Analog Integrated Circuits and Signal Processing 3, 73–83 (1993)

More Efficient Algorithms for Symbolic Network Analysis: Supernodes and Reduced Loop Analysis

RALF SOMMER, DIRK AMMERMANN, AND ECKHARD HENNIG
Institut für Netzwerktheorie und Schaltungstechnik, Technische Universität Braunschweig, FRG

"It's funny how many of the best ideas are just an old idea back-to-front."
Douglas Adams

Abstract. In this paper, two efficient approaches will be discussed that support linear network analysis: supernode analysis (SNA) and reduced loop analysis (RLA). By means of some selected example networks, these methods will be demonstrated and, thus, it will be shown that calculations can be dramatically simplified. In this way, all network situations can be handled. There are obvious advantages to SNA as it combines the MNA and the straightforward manual processing of the network. A very efficient solution strategy is obtained without source shifting and other common, less directed methods being used. SNA/RLA and symbolic algebra fit extremely well together. Thus an algorithm that supports the symbolic calculation of networks by means of supernodes which has been conceptualized and implemented in the analog design expert system EASY will be presented in detail. Above the educational aspect, it should be noted that the computer can now take a systematic approach to MNA and network analysis in general.

1. Introduction

In recent years, symbolic methods have become increasingly important in the equation-based nonfixed topology approach to the automated design of analog circuits. Contrary to numerical calculations, symbolic calculations tend to be very complicated and time-consuming, especially if inversion of large matrices—typically generated—must be performed. Thus, in order to make symbolic network analysis feasible, it is necessary to reduce the amount of mathematical work the task requires, to an absolute minimum. This can be achieved by first selecting an analysis method appropriate to the task, and then, systematically taking advantage of forced conditions implied by the network.

In this paper, two efficient approaches will be discussed that support linear network analysis: supernode analysis (SNA) and reduced loop analysis (RLA). By means of some selected example networks, these methods will be demonstrated and, thus, it will be shown that calculations can be dramatically simplified. Since the simplified linear modeling of amplifier circuits such as OpAmps and transistors through controlled sources is becoming increasingly important, this aspect will also be covered by the paper. Special importance will be attached to the treatment of controlled sources with infinite gain and their nullator/norator equivalents. A few additional rules are appended to SNA and RLA which make these approaches cope with nullors. An example network containing nullors will then be analyzed with SNA and RLA to demonstrate the universality and efficiency of the methods in all conceivable situations. Finally, some ideas will be presented on how SNA can be implemented on a computer; it will be shown how to automatically generate a smaller system of equations in comparison to popular methods. This can be achieved by using a set of general rules for each type of network element combined with the topological information supplied by the network.

2. Basic Relations and Theory

Before launching into the discussion of efficient network analysis techniques, a remark must be made on an old bad habit which can be traced back to the time of Kirchhoff himself and has still not been rooted out. Very often, in literature as well as in lectures, loop equations are set up as sums of branch voltages, giving something like the following for each loop:

$$u_1 + u_2 + \ldots + u_k = 0$$

Then, the u_i are expressed one by one in terms of loop currents, and finally, those intermediate results are reinserted in the above equation. This procedure is not actually incorrect, but, it is contrary to all efforts that try to avoid the introduction of unnecessary variables and equations.

To introduce unnecessary variables and equations complicates the system of equations that need to be solved. Furthermore, it obstructs any attempt towards applying loop analysis to networks containing elements that have no impedance representation at all, such as current sources and open-circuit branches. In a similar way, everything mentioned here applies to the setting up of node equations as well.

There exist several approaches to the systematic formulation of network equations which either require the network to contain certain types of elements only or which yield a larger number of equations but can handle all types of elements. A well-known example for the latter is modified nodal analysis (MNA). Regrettably, MNA introduces additional equations although there should be no need to set up more than $n - 1$ equations if there are n nodes[1] in the network. Likewise, l loops should not yield more than l equations for any given network. Actually, there is no reason at all why current sources should not appear in networks to be analyzed with loop analysis (LA). For example, if a source current i could be identified with one independent loop current j_k, the loop equation to obtain j_k would not even need to be set up, as j_k is equal to i. The same applies to nodal analysis (NA). If a voltage source is considered as a forced potential difference of u between its two terminal nodes (A) and (B), then one potential would be immediately known if either of the potentials V_B or V_A were known. It turns out that every current source sets a forced condition for LA much in the same way that every voltage source does for NA. Consequently, each forced condition reduces the degree of freedom of the independent equations by one. These approaches will be called supernode analysis (SNA) and reduced loop analysis (RLA).

3. Manual Equation Setup and Motivating Examples

Before some examples can be calculated to illustrate the back-to-front idea underlying these approaches, a clear definition of what supernodes are must be made before they are used.

