83-03

CIRCUIT THEORY AND APPLICATIONS, VOL. 11, 241–264 (1983)

EFFICIENT COMPUTER ANALYSIS OF IDEAL SWITCHED-CAPACITOR CIRCUITS USING MATRIX COMPACTION TECHNIQUES

LUC CLAESEN*, JOOS VANDEWALLE AND HUGO DE MAN
Katholieke Universiteit Leuven, Kardinaal Mercierlaan 94, 3030 Heverlee, Belgium

SUMMARY

In this paper we describe how three simple observations can be used in order to obtain an efficient algorithm for the computer analysis of ideal switched-capacitor circuits. The resulting algorithm is linear in the number of phases. The first observation uses the structure of the z-domain MNA-matrix to come up with a new LU-decomposition scheme which is gradual per timeslot. The second observation allows a great reduction in size of the z-domain MNA-matrix by a matrix compaction algorithm which also operates gradually per timeslot and which can be interleaved with the first gradual LU-decomposition process. This leads to a small matrix which can then be used for a time- and direct frequency-domain analysis. Third, the computations of transfer functions, aliasing functions and sensitivities can be optimized by applying appropriate excitations and making appropriate combinations of the terms in the expressions. These algorithms have been implemented in the SC–analysis program DIANA.SC. The usefulness and efficiency of the program is then illustrated with some examples.

1. INTRODUCTION

Except for some trivial examples, the complete frequency, aliasing and sensitivity analysis of switched capacitor (SC) networks is not easy and is best done by computer. Even then a complete analysis can take several minutes. This motivated the direct computation of frequency, aliasing and sensitivity properties of general periodically switched capacitor networks in the DIANA program by making optimal use of the z-domain transfer matrix,[1] the adjoint network[2] and many other ideas. This results in a set of linear equations, with a size which is proportional to the number of phases (N) and to the size of the network (s). By making full use of the matrix structure, a very efficient LU-decomposition and matrix compaction algorithm is obtained which is linear in the number of phases. With the appropriate choices of the right-hand side and the use of the adjoint switched capacitor network we further minimize the number of linear equations to be solved. The general analysis problem is described in Section 2. It is shown that the full frequency behaviour of switched capacitor networks can in general be obtained by solving a set of linear equations obtained by the well known modified nodal analysis (MNA) method used in most CAD programs today. In Section 3 we describe then an efficient LU decomposition which is tailored to this structure and to the sparsity of the matrices. In fact only one gradual LU decomposition has to be performed in advance and one extra LU decomposition is needed at each frequency point. In Section 4 it is shown how these equations can be reduced by using an algebraic matrix compaction algorithm. The frequency, aliasing and sensitivity computations can then be organized in Section 5 such that a minimal number of linear equations has to be solved. This can be achieved by choosing appropriate excitations for the linear equations and by using the adjoint SC network. Some examples illustrating this algorithm are given in Section 6.

* Supported by the Belgian IWONL and Bell Telephone Mfg. Antwerpen, Belgium.

0098–9886/83/030241–24$02.40
© 1983 by John Wiley & Sons, Ltd.

Received 10 June 1982
Revised 1 November 1982

The above described techniques allowed considerable savings compared to the time domain technique[3] which was earlier implemented in the DIANA program[4] and which was then used for frequency analysis by performing FFTs on impulse responses. At this moment the DIANA program has been enhanced by the algorithms described in this paper[5] and has been tested and used for industrial application. Other programs[6,7] do not make use of the full structure of the problem or are designed for special classes of SC networks. It should also be mentioned that other approaches[8,9] to set up the equations (based on graph theoretic arguments instead of modified nodal analysis stamps) can be combined with the techniques of Sections 4 and 5.

By using matrix compaction, the analysis of switched-capacitor circuits is at least as efficient as the method which uses predefined macro models of SC-building blocks[10] while it still has the advantage that the circuit topology and the number of phases are not restricted.

The computations in this paper are very general in the sense that they apply to SC networks with many phases, arbitrary duty cycles and with or without continuous coupling between input and output. The linearity of controlled sources and capacitors, however, is always assumed.

The basic ideas of this paper have been pointed out in Reference 19 by the authors. In what follows, these ideas and especially the matrix compaction process are carried out more formally.

2. BACKGROUND RESULTS ON SC-CIRCUIT ANALYSIS

2.1. SC-circuits

We define SC circuits to be arbitrary linear networks containing ideal switches, capacitors, independent voltage and charge sources and dependent sources VCVS, QCQS, QCVS and VCQS. The switches are controlled by Boolean clock variables $\phi_i(t) = 0$ or 1. $\phi_i(t) = 0$ (resp. $\phi_i(t) = 1$) corresponds to an open (resp. closed) switch at time t if this switch is driven by clock i. The time is partitioned into time slots $\Delta_k = (t_k, t_{k+1}]$ such that the clock signals do not vary in Δ_k, i.e. $\phi_i(t) = \phi_{ik}$ for $t \in \Delta_k$. We assume that the clock signals are T-periodic, with N time slots (called N phases) in one period of duration T.

Suppose, without loss of generality, that the SC-network is connected in each timeslot. This assumption is, moreover, realistic because on each real chip implementation of an SC-circuit no network parts can be completely floating without a (parasitic) capacitive connection with the rest of the circuit. Suppose also that there are no loops of (independent or dependent) voltage sources and closed switches in any timeslot. Cutsets of (independent or dependent) current sources and open switches are also forbidden in any phase. This is not a real restriction since for such loops or cutsets a contradiction can only be avoided if the parameters of certain components have very specific values.

2.2. SC-analysis by using the Z-domain MNA-equations

In this subsection we summarize the results for SC-analysis which have been presented in previous publications.[1,2]

The MNA-framework of SC-circuits in the time domain was introduced in Reference 3. It has been extended towards the z-domain analysis in Reference 1. The results are:

Fact 1.[1] Let $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \ldots \mathbf{v}_N, \mathbf{v}_{N+1} \ldots\}$ (resp. $\{\mathbf{u}_1, \mathbf{u}_2 \ldots\}$) be the piecewise-constant voltage responses at the nodes (resp., voltage sources in some selected branches). In other words $\mathbf{v}_l$ is the node voltage vector in Δ_l. Also, let $\{\mathbf{q}_1, \mathbf{q}_2 \ldots\}$ (resp., $\{\mathbf{w}_1, \mathbf{w}_2, \ldots\}$) be the charges transferred in some selected branches (resp., injected by charge sources in the nodes). More precisely $\mathbf{q}_l$ is the charge vector transferred in these branches between the end t_l^- of the time slot Δ_{l-1} and t_{l+1}^+. Each of these sequences is partitioned into N different sequences one for each phase and z-transformed e.g.

$$\mathbf{V}_k(z) \triangleq \mathscr{Z}\{\mathbf{v}_{k+lN}\} = \sum_{l=0}^{\infty} \mathbf{v}_{k+lN} z^{-l}, \qquad k = 1, 2, \ldots N \tag{1}$$

Then it is proved in Reference 1 that the input z-transforms $\mathbf{W}_k$, $\mathbf{U}_k$, $k = 1, \ldots N$ are related to the output z-transforms $\mathbf{V}_k$, $\mathbf{Q}_k$, $k = 1, \ldots N$ by:

$$\underbrace{\left[\begin{array}{cc|cc|cc|cc|cc} \mathbf{A}_1 & \mathbf{B}_1 & & & & & & & -\mathbf{E}_1 z^{-1} & \mathbf{0} \\ \mathbf{C}_1 & \mathbf{D}_1 & & & & & & & \mathbf{0} & \mathbf{0} \\ \hline -\mathbf{E}_2 & \mathbf{0} & \mathbf{A}_2 & \mathbf{B}_2 & & & & & & \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_2 & \mathbf{D}_2 & & & & & & \\ \hline & & -\mathbf{E}_3 & \mathbf{0} & \mathbf{A}_3 & \mathbf{B}_3 & & & & \\ & & \mathbf{0} & \mathbf{0} & \mathbf{C}_3 & \mathbf{D}_3 & & & & \\ \hline & & & & & & -\mathbf{E}_N & \mathbf{0} & \mathbf{A}_N & \mathbf{B}_N \\ & & & & & & \mathbf{0} & \mathbf{0} & \mathbf{C}_N & \mathbf{D}_N \end{array}\right]}_{\mathbf{M}^{-1}} \cdot \underbrace{\left[\begin{array}{c} \mathbf{V}_1 \\ \mathbf{Q}_1 \\ \mathbf{V}_2 \\ \mathbf{Q}_2 \\ \mathbf{V}_3 \\ \mathbf{Q}_3 \\ \mathbf{V}_N \\ \mathbf{Q}_N \end{array}\right]}_{\xi} = \underbrace{\left[\begin{array}{c} \mathbf{W}_1 \\ \mathbf{U}_1 \\ \mathbf{W}_2 \\ \mathbf{U}_2 \\ \mathbf{W}_3 \\ \mathbf{U}_3 \\ \mathbf{W}_N \\ \mathbf{Q}_N \end{array}\right]}_{\zeta} \tag{2}$$

where the missing entries are zero and where the matrices $\mathbf{A}_k$, $\mathbf{B}_k$, $\mathbf{C}_k$, $\mathbf{D}_k$ and $\mathbf{E}_k$ characterize the SC network and are determined by accumulating the stamps of all branches as described in References 1 and 3.

The equations (2) contain N blocks, one for each phase. In each block the first part consists of node charge equations and the second part are the constitutive equations of voltage sources, dependent sources and switches. Let $\mathbf{M}$ be the inverse of the matrix in (2), call it the *z-domain transfer matrix* and partition it accordingly into

$$\mathbf{M} = \left[\begin{array}{cc|cc|c|cc} \mathbf{G}_{11} & \mathbf{H}_{11} & \mathbf{G}_{12} & \mathbf{H}_{12} & \cdots & \mathbf{G}_{1N} & \mathbf{H}_{1N} \\ \mathbf{K}_{11} & \mathbf{L}_{11} & \mathbf{K}_{12} & \mathbf{L}_{12} & & \mathbf{K}_{1N} & \mathbf{L}_{1N} \\ \hline \mathbf{G}_{21} & \mathbf{H}_{21} & \mathbf{G}_{22} & \mathbf{H}_{22} & \cdots & \mathbf{G}_{2N} & \mathbf{H}_{2N} \\ \mathbf{K}_{21} & \mathbf{L}_{21} & \mathbf{K}_{22} & \mathbf{L}_{22} & & \mathbf{K}_{2N} & \mathbf{L}_{2N} \\ \hline \cdots & & & & & \cdots & \\ \hline \mathbf{G}_{N1} & \mathbf{H}_{N1} & \mathbf{G}_{N2} & \mathbf{H}_{N2} & \cdots & \mathbf{G}_{NN} & \mathbf{H}_{NN} \\ \mathbf{K}_{N1} & \mathbf{L}_{N1} & \mathbf{K}_{N2} & \mathbf{L}_{N2} & \cdots & \mathbf{K}_{NN} & \mathbf{L}_{NN} \end{array}\right] \tag{3}$$

Fact 2.[1] The frequency domain transfer function $H^{(ij)}(\omega)$ is the entry (ij) of the frequency domain transfer matrix $\mathbf{H}(\omega)$ which relates the branch voltage vector $\mathbf{U}(\omega)$ and the node voltage vector $\mathbf{V}(\omega)$ and can be computed from

$$\mathbf{H}(\omega) = \sum_{k=1}^{N} \left\{ \nu_k(\omega)\, \mathrm{e}^{-j\omega t_{k+1}} \left(\sum_{l=1}^{N} \mathrm{e}^{j\omega t_{l+1}} \mathbf{H}_{kl}(\mathrm{e}^{j\omega T}) \right) + \left[\frac{t_{k+1} - t_k}{T} - \nu_k(\omega) \right] \mathbf{H}_{kk}(\infty) \right\}, \tag{4}$$

where

$$\nu_k(\rho) \triangleq 2\{\sin[\rho(t_{k+1} - t_k)/2] \exp[j\rho(t_{k+1} - t_k)/2]\}/T\rho \tag{5}$$

and the matrices $\mathbf{H}_{kl}(z)$ are given in (3).

