

# Calculation of Symbolic Sensitivities for Large-Scale Circuits in the Sequence of Expressions Form via the Transimpedance Method

Franciszek Balik and Benedykt Rodanski

**Abstract** - In this paper we describe a new method for obtaining, in symbolic form, the network functions and their small- and large-change sensitivities. Our method is based on the two-port transimpedance concept and the sequence of expressions approach to calculation of matrix determinant. We show that the network functions as well as their sensitivities can be expressed by appropriate transimpedances. Each transimpedance is given by an algebraic sum of at most four elements of the inverse of the circuit's reduced node admittance matrix. In our method the complexity of the sequence of expressions grows linearly with the circuit size. The extra effort required to obtain sensitivities is minimal and, more importantly, independent of the circuit size.

## I. INTRODUCTION

Sensitivity analysis, an important step in most circuit design, calculates the sensitivity of the circuit performance to variations in the values of the circuit components (parameters). Due to manufacturing process fluctuations, modelling and measurement errors, biasing and environmental (e.g. temperature) changes, these variations are unavoidable in practical realizations. Sometimes, the parameter changes can be large, as in the case of a circuit component exchange or failure, or changes caused by large temperature variations. As a result, the circuit performance can deviate from intended and simulated behaviour. In this paper we will present an unified approach to calculation of both small- and large-scale sensitivities in the frequency domain.

The small-change (differential) sensitivity is a measure of variation in the circuit performance due to an infinitesimally small variation in a circuit parameter (e.g. component value). The circuit performance is simply some network function  $H(s, \mathbf{p})$ , where  $s$  is the complex frequency and  $\mathbf{p} = [p_1, p_2, \dots, p_m]^T$  is a vector of circuit parameters. The most widely used definition of the differential sensitivity of  $H(s, \mathbf{p})$  to changes in a parameter  $p_k$  is [1]:

$$S_{p_k}^H = \frac{\partial \ln H}{\partial \ln p_k} = \frac{\partial H}{\partial p_k} \frac{p_k}{H}. \quad (1)$$

By analogy, the large-change relative sensitivity can be defined as:

$$\delta_{p_k}^H = \frac{\Delta H}{H} \frac{p_k}{\Delta p_k}. \quad (2)$$

If the network function,  $H(s, \mathbf{p})$ , is given symbolically, the required small-change sensitivity to any parameter variation can be obtained in symbolic form by differentiating the expression with respect to the circuit parameter  $p_k$ . This direct approach works well only for the relatively small circuits, for which the network function does not have unreasonably large number of terms.

The well known drawback of symbolic methods, when applied to large-scale circuits, is the exponential growth of the number of terms in the network function with circuit size [2]. Important developments in the symbolic analysis of large-scale circuits have been achieved with the introduction of *hierarchical decomposition* [3] and the concept of the *sequence of expressions* (SOE) [4]. The basic idea is to produce a succession of simple expressions with a backward hierarchical dependency on each other. It was shown [4] that the growth of the number of expressions (and thus the number of arithmetic operations - *flops* - required for numerical evaluation) is linear for practical circuits. Until recently it was believed that in order to minimise the SOE complexity, the circuit must be first partitioned into smaller sub-circuits. In [5, 6] the arguments were presented to the contrary. Partitioning imposes restrictions on the order in which internal variables are suppressed, leading to sub-optimal solutions. When the SOE concept was combined with the technique of (locally) optimal reduction of the modified node admittance matrix (MNAM) to a two-port matrix, it produced sequential formulae of considerably lesser complexity [7].

Once the circuit equations are derived, the relevant derivatives of each equation can be computed symbolically, and then the differential sensitivity can be calculated using the chain rule of differentiation. This solution was first proposed in [8]. It was extended in [9, 10] by representing a SOE as a directed acyclic graph (DAG) and providing an algorithm for analysing the DAG and reducing the number of additional expressions generated. Although the method is conceptually simple, it does not always generate the optimal sequence [10]. This is due to the fact that the number of additional expressions (derivatives), required for sensitivity calculations, heavily depends on the position of the symbol with respect to which the derivatives are calculated. If the symbol appears near the top of the SOE, the number of additional expressions may be much larger than if the symbol is near the bottom (in the worst case, doubling the computing effort). It is possible to re-order the equations as to move the symbol close to the bottom of the SOE. The new ordering may, however, be no longer optimal from the point of view of the initial SOE complexity. Additional computational effort is also required to create and analyse the DAG.

The large-change sensitivity can be calculated by evaluating the network function twice: at the nominal and perturbed value of parameter  $p_k$ . This brute force approach is very inefficient, as it doubles the calculation time.

The purpose of this paper is to present a new method of obtaining small- and large-change symbolic sensitivities in the SOE form. Rather than generating a SOE for a network function and differentiate it symbolically to obtain the required differential sensitivity (or evaluate it twice, to get the large-change sensitivity), we calculate both the network function and its sensitivities simultaneously. This is achieved by the use of the *two-port transimpedance* concept [11] and an efficient method of symbolic calculations of the elements of the inverse of the reduced node admittance matrix (RNAM). We will show that the network functions as well as their derivatives and large-change increments can be expressed by various transimpedances which in turn are represented by linear combinations of elements of the inverse of the RNAM. Symbolic calculation of the inverse of a large matrix is a formidable task and the result is usually a complicated and lengthy expression. It is possible, however, by suppressing internal variables, to reduce the problem to no more than six variables (four in most practical cases), thus reducing the node admittance matrix (NAM) to a matrix of at most sixth order. Moreover, because network functions are always expressed as ratios of various cofactors of a circuit matrix, it is possible to remove factors common to both the numerator and the denominator, further reducing the complexity of the final expression. Apart of being capable of producing less complex results, the new method has other advantages over direct differentiation of the SOE. The most important

of them are:

1. SOE complexity is only weakly dependent on the element's position in the circuit.
2. The extra effort required to obtain sensitivities is minimal and, more importantly, independent of the circuit size; it means that the larger the circuit the (relatively) less expensive it becomes to calculate sensitivities.
3. The use of transimpedance concept allows to simultaneously compute many network functions and their sensitivities at minimal additional cost.
4. Using transimpedances it is possible to obtain both small- and large-change sensitivities.
5. Higher-order differential sensitivities [12] and multiparameter large-change sensitivities [13] can also be obtained using our approach.

The paper is organised as follows. In section II we formulate the problem. Section III introduces the transimpedance concept. Using this concept we derive the network function and sensitivity formulae in section IV. Section V details the algorithm and software developed to implement our technique. Two circuit examples, illustrating the effectiveness of the new method, are presented in section VI.

## II. PROBLEM FORMULATION

Consider a lumped, linear, time-invariant circuit depicted in Fig. 1.