DEFINITION. *Supernodes* are generalized cut-sets[2] enclosing independent and/or dependent voltage sources [1].

These special cut-sets will be used in a very efficient way as will be shown below.

3.1. Algorithm for Setting Up SNA Equations

To set up the equations needed for SNA done by hand, follow these steps:

1. Label each of the n nodes of the network, one of which must be the reference node. Thus $(n - 1)$ node voltages have to be taken as independent variables.
2. Mark all supernodes by surrounding all the cut-sets of voltage sources (s is number, no matter whether dependent or independent) by a closed line.
3. Set up all forced conditions for each supernode. Use the forced conditions to eliminate s node voltages. Take one reference node voltage of each supernode as the independent variable.
4. Set up the remaining $(n - s - 1)$ generalized cut-set equations. (Set up one equation for each supernode as well as for every remaining regular node.)

Remark. Controlling currents have to be expressed in terms of node voltages and element relations. This might require additional node equations.

3.2. Algorithm for Setting Up RLA Equations

The setup of the equations for RLA can be divided into the following steps:

1. Remove all current sources (s in number, dependent or independent) from the network.
2. Introduce $(l - s)$ loop currents for the remaining $(l - s)$ independent closed loops.
3. Reinsert the current sources step by step and assign only one loop current to a closed loop laid across this source in each step.
4. Identify all loop currents flowing through the current sources with the source currents themselves.
5. Set up all remaining loop equations. Express all voltages in these loops in terms of loop currents, the constraint equations from step 4, and element relations.
6. If there are voltage controlled sources, additional equations have to be set up.

Another procedure to obtain an optimized set of loops is to generate a tree from the network in which the current sources are located within the interconnection branches. This tree approach can also be used to prove the correctness of RLA. Furthermore, it should be relevant to support the implementation of RLA on a computer, because there are several methods to find exactly those trees by making use of the nodal incidence matrix **A** [3].

The following examples will hopefully put aside any difficulties.

3.3. Example Demonstrating SNA

In the network shown in figure 1, there are six loops and two independent current sources: this yields $6 - 2 = 4$ equations when RLA is applied. On the other hand, when using SNA, a total of one supernode (SN_1) equation and one element equation for the CCVS are needed. This result is arrived at by identifying five nodes, taking into account the four voltage sources, and deducing the one required supernode equation. The facts listed above favor the supernode approach to reduce the work most efficiently.

Now the steps for the supernode method are applied:

1. All nodes are labeled by $V_1, \ldots, V_5$ respectively.
2. Two supernodes are found and marked (see above).
3. The forced conditions are set up (reference voltage of supernode SN_1 is V_1): The additional equation for the CCVS is $V_3 - V_1 = rG_4(V_1 - V_5)$; this directly applied results in

$$V_2 = V_1 + U_{01} \qquad (f1)$$

$$V_4 = V_1 + U_{02} \qquad (f2)$$

$$V_5 = U_{03} \qquad (f3)$$

$$V_3 = V_1 + r(V_1 - U_{03})G_4 \qquad (f4)$$

4. Now one supernode equation can be written down straightforward, directly expressing node potentials via ($f1$) to ($f4$) (SN_0 belongs to the reference node, so no extra node equation is needed):

$$I_{02} + G_3V_1 + G_4(V_1 - U_{03}) + G_5((V_1 + U_{02}) - U_{03}) - I_{01} = 0$$

which is only *one independent equation* with V_1 as unknown.

3.4. Example Demonstrating RLA

Since the network shown in figure 2 contains no voltage sources or short-cut branches, it seems to be suited perfectly to standard nodal analysis. A 3×3 matrix would have to be set up and inverted, if this method was used. By counting the number of loops and subtracting the number of current sources it turns out, that only equation remains to be solved if RLA is applied.

Following the strategy for the network on the left-hand side results in the steps listed below:

1. Removing the current sources leaves only one closed loop.

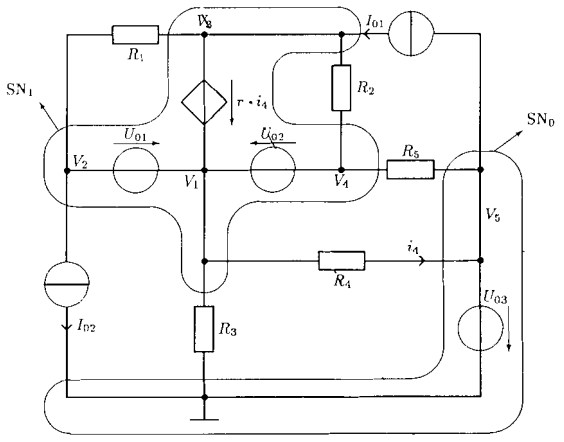

Fig. 1. Example to demonstrate MNA and SNA.

