C. W. Ho

Modified Nodal Approach to DC Network Sensitivity Computation

Abstract: Programming techniques are presented for computing dc sensitivity vectors of nonlinear electronic circuits. The modified nodal approach is used as the method of formulation for the circuit equations, in which multiple performance objectives can be accommodated. Numerical examples illustrate some of the techniques discussed.

1. Introduction

Recently, interest in numerical computation of circuit sensitivity with respect to design parameters has centered on the adjoint approach [1-4, 17], because it is general, fast, and efficient. Computation of the sensitivities relies mainly on generating and solving a set of equations that is adjoint to the original set. A simple and efficient programming method is then desirable for solving the adjoint equations.

Although the adjoint approach has aroused a great deal of interest among designers using computer-aided methods, it has yet to live up to its initial promise. It has not established itself as an indispensible fixture among modern network analysis and design programs [5-8] to the extent of such methods as sparse matrix or implicit integration.

We believe that part of the problem is due to the general difficulty of converting a practical circuit design problem, in a straightforward mathematical manner, into a single performance measure or objective function, because a balanced circuit design usually requires compromises among several, often conflicting, constraints. Another problem is related to program implementation. Although successful solutions to practical problems have been reported [9-12] using the adjoint method, in most cases the computer programs were developed for a specific class of problems only, and it is relatively difficult to adapt them for other applications.

This paper focuses on dc network sensitivity formulation and computation, taking cognizance of both the multiple performance objective and the implementation problem.

It is generally a nontrivial programming problem to generate the various terms of partial derivatives needed for computing the desired sensitivity vector in a simple and straightforward manner. To show this, let us start with a single performance function and consider the network equation $\mathbf{f}$ in a general form [12-13] for the dc case. We use $\mathbf{x}$ to denote the unknown vector of voltages and currents and $\mathbf{p}$ to denote the design parameter vector, so that

$$\mathbf{f}(\mathbf{x}, \mathbf{p}) = 0. \tag{1}$$

Typically the network elements (resistors, capacitors, etc.) are used in the equations as design parameters or functions of design parameters, and the sensitivity vector is the derivative of a performance characteristic that is either a network variable or a function of several network variables. If we assume that the performance characteristic of interest is a network variable $\mathbf{x}_i$, then the basic sensitivity vector can be defined as

$$\mathbf{s}_i = \left(\frac{d\mathbf{x}_i}{d\mathbf{p}}\right)^{\mathrm{T}}, \tag{2}$$

where superscript T denotes vector transpose. The perturbation of Eq. (1) due to vector $\mathbf{p}$ is

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\frac{d\mathbf{x}}{d\mathbf{p}} + \frac{\partial \mathbf{f}}{\partial \mathbf{p}} = 0. \tag{3}$$

We see that the sensitivity vector of interest $\mathbf{s}_i$ defined in Eq. (2) is actually the ith row of the sensitivity matrix $d\mathbf{x}/d\mathbf{p}$ in Eq. (3) after transposing it into a column vector. Alternatively, the transpose of the sensitivity vector $\mathbf{s}_i$ is equal to the ith row of the sensitivity matrix $d\mathbf{x}/d\mathbf{p}$ or

$$\mathbf{s}_i^{\mathrm{T}} = \mathbf{e}_i^{\mathrm{T}}\frac{d\mathbf{x}}{d\mathbf{p}}, \tag{4}$$

where $\mathbf{e}_i$ is an elementary column vector that contains a $+1$ in the ith position and zeros everywhere else. Equation (4) implies that

$$\mathbf{s}_i = \left(\frac{d\mathbf{x}}{d\mathbf{p}}\right)^{\mathrm{T}}\mathbf{e}_i,$$

which can be combined with Eq. (3) to yield

$$\mathbf{s}_i = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{p}}\right)^{\mathrm{T}} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^{-1\mathrm{T}} \mathbf{e}_i. \tag{5}$$

The following substitution in Eq. (5),

$$\mathbf{y} = \left[\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^{-1}\right]^{\mathrm{T}} \mathbf{e}_i,$$

implies that

$$\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^{\mathrm{T}} \mathbf{y} = \mathbf{e}_i \tag{6}$$

and

$$\mathbf{s}_i = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{p}}\right)^{\mathrm{T}} \mathbf{y}. \tag{7}$$

Therefore in order to solve for the basic sensitivity vector $\mathbf{s}_i$ an intermediate vector $\mathbf{y}$ is first evaluated by solving the adjoint equation (6) whereas the elementary vector $\mathbf{e}_i$ is used as the vector on the right hand side. The intermediate vector $\mathbf{y}$ is then substituted into Eq. (7) for the final sensitivity vector, where no matrix inversion is required.

As shown in Eqs. (6) and (7), both the adjoint equation and the sensitivity equation are formulation-dependent, and the corresponding computation speed and storage required to solve these equations may therefore vary for different network formulations. Indeed, a new network formulation, the modified nodal approach, was recently reported [14] and subsequently implemented in a general purpose network analysis and design program [8]. It was shown quantitatively that this approach is equal or superior to some of the existing formulations in terms of storage requirements and execution speed. Furthermore, the resulting Jacobian circuit matrix is basically numerically well-behaved for pivoting on the diagonal.

In the following section, we briefiy discuss the modified nodal approach for network analysis, which defines the Jacobian matrix $\partial\mathbf{f}/\partial\mathbf{x}$ of Eq. (6). (For a more detailed discussion of the modified nodal approach, see [14].) A table of "element stamps" is given for each type of element for constructing the Jacobian matrix. A corresponding sensitivity table, which can be used straightforwardly to construct the matrix $\partial\mathbf{f}/\partial\mathbf{p}$ of Eq. (7), is then presented in Section 4. After obtaining these two matrices, the basic sensitivity vector in Eq. (7) can be evaluated. A theorem of dc sensitivity computation for multiple performance objectives is then proved in Section 4, followed by examples and the summary.

2. Modified nodal analysis

In using the modified nodal approach to develop a general purpose network analysis and design program, the programming code needs to keep track of only two classes of objects with respect to the network being analyzed to generate the Jacobian matrix: all the *nodes* and only those network *elements* whose currents are to be included in the solution vector. The reason is that, in order to be completely general and to keep the matrix size to a minimum, the modified nodal equations need account for only those elements whose currents have to be included in the solution vector.