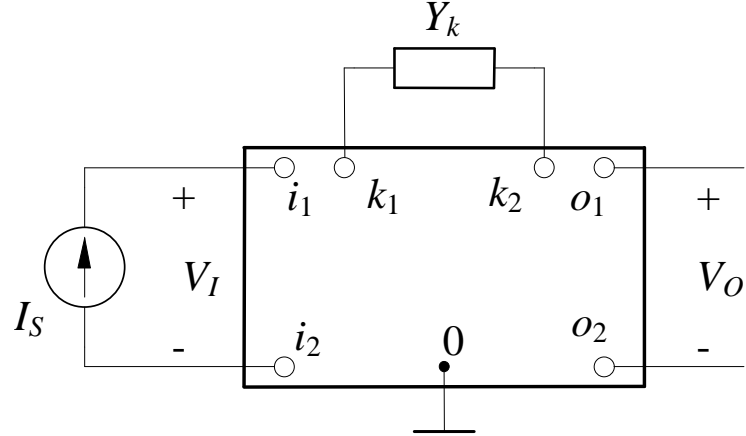


Fig. 1. A general two-port for sensitivity calculations.

The circuit nodes are numbered with natural numbers, starting from 0 (the reference node). The three pairs of terminals (ports) of interest are numbered as follows: the input port  $\mathbf{i} = (i_1, i_2)$  with an independent current source  $I_S$ , the output port  $\mathbf{o} = (o_1, o_2)$  and the port  $\mathbf{k} = (k_1, k_2)$ , terminated with admittance  $Y_k$ , whose effect on a network function is to be calculated. We assume that the circuit is described by a node equation:

$$\mathbf{Y}\mathbf{V} = \mathbf{I}, \quad (3)$$

where:  $\mathbf{Y}$  is a  $n \times n$  node admittance matrix (NAM) of the circuit,

$\mathbf{V}$  is an  $n$ -vector of node potentials, possibly augmented by additional variables, e.g., inductor currents, currents in (dependent) voltage sources, etc., and

$\mathbf{I}$  is an  $n$ -vector of terminal currents.

Note that  $n$  may be less than, equal to or greater than the number of circuit nodes, depending on the circuit and the variation of the nodal method used. The most commonly used method of circuit equation formulation is the *modified nodal method*. It introduces additional (current) variables to handle circuit components like op amps, controlled sources other than VCCS, etc., which are not allowed in the classical nodal analysis. The matrix created with this method is usually referred to as the *modified node admittance matrix* (MNAM). In the modified nodal formulation inductors are usually entered into the MNAM as impedances, thus ensuring that all entries are of the form  $\pm A \pm sB$ . The various ways of formulating the MNAM are described in [1, 14]. When ideal op amps (nullors) and other ideal elements are present in the circuit, the MNAM can be compacted, e.g., for each ideal op amp one variable and one equation is removed from the set of node equations. The matrix thus obtained is referred to as the compacted MNAM or CMNAM. The compacting procedure is detailed in [14]. Because the methods and algorithms described in this paper work equally well with any type of the nodal matrix, to avoid using lengthy acronyms we will simply refer in the remainder to the node admittance matrix (NAM).

Our aim is to symbolically calculate the sensitivities (1) and (2) of a network function  $H(s, \mathbf{p})$  to changes of circuit parameter  $p_k$ . Usually the circuit parameter in question is the value of a circuit component: resistance  $R_k$ , capacitance  $C_k$ , inductance  $L_k$ , conductance  $G_k$ , transconductance  $g_{mk}$ , etc. To maintain generality we will consider network function sensitivity to variation of admittance parameter  $Y_k$ . Depending on the particular type of circuit parameter, the required sensitivity can be easily obtained using the chain rule of differentiation:

$$\frac{\partial H(s, \mathbf{p})}{\partial p_k} = \frac{\partial H(s, \mathbf{p})}{\partial Y_k} \frac{\partial Y_k}{\partial p_k}, \quad (4)$$

where:

$$\frac{\partial Y_k}{\partial p_k} = \begin{cases} 1 ; & p_k = G_k, g_{mk} \\ \frac{-1}{R_k^2} ; & p_k = R_k \\ s ; & p_k = C_k \\ \frac{-1}{sL_k^2} ; & p_k = L_k \end{cases} \quad (5)$$

We will also limit our derivations to two most commonly used network functions: the voltage transmittance  $T_v(s, \mathbf{p}) = V_o(s, \mathbf{p})/V_i(s, \mathbf{p})$  and the input impedance  $Z_{in}(s, \mathbf{p}) = V_i(s, \mathbf{p})/I_s(s, \mathbf{p})$ . Other network functions and their sensitivities can be obtained using similar procedures.

### III. THE TWO-PORT TRANSIMPEDANCE AND ITS PROPERTIES

#### 3.1 Definitions

In order to simplify the notation we will introduce a new function, called the *two-port transimpedance* (or simply *transimpedance*).

##### Definition 1

Let  $\mathbf{Z}$  denote the inverse of the (nonsingular) NAM:  $\mathbf{Z} = \mathbf{Y}^{-1}$ ; let also  $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$ ,  $\boldsymbol{\beta} = (\beta_1, \beta_2)$  be pairs of natural numbers, representing circuit nodes. We define the *transimpedance of the first order* as:

$$Z(\boldsymbol{\alpha}, \boldsymbol{\beta}) = z_{\alpha_1\beta_1} - z_{\alpha_1\beta_2} - z_{\alpha_2\beta_1} + z_{\alpha_2\beta_2}, \quad (6)$$

where  $z_{ij}$  represents the element from the  $i$ -th row and the  $j$ -th column of  $\mathbf{Z}$ .

##### Definition 2

Let  $S = \{s_1, s_2, \dots, s_{n+1}\}$  be an ordered set of pairs of natural numbers. We define the *transimpedance of the  $n$ -th order* as:

$$Z(s_1, s_2, \dots, s_{n+1}) = \prod_{i=1}^n Z(s_i, s_{i+1}), \quad (7)$$

where  $s_i$  is the  $i$ -th pair of natural numbers (representing nodes incident to the  $i$ -th two-terminal circuit element).

#### 3.2 Some properties of the transimpedance

Definition 2 implies the following property of the transimpedance of the  $n$ -th order.

##### Property 1

Any transimpedance of the  $n$ -th order, defined on an ordered set  $S = \{s_1, s_2, \dots, s_{n+1}\}$ , can be represented as a product of two transimpedances: of order  $k$  and  $n-k$ , respectively, defined on the ordered subsets  $S_\alpha = \{s_1, s_2, \dots, s_k, s_{k+1}\}$  and  $S_\beta = \{s_{k+1}, s_{k+2}, \dots, s_{n+1}\}$  of  $S$ . We can write:

$$Z(s_1, s_2, \dots, s_{n+1}) = Z(s_1, s_2, \dots, s_{k+1})Z(s_{k+1}, s_{k+2}, \dots, s_{n+1}). \quad (8)$$

Element  $s_{k+1}$  is common to both subsets:  $s_{k+1} = S_\alpha \cap S_\beta$ .

The transimpedance (6) has the following differential property.