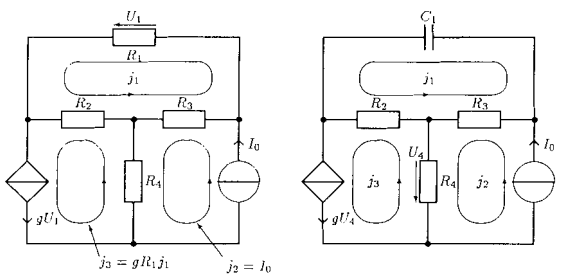

Fig. 2. Example for RLA.

2. The loop current j_1 is assigned to this loop.
3. Reinserting I_0 closes the loop $L2$ (identified with loop current j_2), then the VCCS is inserted, thus completing loop $L3$ (identified with loop current j_3).
4. The loop currents j_2 and j_3 are identified with $j_2 = I_0$ and $j_3 = gU_1$, where $U_1 = R_1 j_1$.
5. By directly inserting these relations the loop equation $L1$ is set up:

$$R_1(j_1) + R_2(j_1 - gR_1 j_1) + R_3(j_1 - I_0) = 0$$

This equation could be solved with ease.

The network on the right-hand side is a little more complicated as the current j_3 through the VCCS flows in the controlling branch (R_4), too. For this reason, one more equation has to be considered: Steps 1 and 2 are the same as above, but in step 3 the voltage U_4 across R_4 has to be expressed in terms of loop currents and element relations:

$$3.\ U_4 = R_4(-gU_4 + I_0) \Rightarrow U_4 = \frac{R_4 I_0}{1 + gR_4}$$

Now the loop equation ($L1$) to derive $\underline{J}_1$[3] is set up:

$$4.\ L1: \frac{1}{j\omega C_1}\underline{J}_1 + R_2\left(\underline{J}_1 - g\frac{R_4 I_0}{1 + gR_4}\right) + R_3(\underline{J}_1 - I_0) = 0$$

$$\Rightarrow \underline{J}_1 = \frac{R_2 R_4 g I_0/(1 + gR_4) + R_3 I_0}{1/j\omega C_1 + R_2 + R_3}$$

3.5. SNA in Comparison to Source Shifting

There is an interesting relation between the equations which result from the SNA and RLA approach and those obtained from standard methods by means of source shifting. The crux of the problem is the question of where to shift which sources? This question is quite simple to answer now: for both methods (SNA and RLA), the number of equations minus the forced conditions is to be determined. The method with the smaller overall number (zero is a trivial case), is the approach which will require the sources to be shifted in that direction. If SNA is the preferred method, then current sources have to be created. This may require more than one shift (see preceeding example). On the other hand, if RLA is the preferred method, then voltage sources have to be created through source shifting.

The circuit (figure 3) is another example for using SNA. On the other hand, when using RLA, six loops are found, and two forced conditions are set by current sources. As a result, source shifting or RLA would result in four equations. Applying the steps of SNA to the network in figure 3:

1. All nodes are labeled by $V_1, \ldots, V_4$ respectively.
2. Two supernodes are found and marked (see above).

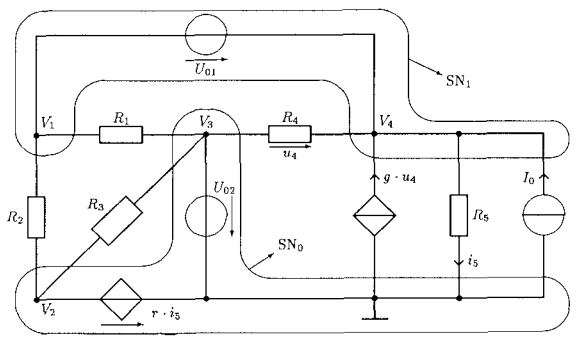

Fig. 3. Network, supernodes marked.

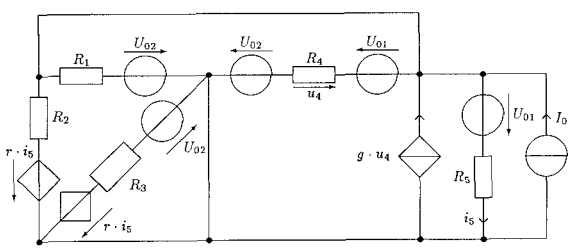

Fig. 4. Network, voltage source shifting.