Network elements whose currents are included in the solution vector become part of the output if at least one of the following three conditions is met:

1. The element is either a voltage source, E, or an inductor, L.
2. Any other nonlinear circuit element depends on its current.
3. Its branch current is requested as an output variable by the user.

After identifying all nodes in the network and all elements whose currents are included in the solution vector because of the above conditions, a labeling process begins. A ground node (usually chosen by the user) is first labeled node 0. All the remaining u nodes in the network are then arbitrarily labeled in some order, e.g., $1, 2, 3, \cdots, u$. Subsequently, all of the remaining q elements whose currents are identified as outputs are labeled as $u+1, u+2, \cdots, u+q$. The remaining elements whose branch currents are not needed as outputs are not labeled. We now arrange the order of the equations such that the first u equations are the nodal equations and the remaining q equations are branch relations. For an element labeled as $u+i$, we reserve the $(u+i)$th equation in the set as its branch relation, and its current becomes naturally the $(u+i)$th member of the solution vector. The size of the Jacobian matrix and hence the length of the solution vector are both equal to the sum of the numbers of the nodal voltages u and branch currents q, i.e., $u+q$.

With the completion of the labeling process, the contribution of each element to the Jacobian matrix can be easily determined. For example, consider a voltage source E connected between nodes labeled i and j. Its branch relation is always included in the equations as required by condition 1. Let us assume that it is labeled k. Due to Kirchoff's current law, the contributions of E to the ith and jth nodal equations are I_E and $-I_E$, respectively, if we assume, for example, that the direction of current flowing out of a node is considered to be positive. The corresponding contributions to the Jacobian matrix are therefore $a+1$ and $a-1$ and the (i, k)th and (j, k)th positions, respectively, because I_E is the kth unknown in the solution vector. Furthermore, its branch relation $V_j - V_i = E$ becomes the kth equation and, in the Jacobian matrix, we have -1 and 1 in the (k, i)th and (k, j)th positions, respectively. The constant E appears in the right hand side of the kth equation.

Table 1 Element stamps for the modified nodal matrix.

Element type	Element stamps: Current not output	Element stamps: Branch current output
G	$\begin{array}{c\|ccc} & V_i & V_j & \text{RHS} \\ \hline i & G & -G & \\ j & -G & G & \end{array}$	$\begin{array}{c\|cccc} & V_i & V_j & I_G & \text{RHS} \\ \hline i & & & 1 & \\ j & & & -1 & \\ k & G & -G & -1 & \end{array}$
C	$\begin{array}{c\|ccc} & V_i & V_j & \text{RHS} \\ \hline i & \frac{C}{h} & -\frac{C}{h} & +\frac{C}{h} V_{cp} \\ j & -\frac{C}{h} & \frac{C}{h} & -\frac{C}{h} V_{cp} \end{array}$	$\begin{array}{c\|cccc} & V_i & V_j & I_C & \text{RHS} \\ \hline i & & & +1 & \\ j & & & -1 & \\ k & \frac{C}{h} & -\frac{C}{h} & -1 & \frac{C}{h} V_{cp} \end{array}$
L		$\begin{array}{c\|cccc} & V_i & V_j & I_L & \text{RHS} \\ \hline i & & & +1 & \\ j & & & -1 & \\ k & 1 & -1 & -\frac{L}{h} & -\frac{L}{h} I_{Lp} \end{array}$
E		$\begin{array}{c\|cccc} & V_i & V_j & I_E & \text{RHS} \\ \hline i & & & +1 & \\ j & & & -1 & \\ k & -1 & 1 & & E \end{array}$
J	$\begin{array}{c\|ccc} & V_i & V_j & \text{RHS} \\ \hline i & & & -J \\ j & & & +J \end{array}$	$\begin{array}{c\|cccc} & V_i & V_j & I_J & \text{RHS} \\ \hline i & & & 1 & \\ j & & & -1 & \\ k & & & 1 & J \end{array}$
Voltage-controlled voltage source $E = uV_{mn}$		$\begin{array}{c\|ccccc} & V_i & V_j & V_m & V_n & I_E \\ \hline i & & & & & 1 \\ j & & & & & -1 \\ k & -1 & 1 & -u & u & \end{array}$
Current-controlled voltage source $E = rI_r$		$\begin{array}{c\|cccc} & V_i & V_j & I_r & I_E \\ \hline i & & & & 1 \\ j & & & & -1 \\ k & -1 & 1 & -r & \end{array}$
Voltage-controlled current source $J = g_m V_{mn}$	$\begin{array}{c\|cccc} & V_i & V_j & V_m & V_n \\ \hline i & & & g_m & -g_m \\ j & & & -g_m & g_m \end{array}$	$\begin{array}{c\|ccccc} & V_i & V_j & V_m & V_n & I_J \\ \hline i & & & & & 1 \\ j & & & & & -1 \\ k & & & g_m & -g_m & -1 \end{array}$
Current-controlled current source $J = \beta I_r$	$\begin{array}{c\|ccc} & V_i & V_j & I_r \\ \hline i & & & \beta \\ j & & & -\beta \end{array}$	$\begin{array}{c\|cccc} & V_i & V_j & I_r & I_J \\ \hline i & & & 1 & \\ j & & & -1 & \\ k & & & 1 & -\beta \end{array}$

The contributions to the matrix of all the remaining element types can be similarly determined. The results are summarized in Table 1 as element stamps, in which the element in each case is assumed to be connected between nodes i and j. The positions of its branch relation in the matrix and its current in the solution vector are both assumed equal to k. For dependent elements, the controlling element is assumed to be connected between nodes m and n; the position of the current of the controlling element is equal to r.

In actual processing, Table 1 is stored in the computer. After the labeling process has been completed, each element in the network being analyzed is considered sequentially. Only if its current is *included* as an output is its contribution to the matrix extracted from the right hand column in the table. Otherwise the left hand column is used to construct the matrix for both its zero-nonzero pattern for pivoting and the numerical values for its evaluation.

Note that, for simplicity, partial derivatives for nonlinear elements are not included in the table. However, their inclusion in the matrix in actual programming can be handled by a cross referencing table containing indices for every dependent element and its independent variables. Our experiences and those of others [5] indicate that partial derivatives are generally not needed for nonlinear R, G, C, L types of elements to achieve effective convergence. This is because the branch relations of these types of elements have good numerical results compared to those of the Newton-Raphson method, which uses partial derivatives. The results are simplicity in programming and economy in storage. However, partial derivatives are needed for E and J type elements for effective convergence of Newton iterations.