##### Property 2

The derivative of a first-order transimpedance, defined on a set  $S = \{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$ , with respect to an

admittance  $Y_\xi$  [connected to a pair of nodes:  $\xi = (\xi_1, \xi_2)$ ] is equal to the second-order transimpedance defined on a set  $\Sigma = \{\alpha, \xi, \beta\}$  taken with the negative sign:

$$\frac{\partial Z(\alpha, \beta)}{\partial Y_\xi} = -Z(\alpha, \xi, \beta) = -Z(\alpha, \xi)Z(\xi, \beta). \quad (9)$$

Proof of this important property is given in Appendix A. Similar formulae can be obtained for derivatives of  $Z(\alpha, \beta)$  with respect to parameters of four types of controlled sources (transconductance  $g_m$  of a VCCS, voltage gain  $\mu$  of a VCVS, etc.).

### 3.3 Large-change increment of the transimpedance

Suppose that an admittance  $Y_\xi$  has changed, and its new value is  $Y_\xi + \Delta Y_\xi$ . This change causes the admittance matrix to change accordingly, and in consequence, also its inverse:

$$(\mathbf{Y} + \Delta \mathbf{Y})^{-1} = \mathbf{Z} + \Delta \mathbf{Z}. \quad (10)$$

Using Householder formula, it has been shown [15] that

$$(\mathbf{Y} + \Delta \mathbf{Y})^{-1} = \mathbf{Z} - \Delta Y_\xi \frac{\mathbf{Z}(\mathbf{e}_\xi \mathbf{e}_\xi^T) \mathbf{Z}}{1 + \Delta Y_\xi Z(\xi, \xi)}. \quad (11)$$

Hence

$$\Delta \mathbf{Z} = -\Delta Y_\xi \frac{\mathbf{Z}(\mathbf{e}_\xi \mathbf{e}_\xi^T) \mathbf{Z}}{1 + \Delta Y_\xi Z(\xi, \xi)}, \quad (12)$$

where  $\mathbf{e}_\xi$  is a column vector ( $1 \times n$ ) containing +1 at position  $\xi_1$ , -1 at position  $\xi_2$  and zeroes everywhere else.

The increment of the transimpedance can be now defined as:

$$\Delta Z(\alpha, \beta) = \Delta z_{\alpha_1 \beta_1} - \Delta z_{\alpha_1 \beta_2} - \Delta z_{\alpha_2 \beta_1} + \Delta z_{\alpha_2 \beta_2}, \quad (13)$$

Each element  $\Delta z_{rs}$  of the matrix  $\Delta \mathbf{Z}$  can be determined from the following relationship:

$$\Delta z_{rs} = K(z_{r\xi_1} - z_{r\xi_2})(z_{\xi_1 s} - z_{\xi_2 s}), \quad (14)$$

where:

$$K = -\frac{\Delta Y_\xi}{1 + \Delta Y_\xi Z(\xi, \xi)} = -\Delta Y_\xi K_0. \quad (15)$$

Substituting (14), with appropriate coefficients  $(r,s) = (\alpha, \beta)$ , into (13) and taking into account the relationship (9), one obtains the increment of the transimpedance:

$$\Delta Z(\alpha, \beta) = K Z(\alpha, \xi) Z(\xi, \beta) = -K \frac{\partial Z(\alpha, \beta)}{\partial Y_\xi}. \quad (16)$$

As can be seen, the increment of a two-port transimpedance due to large change of an admittance is proportional to the derivative of this transimpedance with respect to the admittance changed.

## IV. APPLICATION OF THE TRANSIMPEDANCE TO CALCULATION OF NETWORK FUNCTIONS AND THEIR SENSITIVITIES

### 4.1 Calculating network functions

Using transimpedances it is possible to express the required network functions. Potential at any node  $j$  of a circuit in Fig. 1 can be calculated as  $V_j = (z_{j,i1} - z_{j,i2})I_S$  and the voltage transmittance:

$$T_V = \frac{V_O}{V_I} = \frac{V_{o_1} - V_{o_2}}{V_{i_1} - V_{i_2}} = \frac{z_{o_1 i_1} - z_{o_1 i_2} - z_{o_2 i_1} + z_{o_2 i_2}}{z_{i_1 i_1} - z_{i_1 i_2} - z_{i_2 i_1} + z_{i_2 i_2}} = \frac{Z(o, i)}{Z(i, i)}. \quad (17)$$

Analogically, the input impedance can be expressed as:

$$Z_{in} = \frac{V_I}{I_S} = \frac{V_{i_1} - V_{i_2}}{I_S} = Z(i, i). \quad (18)$$

Other network functions can be obtained in a similar way.

### 4.2 Obtaining network function sensitivities

#### A. Small-change sensitivities

Differentiating (17) and (18) with respect to admittance  $Y_k$  and using Property 2 we can write:

$$\frac{\partial T_V}{\partial Y_k} = \frac{Z(i, k, i) Z(o, i) - Z(o, k, i) Z(i, i)}{Z^2(i, i)} \quad (19)$$

and

$$\frac{\partial Z_{in}}{\partial Y_k} = -Z(i, k, i). \quad (20)$$

Now, utilising Property 1 (8) and Definition 2 (7), we can re-write (19) and (20) in terms of the

elements of  $\mathbf{Z}$ :

$$\frac{\partial T_V}{\partial Y_k} = \frac{[Z(\mathbf{i}, \mathbf{k})Z(\mathbf{o}, \mathbf{i}) - Z(\mathbf{o}, \mathbf{k})Z(\mathbf{i}, \mathbf{i})]Z(\mathbf{k}, \mathbf{i})}{Z^2(\mathbf{i}, \mathbf{i})} = [T_V Z(\mathbf{i}, \mathbf{k}) - Z(\mathbf{o}, \mathbf{k})] \frac{Z(\mathbf{k}, \mathbf{i})}{Z(\mathbf{i}, \mathbf{i})}, \quad (21)$$

$$\frac{\partial Z_{in}}{\partial Y_k} = -Z(\mathbf{i}, \mathbf{k})Z(\mathbf{k}, \mathbf{i}). \quad (22)$$

Substituting (21), (17) and (22), (18) into (1), we obtain the relative differential sensitivities to  $Y_k$ :

$$S_{Y_k}^{T_V} = \frac{\partial \ln T_V}{\partial \ln Y_k} = \frac{\partial T_V}{\partial Y_k} \frac{Y_k}{T_V} = Y_k [T_V Z(\mathbf{i}, \mathbf{k}) - Z(\mathbf{o}, \mathbf{k})] \frac{Z(\mathbf{k}, \mathbf{i})}{Z(\mathbf{o}, \mathbf{i})}, \quad (23)$$

$$S_{Y_k}^{Z_{in}} = \frac{\partial \ln Z_{in}}{\partial \ln Y_k} = \frac{\partial Z_{in}}{\partial Y_k} \frac{Y_k}{Z_{in}} = -Y_k \frac{Z(\mathbf{i}, \mathbf{k})Z(\mathbf{k}, \mathbf{i})}{Z(\mathbf{i}, \mathbf{i})}. \quad (24)$$

Finally, utilising (4) and (5), one can obtain expressions for small-change sensitivities to circuit components:  $R_k$ ,  $C_k$ ,  $L_k$ , etc. For example, when  $p_k = L_k$  we have  $Y_k = 1/(sL_k)$  and:

$$S_{L_k}^{T_V} = \frac{\partial T_V}{\partial Y_k} \frac{\partial Y_k}{\partial L_k} \frac{L_k}{T_V} = -\frac{1}{sL_k} [T_V Z(\mathbf{i}, \mathbf{k}) - Z(\mathbf{o}, \mathbf{k})] \frac{Z(\mathbf{k}, \mathbf{i})}{Z(\mathbf{o}, \mathbf{i})} = -S_{Y_k}^{T_V}. \quad (25)$$