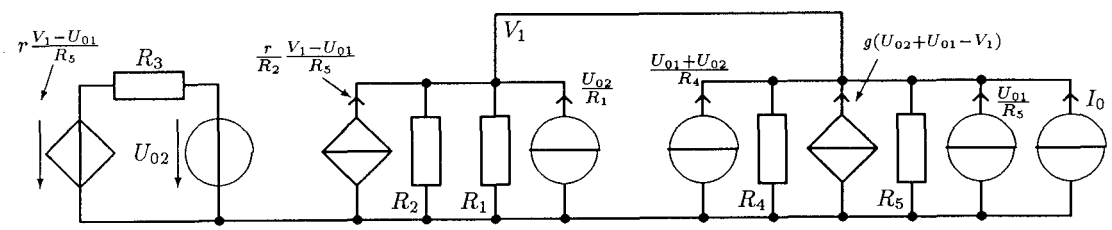

Fig. 5. Transformation to Norton equivalent circuit.

3. The forced conditions are set up (first independent voltage sources, then controlled sources; reference voltage of supernode SN_1 is V_1):

$$V_3 = U_{02} \qquad (f1)$$

$$V_4 = V_1 - U_{01} \qquad (f2)$$

$$V_2 = rG_5V_4 = rG_5(V_1 - U_{01}) \qquad (f3)$$

4. Now one supernode equation can be written down immediately, directly expressing node potentials via ($f1$) to ($f3$) (SN_0 belongs to the reference node, so no extra node equation is needed):

$$G_2(V_1 - rG_5(V_1 - U_{01})) + G_1(V_1 - U_{02}) + G_4((V_1 - U_{01}) - U_{02}) - g(U_{02} - (V_1 - U_{01})) + G_5(V_1 - U_{01}) - I_0 = 0$$

which is one equation with V_1 as unknown.

4. Applying SNA to Nullor Networks

Since there already exist other approaches that make nullors fit into the concept of nodal analysis, it would be interesting to know if the same object could be accomplished with SNA. In fact, only the following two rules have to be observed to achieve the goal.

1. A norator must be considered as a voltage source with an unknown output voltage.[4] Therefore, norators are treated like all other voltage sources as far as their inclusion in a supernode is concerned. However, since their output voltage is arbitrary, they do not furnish any constraint equations. This is logical as each norator causes the rank of the admittance matrix to be reduced by one.
2. Nullators must not be incorporated into a supernode. Each nullator forces the potentials at its two terminal nodes to be equal, thus eliminating one node voltage from the system of equations.

To demonstrate the application of SNA to nullor networks, the example network in figure 6 will be analyzed. The task shall be to compute the node voltages V_3 and V_6.

1. All nodes are given individual labels/variables: $V_1, \ldots, V_6$.
2. All supernodes are marked. In this case, there exists only one supernode, which consists of the two voltage sources U_1 and U_2, and both norators.

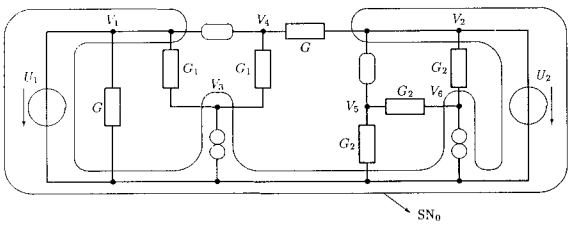

Fig. 6. Example nullor network for SNA.

3. The constraint equations are written down. The voltage sources in SN_0 demand

$$V_1 = U_1, \qquad V_2 = U_2$$

The nullator conditions demand

$$V_4 = V_1, \qquad V_5 = V_2$$

4. All remaining node equations are set up. There is no equation needed for SN_0 because it is the reference node. Moreover, no equations are necessary for nodes 1, 2, 3, and 6, since they belong to SN_0. This leaves only nodes 4 and 5 to supply the missing two independent equations. By immediately inserting the constraints from step 3, the following equations are obtained.

$$N4: \quad G_1(U_1 - V_3) + G(U_1 - U_2) = 0$$

$$N5: \quad GU_2 + G_2(U_2 - V_6) = 0$$

These equations can easily be solved for V_3 and V_6.

It becomes apparent from this example and other research, that SNA is a truly universal and powerful tool for handling all imaginable network elements and configurations.

5. An RLA Approach to Nullor Networks

The following example demonstrates that RLA is also able to handle nullors very well.

Figure 7 shows the network that will be analyzed below. In fact, it is almost the same network on which

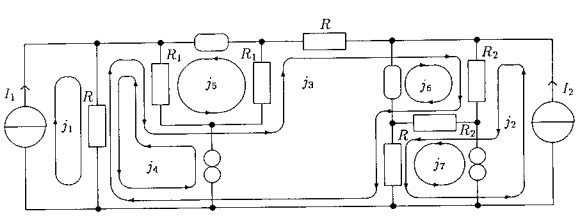

Fig. 7. Gyrator equivalent circuit.

an SNA was already performed in Section 4. The task supposed here is to compute the output current of the norator on the right-hand side. Since the unknown value is a current, it is best to apply RLA. Thus, one loop current has to be defined for each of the seven independent loops as was shown when RLA was introduced. As far as the nullors are concerned, a few additional rules must be observed.

