



# Computationally efficient large-change statistical analysis of linear electronic circuits

T. Ilić, V.B. Litovski \*, I.V. Litovski, S. Stojiljković

*Faculty of Electronic Engineering, University of Niš, Beogradska 14, 18000 Niš, Yugoslavia*

Received 10 November 1999; received in revised form 10 January 2000

## Abstract

A new method is presented for efficient statistical analysis of linear electronic circuits, when small and large parameter tolerances are given. The statistically generated value of the parameter is considered as a faulted value, as it deviates from the nominal thus enabling the application of a simulation method which uses a new approach of concurrent fault simulation. This method adds new elements to the circuit, representing individual parameter increments, while keeping the topology of the original one. The equations for the original and several perturbed circuits are formulated and solved simultaneously. In this way, redundant computations are avoided in both the equation formulation and equation solving phases, which shorten the simulation time. A statistical frequency and time domain tolerance simulator of linear circuits was developed on the basis of this method with effective user-friendly interface. The method is especially suited for yield sensitivity to some selected circuit parameters estimation. Here simulation results of several benchmark circuits are presented. Efficiency analysis is also included. © 2000 Elsevier Science Ltd. All rights reserved.

## 1. Introduction

Realistic system and circuit design must account for the fact that exact realizations of hand calculations are seldom achieved. The reason for this includes the physical and economic constraints in the manufacture, but the consideration should be also focused on aging or varied field environments in which the system must operate. The effects of variations in design parameters, which are usually modeled as random variables, can be investigated via statistical circuit analysis [1]. The conventional approach to statistical analysis is the Monte Carlo method, in which, for a set of specified probability density functions of design parameters, an empirical distribution for various outputs or performance measures is found. As a secondary result, the parametric yield may be extracted from the results obtained. In this way, the designer can obtain results characterizing the output of the manufacturing process and can predict the system performance in different conditions. Tolerance analysis using Monte Carlo method is well established [2], and prominent simulators such as SPICE [3] exploit this method for acquiring statistical data of a circuit.

A specific application of the statistical analysis may be found in the iterative tolerance design methods such as design centering [4], worst case analysis [5], or similar. In these methods, the statistical analysis is repeatedly performed in every iteration in order to estimate the performance of the updated circuit. When design centering is of concern, fast new search algorithms are available [6], aimed to reduce the design time.

The Monte Carlo method is algorithmically straightforward [7]. The tolerance simulator repetitively generates sets of component values, simulates the circuit, and displays the obtained results. This simple method of repetitive simulations is prescribed in many simulators for its ease of implementation. To estimate its computational complexity, however, the following is to be considered. Let us denote the  $m$ -vector of nominal values of the circuit parameters as  $\mathbf{p}_0$  which will be

\* Corresponding author. Tel.: +381-18-529-208; fax: +381-18-46-180.

E-mail address: vanco@elfak.ni.ac.yu (V.B. Litovski).

a subject of statistical variations. In every step of the Monte Carlo analysis, a new vector is generated, designated as  $\mathbf{p}_i$ ,  $i = 1, 2, \dots, k$ ,  $k$  being the number of Monte Carlo simulations. It will be convenient for further use, to express this new value as  $\mathbf{p}_i = \mathbf{p}_0 + \Delta\mathbf{p}_i$ . In such a Monte Carlo analysis, the circuit should be analyzed  $k + 1$  times. If the frequency or time domain analysis is performed in  $n$  distinct time or frequency points, the overall number of analyses will be  $n(k + 1)$ . This may become a serious computational task even for linear circuits and even if no optimization is considered.

Here, we want to consider a special case of statistical analysis when the influence of only one or a small number of parameters to the circuit response or to the yield is investigated. This kind of analysis may be performed in order to highlight the large change sensitivity of the response to a parameter for which small-change adjoint-network based sensitivity analysis [8] (because of the magnitude of the parameter tolerances or for other reason) is not satisfactory. Generally speaking, the new circuit arising in every Monte Carlo analysis step, may be considered as a perturbation of the nominal one, the parameter deviation being  $\Delta\mathbf{p}_i$ . These deviations may be, however, considered as parametric faults of the circuit with  $m$  parameters. So, the circuit analysis within one Monte Carlo step may be considered as a fault analysis and fault analysis method may be applied for this reason. This, of course, stands for the case when only one parameter is also tolerated or faulted.