Moreover the aliasing matrix is given by:

$$\mathbf{X}(\omega, \omega + n\omega_s) = \sum_{k=1}^{N} \left\{ \nu_k(\omega)\, \mathrm{e}^{-j\omega t_{k+1}} \left[\sum_{l=1}^{N} \mathrm{e}^{j(\omega + n\omega_s)t_{l+1}} \mathbf{H}_{kl}(\mathrm{e}^{j(\omega + n\omega_s)T}) \right] \right.$$

$$\left. + [\nu_k(-n\omega_s) - \nu_k(\omega)] \exp[-n\omega_s t_{k+1}] \mathbf{H}_{kk}(\infty) \right\} \tag{6}$$

Fact 3.[2] For any network $\mathcal{N}$ one can construct an adjoint SC network $\hat{\mathcal{N}}$. This can be used for an efficient sensitivity and noise analysis. The setup of the adjoint network and several useful expressions are given in Reference 2.

3. EFFICIENT LU-DECOMPOSITION OF THE z-DOMAIN MNA MATRIX

For the solution of network equations $\mathbf{AX} = \mathbf{B}$, LU-decomposition or the method of Crout and Doolittle is very often used. This is because the L and U coefficients can be stored at the same memory places as the original matrix $\mathbf{A}$. By using appropriate matrix reordering techniques sparsity in $\mathbf{A}$ often results in sparsity in the L and U factors so that sparse matrix techniques can be used in the solution process.[12] The advantage of this approach is that much less core-memory[12] is required to store the matrix and that the computation time increases linearly with the circuit complexity compared to a full matrix implementation.

The z-domain MNA-matrix $\mathbf{M}^{-1}$ can be partitioned as an $N \times N$ matrix consisting of submatrices $\mathbf{F}_{ij}$, where:

$$\begin{aligned} \mathbf{F}_{ii} &\neq \mathbf{0}, \qquad \forall i = 1 \ldots N-1 \\ \mathbf{F}_{i+1\,i} &\neq \mathbf{0}, \qquad \forall i = 1 \ldots N-1 \\ \mathbf{F}_{1N} &\neq \mathbf{0} \end{aligned}$$

and all other $\mathbf{F}_{ij} = \mathbf{0}$

So the z-domain MNA equations assume the form

$$\underbrace{\begin{bmatrix} \mathbf{F}_{11} & & & & & \mathbf{F}_{1N}z^{-1} \\ \mathbf{F}_{21} & \mathbf{F}_{22} & & & & \\ & \mathbf{F}_{32} & \mathbf{F}_{33} & & & \\ & & \ddots & \ddots & & \\ & & & \mathbf{F}_{N-1\,N-1} & & \\ & & & \mathbf{F}_{N\,N-1} & & \mathbf{F}_{NN} \end{bmatrix}}_{\mathbf{M}^{-1}} \underbrace{\begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \\ \\ \mathbf{X}_{N-1} \\ \mathbf{X}_N \end{bmatrix}}_{\xi} = \underbrace{\begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \mathbf{Y}_3 \\ \\ \mathbf{Y}_{N-1} \\ \mathbf{Y}_N \end{bmatrix}}_{\zeta} \tag{7}$$

where $\mathbf{F}_i$ is a main submatrix of phase i. For the MNA formulation the submatrices $\mathbf{F}_{ii}$, $\mathbf{F}_{ij}$, the unknown vector $\mathbf{X}_i$ and the source vector $\mathbf{Y}_i$ are

$$\mathbf{F}_{ii} = \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{C}_i & \mathbf{D}_i \end{bmatrix} \qquad \mathbf{F}_{ij} = \begin{bmatrix} -\mathbf{E}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \qquad \mathbf{X}_i = \begin{bmatrix} \mathbf{V}_i \\ \mathbf{Q}_i \end{bmatrix} \qquad \mathbf{Y}_i = \begin{bmatrix} \mathbf{W}_i \\ \mathbf{U}_i \end{bmatrix}$$

From the assumptions made about the network topology we may deduce that the submatrices $\mathbf{F}_{ii}$ are all non-singular. Therefore one can perform the pivoting for non-zero diagonal elements during LU-decomposition[13] and the reordering for reducing the number of fillins[12] in the sparse matrix storage scheme gradually one timeslot i at a time on the main submatrices $\mathbf{F}_{ii}$. Compared to the case where the matrix (7) is treated as a whole this may result in small expense of some fillins, and in much more efficient algorithms for pivoting and reordering. This is due to the fact that the complexity of these algorithms is more than linear with respect to the matrix size.

Suppose that (7) has been gradually reordered for non-zero pivots and near-minimum fillins. After LU decomposition of the first $N-1$ phases, (7) becomes:

$$\begin{bmatrix} \mathbf{L}_{11} & & & \\ \mathbf{L}_{12} & \mathbf{L}_{22} & \mathbf{0} & \\ & & & \\ \mathbf{0} & & \mathbf{L}_{N\,N-1} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{U}_{11} & & \mathbf{U}_{1N}z^{-1} \\ & \mathbf{U}_{22} & \mathbf{U}_{2N}z^{-1} \\ & & \\ \mathbf{0} & & \mathbf{F}_{NN} + \mathbf{P}z^{-1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \\ \mathbf{X}_N \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \\ \mathbf{Y}_N \end{bmatrix} \tag{8}$$

where all $\mathbf{L}_{ii}$ are lower and all $\mathbf{U}_{ii}$ are upper triangular submatrices. Remark that up to this point no value for z^{-1} has been specified. Thus, for a frequency analysis the LU-decomposition of the submatrices of the first $N-1$ timeslots need only be done once!

Hereafter the decomposition $\mathbf{L}_{NN}\mathbf{U}_{NN} = \mathbf{F}_{NN} + \mathbf{P}z^{-1}$ of the last timeslot submatrix can be done for every frequency value ω of interest ($z^{-1} = e^{-j\omega T}$) followed by a forward and backward substitution.

In a computer implementation using sparse matrix storage, the LU factors in (8) can be stored in the same pointer system as that used for (7). Of course the submatrices of the last phase $\mathbf{F}_{NN}$, $\mathbf{P}$, $\mathbf{L}_{NN}$, $\mathbf{U}_{NN}$ cannot be treated the same way but can be stored in an extended storage place of the original pointer system.

4. EFFICIENT MATRIX COMPACTION

4.1. Generalities about matrix compaction

The compaction process described below is very general in the sense that it can be applied to all matrix equations of the pencil form:

$$[\mathbf{A} + z^{-1}\mathbf{B}]\mathbf{X} = \mathbf{Y} \tag{9}$$

where z^{-1} is a complex parameter or a linear operator. In our case the matrices $\mathbf{A}$ and $\mathbf{B}$ are directly obtained by accumulating the contributions of the stamps of the components.

In this section an efficient algebraic matrix compaction algorithm is described which usually yields a matrix for SC circuits which is smaller than the sum of the number of applied excitations and observed output variables in the different phases plus the order of the prototype filter. This matrix compaction technique results in matrices which are most often smaller than these obtained by topological methods for matrix compaction of SC-circuits[9] and considerably smaller than these obtained in other approaches which do not take any advantage of matrix compaction.[6–8]

Observe that our topological assumptions guarantee that equation (7) can be solved and the matrix $\mathbf{A}$ will be non-singular.

In the compaction process which is described below, certain internal variables and equations in (9) are removed by Gauss elimination. This removal can only be done for these variables and equations that are of no further interest for the ensuing calculations. After compaction, certain rows and columns in (9) are eliminated by Gauss elimination and (9) is reduced to

$$[\mathbf{A}' + z^{-1}\mathbf{B}']\boldsymbol{\xi}' = \boldsymbol{\zeta}' \tag{10}$$

The method presented in this paper makes full use of the structure of the z-domain matrix (2). After compaction the compacted matrix (10) will still be in a similar form to (2).

The variables in the solution which are of further interest are the desired output variables and certain variables necessary for the sensitivity analysis. The variables in the right-hand side which are of further interest are the non-zero components. Corresponding to these two sets of variables the matrix has certain 'columns and rows of interest'.

Notice the duality between rows and columns in the elimination process because of the use of the adjoint switched capacitor network.

The analysis below can even be adapted to the time-domain analysis of ideal SC circuits, (which do not have transient effects due to opamp poles or switch resistances) by a back transformation of the z-domain equations (9) to the time domain where the z^{-1}-operator corresponds to a delay in time:

$$\mathbf{A}'\boldsymbol{\xi}'(nT) = -\mathbf{B}'\boldsymbol{\xi}'((n-1)T) + \boldsymbol{\zeta}'(nT) \tag{11}$$

Variables in different phases of the solution vector $\boldsymbol{\xi}'$ must of course be positioned into the right time sequence.

4.2. Algebraic observations about Gauss elimination and LU-decomposition of a set of equations (9)

In this section some observations are made concerning LU decomposition and Gauss elimination which can be used in the compaction process.

In order to facilitate the use in a computer program both CPU time and memory requirements have to be taken into account. To avoid large memory storage, sparse matrix techniques have to be used, but also redundant storage has to be avoided. Therefore we use LU decomposition where the LU factors are gradually stored in the space of the original matrix $\mathbf{A}$ as $\mathbf{A}^* = \mathbf{L} + \mathbf{U} - \mathbf{I}$. The next theorem provides conditions which allow an interleaving of an LU-decomposition of a part of a matrix and Gauss eliminating certain parts.

Theorem 1

Consider any set of equations $\mathbf{AX} = \mathbf{Y}$($\mathbf{A}$ non-singular) which is partitioned as:

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{A}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \mathbf{Y}_3 \end{bmatrix} \tag{12}$$

If $\mathbf{A}_{11}$ and $\mathbf{A}_{22}$ are invertible and $\mathbf{Y}_2 = \mathbf{0}$ and $\mathbf{A}_{12} = \mathbf{0}$ or $\mathbf{A}_{21} = \mathbf{0}$, then it does not matter whether one first does a partial LU decomposition with $\mathbf{A}_{11}$ and then a Gauss elimination of $\mathbf{X}_2$ with the second set of equations or the other way round.

Proof. See Appendix I.

The requirements of this theorem are satisfied if one proceeds from the first timeslot to the $(N-1)$th and performs LU-decompositions and Gauss eliminations within each timeslot in the z-domain MNA matrix (2).

The second block row and the second block column correspond to the elementary rows and columns in one timeslot i in (2) that may be eliminated. Only the equations without input source term in timeslot i are eliminated and therefore the requirement of $\mathbf{Y}_2 = \mathbf{0}$ is satisfied in that timeslot.

As already indicated in Section 3, we keep postponing calculations with specific values of z^{-1}. Therefore matrix reductions will also be done as long as possible without specific values of z^{-1}.

It is our intention to compact the equations of phase N so that the resulting equation is still of the form (9). More generally it is possible, but not advisable, to compact further until transfer functions are obtained if one works with formula manipulation in the field of rational functions of z.

In Appendix II some definitions are given about λ-row and λ-column independence. Theorems 2 and 3 state how z^{-1} independent rows and columns can be treated so that Gauss-elimination will only generate matrix elements with terms in z^i where $i = 0$ or -1. Theorems 4 and 5 state under which conditions on the rows and columns these properties are preserved.