1. A nullator is a special case of a current source. Hence, only one single loop current may pass through each nullator. Nullator loop equations are thus set up like regular loop equations. The nullator loop current is forced to be equal to zero, and consequently, does not appear in the equation. In this way, a nullator loop furnishes two equations: one loop equation and one constraint equation for the loop current.
2. In spite of the fact that a norator may be traversed by any number of loop currents, it is recommended, however, to let only one loop current flow through each norator. The idea is that every norator reduces the rank of the impedance matrix by one, because its voltage as well as its current are arbitrary. Thus, there will be no need to set up the norator loop equation at all, because it is linearly dependent on the other equations.
3. Otherwise, if for any reason, more than one loop current must be laid across a norator, it is necessary to set up all norator loop equations with an unknown norator voltage u that must first be eliminated from all equations but one. Then this last equation which still contains the unwanted unknown u may be deleted because it is no longer needed.

Thus, a nullator reduces the number of variable loop currents by one, whereas a norator saves one entire loop equation.

From figure 7, the following four constraint equations are obtained:

$$j_1 = I_1, \quad j_2 = I_2, \quad j_5 = 0, \quad j_6 = 0$$

Hence, only j_3, j_4, and j_7 are left as unknowns. Loop equations must be set up for loops 3, 5, and 6. There are no equations needed for loops 4 and 7 because of the reasons mentioned in rule 2.

$$L3: \quad R(j_3 - j_4 - I_1) + R_1(j_3 - j_4) + R_1 j_3 + Rj_3 + R_2(j_3 + I_2) + R_2(j_3 + j_7 + I_2) + R(j_3 + j_7 + I_2) = 0$$

$$L5: \quad R_1(j_3 - j_4) + R_1 j_3 = 0$$

$$L6: \quad R_2(j_3 + I_2) + R_2(j_3 + j_7 + I_2) = 0$$

Sorting the equations and variables results in the following 3×3 system that must be solved for the norator output current j_7.

$$\begin{bmatrix} 3R + 2R_1 + 2R_2 & -R - R_1 & R + R_2 \\ 2R_1 & -R_1 & 0 \\ -2R_2 & 0 & -R_2 \end{bmatrix} \begin{bmatrix} j_3 \\ j_4 \\ j_7 \end{bmatrix} = \begin{bmatrix} RI_1 - (2R_2 + R)I_2 \\ 0 \\ 2R_2 I_2 \end{bmatrix}$$

The unknown op amp output current j_7 can be obtained as $j_7 = 2I_1$ as a result of only a few mathematical steps. In this case, RLA proves to be even more efficient than SNA because SNA would have required another node equation to express the norator current in terms of node voltages and element relations.

6. Correlation of MNA and SNA

6.1. Motivating Example: Supernodes for Use in Computer-Aided Analysis

Independent of the network type and size, the supernode method is an important and extremely useful tool for circuit analysis. This method is usually done by hand but can be easily adapted for use by computers with symbolic network analysis programs. This is especially relevant, as compact equations are much more important for symbolic calculations as they would be for numerical calculations. The cost benefit of preprocessing the equations, and thereby reducing them in number, is strongly noticed later in the much simplified arithmetic that must be done. For example, the symbolic solution present in figure 1 requires the setting up of one equation, which is also linear. The additional four equations are "forced conditions" which could, with the result of a single equation mentioned, be simply solved. On the other hand, the use of MNA would require the setting up of 10 equations with 10 unknowns. The latter is obviously more difficult.

6.1.1. Supernode Approach. A supernode analysis of this network has already been performed in Section 3.3, resulting in only one independent equation which must be solved. This poses an interesting question: Is it possible to interpret and to derive the supernode method from the MNA or the general system of equations? If this is possible, then the topological information contained in the supernode(s) could be used before (or in) the MNA, and thereby simplify the amount and

type of mathematics needed to solve the system of equations. In the following sections, this aspect will be investigated in detail. Furthermore, an algorithm already implemented in EASY [4, 5], will be presented and discussed.

The MNA is a well known and commonly implemented method for the analysis and calculation of networks: for example, SPICE. At this moment, only the results of the theory behind the MNA are needed, so that the matrix can be filled appropriately for each element. The node-based equations (KCL) represent each row of the aforementioned matrix.