There were many attempts to reduce the computational complexity of the Monte Carlo method, for example see Ref. [9]. We find that one of the ways in this direction should be to reduce the time needed for evaluation of the circuit response [10]. In this sense, we would prefer a method performing equation formulation and solving simultaneously (or concurrently) by avoiding repetitive computation which were supposed to be done when the original circuit was analyzed.

We shall group the existing methods attempting to apply such concepts into three groups. First, the difference form of Tellegen's theorem [11] was applied for large change sensitivity analysis [12]. This, in fact, stands for tolerance analysis in the frequency domain. The method is based on an analysis of the adjoint network (equally complex as the analysis of the original one) followed by a solution of a newly constituted system of linear equations of order  $m$ . The new system being relatively small and sparse at the same time gives to the method, high efficiency. The only disadvantage of this method is lack of applicability in the time domain. As to our knowledge, no refinements of this method were published up to date. As the second method, we will mention a set of attempts based on Householder's formula [13] which was analyzed in Ref. [1]. Application of this method again leads to a solution of the original circuit equations and, again, produces an  $m$ th order new system of equations. The computational overheads of this procedure, as shown in Ref. [1], are proportional to  $m^3$  which is not efficient enough for linear circuits tolerance analysis, whereas successfully applied to nonlinear concurrent fault simulation [14,15]. The third approach [16] is based on the modified nodal concept of equation formulation [17] and formally is reminiscent of Ref. [18]. It is equally applicable to time and frequency domain analysis. The following advantages of this approach will be shown in this article: exceptional efficiency in the equation formulation phase of the analysis, low overheads in the equation solution phase, multiple output observation, and catastrophic faults (i.e.  $\pm 100\%$ ) modeling.

Based on this, this article proposes a new method for statistical simulation of linear analog circuits using a concurrent fault simulation. The method performs Monte Carlo analysis in order to achieve appropriate statistical properties of a circuit response. The basic idea is to keep the original structure of a circuit when statistical simulation is performed. Toleranced elements are modeled by added branches in the way that original system of equations is preserved, thus giving an opportunity to use decomposed matrix of the original system. Specific advantages offered by the structure of the faulted circuit system of equations is exploited in order to raise the speed of the computations. According to this idea, a new statistical linear electronic circuit analyzer is developed and implemented. Special efforts were paid to ensure user friendliness. Application of this tool to the analysis of specific high selectivity filters shows its effectiveness and usefulness.

The article is organized as follows: Section 2 describes the method of concurrent fault simulation. The method is first described for one tolerated element, and then extended to a general procedure for introducing perturbations into other lumped network elements in the frequency and time domain. Section 3 is focused on application of this method to statistical circuit simulation. After that, some aspects of the method implementation are given, as well as the simulation results obtained by concurrent statistical electronic linear analog simulation (CONSTELLATION), a simulator developed according to the methods described in this article.

## 2. Concurrent fault simulation in linear electronic circuits

The method of concurrent fault simulation is based on the modified nodal analysis (MNA) [17] approach. This method [16] enables parametric fault analysis, but also makes possible the catastrophic fault simulation. The main advantages of the concurrent fault simulation are in time savings when each defect is considered. These time savings relate to the steps of equations' formulation and system decomposition.

This section describes the basics of the concurrent fault simulation, and further extends the discussion to specific elements in frequency and time domain.

2.1. Basics of the method

Let us examine a conductance  $G$  in a circuit  $N$ , consisting of  $n$  variables (node voltages and branch currents). Consider that  $G$  is perturbed in the way that its change is  $\Delta G = g$ . The representations of the original and perturbed circuit are given in Fig. 1, where a port is created parallel to  $G$ . In this way, a new equation representing the new branch current is added to the original system of equations. The equations representing this branch in the original and perturbed circuit are

$$i_A = 0, \tag{1a}$$

$$g\hat{v}_A - \hat{i}_A = g\hat{v}_k - g\hat{v}_l - \hat{i}_A = 0, \tag{1b}$$

where  $\hat{\phantom{x}}$  denotes a “perturbed” circuit variable, and  $v_k$  and  $v_l$  stand for the  $k$ th and  $l$ th node voltages, respectively.