3. Sensitivity matrix

In the last section, an approach was presented that can be used to construct the Jacobian matrix $\partial\mathbf{f}/\partial\mathbf{x}$ of Eq. (6). In this section, a corresponding scheme is presented that leads to the forming and evaluation of the sensitivity matrix $\partial\mathbf{f}/\partial\mathbf{p}$ of Eq. (7).

As shown in Section 2 the dimension of the network equation f is the summation of the number of nodes and those of currents that are used as outputs, i.e., $u + q$. From Eqs. (6) and (7) it is seen that the number of rows in the sensitivity matrix $\partial\mathbf{f}/\partial\mathbf{p}$ is also equal to $u + q$. The number of columns is equal to the number of design parameters in the network. Hence in addition to labeling nodes and currents for formulating the modifying nodal equations, a third labeling process is needed for the sensitivity matrix, which assigns $1, 2, 3, \cdots, n$ to the n given design parameters. As mentioned previously, design parameters are either circuit elements themselves or functions of circuit elements. In either case, partial derivatives of the network function $\mathbf{f}$ with respect to circuit elements are needed. As with the Jacobian matrix shown in the previous section, construction of the sensitivity matrix on an element-by-element basis is straightforward.

Consider a conductor G connected between nodes labeled i and j. Further, assume that the branch current of G is not needed as an output. Its contributions to the modified nodal matrix are therefore G, $-G$, $-G$, and G in locations (i, i), (i, j), (j, i), and (j, j), respectively, as shown in Table 1. This is so because the element G contributes to the ith and the jth nodal equations the term $\pm G(V_i - V_j)$, and partial derivatives are taken with respect to V_i and V_j. Now if G is considered to be a design parameter and partial derivatives are thus taken with respect to G for those equations to generate the sensitivity matrix, the terms $\pm(V_i - V_j)$ result in the (i, n)th and (j, n)th positions in the sensitivity matrix if we assume that G is the nth design parameter. On the other hand, if the current of G is needed as an output, then its contribution to the sensitivity matrix is simply $(V_i - V_j)$ in the (k, n)th position.

Other types of elements can be similarly considered for the derivation of their contributions to the sensitivity matrix. The resulting element stamps for this case are summarized in Table 2. For the computation of the dc sensitivity vector, the capacitors and inductors have no effect and hence are not included in the table.

Note that in Table 2 design parameters are network elements themselves. For the more general case, in which several network elements are functions of a single design parameter, the column in the sensitivity matrix corresponding to that design parameter is obtained as follows. The contribution to the sensitivity matrix for each network element is first determined by multiplying the appropriate terms in Table 2 by the derivative of that element with respect to the design parameter. Then, by addition, the resultant vectors are merged into one. For example, if two resistors, R_1 connected between nodes 1 and 2 and R_2 connected between nodes 2 and 3, are both dependent on a design parameter p_1, then the column vector in the sensitivity matrix corresponding to p_1 is

$$\left(\frac{\partial\mathbf{f}}{\partial\mathbf{p}}\right)_{\text{col } 1} = \begin{Bmatrix} (V_1 - V_2)\dfrac{-1}{R_1^2}\dfrac{dR_1}{dP_1} \\[2ex] (V_1 - V_2)\dfrac{1}{R_1^2}\dfrac{dR_1}{dP_1} - (V_2 - V_3)\dfrac{1}{R_2^2}\dfrac{dR_2}{dP_1} \\[2ex] (V_2 - V_3)\dfrac{1}{R_2^2}\dfrac{dR_2}{dP_1} \\[2ex] 0 \\ \vdots \\ 0 \end{Bmatrix}. \quad (8)$$

In computation, the network equation (1) is first solved by using the modified nodal matrix to yield the solution vector $\mathbf{x}$. It is used in conjunction with Table 2, stored in the computer, to generate the sensitivity matrix columnwise with respect to each design parameter. Note that the derivatives of network elements with respect to design parameters can be evaluated either analytically or by numerical perturbation, depending on the problem.

In the general case in which a scalar performance characteristic is not simply a network variable but a function of several network variables x_i, x_j, x_k, etc.,

$$P = g(x_i, x_j, x_k, \cdots). \tag{9}$$

Then the sensitivity vector of performance characteristic P versus design parameters $p_1, p_2, p_3, \cdots, p_n$ is

$$\frac{\partial \mathbf{P}}{\partial \mathbf{p}} = \frac{\partial g}{\partial x_i}\frac{\partial x_i}{\partial \mathbf{p}} + \frac{\partial g}{\partial x_j}\frac{\partial x_j}{\partial \mathbf{p}} + \frac{\partial g}{\partial x_k}\frac{\partial x_k}{\partial \mathbf{p}} + \cdots, \tag{10}$$

which is clearly a combination of the basic sensitivity vectors defined in Eq. (2). Substituting Eq. (5) for each of the basic sensitivity vectors into Eq. (10), after some manipulation we obtain the sensitivity vector

$$\begin{aligned} \mathbf{S} = \left(\frac{\partial P}{\partial \mathbf{p}}\right)^{\mathrm{T}} = &-\left(\frac{\partial \mathbf{f}}{\partial \mathbf{p}}\right)^{\mathrm{T}} \left[\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^{-1}\right]^{\mathrm{T}} \left(\frac{\partial g}{\partial x_i}\right) \mathbf{e}_i \\ &-\left(\frac{\partial \mathbf{f}}{\partial \mathbf{p}}\right)^{\mathrm{T}} \left[\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^{-1}\right]^{\mathrm{T}} \left(\frac{\partial g}{\partial x_j}\right) \mathbf{e}_j \\ &-\left(\frac{\partial \mathbf{f}}{\partial \mathbf{p}}\right)^{\mathrm{T}} \left[\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^{-1}\right]^{\mathrm{T}} \left(\frac{\partial g}{\partial x_k}\right) \mathbf{e}_k + \cdots \end{aligned}$$

or

$$\begin{aligned} \mathbf{S} = -\left(\frac{\partial \mathbf{f}}{\partial p}\right)^{\mathrm{T}} \left[\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^{-1}\right]^{\mathrm{T}} &\left[\left(\frac{\partial g}{\partial x_i}\right) \mathbf{e}_i + \left(\frac{\partial g}{\partial x_j}\right) \mathbf{e}_j \right. \\ &\left. + \left(\frac{\partial g}{\partial x_k}\right) \mathbf{e}_k + \cdots\right]. \end{aligned} \tag{11}$$