6.2. *The Interpretation of a Supernode*

Consider a simple supernode with an independent voltage source (figure 8). The unknowns needed for the usual approach are V_p, V_q, and the supporting current i_k. The following two equations are then set up:

$$\text{Node } p: \quad i_1 + i_2 + i_k = 0$$

$$\text{Node } q: \quad i_3 + i_4 - i_k = 0$$

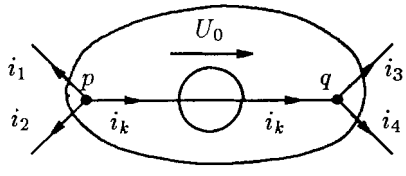

Fig. 8. Independent voltage sourced as supernode.

The currents i_1, i_2, i_3, and i_4 are then soleley functions of nodal potentials and element relations (e.g., Ohm's law). But, a new variable, i_k, has appeared, thus requiring an additional equation:

$$V_p - V_q = U_0 \qquad (*)$$

The construction of a supernode means that a cut-set equation has to be set up, and this is nothing more than the addition of the two node equations in nodes p, q. In this way, the current i_k must eliminate itself as it is present in a positive sense in one equation and negative in the other one. This results in only one cut-set equation, in which the potentials V_p and V_q are still present as unknowns:

$$i_1 + i_2 + i_3 + i_4 = 0$$

This degree of freedom can be immediately reduced by applying the forced condition (*). Consequently, one of the two potentials remains as an unknown, whereas the other potential is well defined (in terms of the first), and is no longer an equation in the system that needs to be solved. The number of unknowns has been reduced from three to one. For example, V_q can be expressed as $V_q = V_p - U_0$. Exactly this elimination is recognizable in the filling pattern of the MNA matrix (see figure 9). It is possible to add row p to row q. One of the original two rows may then be deleted. In this manner, the variable i_k can be eliminated, so that this column may be deleted as well. This seems to be the RMNA approach sometimes referenced in the literature [6].

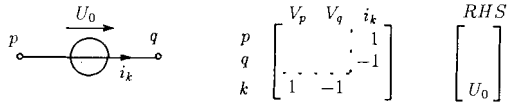

Fig. 9. MNA fill-in patern of an independent voltage source.

It is possible to obtain another simplification by using the row k as a forced condition to eliminate either V_q or V_p. Consequently another column has disappeared.[5] This latter step may not be suited for a numerically based program as it is not able to perform simple equation manipulation. On the other hand, a symbolically based program is able to perform this extraction of subexpressions in terms of one or more variables quite simply. As a result, only one of three equations remain to be solved, and there is only one as opposed to three unknowns left to be solved for. If this were extrapolated onto a large system, the benefit would become quite apparent [8, p. 125]. In the same way, VCVS and CCVS can be handled.

The description of the algorithm in Section 8 will clarify any doubts that may exist.

7. Nullator, Norator, and Nullor

Nullators and norators can be easily incorporated into the analysis. A nullor consists of a norator and a nullator. The schematic and matrix fill-in model are presented in figure 10. Two properties are easily recognized.

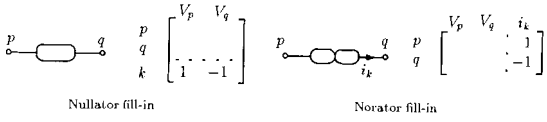

Fig. 10. Fill-in patterns of a nullator and a norator.

1. Each of the nullator and norator introduce only one new row or column, but not both. Consequently, the matrix is no longer square, and the system of equations is singular.
2. A nullor (a nullator and a norator) eliminates one row and one column from the matrix. This is equivalent to constructing a supernode around the norator. Mathematically, row r is summed to row s, resulting in row r being removed from the matrix. In this way, the norator current is eliminated from the list of unknowns. A nullator equates V_p to V_q. This collection of unknowns means that column p is added to column q, and column p can be removed. Likewise, V_p is removed from the list of unknowns.

8. Algorithm Implemented in EASY

EASY [4, 5] is an experimental analog design expert system developed at the Institute of Network Theory at the Technical University of Braunschweig. The following algorithm was conceptualized and implemented in EASY. The algorithm supports the symbolic calculation of networks by means of supernodes. It offers the possibility to express the results in the Belevitch form, which is needed to support the calculation of networks containing nonlinear elements via a piecewise linear representation. This will be referred to as the PWL Tool, and is described in more detail in [9].