### B. Large-change sensitivities

If admittance  $Y_k$  undergoes a large change  $\Delta Y_k$ , the resulting change in the voltage transmittance is:

$$\Delta T_V = \frac{Z(\mathbf{o}, \mathbf{i}) + \Delta Z(\mathbf{o}, \mathbf{i})}{Z(\mathbf{i}, \mathbf{i}) + \Delta Z(\mathbf{i}, \mathbf{i})} - \frac{Z(\mathbf{o}, \mathbf{i})}{Z(\mathbf{i}, \mathbf{i})} = \frac{\Delta Z(\mathbf{o}, \mathbf{i}) - T_V \Delta Z(\mathbf{i}, \mathbf{i})}{Z(\mathbf{i}, \mathbf{i}) + \Delta Z(\mathbf{i}, \mathbf{i})}. \quad (26)$$

Taking (16) into account, one obtains:

$$\Delta T_V = K \frac{[Z(\mathbf{o}, \mathbf{k}) - T_V Z(\mathbf{i}, \mathbf{k})]Z(\mathbf{k}, \mathbf{i})}{Z(\mathbf{i}, \mathbf{i}) + K Z(\mathbf{i}, \mathbf{k})Z(\mathbf{k}, \mathbf{i})}. \quad (27)$$

The large-change sensitivity measure can now be expressed in a straightforward manner:

$$\delta_{Y_k}^{T_V} = \frac{K}{T_V} \frac{Y_k}{\Delta Y_k} \frac{[Z(\mathbf{o}, \mathbf{k}) - T_V Z(\mathbf{i}, \mathbf{k})]Z(\mathbf{k}, \mathbf{i})}{Z(\mathbf{i}, \mathbf{i}) + K Z(\mathbf{i}, \mathbf{k})Z(\mathbf{k}, \mathbf{i})}. \quad (28)$$

Utilising (21), we can write finally:



$$\delta_{Y_k}^{T_V} = -\frac{K}{T_V} \frac{Y_k}{\Delta Y_k} \frac{\partial T_V}{\partial Y_k} \frac{Z(i,i)}{Z(i,i) + KZ(i,k)Z(k,i)} = \frac{K_0 S_{Y_k}^{T_V}}{1 - \Delta Y_k K_0 \frac{Z(i,k)Z(k,i)}{Z(i,i)}}, \quad (29)$$

where:  $S_{Y_k}^{T_V} = \frac{\partial T_V}{\partial Y_k} \frac{Y_k}{T_V}$  is the relative differential sensitivity, given by (23), and

$$K_0 = \frac{1}{1 + \Delta Y_k Z(k,k)}.$$

As can be seen, only one additional transimpedance, namely  $Z(k,k)$ , is needed to obtain the large-change sensitivity when the voltage transmittance and its derivative have been already calculated.

In similar way the increment and large-change sensitivity for the input impedance can be expressed with appropriate transimpedances:

$$\begin{aligned} \Delta Z_{in} &= KZ(i,k)Z(k,i), \\ \frac{\Delta Z_{in}}{Z_{in}} &= K \frac{Z(i,k)Z(k,i)}{Z(i,i)}, \end{aligned} \quad (30)$$

$$\delta_{Y_k}^{Z_{in}} = K \frac{Y_k}{\Delta Y_k} \frac{Z(i,k)Z(k,i)}{Z(i,i)} = K_0 S_{Y_k}^{Z_{in}}; \quad (31)$$

In practice it may be more convenient to work with component variations:  $\Delta R_k$ ,  $\Delta L_k$ ,  $\Delta C_k$ , etc., rather than admittance variations  $\Delta Y_k$ . To facilitate this requirement, one of the simple conversion formulae can be used:

$$Y_k = \frac{1}{R_k} \Rightarrow \Delta Y_k = \frac{-\Delta R_k}{R_k(R_k + \Delta R_k)}, \quad (32)$$

$$Y_k = \frac{1}{sL_k} \Rightarrow \Delta Y_k = \frac{-\Delta L_k}{sL_k(L_k + \Delta L_k)}, \quad (33)$$

$$Y_k = sC_k \Rightarrow \Delta Y_k = s\Delta C_k. \quad (34)$$

### 4.3 Expressing transimpedances in terms of elements of NAM

In order to calculate the elements of the inverse of  $\mathbf{Y}$  we will use the well-known formula:

$$\mathbf{Z} = \mathbf{Y}^{-1} = \frac{\text{adj}(\mathbf{Y})}{|\mathbf{Y}|}. \quad (35)$$

Thus the elements of  $\mathbf{Z}$  are obtained from:

$$z_{ij} = \frac{(-1)^{i+j}\Delta_{ji}}{\Delta}, \quad (36)$$

where:  $\Delta_{ji}$  is the  $ji$ -th minor of  $\mathbf{Y}$  - the determinant of a matrix obtained from  $\mathbf{Y}$  by deleting row  $j$  and column  $i$ ,  
 $\Delta = |\mathbf{Y}| \neq 0$  is the determinant of  $\mathbf{Y}$ .

Utilising (36) a transimpedance (6) can be rewritten as

$$\mathbf{Z}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{(-1)^{\alpha_1+\beta_1}\Delta_{\beta_1\alpha_1} - (-1)^{\alpha_1+\beta_2}\Delta_{\beta_2\alpha_1} - (-1)^{\alpha_2+\beta_1}\Delta_{\beta_1\alpha_2} + (-1)^{\alpha_2+\beta_2}\Delta_{\beta_2\alpha_2}}{\Delta}. \quad (37)$$

For large circuits, symbolic expressions for the cofactors in (37) could be unacceptably long, rendering the method impractical. It is, however, possible to suppress all internal variables of (3), leaving at most six variables associated with the external terminals. The reduced set of equations is called the *reduced node equation* (RNE) and the coefficient matrix is appropriately termed the *reduced node admittance matrix* (RNAM). Formulae derived above are also valid for the RNAM. The last statement is true because of the following RNAM property:

The determinant and each cofactor of RNAM are equal to the determinant and the corresponding cofactor of NAM divided by the product of pivots used in the reduction process.

The reduced set of nodal equations can be obtained in the SOE form by symbolic Gaussian elimination. Although this technique has been recently described in [7, 16], for completeness, a brief outline is presented in the following paragraph.

