Nullators and norators in voltage to current mode transformations. Int J Circuit Theory Appl 21:421–424
32. Sánchez-López C, Tlelo-Cuautle E (2005) Behavioral model generation for symbolic analysis of analog integrated circuits. In: IEEE International Symposium on Signals, Circuits and Systems, pp 327–330
33. Tlelo-Cuautle E, Sánchez-López C, Martínez-Romero E, Tan SXD (2010) Symbolic analysis of analog circuits containing voltage mirrors and current mirrors. Analog Integr Circ Sig Process https://doi.org/10.1007/s10470-010-9455-y
34. Tlelo-Cuautle E, Martinez-Romero E, Sánchez-López C, Tan SXD (2009) Symbolic formulation method for mixed-mode analog circuits using nullors. In IEEE International Conference on Electronics, Circuits and Systems, pp 856–859