The algorithm contains the following steps:

1. Read in network/netlists (to calculate currents, a short circuit branch must be identified).
2. Set up standard matrix for MNA.
3. Create lists needed for the evaluation.
   a. Create list of all control currents (L_1).
   b. Create list of all generated currents in the MNA (L_2).
   c. Deduce list of desired currents (L_3).
   d. Generate the union list/set of L_1 and L_3 (L_{union}).
4. For all currents from node k to k', $i_{k,k'} \notin L_{\text{union}}$:
   a. Add row k to row k'.
   b. Rename row k' to k, k'.
   c. Delete row k and column $I_{k,k'}$, i.e., the column belonging to the eliminated current.

   Explanation (4a). By the addition of row k and row k', a cut-set of node k and k' is generated: the supernode. The internal currents through the voltage sources and the short-circuits will be eliminated in this way.

   Explanation (4b). The inclusion of the row k' to the supernode k, k' still allows access to the original node(s), and thereby does not hinder the collection of several nodes into a supernode.

   Note. If k or k' is the reference node (ground) then step 4a is not performed. In place of step 4b, the computer generates supernode SN_0 labeled with 0, k.

   Explanation. The reference node row is linearly dependent on the other nodal rows. It may be deleted because it has been incorporated into SN_0. The voltage reference is remembered.
5. For all short-circuits between node j and node j':
   a. Add column j to column j'.
   b. Delete column j which is redundant.
   c. Remove row $m + 1$ which is a zero row. (Row $m + 1$ denotes the row belonging to the short-circuit.)

   Explanation (5a). A short-circuit means $v_j = v_{j'}$, and therefore, the columns are combined.

   Explanation (5b). Row $m + 1$ informs that $v_j = v_{j'}$. This is the forced condition.
6. Fixing of desired voltages:
   a. Output of all remaining nodal voltages.
   b. User input of desired voltages in terms of node voltage differences.
   c. Apply the following scheme for the substitution of nodal voltages by branch voltages:
      (i) Potentials not needed for the description of the desired voltages must remain.
      (ii) Of the r branch voltages that exist, as many (if not all) as possible are to be used to replace the node voltages (s) being their number). For this reason, the rank of the matrix V must be r, otherwise the voltages are linearly dependent on each other.

$$\begin{bmatrix} 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_{..} \\ \vdots \\ V_r \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & 0 & \dots \\ 1 & 1 & 0 & \dots \\ 0 & 1 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} U_{1,..} \\ U_{2,..} \\ U_{..,..} \\ \vdots \\ U_s \end{bmatrix} \quad (1)$$

By using Gauss-Seidel elimination techniques, the matrix will be restructured to appear in the form:

$$\begin{bmatrix} 1 & 0 & 0 & \dots \\ 1 & 1 & 0 & \dots \\ 0 & 1 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} U_1,\,. \\ U_2,\,. \\ U.\,.,\,. \\ \vdots \\ V_4 \\ V.\,. \\ \vdots \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \\ V.\,. \\ \vdots \end{bmatrix} \quad (2)$$

In this way the r node voltages are replaced by r desired branch voltages and $s - r$ nodal voltages.

$$[\mathbf{v}] = \mathbf{T} \begin{bmatrix} \mathbf{u} \\ \mathbf{v}' \end{bmatrix}$$

7. By means of Gauss-Seidel elimination, the potentials of $\mathbf{v}'$ are removed and a set of equations results containing only the desired quantities.
8. With user-input desired output format, the solution of the set of equations is constructed.
   a. Explicit solution
   b. Belevitch form (n-port equations)

These procedures written in MACSYMA [10] for EASY [4, 5] directly correspond to the described algorithm. In EASY, the results are used for

- fast simulation based on symbolic expressions to display the results in oscilloscope-like icons that may be directly manipulated.
- piecewise linear tool, which needs a special preprocessed Belevitch form [9].

9. Cookbook Approaches, Educational Aspects

In this section, the educational value of these ideas will be discussed. It has been shown that source shifting, Norton and Thévenin equivalent circuits, superpositioning of sources, calculating op amp circuits, wise use of approximations and simplifications in practical circuits, and some aspects of circuit design are easy to embed into the global concept of SNA and RLA. Students have often commented that the above listed ideas have been juggled in a haphazard somewhat nonpredictable way resulting in a poor understanding of circuit analysis. Consequently, many may now be able to interpret several of the SNA equations as one or the other of the above listed "magical tricks."

10. Conclusions

The supernode method, when applied manually, allows for a strongly reduced number of unknown voltages and/or currents. By not calculating the currents with the aid of voltage sources and short circuits, it is possible to find efficient generalized cut-sets which consist only of voltage sources (either dependent or independent). These cut-sets are called supernodes. The descriptive equations inside the supernodes are mostly simplistic relationships. These should be used at a very early stage to simplify the required linear algebra. In the case of current controlled sources, the currents should be expressed directly from the network by use of element relations in terms of node potentials.