This is similar to Eq. (5) except that the elementary vector $\mathbf{e}_i$ in Eq. (5) is replaced by a summation of elementary vectors multiplied by the partial derivatives of the performance function with respect to each of the network variables upon which it depends. We use λ to denote the summation of vectors so that

$$\lambda = \left(\frac{\partial g}{\partial x_i}\right) \mathbf{e}_i + \left(\frac{\partial g}{\partial x_j}\right) \mathbf{e}_j + \left(\frac{\partial g}{\partial x_k}\right) \mathbf{e}_k + \cdots, \tag{12}$$

and combine it with Eq. (11) to yield

$$\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^{\mathrm{T}} \mathbf{y} = \lambda; \tag{13}$$

$$\mathbf{S} = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{p}}\right)^{\mathrm{T}} \mathbf{y}. \tag{14}$$

If some of the network variables in Eq. (9) are branch currents and are not included in the solution vector by conditions 1-3 in Section 2, those branch currents must be added to the solution vector in order to compute the sensitivity vector. It should be emphasized, however, that for this general case the computation cost for the sensitivity vector is still only one analysis for the adjoint equation (13) and one matrix multiplication of Eq. (14).

Table 2 Element stamps for the sensitivity matrix.

Element type	*Element stamps*: *Current output*	*Element stamps*: *Branch current output*
G	$\begin{array}{cc} & m \\ \hline i & (V_i - V_j) \\ j & -(V_i - V_j) \end{array}$	$\begin{array}{cc} & m \\ \hline i & \\ j & \\ k & (V_i - V_j) \end{array}$
R	$\begin{array}{cc} & m \\ \hline i & (V_i - V_j)\left(-\frac{1}{R^2}\right) \\ j & (V_i - V_j)\frac{1}{R^2} \end{array}$	$\begin{array}{cc} & m \\ \hline i & \\ j & \\ k & (V_i - V_j)\frac{1}{R^2} \end{array}$
E		$\begin{array}{cc} & m \\ \hline i & \\ j & \\ k & -1 \end{array}$
J	$\begin{array}{cc} & m \\ \hline i & 1 \\ j & -1 \end{array}$	$\begin{array}{cc} & m \\ \hline i & \\ j & \\ k & -1 \end{array}$
Voltage-controlled voltage source, $E = uV_{mn}$		$\begin{array}{cc} & m \\ \hline i & \\ j & \\ k & -(V_m - V_n) \end{array}$
Current-controlled voltage source $E = rI_r$		$\begin{array}{cc} & m \\ \hline i & \\ j & \\ k & -I_r \end{array}$
Voltage-controlled current, source $J = g_m V_{mn}$	$\begin{array}{cc} & m \\ \hline i & (V_m - V_n) \\ j & -(V_m - V_n) \end{array}$	$\begin{array}{cc} & m \\ \hline i & \\ j & \\ k & -(V_m - V_n) \end{array}$
Current-controlled current source $J = \beta I_r$	$\begin{array}{cc} & m \\ \hline i & I_r \\ j & -I_r \end{array}$	$\begin{array}{cc} & m \\ \hline i & \\ j & \\ k & -I_r \end{array}$

Sensitivity for multiple performance functions

So far we have considered the formulation of the sensitivity vector for a single dc performance function. For a more general case, a practical circuit design problem usually involves more than one performance function. Furthermore, those performance functions, such as power dissipation or voltage levels, often cannot be simply weighted and added together to form a single performance function throughout the design process. The weights for different performance functions may not be optimally chosen prior to the design process and may need constant updating. Also, during the design process the weights have to be adjusted accordingly if the maximal or minimal permissible values of some of the performance measures have been reached. Moreover, in some design parameter regions, some of the performance functions may have become relatively insensitive to the design parameters. Hence, in a computer aided design program, the capability of computing the sensitivity vector for an appropriate set of weights needs to be very flexible. The sensitivity vector for the given performance functions can be either computed individually or combined in some manner. The user can then use his own judgement, with the help of sensitivity vectors, to provide some direction (preferably in an interactive computing environment) to update his design to meet the design requirements.

Let us denote the network equations of Eq. (1) to be $\mathbf{f}_1$; the performance function equations, each as given by Eq. (9), to be $\mathbf{f}_2$; and the vectors of network variables, performance functions, and design parameters to be $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$, respectively. We now present a mathematical theorem concerning the dc sensitivity computations for multiple performance functions.

Theorem

Given is a vector

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{pmatrix},$$

where the dimensions of subvectors $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ are denoted as n_1, n_2 and n_3, respectively. Given also are a set of nonlinear algebraic equations

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \mathbf{f}_1\ (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \\ \mathbf{f}_2\ (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \\ \mathbf{x}_3 - \mathbf{c} \end{pmatrix} = 0, \tag{15}$$

where $\mathbf{c}$ is a known constant vector, and the following set of adjoint equations

$$\left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right)^{\mathrm{T}} \begin{pmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \\ \hat{\mathbf{x}}_3 \end{pmatrix} = \begin{pmatrix} \mathbf{Y} \\ \mathbf{Z} \\ 0 \end{pmatrix}, \tag{16}$$

where $\partial \mathbf{f}/\partial \mathbf{x}$ is the Jacobian matrix of $\mathbf{f}$ evaluated at the solution point of Eq. (15), which is assumed to exist. Here $\mathbf{Y}$ and $\mathbf{Z}$ are constant vectors. We have

$$\hat{\mathbf{x}}_3 = \left(\frac{\partial \mathbf{x}_1}{\partial \mathbf{x}_3}\right)^{\mathrm{T}} \mathbf{Y} + \left(\frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_3}\right)^{\mathrm{T}} \mathbf{Z}. \tag{17}$$

A proof is given in the Appendix.