These equations are used in the MNA circuit equations’ formulation. If the MNA is applied to the original circuit of Fig. 1(a), we obtain

$$\begin{bmatrix} \mathbf{Y} & \mathbf{A} \\ \mathbf{B} & -1 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ i_A \end{bmatrix} = \begin{bmatrix} \mathbf{j} \\ 0 \end{bmatrix}, \tag{2}$$

where  $\mathbf{Y}$  is the MNA matrix of the circuit  $N$ ,  $\mathbf{A}$ , the column vector with all the elements equal to zero except for the  $k$ th and  $l$ th, which are 1 and  $-1$ , respectively,  $\mathbf{B}$ , the row vector containing all zeros,  $\mathbf{v}$ , the vector of unknown circuit variables, and  $\mathbf{j}$ , the vector excitation. Given that  $i_A = 0$ , it does not matter which conductance is perturbed and  $\mathbf{v}$  (in the original circuit) remains unchanged. The same observation will be valid if more than one port is created simultaneously.

The perturbed circuit shown in Fig. 1(b) can be represented by the following system of equations:

$$\begin{bmatrix} \mathbf{Y} & \mathbf{A} \\ \mathbf{D} & -1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{v}} \\ \hat{i}_A \end{bmatrix} = \begin{bmatrix} \mathbf{j} \\ 0 \end{bmatrix}, \tag{3}$$

where  $\mathbf{D} = g\mathbf{A}^T$ . In order to find the voltage and current increments for the changed value of conductance, the following convention is used:

$$\begin{bmatrix} \hat{\mathbf{v}} \\ \hat{i}_A \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ i_A \end{bmatrix} + \begin{bmatrix} \Delta\mathbf{v} \\ \Delta i_A \end{bmatrix}. \tag{4}$$

The next system of equations is obtained when Eq. (4) is substituted into Eq. (3) and the known values transferred to the right-hand side

$$\begin{bmatrix} \mathbf{Y} & \mathbf{A} \\ \mathbf{D} & -1 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{v} \\ \Delta i_A \end{bmatrix} = \begin{bmatrix} \mathbf{j} \\ 0 \end{bmatrix} - \begin{bmatrix} \mathbf{Y} & \mathbf{A} \\ \mathbf{D} & -1 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ i_A \end{bmatrix} = \begin{bmatrix} \mathbf{j} \\ 0 \end{bmatrix} - \begin{bmatrix} \mathbf{Y} & \mathbf{A} \\ \mathbf{B} & -1 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ i_A \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{D} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ i_A \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ -g(v_k - v_l) \end{bmatrix}. \tag{5}$$

This system defines the relations between the voltage and current increments in the modified circuit as a function of excitation defined by the original circuit response.

Looking at Eq. (5), we see that solving this system means computing all node voltage (current) increments, which is not the case when the adjoint network method [8] is applied. Further, the method of concurrent fault simulation does

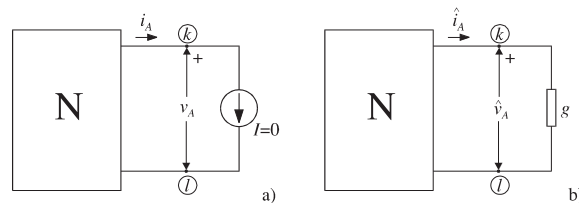


Fig. 1. (a) The original and (b) perturbed circuit.

not impose any limitations on the types and number of excitations in  $N$ , in contrast to the adjoint network method where only one excitation is allowed.

The benefit derived by use of the concurrent fault simulation method is easily seen as only one decomposition of  $\mathbf{Y}$  matrix is necessary. Once decomposed,  $\mathbf{Y}$  matrix can be further used in the process of solving the perturbed circuits. Since  $\mathbf{D}$  and  $\mathbf{A}$  are vectors, the time necessary for factorization of  $\mathbf{D}$  and  $\mathbf{A}$  is small compared to the time consumed by factorization of  $\mathbf{Y}$  matrix. Also, we will see that, in some cases, sparsity of the right-hand side vector in Eq. (5) can be exploited for speeding up a forward substitution [19].

If the number of elements with given tolerance is  $m$ , the structure of the obtained system matrix is shown in Fig. 2. The circuit behavior can be observed in two ways: when faults are inserted one at a time, or by including all element's tolerances at the same time. The difference lies in the dimension of the analyzed system. In the first case, only one additional row and column for each fault will be substituted when that fault is inspected. Such a fault analysis requires one factorization and  $m+1$  forward and back substitutions. In the second case, all additional rows and columns from the Fig. 2 will be added at the same time, increasing the size of the system to  $n + m$ .