#### 4.4 Symbolic suppression of internal variables

In order to illustrate the internal variable suppression process in a concise manner, we need to further simplify the problem. Consider again the circuit shown in Fig. 1. Let us assume that the input and output ports, and the admittance  $Y_k$  share a common terminal which is grounded. Furthermore, since node numbering is arbitrary, let  $\mathbf{i} = (1,0)$ ,  $\mathbf{o} = (2,0)$  and  $\mathbf{k} = (3,0)$ . The circuit can be described by the set of symbolic equations:

$$\begin{bmatrix} y_{11} & y_{12} & y_{13} & \cdots & y_{1p} & \cdots & y_{1n} \\ y_{21} & y_{22} & y_{23} & \cdots & y_{2p} & \cdots & y_{2n} \\ y_{31} & y_{32} & y_{33} & \cdots & y_{3p} & \cdots & y_{3n} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ y_{q1} & y_{q2} & y_{q3} & \cdots & y_{qp} & \cdots & y_{qn} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ y_{n1} & y_{n2} & y_{n3} & \cdots & y_{np} & \cdots & y_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ \vdots \\ V_p \\ \vdots \\ V_n \end{bmatrix} = \begin{bmatrix} I_S \\ 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (38)$$

where  $y_{ij}$  are symbolic entries of general form:  $\pm G \pm sC$ , and some of the internal variables  $V_4$ ,

...,  $V_n$  may represent currents in certain circuit elements. It is important to note that the first entry,  $I_s$ , of the right-hand-side vector is its only nonzero. This form of the node equation will enable us to eliminate all internal variables without modifying the rhs vector, i.e., we will be able to work with the coefficient matrix only.

Since we are only interested in the three external variables ( $V_1$ ,  $V_2$  and  $V_3$ ), all other (internal) variables can be suppressed. Suppose that we wish to suppress the variable  $V_p$ . To achieve this we can use any equation from the set (38), except the first three equations, that has a nonzero coefficient at  $V_p$ . Let us choose it to be equation  $q > 3$  (the method of selecting the right variable and the right equation at each elimination step is called *pivoting*; it plays crucial role in achieving efficiency of the technique [7, 16]). The  $q$ th equation can be written in the expanded form:

$$y_{q1}V_1 + y_{q2}V_2 + y_{q3}V_3 + \dots + y_{qp}V_p + \dots + y_{qn}V_n = 0. \quad (39)$$

Provided that  $|y_{qp}| \neq 0$ , we can calculate  $V_p$  from (39) as

$$V_p = -\frac{y_{q1}}{y_{qp}}V_1 - \frac{y_{q2}}{y_{qp}}V_2 - \frac{y_{q3}}{y_{qp}}V_3 - \dots - \frac{y_{qn}}{y_{qp}}V_n. \quad (40)$$

Substituting (40) into (38) will eliminate the variable  $V_p$  and equation  $q$  from the set. During the elimination, each element  $y_{ij}$  of  $\mathbf{Y}$  undergoes the transformation:

$$y_{ij} := y_{ij} - \frac{y_{qj}y_{ip}}{y_{qp}}, \quad (41)$$

where:  $i, j = 1, 2, \dots, n$ ;  $i \neq q, j \neq p$ . This process of suppression of a variable is the very well known Gaussian elimination. The only difference from the usual appearance of the elimination formula (41) in the literature is the fact that the pivot,  $y_{qp}$ , may be off-diagonal ( $p \neq q$ ). In practice, the transformation (41) is only applied to the matrix elements  $y_{ij}$  for which  $|y_{qj}|/|y_{ip}| \neq 0$ . Every application of the Gaussian elimination formula (41) produces a symbolic expression. This procedure is carried out until all internal variables are suppressed and we obtain a set of three equations (the reduced node equation):

$$\begin{bmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} I_s \\ 0 \\ 0 \end{bmatrix}, \quad (42)$$

where each element  $y_{ij}$  is calculated by a sequence of expressions.

## V. IMPLEMENTATION

### A. The Algorithm

The process of obtaining sequential symbolic formulae for network functions and their sensitivities can be easily automated. The algorithm is presented below.

1. Obtain the small-signal equivalent circuit.
2. Get the information about the network function required and the element(s) with respect to which the sensitivity is to be calculated.
3. Formulate the symbolic NAM using the element stamp method.
4. If there are ideal op amps then compact the NAM (NAM  $\rightarrow$  CNAM).
5. Perform symbolic reduction of the CNAM to a matrix containing only rows and columns associated with the external nodes (CNAM  $\rightarrow$  RNAM).
6. Calculate the relevant cofactors and the determinant of the RNAM.
7. Apply the appropriate formulae to obtain the required network functions [e.g., (17) and (18) for the voltage transmittance and input impedance].
8. Calculate the differential sensitivity of the transfer function(s) by applying (23), (24).
9. Calculate the large-change sensitivity of the transfer function(s) using (29), (31).

### B. Software

Due to its excellent user interface, ability to display and handle large symbolic arrays and ease of programming (in VBA and/or C/C++), we have chosen Microsoft Excel as a software platform to test our algorithm. Circuit data can be either generated automatically from the output file of Cadence PSpice or entered manually onto an input spreadsheet in the familiar Spice-like format. The user then selects the pivoting criteria, the way the output is to be presented, the required network function and the circuit component with respect to which the sensitivity is to be calculated. The CNAM and the calculated expressions are displayed on another spreadsheet. Three text files may also be created. They are: component values (if any), CNAM elements and the sequence of expressions. These files can be used for further processing (e.g., by MATLAB, Maple, Mathematica, etc.). Our implementation, which is essentially an Excel workbook, is called STAINS 3 - Symbolic Two-port Analysis via Internal Node Suppression. STAINS can handle any linear circuit that can be described by the MNAM. Theoretically, the size of the circuit is only limited by the available computer memory. Our programme, however, has a size limit of 256 nodes due to maximum number of columns on the Excel spreadsheet. A copy of the software can be obtained via Internet from the authors (benr@eng.uts.edu.au).

## VI. CIRCUIT EXAMPLES

To illustrate the effectiveness of our approach to sensitivity calculations we present two circuit examples.

*Example 1.* Consider a circuit in Fig. 2. The circuit is deliberately small to allow for inclusion of the entire SOE. We wish to obtain a sequential expression for the voltage ratio,  $T_V = V_o/V_i$  ( $T_V$ ) and its sensitivity to  $R_1$  ( $ST_V R_1$ ). Here we have:  $\mathbf{i} = (1,2)$ ,  $\mathbf{o} = (3,4)$  and  $\mathbf{k} = (1,3)$ ; there is only one internal variable to eliminate ( $V_5$ ). The result generated by STAINS 3 is shown in Fig. 3. The first six expressions suppress  $V_5$ . Then the cofactors required to calculate the transimpedances  $Z(\mathbf{o},\mathbf{i})$  and  $Z(\mathbf{i},\mathbf{i})$  are expressed in terms of the elements of the RNAM. Note that the division by the RNAM's determinant is not performed, as the terms cancel out in the formula for  $T_V$ . This determinant ( $D_{00}$ ) is only used in the final sensitivity formula. Numerical evaluation of this sequence requires 97 flops. To compare our method with the direct differentiation of sequential formulae, the compact SOE was first generated by our other programme, STAINS 2 [7], and then the relevant derivatives were inserted manually. The