Observe that the columns contained in the $N-1$ first block columns of matrix (2) do not depend on z^{-1}. This allows the performance of Gauss eliminations which are valid for all z^{-1}. Also LU decomposition may be performed when taking pivot elements in parts which do not depend on z^{-1} because LU-decomposition is essentially the same as Gauss elimination.

In the last block column the columns corresponding to the charge variables are also z^{-1}-independent. The rows in the last block column corresponding to the branch constitutive relations are also z^{-1}-independent.

With the above observations in mind the compaction process for the z-domain MNA matrix (7) is derived in the next section.

4.3. The compaction process

Depending upon the available computer memory the compaction of the equations of the first $N-1$ phases can be done one at a time (gradual) or all at once. If compaction is performed gradually less memory is necessary but more sparse matrix fillins are generated in the reordering process. If compaction of all first $N-1$ timeslots is done at once fewer fillins occur but the ordering and pivoting algorithms will take more time. The time necessary for matrix reordering and pivoting increases with the third power

of the submatrix dimension to be ordered. Therefore for circuits with many timeslots it is better to use the gradual compaction process. In the DIANA program both kinds of analyses can be specified (default: gradual, can be overridden by the '—GRADUAL' option).

When it is said in what follows that processing (such as reordering, LU-decomposition and Gauss elimination) is done on a certain part of a submatrix $\mathbf{F}_{ij}$ in (7), all the submatrices which are in the same row or column must also be processed according to the selected operation. E.g. in the LU-decomposition of matrix $\mathbf{F}_{ii}$ also the elements of the matrix $\mathbf{F}_{ii}$ and $\mathbf{F}_{i+1\,i}$ are processed at the same time. This must be done of course because they are part of the same matrix (7).

Step 1. Preparation.

1. Set up and enter the submatrix $\mathbf{F}_{1N}$ in the sparse matrix system.

Step 2. Compaction of the first $N-1$ phases.

Perform for $i = 1 \ldots N-1$ steps 2.1 to 2.8

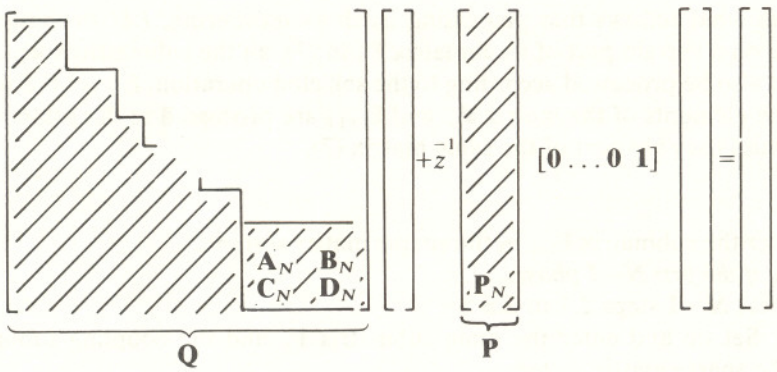
1. *Matrix set up*: Set up and enter the main submatrix $\mathbf{F}_{ii}$ and the coupling submatrix $\mathbf{F}_{i+1\,i}$ of the timeslot i in the sparse matrix system.

   If i is equal to $N-1$ then also set up and enter the main submatrix of the last timeslot $\mathbf{F}_{NN}$ in the sparse matrix system. This will allow to determine a better 'pseudo' minimum number of fillins during the reordering process.
2. *Marking of rows and columns*: 'Mark' the rows in $\mathbf{F}_{ii}$ corresponding to input sources in the direct network and desired outputs of the adjoint network in timeslot i, so that they cannot be eliminated. In the following all rows and columns which may not be removed by Gauss elimination will be 'marked'. The remaining rows and columns are 'unmarked' and can become candidates for the Gauss elimination in the compaction process.

   Mark the columns in $\mathbf{F}_{ii}$ corresponding to desired outputs in the direct network and input sources of the adjoint network in timeslot i, so that they will not be eliminated.
3. *Singleton algorithm*:[11] Reorder singletons (rows and columns with only one non-zero element) which are one in $\mathbf{F}_{ii}$ in the first part of the matrix $\mathbf{F}_{ii}$. These singletons can be assigned as initial pivots for either 'unmarked' rows and columns or marked rows or columns. The singletons originate either from grounded voltage sources and grounded closed switches or from current sources and open switches. Delete rows and columns corresponding to singletons of unmarked rows and unmarked columns.
4. *Preliminary pivot selection*: Perform a pivoting algorithm on $\mathbf{F}_{ii}$ in order to obtain a matrix diagonal which is different from zero during the LU-decomposition process. For this purpose the algorithm[13] can be applied to the SC-circuit in each timeslot i.
5. *Construction of a submatrix with free pivots*: At the beginning of the step no pivots are free. Make pivots free which correspond to rows *and* columns which are unmarked and order them in the upper left corner of the matrix $\mathbf{F}_{ii}$.
6. *Reordering step of the freed pivots*: Reorder the pivots of the freed part in order to cause a 'pseudo' minimum number of fillins in the sparse matrix system[12] during Gauss elimination or LU-decomposition.
7. *Gauss elimination*: Perform a partial Gauss elimination of the whole matrix (7) with pivots that have just been freed. Hereafter entries of the sparse matrix system corresponding to eliminated rows and columns can be deleted in a garbage collection in order to obtain an efficient memory use and fast execution time.

   At this point the original pivots of the remaining matrix either have a marked row or column. However by performing a column and/or row interchange a new submatrix with free pivots can be constructed in the upper left corner. For numerical reasons the new pivots are chosen so that their absolute value is greater than a certain threshold value V_T. If there is at least one pivot freed in this search then repeat steps 2.6 and 2.7.
8. *LU decomposition of the remaining submatrix*: Reorder the remaining non-feed part of $\mathbf{F}_{ii}$ for a minimum number of fillins, and perform a partial LU decomposition in this part.

At this point in the compaction process the matrix equation (2) is now reduced to :



(13)

where only non-zero elements occur in the shaded areas of **Q** and **P**.

Step 3. Compaction of the last phase.

1. *Marking of rows and columns*: Mark rows in $\mathbf{F}_{NN}$ corresponding to input sources and desired outputs of the adjoint network in the last phase. Mark columns in $\mathbf{F}_{NN}$ corresponding to input sources of the adjoint network and desired outputs of the direct network in the last phase.
2. *Singleton algorithm*: Detect singletons and order them in the first part of $\mathbf{F}_{NN}$. Delete rows and columns corresponding to singletons of unmarked rows and unmarked columns.
3. *Preliminary pivot selection*: Select pivots in the last part of **Q** in order to obtain a non-zero diagonal.
4. *Row algorithm*:
   1. *Construction of a submatrix with free pivots.* Make pivots free which correspond to unmarked rows which are z^{-1}-independent and to unmarked columns. See in Appendix II for the definition of z^{-1}-independent row and column. Notice that rows of $\mathbf{C}_N$ and $\mathbf{D}_N$ are z^{-1}-independent during the first pass of the row algorithm.
   2. *Reordering and Gauss elimination of the free pivots.* Reorder the last free pivots for a minimum number of fillins. Perform a Gauss elimination on the free part. Delete the matrix entries corresponding to the eliminated rows and columns. This can be done all at once because Theorem 4 in appendix II guarantees that z^{-1}-independent rows remain z^{-1}-independent during Gauss elimination.
   3. *Construction of a submatrix with free pivots.* Examine for unmarked rows if they are z^{-1}-independent and see if a column interchange of the pivot column with an unmarked column can result in a new free pivot. Search all new pivots in this way. The new pivots are chosen in **Q** (13) so that a strong diagonal can be maintained. If there is at least one pivot free in this search then repeat from the reordering step in this row algorithm.
5. *Column algorithm*: dual to the row algorithm step 3.4. Repeat row- and column algorithms until no additional pivots can be freed and eliminated anymore.
6. At this stage still a row and column algorithm could be performed but now also with marked rows and marked columns and by using LU decomposition instead of Gauss elimination and without deleting matrix entries. This would result in the fact that for each frequency value an LU decomposition has to be done on a minimal dimension matrix. This step is not implemented in the current DIANA version because only little computational gain is expected here.
7. *Reordering and LU decomposition of the remaining part*: Reorder the remaining rows and columns of phase N so that a minimum number of 'fillins' is created by LU decomposition of this part.

The result of this compaction process can be written as:

$$\begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{S} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{R}z^{-1} \\ \mathbf{0} & \mathbf{H}+z^{-1}\mathbf{P} \end{bmatrix} \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{X}' \\ \mathbf{X}'_N \end{bmatrix} = \begin{bmatrix} \mathbf{Y}' \\ \mathbf{Y}'_N \end{bmatrix} \tag{14}$$

where **L** is a lower triangular matrix and **U** is an upper triangular matrix. Only the real matrices **L**, **U**, **Q**, **R**, **H** and **P** need to be stored.

By using this matrix compaction algorithm, the z-domain MNA-matrix can usually be compacted to a matrix whose size is equal to or smaller than the sum of the number of desired inputs and outputs in the different timeslots and the number of independent capacitors in the last timeslot. For two phase SC-circuits this number is usually equal to the order of the reference filter.

Because the interleaved Gauss elimination and LU-decomposition is performed in one phase at a time the above algorithm is linear in the number of phases.

The matrix compaction requires a minimal effort because one only needs to select the equations to be compacted and eliminate them by Gauss elimination. The operations in the Gauss elimination are the same as in LU-decomposition in this interleaved process, only they have a bigger scope. In the last phase part an additional search for z^{-1} independent rows and columns needs to be done.

4.4. Solution process for a certain ω_m-value

Step 1. For any ω_m, set $z = e^{j\omega_m T}$ and perform the LU decomposition of the last phase, which is useful for all excitations $\mathbf{Y}'$, $\mathbf{Y}'_N$. Since the matrix **M** is non-singular for almost all values of ω_m, this LU decomposition can be performed:

$$\mathbf{H} + e^{-j\omega_m T}\mathbf{P} = \mathbf{L}_N \mathbf{U}_N$$

Step 2. For any excitation $\mathbf{Y}'$, $\mathbf{Y}'_N$ the response $\mathbf{X}'$, $\mathbf{X}'_N$ can now be computed as follows:

$$\begin{aligned} \mathbf{Z} &= \mathbf{L}^{-1}\mathbf{Y}' \\ \mathbf{F}_N &= \mathbf{Y}'_N - \mathbf{Q}\mathbf{Z} \\ \mathbf{X}'_N &= \mathbf{U}_N^{-1}\mathbf{L}_N^{-1}\mathbf{F}_N \\ \mathbf{M} &= \mathbf{Z} - e^{-j\omega_m T}\mathbf{R}\mathbf{X}'_N \\ \mathbf{X}' &= \mathbf{U}^{-1}\mathbf{M} \end{aligned} \tag{15}$$

The reader can verify for himself by simple substitutions in (14) that this is indeed a correct solution.

If one is interested in a solution of the adjoint equations, step 1 does not have to be repeated. If the contributions of the different phases to the vectors **X** and **Y** are organized in decreasing order of switching phases, we have to solve $\mathbf{M}^{-1T}\mathbf{X} = \mathbf{Y}$. Using the matrices of (14) this equation can be decomposed as:

$$\begin{bmatrix} \mathbf{U}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{R}^T z^{-1} & \mathbf{H}^T + z^{-1}\mathbf{P}^T \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{Q}^T \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{L}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{X}}' \\ \tilde{\mathbf{X}}'_1 \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{Y}}' \\ \tilde{\mathbf{Y}}'_1 \end{bmatrix} \tag{16}$$

This allows description of the different computations for the adjoint network.