The methods encompassing nullors and their implementation into SNA/RLA have been made full use of in the development of these algorithms. Consequently, the use of nullors allows for the construction of simplified networks and avoids complicated limit calculations.[6] In this way, all network situations can be handled. There are obvious advantages to SNA as it combines the MNA and the straightforward manual processing of the network. A very efficient solution strategy is obtained without source shifting and other common, less directed methods being used. SNA/RLA and symbolic algebra fit extremely well together. As symbolic algebra is able to identify and perform matrix row operations to reduce the degree of the system it strongly supports the ideas of SNA/RLA. Numerical methods, on the other hand, may be able to identify elementary matrix operations but cannot perform them on symbolic quantities.

So far, the scope of the independent node potentials and loop currents has been underestimated. It has been the intention of this paper to look at the basic principles behind loop currents and nodal voltages. In many ways, this paper covers the very basics and may be considered trivial, but it identifies some very simple ideas. These ideas have contributed more to circuit analysis than the various techniques and aids commonly known. It is hoped that the various aspects of circuit analysis have been tied together in an algorithm which produces an efficient and compact representation of the mathematics.

The resulting amount of work is much less in comparison to sparse tableau or MNA approaches. Above the educational aspect, it should be noted that the computer can now take a systematic approach to MNA and network analysis in general. The fact that the current centered representation has proven so fruitful may result in more research in this area.

Acknowledgments

The authors would like to acknowledge the students C. Beckmann, H. Trispel, and G. Weinerth for their contributions toward the project. We would especially like to thank A. Reibiger, Technical University Dresden, for profound discussions and for providing the theoretical background, and H. Ziemann for his invaluable help during the translation of this paper.

Notes

1. There exist some extreme situations in which these additional equations are needed to express controlling currents.
2. Generalized cut-sets are not necessarily minimal cut-sets [2]. This means that the removal of a generalized cut-set may split the network graph into more than only two components.
3. Remark: This notation means that the current is in the frequency domain, commonly known as a phasor.
4. This intuitive explanation will be confirmed in Section 7.
5. These compactions are exactly the same as those applied by the CMNA implemented in ISAAC [7, 8, 11]. In fact, the CMNA is isomorphic to the SNA.
6. Not subject of this paper.

References

1. J. David Irwin, *Basic Engineering Circuit Analysis*, Macmillan: New York, 1987.
2. A. Reibiger, Private communication, Dresden, Braunschweig, 1991.
3. L.O. Chua and P.M. Lin, *Computer-Aided Analysis of Electronic Circuit Analysis*, Prentice Hall: Englewood Cliffs, 1975.
4. R. Sommer, R. Kamitz, and E.-H. Horneber, "Qualitative reasoning in the analog design expert system EASY," in *Proc. ECCTD'91*, Copenhagen, 1991.
5. R. Sommer and E.-H. Horneber, "EASY—an experimental analog design system framework," in *Proc. Int. Workshop Symbolic Methods and Applications to Circuit Design*, Paris/Bagneux, 1991.
6. K. Lee and S. Park, "Reduced modified nodal approach to circuit analysis," *IEEE CAS-32*, No. 10, 1985.
7. W. Sansen and G. Gielen, "ISAAC: A symbolic simulator for analog integrated circuits," *IEEE J. Solid-State Circuits*, Vol. 24, No. 6, pp. 1587–1597, 1989.
8. G. Gielen and W. Sansen, *Symbolic Analysis for Automated Design of Analog Integrated Circuits*, Kluwer: Boston, 1991.
9. R. Sommer, D. Ammermann, and E.-H. Horneber, "Qualitative reasoning and nonlinear effects in analog design expert system EASY," in *Proc. ICM'91*, Kairo, 1991.
10. Symbolics *MACSYMA Reference Manual*, Version 13, 1988.
11. W. Sansen and G. Gielen, in *Proc. Summer Course in Systematic Analogue Design*, Katholiek Universiteit Leuven, 1990.

Ralf Sommer was born in Reinbek, Germany, on October 17, 1961. He received his degree in electrical engineering from the Technical University of Braunschweig in 1988. Since 1988 he has been a Ph.D. candidate at the Institute of Network-Theory and Circuit-Design at TU-Braunschweig. His current researches are development of an expert system for analog circuit design; topics of interest include network theory, symbolic computer algebra, and qualitative circuit analysis.

Eckhard Hennig was born in Westerstede, Germany, on January 14, 1969. He is a graduate student in electrical engineering at the Technical University of Braunschweig.

Dirk Ammermann was born in Bremen, Germany, on March 16, 1967. He started studying electrical engineering at the Technical University of Braunschweig in 1987. In 1991–92, he attended Georgia Institute of Technology, where he received his M.S. in electrical engineering.