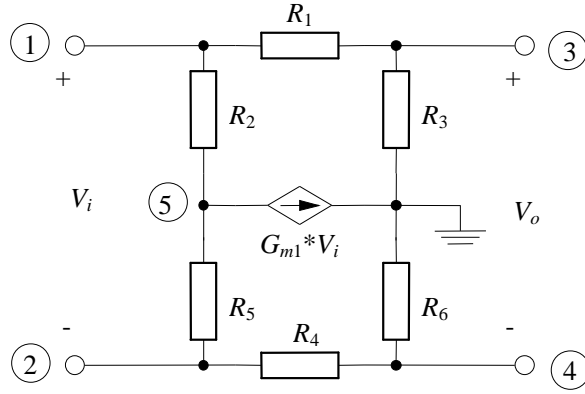


Fig. 2. Small example circuit for sensitivity analysis.

$$\begin{aligned}
 d1 &= (Gm1-G2)/(G2+G5); & Z22 &= (G4+G6)*(x1*(G1+G3)-G1*G1); \\
 x1 &= G1+G2+G2*d1; & Zii &= Z11+Z22-Z12-Z21; \\
 x2 &= G5*d1; & Tv &= Zoi/Zii \\
 d2 &= -(G5+Gm1)/(G2+G5); & Z13 &= G1*(x4*(G4+G6)-G4*G4); \\
 x3 &= G2*d2; & Z23 &= -G1*x2*(G4+G6); \\
 x4 &= G4+G5+G5*d2; & Zik &= Z11+Z23-Z13-Z21; \\
 Z31 &= G1*(x4*(G4+G6)-G4*G4); & Z33 &= x1*(x4*(G4+G6)-G4*G4)-... \\
 Z32 &= -(G4+G6)*x3*G1; & & x2*(G4+G6)*x3; \\
 Z41 &= -(G1+G3)*x2*G4; & Z43 &= -G1*x2*G4; \\
 Z42 &= G4*(x1*(G1+G3)-G1*G1); & Zok &= Z31+Z43-Z33-Z41; \\
 Zoi &= Z31+Z42-Z32-Z41; & Zki &= Z11+Z32-Z12-Z31; \\
 Z11 &= (G1+G3)*(x4*(G4+G6)-G4*G4); & D00 &= (G1+G3)*Z33-G1*Z13; \\
 Z12 &= -(G4+G6)*x3*(G1+G3); & STvR1 &= G1*Zki*(Zok-Tv*Zik)/(Zoi*D00) \\
 Z21 &= -(G1+G3)*x2*(G4+G6); & &
 \end{aligned}$$

Fig. 3. The SOE generated by STAINS 3 for the voltage transmittance of the circuit in Fig. 2 and its sensitivity to  $R_1$  (note:  $G_i = 1/R_i$ ,  $i = 1, 2, \dots, 6$ ).

resulting sequence is not presented here due to space restrictions. It requires 149 flops. (It should be noted that the number of flops is only an approximate measure of computational complexity, especially when operations are performed with complex numbers. Issues related to SoE computational efficiency are discussed in [17].)

*Example 2.* Consider a notch filter shown in Fig. 4. Both op amps are modelled at the device level with the full small-signal model of  $\mu A$  741 (the details of the model can be accessed at <http://www.eng.uts.edu.au/~benr/symbolic/index.htm>). Thus, the equivalent circuit has more than 150 nodes and more than 440 symbolic components. Our goal is to find the sequence of symbolic

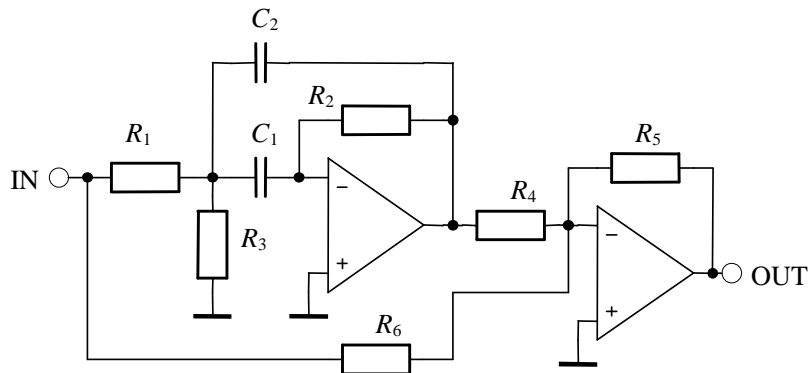


Fig. 4. Notch filter.

expressions allowing us to calculate the voltage transmittance, its differential sensitivity,  $S_{R_3}^{T_V}$ , and large-change sensitivity,  $\delta_{R_3}^{T_V}$ , with respect to resistance  $R_3$ . Since the two op amps are identical, the symbolic reduction process can be simplified considerably. This is achieved by representing each op amp with a  $2 \times 2$  RMNAM (parameters  $y_{11}, \dots, y_{22}$  in Fig. 5), obtained by suppressing all internal nodes of the small-signal model of  $\mu A 741$ . These reduced matrices are then incorporated into a  $6 \times 6$  MNAM of the filter. Then, another sequence of expressions is generated for suppressing all internal variables (potentials at nodes other than input, output and the node to which  $R_3$  is connected), reducing the MNAM of the circuit to a  $3 \times 3$  RMNAM. The second reduction process suppresses the remaining three internal nodes, resulting in a SoE with over 600 expressions.

Next, the formulae for all required cofactors and the determinant of the RMNAM are generated. Finally, the requested network function and its small- and large-change sensitivities are calculated. A small fragment of the SOE, generated by STAINS, is presented in Fig. 5. This SoE, augmented by circuit component values (not shown), can be used in MATLAB (or other mathematical/visualisation software) to gain insight into circuit behaviour unattainable in purely numerical simulators. As an example, we can examine the magnitude of voltage gain variations,  $\Delta T_V$ , as a function of frequency  $f$  and large changes of  $R_3$ . The result is shown in Fig. 6. As can be seen, the maxima of  $|\Delta T_V|$  follow their characteristic paths, different for positive and negative variations of  $R_3$ .