Step 3. Compute

$$\begin{aligned}
\tilde{\mathbf{Z}} &= \mathbf{U}^{T^{-1}}\tilde{\mathbf{Y}}' \\
\tilde{\mathbf{F}}_1 &= \tilde{\mathbf{Y}}_1' - \mathrm{e}^{-j\omega_m T}\mathbf{R}^T\tilde{\mathbf{Z}} \\
\tilde{\mathbf{X}}_1' &= \mathbf{L}_N^{T^{-1}}\mathbf{U}_N^{T^{-1}}\tilde{\mathbf{F}}_1 \\
\tilde{\mathbf{M}} &= \tilde{\mathbf{Z}} - \mathbf{Q}^T\tilde{\mathbf{X}}_1' \\
\tilde{\mathbf{X}}' &= \mathbf{L}^{T^{-1}}\tilde{\mathbf{M}}
\end{aligned} \tag{17}$$

Careful inspection shows that the number of computations required by this algorithm increases linearly with the number N of phases. This is certainly important for circuits with many phases, but also for circuits with two phases the algorithm is very fast.

Observe that the matrix $\mathbf{P}$ characterizes the essential signal circulation phenomena of a SC network.

4.5. Graphical interpretation of this matrix compaction algorithm

The matrix reduction method used in this paper has a purely algebraic nature. It has the advantage with respect to purely topological matrix reduction methods such as those used in[9] that matrix reduction can be carried out much further because the reduction is not restricted to nodes with their corresponding node equations.

For nodal matrices elimination of equations and variables with pivots on the diagonal can be interpreted as a star-mesh transform.[14,15] When using off-diagonal pivots and with the MNA-matrix such a representation is not adequate any more. Then a digraph representation must be used.[16] Beyond this argument which is valid for time-invariant as well as time varying networks there is another argument which is related to the periodicity of the network and the way the z-domain MNA equations are set up. To each node correspond N equations and variables, one for each phase. Because the compaction is algebraic, one has the flexibility to eliminate some of the N and keep the others, which is not the case for topological reductions.

5. EFFICIENT ORGANIZATION OF FREQUENCY, ALIASING AND SENSITIVITY COMPUTATIONS

We show that only two sets of linear equations of the form (14) have to be solved at each ω_m of interest in order to obtain the frequency, aliasing and sensitivity characteristics at ω_m. Our algorithm makes full advantage of the following observations. The frequency and aliasing characteristics require linear combinations of certain entries of the matrix $\mathbf{M}$. These entries and those needed for the sensitivities can often be computed more efficiently from the matrix $\mathbf{M}^T$ of the adjoint switched capacitor network. By choosing the excitation $\mathbf{Y}$ appropriately certain linear combinations are automatically obtained in (14). The contribution from the continuous coupling can be obtained by setting $z = \infty$ in (14).

Step 1

Apply an excitation $\mathbf{Y}$ where component j of $\mathbf{U}_k$ is 1 and all other components of $\mathbf{Y}$ are zero. Solve (14) with $z = \infty$. The solution $\mathbf{X}$ is then partitioned into N vectors.

$$\begin{bmatrix} \boldsymbol{\lambda}_k \\ \boldsymbol{\sigma}_k \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{kk}^{(.j)}(\infty) \\ \mathbf{L}_{kk}^{(.j)}(\infty) \end{bmatrix}, \quad \text{for } k = 1, 2, \ldots N$$

where $\mathbf{S}^{(.j)}$ stands for the jth column of the matrix $\mathbf{S}$.

Apply an excitation $\tilde{\mathbf{Y}}$ where component i of $\tilde{\mathbf{W}}_{\tilde{k}}$, $\tilde{k} = N - k + 1$ is 1 and all other components of $\tilde{\mathbf{Y}}$ are zero and solve these equations for $z = \infty$. The solution $\tilde{\mathbf{X}}$ is then partitioned into N vectors.

$$\begin{bmatrix} \tilde{\boldsymbol{\lambda}}_k \\ \tilde{\boldsymbol{\sigma}}_k \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{G}}_{kk}^{(i.)T}(\infty) \\ \tilde{\mathbf{K}}_{kk}^{(i.)T}(\infty) \end{bmatrix}, \qquad k = 1, 2, \ldots N$$

Step 2

Solve (24) at each ω_m for the original circuit under the following voltage excitation in branch j (not to be confused with the $j=\sqrt{(-1)}$).

$$\mathbf{U}^{(j)} = e^{j\omega_m t_{l+1}} \quad \text{for } l = 1, 2, \ldots N$$

and all other components of $\mathbf{Y}$ zero. The solution $\mathbf{X}$ is then partitioned into N vectors with

$$\begin{bmatrix} \mathbf{\Lambda}_k \\ \mathbf{\Sigma}_k \end{bmatrix} = \sum_{l=1}^{N} e^{j\omega_m t_{l+1}} \begin{bmatrix} \mathbf{H}_{kl}^{(.j)}(e^{j\omega_m T}) \\ \mathbf{L}_{kl}^{(.j)}(e^{j\omega_m T}) \end{bmatrix}, \qquad \text{for } k = 1, 2, \ldots N$$

Solve the adjoint equations at ω_m for

$$\tilde{\mathbf{W}}_k^{(i)} = \nu_k(\omega_m)\, e^{-j\omega_m t_{k+1}} \quad \text{for } k = 1, \ldots N$$

where

$$\nu_k(\rho) = 2\{\sin[\rho(t_{k+1}-t_k)/2] \exp[j\rho(t_{k+1}-t_k)/2]\}/T$$

and all other components of $\tilde{\mathbf{Y}}$ zero. The solution is then partitioned into N vectors with

$$\begin{bmatrix} \tilde{\mathbf{\Lambda}}_l \\ \tilde{\mathbf{\Sigma}}_l \end{bmatrix} = \sum_{k=1}^{N} \nu_k(\omega_m)\, e^{-j\omega_m t_{k+1}} \begin{bmatrix} \tilde{\mathbf{G}}_{kl}^{(i.)T}(e^{j\omega_m T}) \\ \tilde{\mathbf{K}}_{kl}^{(i.)T}(e^{j\omega_m T}) \end{bmatrix}, \qquad \text{for } l = 1, 2, \ldots N$$

Step 3

The desired information at ω_m is then obtained by making the following combinations

$$H^{(ij)}(\omega_m) = \sum_{l=1}^{N} e^{j\omega_m t_{l+1}} \tilde{\mathbf{\Sigma}}_l^{(j)} + \sum_{k=1}^{N} \left[\frac{t_{k+1}-t}{T} - \nu_k(\omega_m)\right] \tilde{\boldsymbol{\sigma}}_k(j) \tag{18}$$

$$X^{(ij)}(\omega_m, \omega_m + n\omega_s) = \sum_{l=1}^{N} e^{j(\omega_m + n\omega_s)t_{l+1}} \tilde{\mathbf{\Sigma}}_l^{(j)} + \sum_{k=1}^{N} [\nu_k(-n\omega_s) - \nu_k(\omega_m)]\, e^{jn\omega_s t_{k+1}} \tilde{\boldsymbol{\sigma}}_k(j)$$

where $x^{(j)}$ is the jth component of the vector x.

As an example of the computation of sensitivities we give the sensitivity of $H^{(ij)}(\omega_m)$ with respect to the variation of the amplification A of a voltage controlled voltage source, where the voltage in branch q is A times the voltage of node m with respect to node n

$$s_A^{H^{(ij)}}(\omega_m) = \frac{A}{H^{(ij)}(\omega_m)} \left[\sum_{l=1}^{N} \tilde{\mathbf{\Sigma}}_r^{(q)} (\mathbf{\Lambda}_r^{(m)} - \mathbf{\Lambda}_r^{(n)}) - \sum_{k=1}^{N} \left(\frac{t_{k+1}-t_k}{T} - \nu_k(\omega_m) \right) \tilde{\boldsymbol{\sigma}}_k^{(q)} (\boldsymbol{\lambda}_k^{(m)} - \boldsymbol{\lambda}_k^{(n)}) \right]$$

Another useful sensitivity is that with respect to a parameter λ which affects all capacitors

$$\begin{aligned} s_\lambda^{H_{(ij)}}(\omega_m) = \frac{\lambda}{H^{(ij)}(\omega_m)} \Bigg[& - \sum_{r=1}^{N} \tilde{\mathbf{\Lambda}}_r^T \frac{\partial \mathbf{A}_r}{\partial \lambda} \mathbf{\Lambda}_r \Bigg] \\ & + \sum_{r=1}^{N-1} \tilde{\mathbf{\Lambda}}_{r+1}^T \frac{\partial \mathbf{A}_{r+1}}{\partial \lambda} \mathbf{\Lambda}_r + e^{-j\omega_m T} \tilde{\mathbf{\Lambda}}_1^T \frac{\partial \mathbf{A}_1}{\partial \lambda} \mathbf{\Lambda}_N \\ & - \sum_{k=1}^{N} \left(\frac{t_{k+1}-t_k}{T} - \nu_k(\omega_m) \right) \left(\tilde{\boldsymbol{\lambda}}_k^T \frac{\partial \mathbf{A}_k}{\partial \lambda} \boldsymbol{\lambda}_k \right) \end{aligned}$$

where $\partial A_k/\partial\lambda$ is known from the design. Observe that also this sensitivity is obtained with the solutions of two sets of linear equations at ω_m (step 2).

6. APPLICATION AND COMPUTER IMPLEMENTATION

6.1. *Example 1: a first order lowpass filter*

This first example is used to illustrate the compaction process which is described in 4.3.

A first order two timeslot SC lowpass filter is shown in Figure 1. Suppose that one wants to know the transfer function from a voltage source in node 1 in timeslot 2 to node 3 in both timeslots.

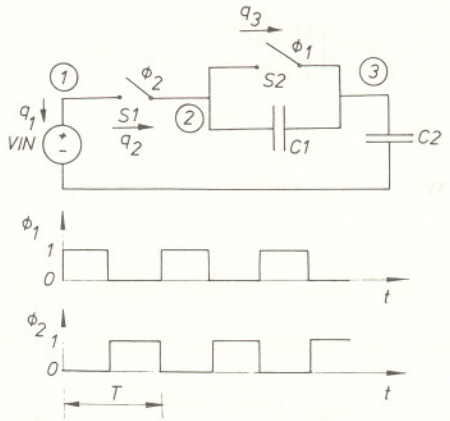


Figure 1. First order 2-phase SC lowpass filter

In the following we refer to elements in the matrix by their original row and column numbers as indicated in (19), even if the rows and columns are reordered in the matrix or if some other rows and columns have been deleted.

After steps 1 and 2.1 the z-domain MNA-matrix has been set up as shown in (19). The input is only applied during the second timeslot. Therefore we set $V_{i1}=0$ and only equation 4′ needs to be marked (step 3.2). The output node 3 is observed during both timeslots, therefore column 3 and 3′ (afterwards in step 3.2) are marked. The marked rows and columns are indicated by a #-sign.

After the application of the singleton algorithm (step 2.3) in the first timeslot submatrix, the singletons (4, 1), (1, 4) and (5, 5) are reordered first in the first timeslot submatrix. All of the rows and columns of the singletons are unmarked, so they can be deleted from $\mathbf{F}_{11}$.

Now the pivoting algorithm is done on the remaining part of $\mathbf{F}_{11}$ (step 2.4). Here $\mathbf{F}_{11}$ is reordered so that the next pivot elements: (2, 6), (3, 3) and (6, 2) appear on the diagonal.