We now explain the practical use of the theorem. After the solution of the nonlinear algebraic equation (15), the Jacobian matrix $\partial \mathbf{f}/\partial \mathbf{x}$ evaluated at the solution point is obtained. The adjoint equation (16) can easily be solved by choosing "arbitrarily" the constant vectors $\mathbf{Y}$ and $\mathbf{Z}$. Equation (17) indicates that as part of the adjoint solution, $\hat{\mathbf{x}}_3$ is the combined sensitivities of $\mathbf{x}_1$ and $\mathbf{x}_2$ with respect to $\mathbf{x}_3$ subject to weighting vectors $\mathbf{Y}$ and $\mathbf{Z}$ in a straightforward and flexible manner. For example, if the vector $\mathbf{Z}$ is set to zero in Eq. (16), the resulting vector $\hat{\mathbf{x}}_3$ is the sensitivity vector of all the network variables $\mathbf{x}_1$ with respect to design parameters $\mathbf{x}_3$, either singularly or combined, by changing the value of $\mathbf{Y}$. Specifically, if $\mathbf{Y}$ is chosen as

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

$\mathbf{x}_3$ becomes the sensitivity vector of the first nodal voltage with respect to all design parameters; if $\mathbf{Y}$ is chosen as

$$\begin{pmatrix} 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

$\hat{\mathbf{x}}_3$ becomes the sensitivity vector of the sum of the first two nodal voltages with respect to all design parameters, etc. Alternatively, by using the same Jacobian matrix but setting vector $\mathbf{Y}$ to zero, the sensitivity vectors of all the performance functions $\mathbf{x}_2$ can be computed, singularly or combined, by changing the value $\mathbf{Z}$. Of course, $\mathbf{Y}$ and $\mathbf{Z}$ can both be chosen to be nonzero for any desired combination of the various sensitivity vectors.

Although the theorem does not necessarily provide any further theoretical insights compared with what has already been discussed in Sections 2 and 3, it can be used in programming development to avoid a great deal of bookkeeping work by lumping multiple performance functions together and yet maintaining their independence. As mentioned earlier, the use of the theorem is especially flexible for sensitivity computation in an interactive environment.

The theorem presented is in a very general form. For most applications, the network function $\mathbf{f}_1$ in Eq. (15) is not dependent on the performance functions $\mathbf{x}_2$, and $\mathbf{f}_2$ is not dependent on the design parameters $\mathbf{x}_3$. The function for performance, $\mathbf{f}_2$, can be so defined that $\partial\mathbf{f}_2/\partial\mathbf{f}_2$ is equal to an identity matrix. Therefore, the evaluation of the Jacobian matrix in Eq. (A1) in the Appendix is actually the evaluation of the modified nodal matrix $\partial\mathbf{f}_1/\partial\mathbf{x}_1$ discussed in Section 2, the sensitivity matrix $\partial\mathbf{f}_1/\partial\mathbf{x}_3$ discussed in Section 3, and the performance matrix $\partial\mathbf{f}_2/\partial\mathbf{x}_1$, which can be obtained either analytically or by numerical perturbation.

5. Examples

This section provides two illustrative examples.

Example 1

A three-port resistive network is to be synthesized such that its short-circuit conductance matrix [15]

$$\mathbf{Y} = \begin{pmatrix} 4 & 1 & 2 \\ 1 & 6 & 3 \\ 2 & 3 & 8 \end{pmatrix}$$

(See [16] for related work.) Refer to Fig. 1, where we have assumed that between every pair of nodes in this three-port network there lies a 1-Ω conductor G_{12}, G_{13}, etc. The total number of conductors is 15. At each port there is also a voltage source, E_1, E_2, E_3. The approach used here is to compute the sensitivity vector of some of the performance functions to be defined later with respect to each of the 15 conductors such that the values of the conductors are changed by using the sensitivities to minimize the difference between the $\mathbf{Y}$ matrix of the network and the given $\mathbf{Y}$ matrix.

In order to synthesize the three-port matrix, we first set $E_1 = 1$, $E_2 = E_3 = 0$ and evaluate the three-port currents, I_{E_1}, I_{E_2}, and I_{E_3}. The three currents should be as close to the entries in the first row of the given $\mathbf{Y}$ as possible, i.e., 4, 1, and 2. Then the process is repeated by setting $E_2 = 1$, $E_1 = E_3 = 0$ and then $E_3 = 1$, $E_1 = E_2 = 0$. The three performance functions are thus defined as

$$\begin{aligned} P_1 &= (I_{E_1} - 4)^2 + (I_{E_2} - 1)^2 + (I_{E_3} - 2)^2, \\ P_2 &= (I_{E_1} - 1)^2 + (I_{E_2} - 6)^2 + (I_{E_3} - 3)^2, \text{ and} \\ P_3 &= (I_{E_1} - 2)^2 + (I_{E_2} - 3)^2 + (I_{E_3} - 8)^2 \end{aligned} \tag{18}$$

for the three different voltage sources.

To construct the modified nodal matrix $\partial\mathbf{f}_1/\partial\mathbf{x}_1$ of Eq. (A1) we choose node 1 as the ground node. It is labeled node 0 in the program. Nodes 2, 3, $\cdots$, 6 in Fig. 1

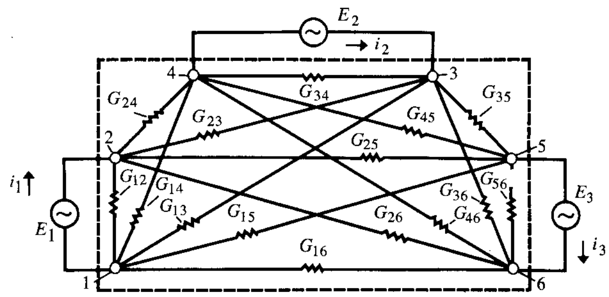

Figure 1 Three-port resistive network.

are then labeled 1, 2, $\cdots$, 5. There are therefore five nodal equations. The branch relations for voltage sources E_1, E_2 and E_3 are then included in the equation set. They are labeled 6, 7, and 8. Table 1 is then used for each network element. Because branch currents for the conductors are not needed, the corresponding element stamp for G is used. For the voltage sources, there is only one element stamp, so that stamp is used. The resultant matrix is shown in Table 3. The sensitivity matrix $\partial\mathbf{f}_1/\partial\mathbf{x}_3$ in Eq. (A1) is similarly formed by using Table 2 as we labeled the 15 design parameters 1 through 15. The resultant matrix is shown in Table 4, where V_2 is the nodal voltage for node 2, etc. Finally, the performance matrix $\partial\mathbf{f}_2/\partial\mathbf{x}_1$ is formed by differentiating Eq. (18) with respect to the network variables as shown in Table 5.