The advantage of such an approach to gaining insight into circuit behaviour cannot be overemphasised. Firstly, additional cost incurred by calculating sensitivities directly is minimal and, more importantly, independent on circuit size (circuits with hundreds of symbolic components have been successfully analysed using this method). Secondly, once the desired circuit parameters are expressed by a SoE, visualising complex relationships is made simple by utilising powerful mathematical software.

```

% OpAmp reduction to a two-port
d1 = -Gx11/(Gp11+Gx11+s*(Cp11+Cu11));
x1 = G5o+Gx10+Gx11+Gc11+Gx11*d1;
x2 = (s*Cu11-Gm11)*d1-Gc11;
d2 = -s*Cu11/(Gp11+Gx11+s*(Cp11+Cu11));
...
d160 = x387/(x393);
y11 = x385-x391*d160;
y21 = x386-x392*d160;
d161 = x390/(x393);
y12 = x388-x391*d161;
y22 = x389-x392*d161;
% Filter reduction to a 3x3 matrix
d1 = -s*C1/(G2+y11+s*C1);
x1 = G1+G3+s*(C1+C2)+s*C1*d1;
x2 = (G2-y21)*d1-s*C2;
d2 = (y12-G2)/(G2+y11+s*C1);
x3 = s*C1*d2-s*C2;
x4 = G2+G4+y22+s*C2+(G2-y21)*d2;
d3 = x2/(x4);
x5 = x1-x3*d3;
x6 = G4*d3;
d4 = -G4/(x4);
x7 = -x3*d4;
x8 = G4+G5+G6+y11+G4*d4;
d5 = -G6/(x8);
x9 = G1+G6+G6*d5;
x10 = (G5-y21)*d5;
x11 = -G1-x7*d5;
d6 = (y12-G5)/(x8);
x12 = G6*d6;
x13 = G5+y22+(G5-y21)*d6;
x14 = -x7*d6;
d7 = x6/(x8);
x15 = G6*d7-G1;
x16 = (G5-y21)*d7;
x17 = x5-x7*d7;
% Voltage Transmittance
Z21 = x11*x16-x10*x17; Zoi = Z21;
Z11 = x13*x17-x14*x16; Zii = Z11;
Tv = Zoi/Zii;
% Differential sensitivity to R3
Z13 = x12*x16-x13*x15; Zik = Z13;
Z23 = x10*x15-x9*x16; Zok = Z23;
Z31 = x10*x14-x11*x13; Zki = Z31;
D00 = x9*Z11+x12*Z21+x15*Z31;
STvR3 = G3*Zki*(Zok-Tv*Zik)/(Zoi*D00);
% Large-change sensitivity to R3
Z33 = x9*x13-x10*x12; Zkk = Z33;
DYk = -DR3/(R3*(R3+DR3));
K0 = 1/(1+DYk*Zkk/D00); K = -DYk*K0;
DTv = K*Zki*(Zok-Tv*Zki)/...
      (Zii*D00+K*Zik*Zki);
DTvR3 = DTv*R3/(Tv*DR3);

```

Fig. 5. Fragment of SOE generated for the notch filter in Fig. 4.

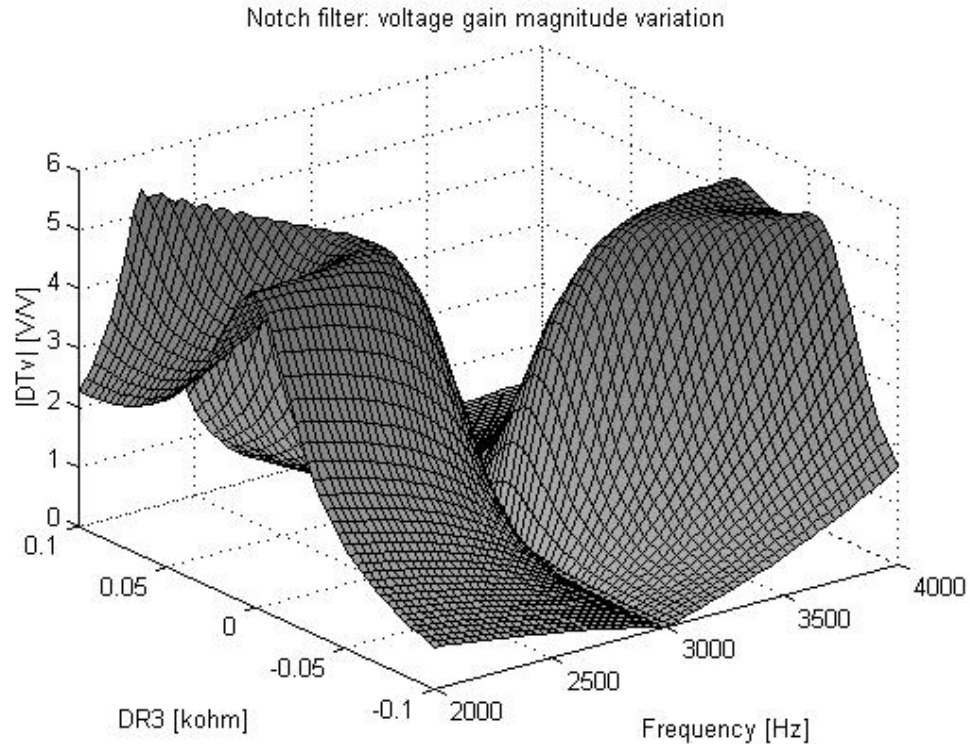


Fig. 6. Variation of the voltage gain as a function of frequency and large changes of  $R_3$  for the notch filter in Fig. 4.

## VII. CONCLUSION

In this paper we have shown that using the two-port transimpedance concept, the network functions together with their differential and large-change sensitivities can be easily obtained in sequential symbolic form. Furthermore, we have demonstrated that once the network function is obtained, very little additional effort is needed to calculate its differential sensitivity. If the sequence of expressions for the network function and its differential sensitivity have been already generated, only a few additional expressions are needed to determine the large-change sensitivity. This additional effort to calculate sensitivities (both small- and large-change) is practically independent of circuit size, making our approach attractive for large-scale circuits. The technique is particularly effective if it is combined with the use of powerful mathematical/visualisation packages, like MATLAB, to gain insight into circuit behaviour unattainable with numerical simulators. It can be applied in circuit optimization, statistical analysis, fault diagnosis, systematic exploration and similar tasks requiring large number of repetitive calculations.

## REFERENCES

- [1] J. Vlach, K. Singhal, *Computer Methods for Circuit Analysis and Design*, 2-nd ed. New York: Van Nostrand Reinhold, 1994.
- [2] G. Gielen, W. Sansen, *Symbolic Analysis for Automated Design of Analog Integrated Circuits*. Boston: Kluwer, 1991.

- [3] J.A. Starzyk, A. Konczykowska, "Flowgraph analysis of large electronic networks," *IEEE Trans. on Circuits and Systems*, vol. 33, no. 3, March 1986, pp. 302-315.
- [4] M.M. Hassoun, P.M. Lin, "A hierarchical network approach to symbolic analysis of large-scale networks," *IEEE Trans. on Circuits and Systems - I: Fundamental Theory and Applications*, vol. 42, no. 4, April 1995, pp.201-211.
- [5] M. Pierzchala, B. Rodanski, "Efficient generation of symbolic network functions for large-scale circuits," *Proc. MWSCAS'96*, pp. 425-428.
- [6] J.A. Starzyk, J. Zou "Direct symbolic analysis of large analog networks," *Proc. MWSCAS'96*, pp. 421-424.
- [7] M. Pierzchala, B. Rodanski, "Generation of sequential symbolic network functions for large-scale networks by circuit reduction to a two-port," *IEEE Trans. on Circuits and Systems - I: Fundamental Theory and Applications*, vol. 48, no. 7, July 2001, pp. 906-909.
- [8] P.M. Lin, "Sensitivity analysis of large linear networks using symbolic programs," *Proc. ISCAS'92*, pp. 1145-1148.
- [9] J.A. Echtenkamp *et al.*, "Hierarchical sensitivity analysis for sequence of expression method," *Proc. ECCTD*, Istanbul, Turkey, August 1995, pp. 75-78,.
- [10] J. Echtenkamp, M. Hassoun, "Implementation issues for symbolic sensitivity analysis," *Proc. MWSCAS'96*, pp. 429-432, 1997.
- [11] F. Balik, "Sensitivity analysis and temperature coefficients calculation for models of electronic devices," *Proc. Intl. AMSE Conf. "Modelling and Simulation,"*, Nice, Sept. 1993, pp. 85-99.
- [12] F. Balik, B. Rodanski, "Calculation of first- and second-order symbolic sensitivities in sequential form via the transimpedance method," *Proc. ECCTD*, Stresa, Italy, August 1999, pp. 70-73.
- [13] F. Balik, B. Rodanski, "Calculation of Large-Change Multiparameter Symbolic Sensitivities via the Transimpedance Method for Large-Scale Circuits," *Proc. SMACD*, Sinaia, Romania, October 2002, pp. 9-12.
- [14] P.M. Lin, *Symbolic Network Analysis*. Amsterdam: Elsevier, 1991.
- [15] R.K. Brayton, R. Spence, *Sensitivity and Optimization*. New York: Elsevier, 1980.
- [16] B. Rodanski, M. Hassoun, "Symbolic Analysis Methods," in *The Circuits and Filters Handbook*, 2-nd ed., W.-K. Chen, Editor. Boca Raton: CRC Press, 2003, pp. 1263-1282.
- [17] B. Rodanski, "Computational Efficiency of Symbolic Sequential Formulae," *Proc. SMACD*, Lisbon, Portugal, October 2000, pp. 45-50.