In step 2.5 pivots (2, 6 and 6, 2) are made free. Pivot (3, 3) is not free because column 3 is marked. After the reordering of the freed part in step 2.6 the matrix looks like:

$$
\begin{array}{c}
 \\
1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 1' \\ 2' \\ 3' \\ \#4' \\ 5' \\ 6'
\end{array}
\begin{array}{c}
\begin{array}{cccccc} 1 & 2 & \# & 4 & 5 & 6 \end{array} \\
\left[\begin{array}{cccccc}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & C_1 & -C_1 & 0 & 0 & 1 \\
0 & -C_1 & C_1+C_2 & 0 & 0 & -1 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 \\
0 & -C_1 & C_1 & 0 & 0 & 0 \\
0 & C_1 & -C_1-C_2 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}\right.
\end{array}
\begin{array}{c}
\begin{array}{ccc} 1' & 2' & \# \\ 3' \end{array} \\
\left[\begin{array}{ccc}
0 & 0 & 0 \\
0 & -C_1z^{-1} & C_1z^{-1} \\
0 & C_1z^{-1} & (-C_1-C_2)z^{-1} \\
\hline
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\hline
0 & 0 & 0 \\
0 & C_1 & -C_1 \\
0 & -C_1 & C_1+C_2 \\
\hline
1 & 0 & 0 \\
0 & -1 & 0 \\
0 & 0 & 0
\end{array}\right.
\end{array}
\cdot
\begin{array}{c}
\begin{array}{ccc} 4' & 5' & 6' \end{array} \\
\left[\begin{array}{ccc}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\hline
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\hline
1 & 1 & 0 \\
0 & -1 & 0 \\
0 & 0 & 0 \\
\hline
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 1
\end{array}\right.
\end{array}
\begin{array}{c}
 \\
\left[\begin{array}{c}
V_{11} \\ V_{21} \\ V_{31} \\ \hline Q_{11} \\ Q_{21} \\ Q_{31} \\ \hline V_{12} \\ V_{22} \\ V_{32} \\ \hline Q_{12} \\ Q_{22} \\ Q_{32}
\end{array}\right.
\end{array}
=
\begin{array}{c}
 \\
\left[\begin{array}{c}
0 \\ 0 \\ 0 \\ \hline V_{i1} \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ 0 \\ \hline V_{i2} \\ 0 \\ 0
\end{array}\right.
\end{array}
\tag{19}
$$

$$
\begin{array}{c|ccc:ccc:ccc}
 & & \# & & & & \# & & & \\
 & 6 & 2 & 3 & 1' & 2' & 3' & 4' & 5' & 6' \\
2 & 1 & C_1 & -C_1 & \cdot & -C_1z^{-1} & C_1z^{-1} & \cdot & \cdot & \cdot \\
6 & \cdot & 1 & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
3 & -1 & -C_1 & C_1+C_2 & \cdot & C_1z^{-1} & -(C_1+C_2)z^{-1} & \cdot & \cdot & \cdot \\
\hdashline
1' & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 1 & \cdot \\
2' & \cdot & -C_1 & C_1 & \cdot & C_1 & -C_1 & \cdot & -1 & \cdot \\
3' & \cdot & C_1 & -C_1-C_2 & \cdot & -C_1 & C_1+C_2 & \cdot & \cdot & \cdot \\
\#4' & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\
5' & \cdot & \cdot & \cdot & 1 & -1 & \cdot & \cdot & \cdot & \cdot \\
6' & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1
\end{array}
\tag{20}
$$

Now a partial Gauss elimination is done (step 2.7) on the whole matrix with as (freed) pivots (2, 6) and (6, 2). Then the rows and columns corresponding to these pivots are deleted.

There are no unmarked columns any more in the non-free part. The compaction process is finished for timeslot 1 and an *LU*-decomposition can be done on the non-free part (pivot (3, 3)) of $\mathbf{F}_{11}$. The matrix is now reduced to:

$$
\begin{array}{c|ccc:ccc:ccc}
 & \# & & & \# & & & & & \\
 & 3 & 1' & 2' & 3' & 4' & 5' & 6' \\
3 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\hdashline
1' & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\
2' & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\
3' & -1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\
\#4' & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\
5' & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\
6' & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1
\end{array}
\cdot
\begin{array}{c|ccc:ccc:ccc}
 & \# & & & \# & & & & & \\
 & 3 & 1' & 2' & 3' & 4' & 5' & 6' \\
3 & C_2 & \cdot & \cdot & -C_2z^{-1} & \cdot & \cdot & \cdot \\
\hdashline
1' & \cdot & \cdot & \cdot & \cdot & 1 & 1 & \cdot \\
2' & \cdot & \cdot & C_1 & -C_1 & \cdot & -1 & \cdot \\
3' & \cdot & \cdot & -C_1 & C_1+C_2-z^{-1}C_2 & \cdot & \cdot & \cdot \\
\#4' & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\
5' & \cdot & 1 & -1 & \cdot & \cdot & \cdot & \cdot \\
6' & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1
\end{array}
\tag{21}
$$

Step 3 of the algorithm can start the compaction of the last phase. The singletons (6', 6'), (1', 4') and (4', 1') are ordered first (step 3.2). The rows and columns corresponding to the singletons (6', 6') and (1', 4') can be deleted from $\mathbf{F}_{22}$ because they are unmarked. In order to obtain a non-zero diagonal the pivots (2', 5'), (5', 2') and (3', 3') and singleton (4', 1') are placed on the diagonal (step 3.3).

In the application of the row algorithm it is found that rows 2', 5' are z^{-1}-independent and unmarked (step 3.4.1). These rows are reordered, Gauss eliminated and deleted by using (2', 5') and (5', 2') as pivots (step 3.4.2). Now the original matrix has been reduced to:

$$
\begin{array}{c|ccc:ccc}
 & \# & \# & & \# & \# & \\
 & 3 & 3' & 1' & 3 & 3' & 1' \\
3 & 1 & \cdot & \cdot & C_2 & -C_2z^{-1} & \cdot \\
\hdashline
3' & -1 & 1 & \cdot & -1 & C_1+C_2-C_2z^{-1} & -C_1 \\
\#4' & \cdot & \cdot & 1 & \cdot & & 1
\end{array}
\tag{22}
$$

In the application of the column algorithm (3.5) it is found that column 1' is z^{-1}-independent. This column and row 3' can be eliminated by Gauss elimination by using element (3', 1') as pivot. At this

stage the compaction process is completed and the compacted equations are given by:

$$\begin{array}{c} \\ 3 \\ \#4' \end{array} \begin{array}{c} \begin{array}{cc} \# & \# \\ 3 & 3' \end{array} \\ \left[\begin{array}{c|c} 1 & 0 \\ \hline -1/C_1 & 1 \end{array}\right] \end{array} \cdot \begin{array}{c} \begin{array}{cc} \# & \# \\ 3 & 3' \end{array} \\ \left[\begin{array}{c|c} C_2 & -C_2 z^{-1} \\ \hline 0 & \dfrac{C_1+C_2-C_2 z^{-1}}{C_1} \end{array}\right] \end{array} \begin{bmatrix} V_{31} \\ \hline V_{32} \end{bmatrix} \quad \begin{bmatrix} 0 \\ \hline V_{i2} \end{bmatrix} \tag{23}$$

Notice that by interleaving matrix compaction and LU-decomposition, the reduced matrix is already in LU-decomposed form for the first $N-1$ timeslots. For this simple example, the matrix (19) of 12×12 could be reduced to a 2×2 matrix (23).

6.2. Example 2. Third order lowpass filter[18]

Figure 2 shows a third order reference filter.[18] This filter is converted into the SC-filter in Figure 3 according to the principles described in Reference 18. From the clocking scheme in Figure 3 it is seen that there are 12 phases. The opamp has been modelled with a finite amplification of 1000. This circuit results in 348 z-domain equations.

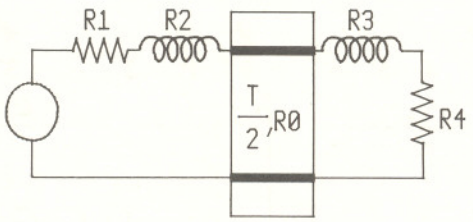


Figure 2. Analogue reference filter of a third order elliptic lowpass filter

For the analysis of the transfer function from the input node in timeslots 1 and 7 to the output node in timeslots 6 and 12, the z-domain MNA-matrix is LU-decomposed and compacted with the above algorithm as implemented in DIANA.SC from a dimension of 348×348 to 9×9. This dimension corresponds to the number of input sources in the different timeslots (=2) plus the number of output nodes in the different timeslots (=2) plus the number of independent capacitors in the last timeslot (=5). This last number is equal to 5 because capacitors CI, $C0$, $C3$ and $C4$ form a loop in the last timeslot and therefore one capacitor voltage will be dependent upon the others. The resulting number of matrix entries is 57. This compaction took 26·1 s VAX 11/780-cpu time. An analysis with this matrix over 100 frequency points took 12.2 cpu s.

Figure 4 shows some analysis results of this filter which were obtained by the DIANA.SC program. Plot (a) shows the amplitude and phase characteristics of the transfer function from all the timeslots at the input to all the timeslots at the output node as defined by equation (4). Plot (b) shows the sensitivity of the transfer function to a parasitic capacitance in node 12 defined as:

$$S_{C_{12}}^{H} = \frac{\mathrm{d}H}{\mathrm{d}C_{12}} \frac{(1\mathrm{pF})}{H}$$

Plots (c) and (d) show the sensitivity of the transfer function to the element values of capacitor $C3$ and the opamp. Finally in (e) the group delay and the amplitude slope is given. The matrix for this whole analysis could be compacted to a dimension of 63×63. This compaction took 25·7 s cpu time. The complete analysis in Figure 4 took 20·6 s cpu time over 100 frequency points.

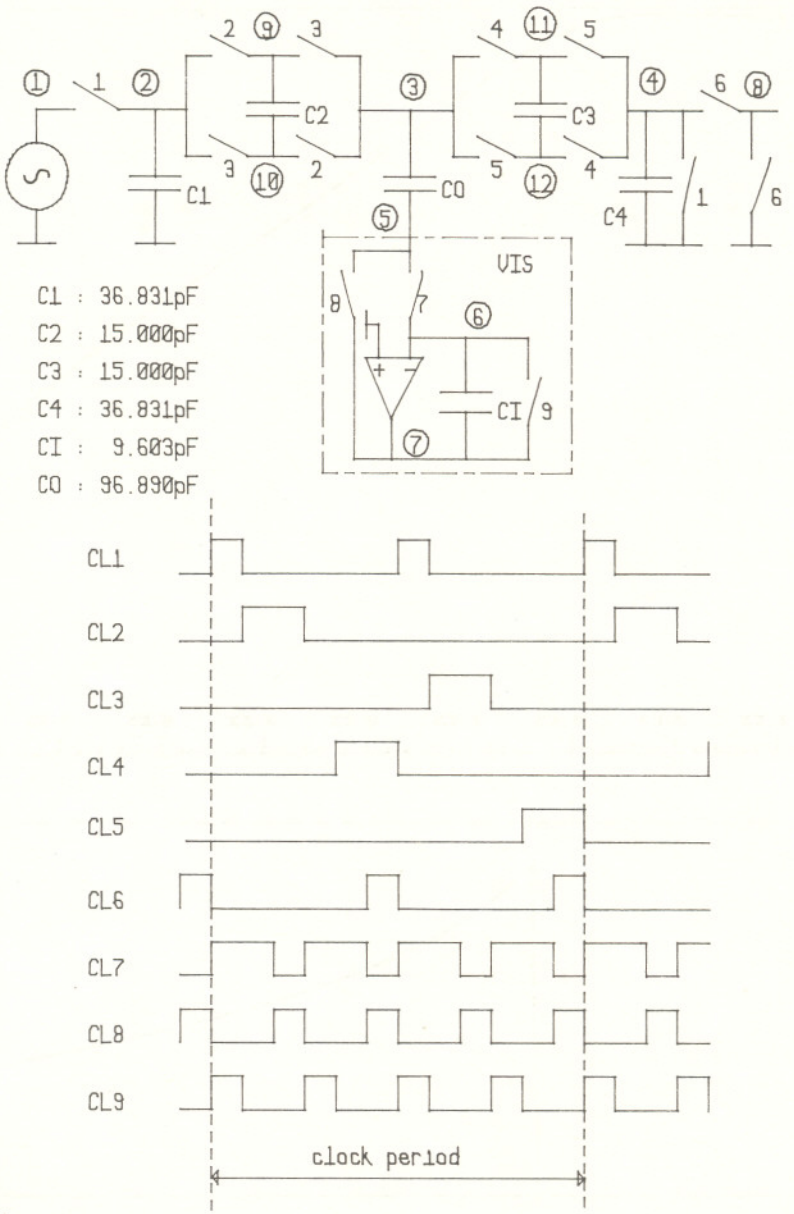


Figure 3. Third order SC elliptic lowpass filter with 12 phases and associated clocking scheme

7. CONCLUSIONS

In this paper we have presented several techniques which are combined in a very efficient algorithm for switched capacitor circuits. This is accomplished by using the z-domain MNA-matrix formalism. The gradual compaction of the z-domain MNA-matrix, by the use of the adjoint network and the appropriate organization of the calculations results in a very efficient and handsome tool for the computation of sensitivities, frequency- and aliasing characteristics. No constraints on the number of phases, duty cycles and continuous I/O-coupling were imposed. The algorithms have all been implemented and tested in the user oriented switched capacitor analysis program DIANA.SC.