The modified nodal equations are solved three times for the three different voltage source value combinations mentioned above. The resultant values of V and I are used to evaluate the matrices in Tables 4 and 5. At each time only one of the three performance functions in Eq. (18) is pertinent. The obvious choices for the $\mathbf{Z}$ vector in Eq. (17) for the three different cases are

$$\mathbf{Z} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \mathbf{Z} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \text{ and } \mathbf{Z} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix};$$

and the vector $\mathbf{Y}$ is always chosen as zero. Three sensitivity vectors are obtained for the three computations. They are summed to form the total sensitivity vector for updating the design. For example, the sensitivity vectors $\mathbf{S}_1$, $\mathbf{S}_2$, and $\mathbf{S}_3$, computed when all the conductors having a value of one, are shown in Table 6. A Fletcher-Powell algorithm has been programmed that updates the conductor values such that the resultant $\mathbf{Y}$ matrix after 15 iterations is

$$\mathbf{Y} = \begin{pmatrix} 3.9999 & 1 & 1.9999 \\ 1 & 5.9999 & 3 \\ 1.9999 & 3 & 7.9999 \end{pmatrix},$$

which is very close to the $\mathbf{Y}$ matrix given.

Table 3 Elements of matrix $\partial \mathbf{f}/\partial \mathbf{x}_1$.

$G_{21}+G_{23}+G_{24}+G_{25}+G_{26}$	$-G_{23}$	$-G_{24}$	$-G_{25}$	$-G_{26}$	-1	0	0
$-G_{23}$	$G_{31}+G_{32}+G_{34}+G_{35}+G_{36}$	$-G_{34}$	$-G_{35}$	$-G_{36}$	0	1	0
$-G_{24}$	$-G_{34}$	$G_{41}+G_{42}+G_{43}+G_{45}+G_{46}$	$-G_{45}$	$-G_{46}$	0	-1	0
$-G_{25}$	$-G_{35}$	$-G_{45}$	$G_{51}-G_{52}+G_{53}+G_{45}+G_{46}$	$-G_{56}$	0	0	1
$-G_{26}$	$-G_{36}$	$-G_{46}$	$-G_{52}$	$G_{61}+G_{62}+G_{63}+G_{46}+G_{56}$	0	0	-1
-1	0	0	0	0	0	0	0
0	1	-1	0	0	0	0	0
0	0	0	1	-1	0	0	0

Table 4 Elements of $\partial \mathbf{f}_1/\partial \mathbf{x}_3$.

V_2	V_2-V_3	0	0	0	0	0	V_2-V_4	0	0	0	V_6-V_2	0	V_2-V_5	0
0	V_3-V_2	V_3-V_4	0	0	0	V_3	0	V_3-V_5	0	0	0	0	0	V_3-V_6
0	0	$-V_3+V_4$	V_4-V_5	0	0	0	$-V_2+V_4$	0	V_4-V_6	0	0	V_4	0	0
0	0	0	$-V_4-V_5$	$-V_5-V_6$	0	0	0	$-V_3+V_5$	0	V_5	0	0	$-V_5+V_5$	0
0	0	0	0	$-V_6+V_5$	V_2	0	0	0	$-V_4+V_6$	0	V_2-V_6	0	0	$-V_3+V_6$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5 Elements of performance matrix $\partial \mathbf{f}_2/\partial \mathbf{f}_1$.

0	0	0	0	0	$2 \cdot (I_{E_1}-4)$	$2 \cdot (I_{E_2}-1)$	$2 \cdot (I_{E_3}-2)$
0	0	0	0	0	$2 \cdot (I_{E_1}-1)$	$2 \cdot (I_{E_2}-6)$	$2 \cdot (I_{E_3}-3)$
0	0	0	0	0	$2 \cdot (I_{E_1}-2)$	$2 \cdot (I_{E_2}-3)$	$2 \cdot (I_{E_3}-8)$

Table 6 Vertical columns containing elements of sensitivity vectors.

$\mathbf{S}_1$	$\mathbf{S}_2$	$\mathbf{S}_3$
−2	0	0
−1	−2	0
0	−6	0
0	−3	−4
0	0	−10
−1.5	0	−3.5
0	−1	0
0	−1	0
0	0	−1
0	0	−1
0.5	0	−1.5
0.5	0	−1.5
−1	−2	0
−1.5	0	−3.5
0	−3	−4

Example 2

A current switch emitter follower circuit is shown in Fig. 2, in which input to the circuit is represented by the voltage source E_{in}. Two output voltages are taken from nodes OUT_1 and OUT_2. The resistor values are also shown in Fig. 2. The transistor parameters, using an Ebers and Moll model, are

$$I_s = 10^{-11} \text{ mA},$$

$$\frac{kT}{q} = 25.8 \text{ mV, and}$$

$$\beta = 100.$$

The input voltage source E_{in} switches the voltage potential from −1.9 to −0.8 V.

The sensitivities of the output voltage at node OUT_1 are to be computed as a function of each of the 15 resistors at both of the two input voltage levels if the power dissipated in the circuit is less than 0.345 W. However, if the power dissipation is more than 0.345 W, then the *sum* of the sensitivities of the voltage at node OUT_1 and the power dissipation as a function of the resistors should be computed instead.

For this problem, the modified nodal matrix can be written similarly to Table 3 and is omitted here. Two performance functions, the output voltage and the power dissipation, are defined by

$$P_1 = VR_3.$$

$$P_2 = (4.1 \times I_{E_{EC}} + E_{in} \times I_{E_{in}}),$$

where 4.1 is the value of the power supply E_{EC}.

The matrices $\partial \mathbf{f}_1/\partial \mathbf{x}_3$ and $\partial \mathbf{f}_2/\partial \mathbf{x}_1$ are also developed similarly to Example 1. They have dimensions of 14×15, 15×15, and 2×15. In the computation, a value of E_{in} equal to $4.1 - 1.9 = 2.2$ V is first used; the solution of the circuit equation yields a power dissipation of 0.34059 W, which is below the critical 0.345 W. Hence a $\mathbf{Z}$ vector $\begin{pmatrix}1\\0\end{pmatrix}$ is chosen that leads to the sensitivities of P_1 with respect to the resistors by evaluating Eq. (17).