## APPENDIX A

### Proof of Property 2 (derivative of the transimpedance)

Consider an admittance  $Y_\xi$ , connected to a pair of nodes  $\xi = (\xi_1, \xi_2)$ . In the modified nodal formulation the element stamp for this admittance is:

$$\mathbf{Y}_Y = \begin{matrix} & \xi_1 & \xi_2 \\ \xi_1 & \begin{bmatrix} Y_\xi & -Y_\xi \end{bmatrix} \\ \xi_2 & \begin{bmatrix} -Y_\xi & Y_\xi \end{bmatrix} \end{matrix} \quad (\text{A1})$$

The derivative of MNAM with respect to the admittance  $Y_\xi$  is a  $n \times n$  matrix ( $n$  is the total number



of variables in the modified nodal formulation) with only four nonzero entries:

$$\frac{\partial \mathbf{Y}}{\partial Y_k} = \begin{matrix} \xi_1 & \xi_2 \\ \xi_1 & \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \\ \xi_2 & \end{matrix} = \mathbf{e}_\xi^T \mathbf{e}_\xi, \quad (\text{A2})$$

where vector  $\mathbf{e}_\xi$  is given by:

$$\mathbf{e}_\xi = [0 \ \dots \ 1 \ \dots \ -1 \ \dots \ 0]. \quad (\text{A3})$$

If  $\mathbf{Z}$  is the inverse of the MNAM ( $\mathbf{Z} = \mathbf{Y}^{-1}$ ), then the theorem of the derivative of the matrix inverse gives:

$$\frac{\partial \mathbf{Z}}{\partial Y_\xi} = -\mathbf{Z} \frac{\partial \mathbf{Y}}{\partial Y_\xi} \mathbf{Z} = -\mathbf{Z} \mathbf{e}_\xi^T \mathbf{e}_\xi \mathbf{Z} = -\mathbf{z}_c \mathbf{z}_r, \quad (\text{A4})$$

where the vectors  $\mathbf{z}_c$  and  $\mathbf{z}_r$  are given by:

$$\begin{aligned} \mathbf{z}_c &= \mathbf{Z} \mathbf{e}_\xi^T = [z_{1\xi_1} - z_{1\xi_2} \quad z_{2\xi_1} - z_{2\xi_2} \quad \dots \quad z_{n\xi_1} - z_{n\xi_2}]^T \\ \mathbf{z}_r &= \mathbf{e}_\xi \mathbf{Z} = [z_{\xi_1 1} - z_{\xi_2 1} \quad z_{\xi_1 2} - z_{\xi_2 2} \quad \dots \quad z_{\xi_1 n} - z_{\xi_2 n}] \end{aligned} \quad (\text{A5})$$

From (A4) and (A5) the expression for the derivative of any matrix element,  $z_{pq}$ , with respect to an admittance  $Y_\xi$  can now be derived as:

$$\frac{\partial z_{pq}}{\partial Y_\xi} = -(z_{p\xi_1} - z_{p\xi_2})(z_{\xi_1 q} - z_{\xi_2 q}) \quad (\text{A6})$$

Using (A6) and (6) we will calculate the derivative of a transimpedance  $Z(\boldsymbol{\alpha}, \boldsymbol{\beta})$ :

$$\begin{aligned} \frac{\partial Z(\boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial Y_\xi} &= \frac{\partial z_{\alpha_1 \beta_1}}{\partial Y_\xi} - \frac{\partial z_{\alpha_2 \beta_1}}{\partial Y_\xi} - \frac{\partial z_{\alpha_1 \beta_2}}{\partial Y_\xi} + \frac{\partial z_{\alpha_2 \beta_2}}{\partial Y_\xi} \\ &= -\left( z_{\alpha_1 \xi_1} - z_{\alpha_1 \xi_2} \right) \left( z_{\xi_1 \beta_1} - z_{\xi_2 \beta_1} \right) + \left( z_{\alpha_2 \xi_1} - z_{\alpha_2 \xi_2} \right) \left( z_{\xi_1 \beta_1} - z_{\xi_2 \beta_1} \right) \\ &\quad + \left( z_{\alpha_1 \xi_1} - z_{\alpha_1 \xi_2} \right) \left( z_{\xi_1 \beta_2} - z_{\xi_2 \beta_2} \right) - \left( z_{\alpha_2 \xi_1} - z_{\alpha_2 \xi_2} \right) \left( z_{\xi_1 \beta_2} - z_{\xi_2 \beta_2} \right) \\ &= -\left( z_{\alpha_1 \xi_1} - z_{\alpha_1 \xi_2} - z_{\alpha_2 \xi_1} + z_{\alpha_2 \xi_2} \right) \left( z_{\xi_1 \beta_1} - z_{\xi_1 \beta_2} - z_{\xi_2 \beta_1} + z_{\xi_2 \beta_2} \right) \\ &= -Z(\boldsymbol{\alpha}, \boldsymbol{\xi}) Z(\boldsymbol{\xi}, \boldsymbol{\beta}) = -Z(\boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\beta}) \end{aligned}$$

□

Formulae for derivatives of the transimpedance with respect to values of other circuit components, especially four controlled sources: VCCS ( $g_m$ ), VCVS ( $\mu$ ), CCCS ( $\alpha$ ) and CCVS ( $r_m$ ), can be derived using identical technique.

**Franciszek Balik**

Institute of Telecommunication & Acoustics  
Wroclaw University of Technology  
50-370 Wroclaw, ul. Wyb. Wyspiańskiego 27  
Poland  
Franciszek.Balik@poczta.pwr.wroc.pl

**Benedykt Rodanski**

Faculty of Engineering  
University of Technology, Sydney  
P.O. Box 123, Broadway, NSW  
Australia  
benr@eng.uts.edu.au