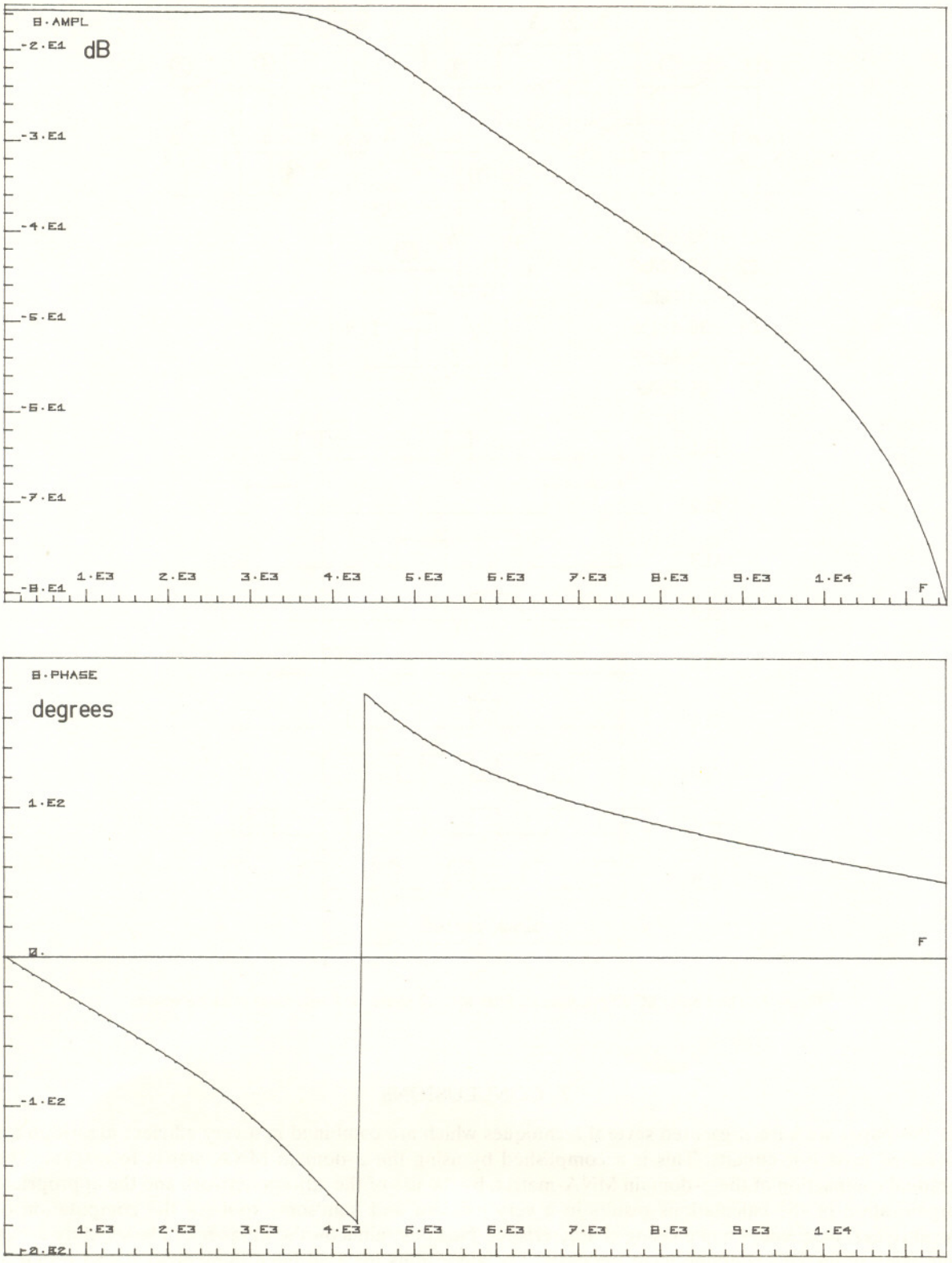


Figure 4. Analysis results of the third order filter in Figure 3: (a) transfer function

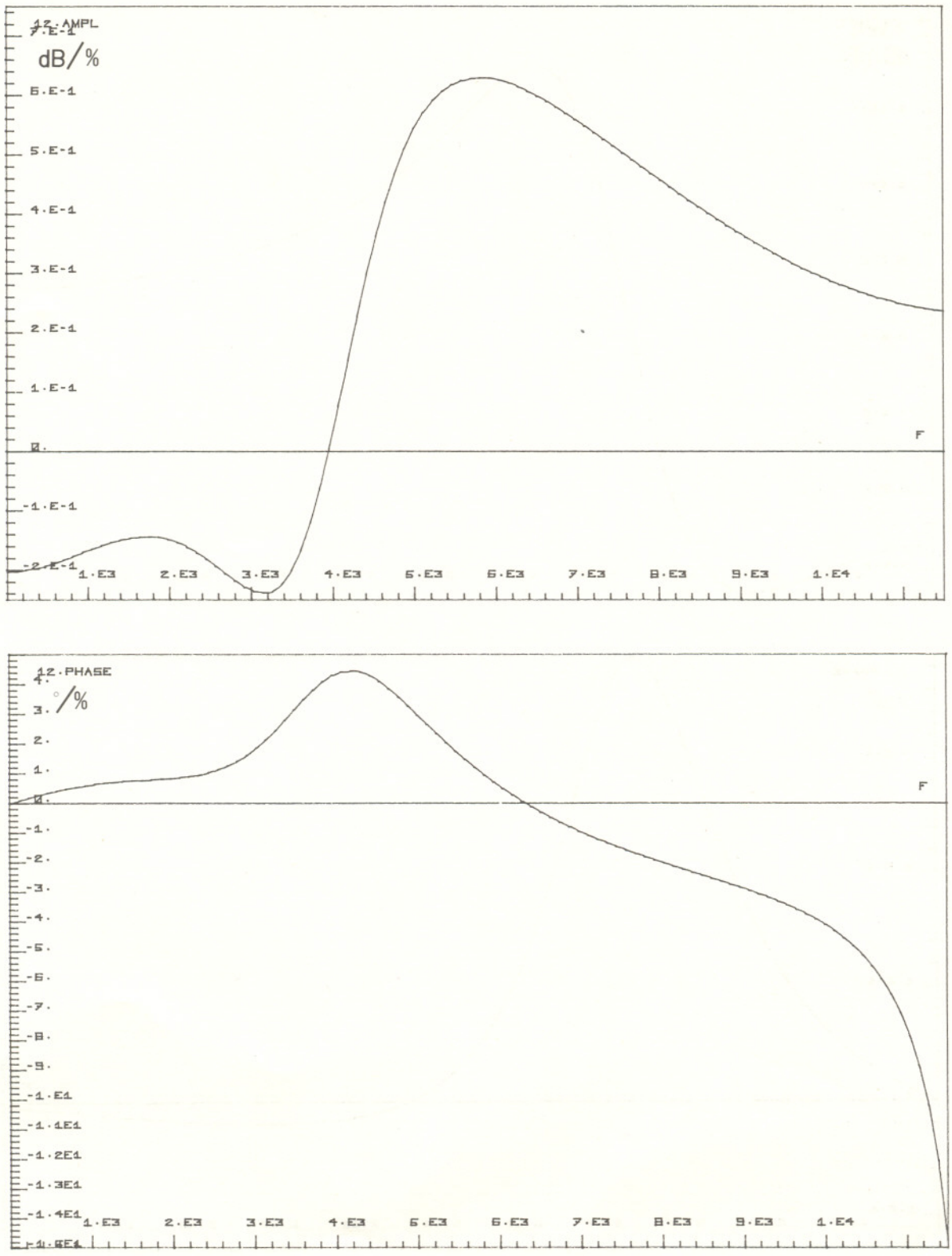


Figure 4(b). Sensitivity of the transfer function to a parasitic capacitance in node 12