The price paid, in the second case, is the additional time needed for forward and back substitutions as the system is expanded. Also, time is consumed by factorization of added parts of the system matrix. Therefore, the effectiveness of the new approach is apparent when the number of tolerated elements observed at a time is smaller than the system matrix dimension. It will perform much faster than the repetitive circuit analysis when the mapping of one or a few parameter tolerances onto the circuit response is of main interest.

2.2. Concurrent fault simulation in frequency domain

If we extend the previous discussion to the frequency domain,  $\mathbf{Y}$ ,  $\mathbf{v}$ ,  $\mathbf{j}$ ,  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{D}$  in Eqs. (2) and (4) become complex:  $\mathbf{Y} = \mathbf{Y}_r + j\mathbf{Y}_i$ ,  $\mathbf{v} = \mathbf{v}_r + j\mathbf{v}_i$ ,  $\mathbf{j} = \mathbf{j}_r + j\mathbf{j}_i$ ,  $\mathbf{A} = \mathbf{A}_r + j\mathbf{A}_i = \mathbf{A}_r + j\mathbf{0}$ ,  $\mathbf{B} = \mathbf{B}_r + j\mathbf{B}_i = \mathbf{0} + j\mathbf{0}$ , and  $\mathbf{D} = \mathbf{D}_r + j\mathbf{D}_i$ . Substituting this in Eq. (2) gives

$$\begin{bmatrix} \mathbf{Y}_r & -\mathbf{Y}_i & \mathbf{A}_r & -\mathbf{A}_i \\ \mathbf{Y}_i & \mathbf{Y}_r & \mathbf{A}_i & \mathbf{A}_r \\ \mathbf{B}_r & -\mathbf{B}_i & -1 & 0 \\ \mathbf{B}_i & \mathbf{B}_r & 0 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{v}_r \\ \mathbf{v}_i \\ i_{Ar} \\ i_{Ai} \end{bmatrix} = \begin{bmatrix} \mathbf{j}_r \\ \mathbf{j}_i \\ 0 \\ 0 \end{bmatrix}, \tag{6}$$

which is describing the original circuit in frequency domain. Substitution in Eq. (5) gives

$$\begin{bmatrix} \mathbf{Y}_r & -\mathbf{Y}_i & \mathbf{A}_r & -\mathbf{A}_i \\ \mathbf{Y}_i & \mathbf{Y}_r & \mathbf{A}_i & \mathbf{A}_r \\ \mathbf{D}_r & -\mathbf{D}_i & -1 & 0 \\ \mathbf{D}_i & \mathbf{D}_r & 0 & -1 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{v}_r \\ \Delta \mathbf{v}_i \\ \Delta i_{Ar} \\ \Delta i_{Ai} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ -g(v_{kr} - v_{lr}) \\ -g(v_{ki} - v_{li}) \end{bmatrix} \tag{7}$$

representing extraction from the perturbed circuit's system of equations. The difference, when frequency domain analysis is considered, is in the dimension of the system which is twice greater than in the DC domain.

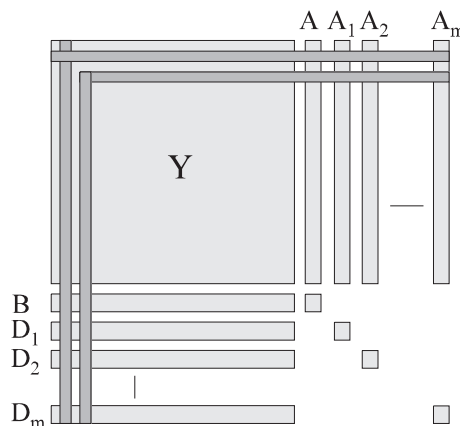


Fig. 2. Structure of the system matrix, where the newly added rows–columns represent particular faults.