In the next pass, the value of E_{in} is changed to $4.1 - 0.8 = 3.3$ V. Power dissipation this time exceeds 0.345 W and reaches a value of 0.35152 W. A $\mathbf{Z}$ vector of $\begin{pmatrix}1\\1\end{pmatrix}$ is chosen, which automatically gives the sensitivities $\mathbf{S}_2$ of P_1 and P_2 with respect to the resistors by evaluating Eq. (17). The sensitivities $\mathbf{S}_1$ and $\mathbf{S}_2$ are listed in Table 7 together with the corresponding resistor names. The vector $\mathbf{Y}$ is set to zero for both passes.

Summary

In this paper, we have presented the modified nodal approach to dc sensitivity computation. In formulating the equations for the network sensitivities, two matrices—the modified nodal matrix and the sensitivity matrix—are needed. Simple tables are provided that can be stored in the computer for use in a general purpose program to construct both matrices in a straightforward manner. Furthermore, the formulation permits multiple performance functions for a given network such that both the performance functions and the constraints required by the design can be considered. Numerical examples have been given to illustrate the methods and techniques presented in the paper. Finally, the same basic approach presented in this paper can be extended to frequency domain computation.

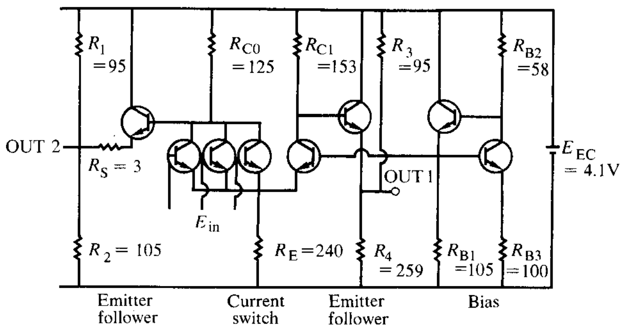

Figure 2 Current-switch emitter follower circuit.

Table 7 Sensitivities are shown with corresponding resistances.

Resistances	*Sensitivities* S1	S2
R_1	−0.017	−0.32
R_S	−0.778	−0.0014
R_2	−1.16	−0.51
R_E	−0.0068	−0.131
R_{C0}	−0.0079	−0.0098
R_{C1}	−0.21	−0.0085
R_3	−1.129	−0.008
R_4	−0.698	−1.21
R_{B3}	−0.16	−0.16
R_{B1}	−0.45	−0.44
R_{B4}	−0.0016	−0.0015
R_{B2}	−0.24	−0.25

Appendix: Proof of theorem

Proof: We first note that the Jacobian matrix $\mathbf{J}$ of $\mathbf{f}(\mathbf{x}) = 0$ is

$$\mathbf{J} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \begin{pmatrix} \dfrac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1} & \dfrac{\partial \mathbf{f}_1}{\partial \mathbf{x}_2} & \dfrac{\partial \mathbf{f}_1}{\partial \mathbf{x}_3} \\ \dfrac{\partial \mathbf{f}_2}{\partial \mathbf{x}_1} & \dfrac{\partial \mathbf{f}_2}{\partial \mathbf{x}_2} & \dfrac{\partial \mathbf{f}_2}{\partial \mathbf{x}_3} \\ 0 & 0 & \mathbf{I} \end{pmatrix} \tag{A1}$$

If we make the following substitutions in matrix $\mathbf{J}$:

$$\mathbf{A} = \begin{pmatrix} \dfrac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1} & \dfrac{\partial \mathbf{f}_1}{\partial \mathbf{x}_2} \\ \dfrac{\partial \mathbf{f}_2}{\partial \mathbf{x}_1} & \dfrac{\partial \mathbf{f}_2}{\partial \mathbf{x}_2} \end{pmatrix} \text{ and } \mathbf{B} = \begin{pmatrix} \dfrac{\partial \mathbf{f}_1}{\partial \mathbf{x}_3} \\ \dfrac{\partial \mathbf{f}_2}{\partial \mathbf{x}_3} \end{pmatrix}, \tag{A2}$$

we obtain

$$\mathbf{J} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ 0 & \mathbf{I} \end{pmatrix}.$$

It can easily be shown that the inverse of $\mathbf{J}$ is

$$\mathbf{J}^{-1} = \begin{pmatrix} \mathbf{A}_1 & -\mathbf{A}_1\mathbf{B} \\ 0 & \mathbf{I} \end{pmatrix},$$

where the inverse of $\mathbf{A}$, denoted by $\mathbf{A}_1$, is assumed to exist. Hence Eq. (16) becomes

$$\begin{pmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \\ \hat{\mathbf{x}}_3 \end{pmatrix} = \begin{pmatrix} \mathbf{A}_1{}^{\mathrm{T}} & 0 \\ -\mathbf{B}^{\mathrm{T}}\mathbf{A}_1{}^{\mathrm{T}} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{Y} \\ \mathbf{Z} \\ 0 \end{pmatrix}. \tag{A3}$$

Because the matrix $\mathbf{A}_1$ has dimensions $(n_1 + n_2) \times (n_1 + n_2)$, where n_1 and n_2 are the dimensions of $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively, we can partition $\mathbf{A}_1{}^{\mathrm{T}}$ into

$$\mathbf{A}_1^{\mathrm{T}} = (\mathbf{A}_{11}^{\mathrm{T}}, \mathbf{A}_{12}^{\mathrm{T}}), \tag{A4}$$

where the number of the columns of the two submatrices are, respectively, n_1 and n_2. From Eqs. (A3) and (A4), we obtain

$$\hat{\mathbf{x}}_3 = -\mathbf{B}^{\mathrm{T}} \mathbf{A}_{11}^{\mathrm{T}} \mathbf{Y} - \mathbf{B}^{\mathrm{T}}\mathbf{A}_{12}^{\mathrm{T}} \mathbf{Z}. \tag{A5}$$

Now we return to Eq. (15) and assume that the constant vector $\mathbf{c}$ is perturbed slightly to $\mathbf{c} + \delta\mathbf{C}$. Then, in order for the vector equation (15) to hold, $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ are also perturbed accordingly to $\mathbf{x}_1 + \delta\mathbf{x}_1$, $\mathbf{x}_2 + \delta\mathbf{x}_2$, and $\mathbf{x}_3 + \delta\mathbf{x}_3$. Because from Eq. (15) $\delta\mathbf{c}$ and $\delta\mathbf{x}_3$ are equal, we substitute $\delta\mathbf{x}_3$ for $\delta\mathbf{c}$, yielding

$$\mathbf{f}(\mathbf{x} + \delta\mathbf{x}) = \begin{pmatrix} \mathbf{f}_1\ (\mathbf{x}_1 + \delta\mathbf{x}_1, \mathbf{x}_2 + \delta\mathbf{x}_2, \mathbf{x}_3 + \delta\mathbf{x}_3) \\ \mathbf{f}_2\ (\mathbf{x}_1 + \delta\mathbf{x}_1, \mathbf{x}_2 + \delta\mathbf{x}_2, \mathbf{x}_3 + \delta\mathbf{x}_3) \\ \mathbf{x}_3 + \delta\mathbf{x}_3 - \mathbf{C} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \delta\mathbf{x}_3 \end{pmatrix}.$$