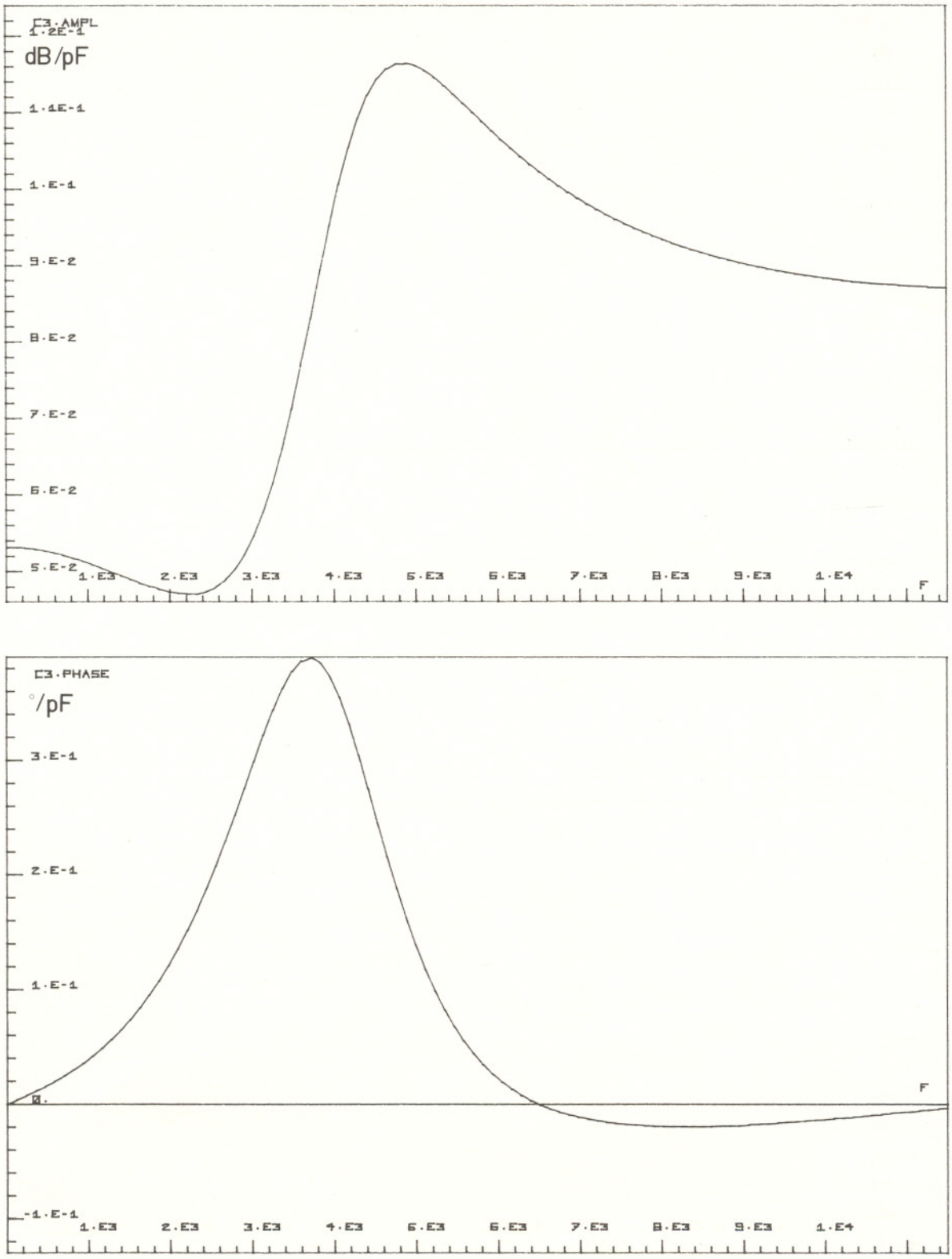


Figure 4(c). Sensitivity of the transfer function to the value of capacitor $C3$

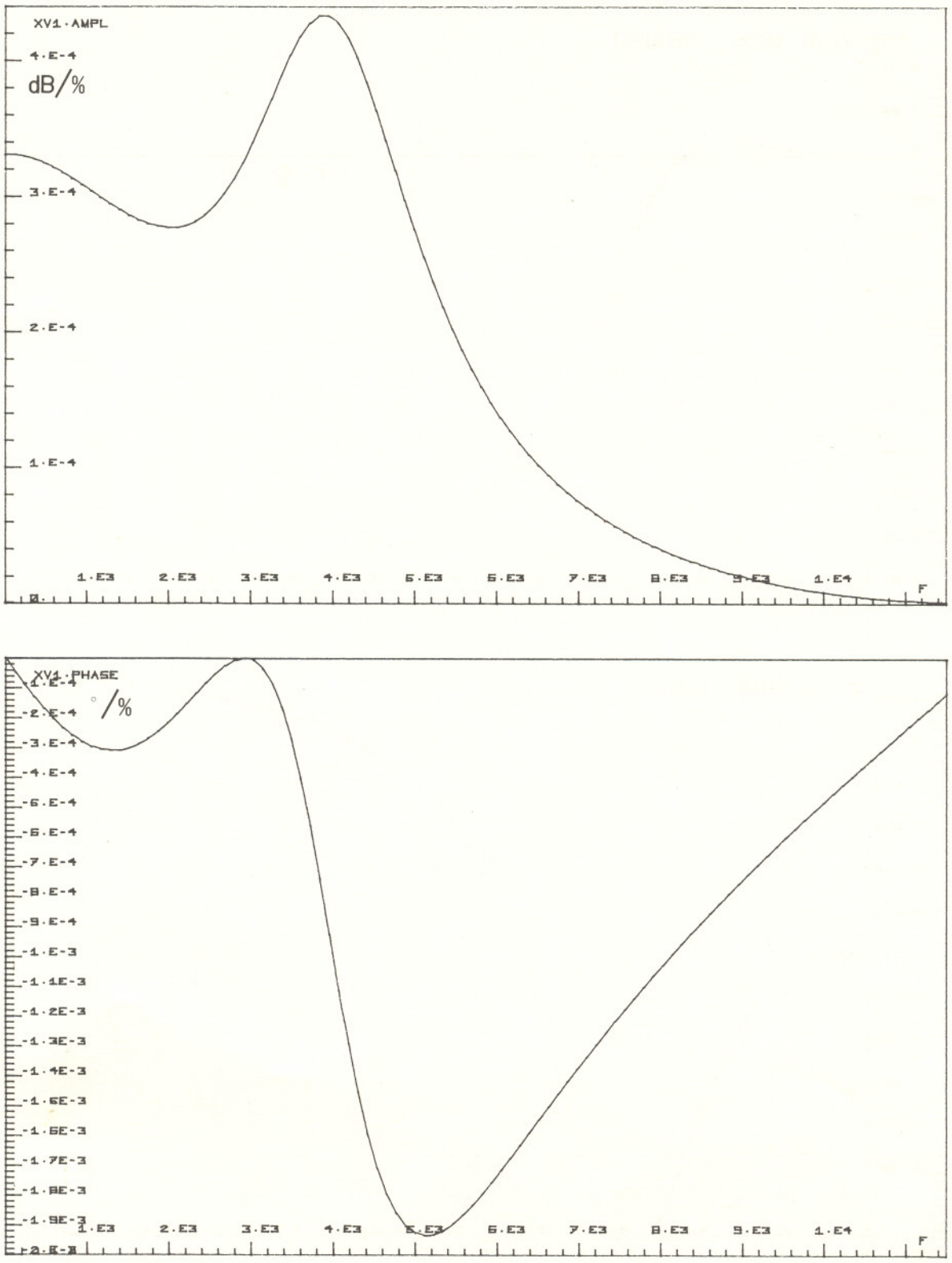


Figure 4(d). Sensitivity of the transfer function to the gain of opamp XV1

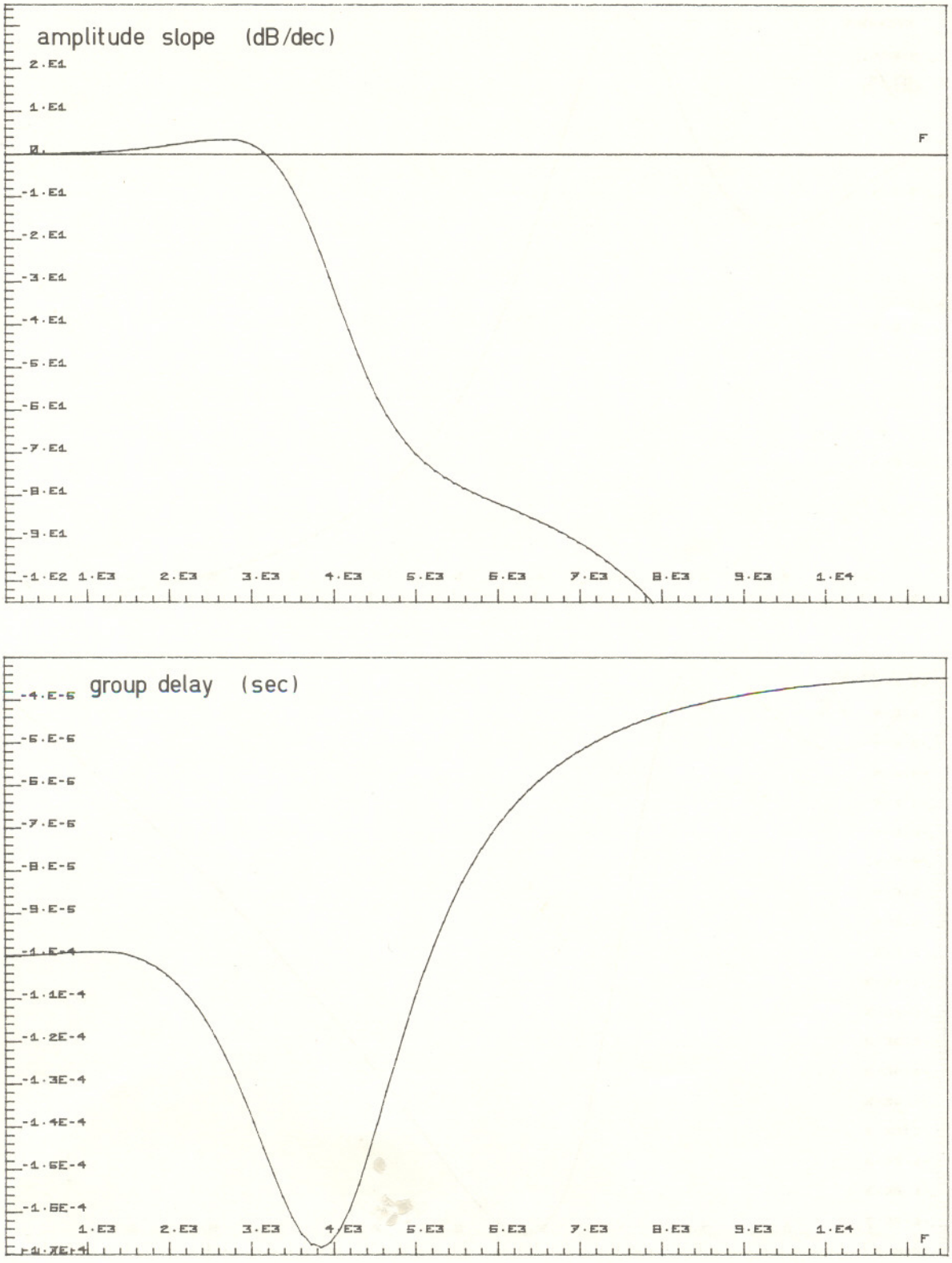


Figure 4(e). Amplitude slope and group delay at the output

The above algorithms form the basis for the direct time- and frequency domain analysis in DIANA.SC. This allows the efficient computations of transfer functions, aliasing functions, sensitivities of the transfer functions to capacitor values, to parasitic capacitances and to values of the four kinds of dependent sources, group delay analysis and amplitude slope analysis. Different analysis operation modes and options are possible as pointed out in Reference 5.

APPENDIX I. PROOF OF THEOREM 1 FOR THE CASE $\mathbf{A}_{21} = \mathbf{0}$

If a partial LU decomposition is performed first, one obtains

$$\begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{L}_{31} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} & \mathbf{U}_{13} \\ \mathbf{0} & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{0} & \mathbf{U}_{32} & \mathbf{U}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{0} \\ \mathbf{Y}_3 \end{bmatrix} \tag{24}$$

with

$$\mathbf{A}_{11} = \mathbf{L}_{11}\mathbf{U}_{11}, \qquad \mathbf{L}_{31} = \mathbf{A}_{31}\mathbf{U}_{11}^{-1}, \qquad \mathbf{U}_{12} = \mathbf{L}_{11}^{-1}\mathbf{A}_{12}, \qquad \mathbf{U}_{13} = \mathbf{L}_{11}^{-1}\mathbf{A}_{13} \tag{25}$$

$$\mathbf{U}_{32} = \mathbf{A}_{32} - \mathbf{A}_{31}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}$$

$$\mathbf{U}_{33} = \mathbf{A}_{33} - \mathbf{A}_{31}\mathbf{A}_{11}^{-1}\mathbf{A}_{13}$$

This set of equations is equivalent to:

$$\begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} & \mathbf{U}_{13} \\ \mathbf{0} & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{0} & \mathbf{U}_{32} & \mathbf{U}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ -\mathbf{L}_{31}\mathbf{L}_{11}^{-1} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{0} \\ \mathbf{Y}_3 \end{bmatrix} \tag{26}$$

In order to perform a Gauss elimination of $\mathbf{X}_2$ with the second set of equations, we use the invertibility of $\mathbf{A}_{22}$ and obtain:

$$\begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{13} - \mathbf{U}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{23} \\ \mathbf{0} & \mathbf{U}_{33} - \mathbf{U}_{32}\mathbf{A}_{22}^{-1}\mathbf{A}_{23} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11}^{-1} & \mathbf{0} \\ -\mathbf{L}_{31}\mathbf{L}_{11}^{-1} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_3 \end{bmatrix} \tag{27}$$

Conversely the Gauss elimination of $\mathbf{X}_2$ using the second set of equations of (19) provides:

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{13} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{23} \\ \mathbf{A}_{31} & \mathbf{A}_{33} - \mathbf{A}_{32}\mathbf{A}_{22}^{-1}\mathbf{A}_{23} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_3 \end{bmatrix} \tag{28}$$

The LU decomposition of the first set of equations in (28) produces:

$$\begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{31} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{31}^{*} \\ \mathbf{0} & \mathbf{U}_{33}^{*} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_3 \end{bmatrix} \tag{29}$$

with (25) and

$$\begin{aligned} \mathbf{U}_{31}^{*} &= \mathbf{L}_{11}^{-1}(\mathbf{A}_{13} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{23}) \\ \mathbf{U}_{33}^{*} &= \mathbf{A}_{33} - \mathbf{A}_{32}\mathbf{A}_{22}^{-1}\mathbf{A}_{23} - \mathbf{L}_{31}\mathbf{L}_{11}^{-1}(\mathbf{A}_{13} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{23}) \end{aligned} \tag{30}$$

This set of equations is then equivalent with:

$$\begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{31}^{*} \\ \mathbf{0} & \mathbf{U}_{33}^{*} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11}^{-1} & \mathbf{0} \\ -\mathbf{L}_{31}\mathbf{L}_{11}^{-1} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{bmatrix} \tag{31}$$

Using (25) and (30) one can easily verify that (27) and (31) are exactly the same equations as claimed in the Theorem. □

APPENDIX II. λ-INDEPENDENCE IN THE COMPACTION PROCESS

Consider a matrix equation:

$$(\mathbf{A}+\lambda\mathbf{B})\mathbf{X}=\mathbf{Y} \tag{32}$$

where $\mathbf{Y}$ is the excitation vector, $\mathbf{X}$ is the unknown vector and $\mathbf{A}$ is a non-singular $(n\times n)$ matrix and $\mathbf{B}$ is an $(n\times n)$ matrix that may be singular. λ is a complex parameter or a linear operator.

Definition 1

A row i of $\mathbf{A}$, $\mathbf{B}$, $\mathbf{Y}$ in (32) is said to be *λ-independent* if the excitation $y_i=0$ and if there is a constant γ_i of such that

$$b_{ij}=\gamma_i a_{ij} \quad \text{for } j=1\ldots n$$

Hence $(1+\lambda\gamma_i)$ is a common factor in equation i and can be eliminated by a division.

Definition 2

A column j of $\mathbf{A}$ and $\mathbf{B}$ in (32) is said to be *λ-independent* if the variable x_j is of no interest and if for all elements a_{ij} $(i=1\ldots n)$ of $\mathbf{A}$ and all elements b_{ij} $(i=1\ldots n)$ of $\mathbf{B}$ the following condition is satisfied.

$$b_{ij}=\gamma_j a_{ij} \quad \text{for } i=1\ldots n$$

This implies that $(1+\lambda\gamma_i)$ is a common factor of the variable x_j. Therefore x_j can be scaled by $(1+\lambda\gamma_i)$.

Observe that the factors γ_i (and γ_j) in Definitions 1 and 2 are finite because the converse would imply that all a_{ij} of row i (resp. column j) are 0. This is not allowed because the matrix $\mathbf{A}$ is assumed to be non-singular.

Theorem 2

If a row i is λ-independent and if $\mathbf{B}'$ is the matrix which one obtains by setting all elements b_{ij} $(j=1\ldots n)$ of $\mathbf{B}$ equal to zero, then the equation:

$$(\mathbf{A}+\lambda\mathbf{B}')\mathbf{X}=\mathbf{Y} \tag{33}$$

is equivalent to (32).

Proof. Row i in matrix equation (32) is given by:

$$\sum_{j=1}^{n} a_{ij}x_j+\lambda\sum_{j=1}^{n} b_{ij}x_j=y_i \tag{34}$$

But because row i is λ-independent $y_i=0$ and $b_{ij}=\gamma_i a_{ij}$ equation (34) can be replaced by:

$$(1+\lambda\gamma_i)\sum_{j=1}^{n} a_{ij}x_j=0 \tag{35}$$

Dividing (35) by $(1+\lambda\gamma_i)$ produces:

$$\sum_{j=1}^{n} a_{ij}x_j=0$$

Hence (32) and (33) are equivalent. □

Theorem 3

If a column j is λ-independent, and if $\mathbf{B}'$ is the matrix which one obtains by setting all elements b_{ij} $(i = 1 \ldots n)$ of $\mathbf{B}$ equal to zero, then the equation with

$$x_j' = x_j(1+\lambda\gamma_j), \qquad x_k' = x_k, \qquad k = 1, \ldots n, k \neq j$$

$$(\mathbf{A}+\lambda\mathbf{B}')\mathbf{X}' = \mathbf{Y}$$

is equivalent to (32).

Proof. Denote by

$$\sum_{l=1}^{n\,j} a_l = \left(\sum_{l=1}^{n} a_l\right) - a_j$$

the summation over all a_l except a_j. The kth equation of (32) can now be written as:

$$\sum_{l=1}^{n\,j} (a_{kl}+\lambda b_{kl})x_l + (a_{kj}+\lambda b_{kj})x_j = y_k \tag{36}$$

By substituting the λ-independent part of column j

$$b_{kj} = \gamma_j a_{kj}, \qquad (k = 1 \ldots n) \tag{37}$$

into (36) and replacing x_j by $x_j'/(1+\lambda\gamma_j)$ we obtain

$$\sum_{l=1}^{n\,j} (a_{kl}+\lambda b_{kl})x_l + a_{kj}x_j' = y_k, \qquad (k = 1 \ldots n)$$

In this equivalent matrix representation we notice that all b_{ij}-terms are equal to zero. □

Theorem 4

All λ-independent rows of (32) remain λ-independent if Gauss elimination is applied on one of these λ-independent rows.

Proof. Let row i and k be λ-independent. Then according to Theorem 2 they can be written as:

$$\sum_{j=1}^{n} a_{ij}x_j = 0 \tag{38}$$

$$\sum_{j=1}^{n} a_{kj}x_j = 0 \tag{39}$$

If a variable x_l by use of equation (38) $(a_{il} \neq 0)$ is Gauss eliminated in the remaining equations, the λ-independent row (39) is replaced by:

$$\sum_{j=1}^{n\,l} \left(a_{kj} - \frac{a_{kl}}{a_{il}} a_{ij}\right) x_j = 0 \tag{40}$$

According to Definition 1, this row k is λ-independent with $\gamma_k = 0$. □

Theorem 5

All λ-independent columns of (32) remain λ-independent if an unknown variable corresponding to a λ-independent column is Gauss-eliminated.

Proof. Let columns j and l be λ-independent. According to Theorem 3 the equations can then be written as:

$$\sum_{m=1}^{n\,j,l} (a_{km}+\lambda b_{km})x_m + a_{kj}x_j' + a_{kl}x_l' = y_k, \qquad (k = 1 \ldots n)$$

Suppose that unknown variable x_j will be eliminated by use of equation i ($a_{ij} \neq 0$). The remaining equations are now replaced by:

$$\sum_{m=1}^{n}{}^{j,l}\left[\left(a_{km}-\frac{a_{kj}a_{im}}{a_{ij}}\right)+\lambda\left(b_{km}-\frac{b_{kj}b_{im}}{a_{ij}}\right)\right]x_m+\left(a_{kl}-\frac{a_{kj}a_{il}}{a_{ij}}\right)x'_l=\left(1-\frac{a_{kj}}{a_{ij}}\right)y_k, \qquad (k=1\ldots n, \text{ and } k\neq j) \tag{41}$$

From Definition 2 the λ-independent column l is still λ-independent with $\gamma_k = 0$.

Remark. Notice that the Gauss elimination of λ-independent rows does not guarantee that λ-independent columns preserve this property and vice versa.

Theorem 6

A Gauss elimination of a λ-independent row or a λ-independent column in equation (32) results in an equation of the form:

$$(\mathbf{A}'+\lambda\mathbf{B}')\mathbf{X}'=\mathbf{Y}'$$

Proof. The proof of this theorem is analogous to the proof of Theorems 4 and 5.

REFERENCES

1. J. Vandewalle, H. De Man and J. Rabaey, 'Time, frequency and z-domain modified nodal analysis of switched capacitor networks', *IEEE Trans. Circuits and Systems*, **CAS-28**, 187–195 (1981).
2. J. Vandewalle, H. De Man and J. Rabaey, 'The adjoint switched capacitor network and its application to frequency, noise and sensitivity analysis', *Int. J. Circuit Theory and Applications*, **9**, 77–88 (1981).
3. H. De Man, J. Rabaey, G. Arnout and J. Vandewalle, 'Practical implementation of a general computer aided design technique for switched capacitor circuits', *IEEE J. Solid State Circuits*, **SC-15**, 190–200 (1980).
4. H. De Man, J. Rabaey, G. Arnout and J. Vandewalle, 'DIANA as a mixed mode simulator for MOSLSI sampled-data circuits', *Proc. IEEE Symp. Circuits and Systems*, 435–438, Houston (1980).
5. L. Claesen, H. De Man, J. Vandewalle and J. Rabaey: 'DIANA.SC a versatile top-down analysis tool for switched-capacitor circuits', *Microelectronics Journal*, **14** (to appear) (1983).
6. F. Brglez, 'SCOP—A switched-capacitor optimization program', *Proc. IEEE Symp. Circuits and Systems*, 985–988 (1980).
7. Y. L. Kuo, M. L. Liou and J. W. Kasinskas, 'Equivalent circuit approach to the computer-aided analysis of switched capacitor circuits', *IEEE Trans. Circuits and Systems*, **CAS-26**, 708–714 (1979).
8. Y. P. Tsividis, 'Analysis of switched capacitive networks', *IEEE Trans. Circuits and Systems*, **CAS-26**, 935–947 (1979).
9. S. C. Fang and Y. P. Tsividis, 'Modified nodal analysis with improved numerical methods for switched capacitive network', *Proc. IEEE Symp. Circuits and Systems*, 977–980, Houston (1980).
10. C. F. Lee and W. K. Jenkins, 'Computer-aided analysis of switched capacitor filters', *IEEE Trans. Circuits and Systems*, **CAS-28**, 681–691 (1981).
11. C-W. Ho, A. Ruehli and P. Brennan, 'The modified nodal approach to network analysis', *IEEE Trans. Circuits and Systems*, **CAS-22**, 504–509 (1975).
12. I. S. Duff, 'A survey of sparse matrix research', *Proc. IEEE*, **65**, 500–535 (1977).
13. I. N. Haij, P. Yang and T. N. Trick, 'Avoiding zero pivots in the modified nodal approach', *IEEE Trans. Circuits and Systems*, **CAS-28**, 271–279 (1981).
14. L. Versfeld, 'Remarks on star-mesh transform of electrical networks', *Electronics Letters*, **6**, 597–599 (1970).
15. M. G. Rezai-Fakhr, G. C. Temes, 'Node elimination in linear active circuits', *Electronics Letters*, **11**, 121–122 (1975).
16. A. L. Sangiovanni-Vincentelli, 'A graph theoretical interpretation of non-symmetric permutation on sparse matrices', *Int. J. Circuit Theory and Applications*, **5**, 139–147 (1977).
17. H. De Man, J. Rabaey, L. Claesen and J. Vandewalle, 'DIANA.SC: A complete CAD system for switched capacitor filters', *Proc. ESSCIRC*, 1981, Freiburg, September 1981, pp. 130–133.
18. A. Fettweis, D. Herbst, B. Hoefflinger, J. Pandel and R. Schweer, 'MOS switched-capacitor filters using voltage invertor switches', *IEEE Trans. Circuits and Systems*, **CAS-27**, 527–538. (1980).
19. J. Vandewalle, L. Claesen and H. De Man, 'A very efficient computer algorithm for direct frequency, aliasing and sensitivity analysis of switched capacitor networks', *IEEE Int. Symposium on Circuits and Systems Conference*, 864–867 Chicago (1981).