Expanding the left hand side into Taylor series and using Eq. (15) results in

$$\frac{\partial\mathbf{f}}{\partial\mathbf{x}_1}\delta\mathbf{x}_1 + \frac{\partial\mathbf{f}}{\partial\mathbf{x}_2}\delta\mathbf{x}_2 + \frac{\partial\mathbf{f}}{\partial\mathbf{x}_3}\delta\mathbf{x}_3 = \begin{pmatrix} 0 \\ 0 \\ \delta\mathbf{x}_3 \end{pmatrix},$$

which can be rearranged into

$$\begin{pmatrix} \dfrac{\partial\mathbf{f}_1}{\partial\mathbf{x}_1} & \dfrac{\partial\mathbf{f}_1}{\partial\mathbf{x}_2} \\ \dfrac{\partial\mathbf{f}_2}{\partial\mathbf{x}_1} & \dfrac{\partial\mathbf{f}_2}{\partial\mathbf{x}_2} \end{pmatrix} \begin{pmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \dfrac{\partial\mathbf{f}_1}{\partial\mathbf{x}_3} \\ \dfrac{\partial\mathbf{f}_2}{\partial\mathbf{x}_3} \end{pmatrix} \delta\mathbf{x}_3.$$

Using Eqs. (A2) and (A3) we have

$$\begin{pmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{A}_{11} \\ \mathbf{A}_{12} \end{pmatrix} \mathbf{B}\ \delta\mathbf{x}_3.$$

It follows that

$$\delta\mathbf{x}_1 = \mathbf{A}_{11}\ \mathbf{B}\ \delta\mathbf{x}_3 \text{ and } \delta\mathbf{x}_2 = \mathbf{A}_{12}\ \mathbf{B}\ \delta\mathbf{x}_3.$$

Hence

$$\left(\frac{\partial\mathbf{x}_1}{\partial\mathbf{x}_3}\right)^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\ \mathbf{A}_{11}^{\mathrm{T}} \text{ and } \left(\frac{\partial\mathbf{x}_2}{\partial\mathbf{x}_3}\right)^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\ \mathbf{A}_{12}^{\mathrm{T}}.$$

Substituting the above equation into Eq. (A5) results in Eq. (17), which is what we set out to prove.

References

1. S. W. Director and R. A. Rohrer, "The Generalized Adjoint Network and Network Sensitivities," *IEEE Trans. Circuit Theory* **CT-16,** 318 (1969).
2. C. W. Ho and G. D. Hachtel, "Automated Optimization of Switching Networks via SCEPTRE Simulation," presented at the 1969 IEEE International Symposium on Circuit Theory, San Francisco, California (December 8-10, 1969).
3. G. D. Hachtel et al, "The Sparse Tableau Approach to Network Analysis and Design," *IEEE Trans. Circuit Theory* **CT-18,** 101 (1971).
4. C. W. Ho, "Time-Domain Sensitivity Computation for Networks Containing Transmission Lines," *IEEE Trans. Circuit Theory* **CT-18,** 114 (1971).
5. W. T. Weeks, et al, "Algorithms for ASTAP—A Network Analysis Program," *IEEE Trans. Circuit Theory* **CT-20,** 628 (1973).
6. F. H. Branin et al, "ECAP II—A New Electronic Circuit Analysis Program," *IEEE J. Solid-State Circuits* **SC-6,** 146 (1971).
7. L. Nagel and R. Rohrer, "Computer Analysis of Nonlinear Circuits, Excluding Radiation (CANCER)" *IEEE J. Solid-State Circuits* **SC-6,** 166 (1971).
8. C. W. Ho, A. E. Ruehli, P. A. Brennan, and D. A. Zein, "A General Purpose Interactive Circuit Analysis and Design Program in APL," *Proceedings of the 1975 IEEE International Symposium on Circuits and Systems*, IEEE Catalog No. 75CH0937-3 CAS, April 1975, p. 216.
9. O. Wing and J. Behar, "Circuit and Design by Minimization Using the Hessian Matrix," *IEEE Trans. Circuit Theory* **CT-21,** 643 (1974).
10. B. A. Wooley, "The Computer-aided Design Optimization of Integrated Broad-band Amplifiers" *ISSCC Dig. Tech. Papers* **13,** 84 (1970).
11. S. W. Director, W. A. Bristol, and A. J. Brodersen, "Fabrication-based Optimization of Linear Integrated Circuits," *IEEE Trans. Circuit Theory* **CT-20,** 690 (1973).
12. D. A. Zein and S. L. Hakimi, "A Computational Approach for Analysis of Nonlinear Circuits" *Proc. 9th Ann. Allerton Conf. Circuits and Syst. Theory*, University of Illinois, Urbana, Ill., (1971), p. 455.
13. F. H. Branin, "Network Sensitivity and Noise Analysis Simplified," *IEEE Trans. Circuit Theory* **CT-20,** 285 (1973).
14. C. W. Ho, A. Ruehli, and P. A. Brennan, "The Modified Nodal Approach to Network Analysis," *IEEE Trans. Circuits and Systems* **CAS-22,** 504 (1975).
15. P. Slepian and L. Weinberg, "Synthesis of Paramount and Dominant Matrices," *Proc. Nat. Electron. Conf.*, Chicago, Ill., (1958).
16. S. W. Director and R. A. Rohrer, "On the Design of Resistance n-Port Networks by Digital Computer," *IEEE Trans. Circuit Theory* **CT-16,** 337 (1969).
17. G. D. Hachtel, M. R. Lightner, and H. J. Kelly, "Application of the Optimization Program AOP to the Design of Memory Circuits," *IEEE Trans. Circuits and Systems* **CAS-22,** 496 (1975).

Received April 18, 1975

The author is located at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598.