Next, fault modeling of some other elements encountered in the frequency domain analysis will be introduced, the capacitance being the first. The perturbed system of equations is formed in the same way as it was for the conductance, by adding a new incremental capacitor branch, shown in Fig. 3. Thus, we have

$$\begin{bmatrix} \mathbf{Y}_r & -\mathbf{Y}_i & \mathbf{A}_r & \mathbf{0} \\ \mathbf{Y}_i & \mathbf{Y}_r & \mathbf{0} & \mathbf{A}_r \\ \mathbf{0} & -\mathbf{D}_i & -1 & 0 \\ \mathbf{D}_i & \mathbf{0} & 0 & -1 \end{bmatrix} \begin{bmatrix} \Delta v_r \\ \Delta v_i \\ \Delta i_{Ar} \\ \Delta i_{Ai} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ +(\omega \Delta C)(v_{ki} - v_{li}) \\ -(\omega \Delta C)(v_{kr} - v_{lr}) \end{bmatrix}, \quad (8)$$

where  $\mathbf{D}_i = [0 \cdots \omega \Delta C \cdots -\omega \Delta C \cdots 0]$ , with non-zero elements in  $k$ th and  $l$ th positions.

When inductance is considered, things are not so simple, for it belongs to the group of elements described by voltage equations. A perturbation in one of these element's values implies addition of an incremental series element, as opposed to the idea described above, and introduces a new node in the circuit. This can be avoided by the introduction of a gyrator as shown in Fig. 4(a) represents the inductance in the original circuit Fig. 4(b) represents the equivalent (original) circuit using a gyrator and a capacitance, where  $C$  is an equivalent to the inductance. The MNA formulation for the transformed original circuit gives

$$i_x = (v_k - v_l)/(j\omega L), \quad (9)$$

where  $v_k$  and  $v_l$  are the node voltages at terminals of the inductor, while  $i_x$  is the inductor branch current. The contribution of the gyrator, i.e. the new equation, belongs to the original circuit which is described by the system matrix  $\mathbf{Y}$ .

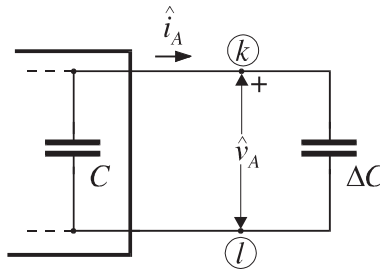
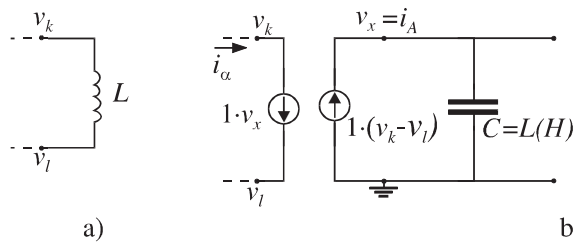
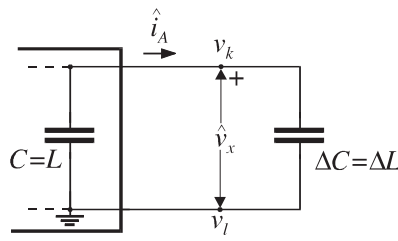


Fig. 3. Incremental capacity branch.



a)

b)



c)

Fig. 4. Tolerancing an inductor: (a) inductance, (b) inductance modeled by gyrator, (c) representation of incremented inductance in modified circuit.

In order to introduce perturbation,  $\Delta L$ , a new port is created in parallel to the equivalent capacitor, as shown in Fig. 4(c). In this way, the series connection discussed above is avoided. The equation representing the perturbed circuit is

$$j\omega \Delta C \hat{v}_x - \hat{i}_A = 0, \quad (10)$$

which is structurally identical to that used to represent conductance. From Eq. (10), considering that  $\hat{v}_x = \hat{i}_x$ ,  $\Delta C = \Delta L$ , and  $\hat{i}_A = \hat{v}_A$ , where  $\hat{v}_A$  is the voltage increment on  $\Delta L$  in the equivalent circuit, we can write

$$j\omega \Delta L \hat{i}_x - \hat{v}_A = 0. \quad (11)$$

For the perturbed circuit system of equations, when an incremental inductance change in the frequency domain is applied, will have the next structure

$$\begin{bmatrix} \mathbf{Y}_r & -\mathbf{Y}_i & \mathbf{A}_r & \mathbf{0} \\ \mathbf{Y}_i & \mathbf{Y}_r & \mathbf{0} & \mathbf{A}_r \\ \mathbf{0} & -\mathbf{D}_i & -1 & 0 \\ \mathbf{D}_i & \mathbf{0} & 0 & -1 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{v}_r \\ \Delta \mathbf{v}_i \\ \Delta i_{Ar} \\ \Delta i_{Ai} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ +\omega \Delta L i_{xi} \\ -\omega \Delta L i_{xr} \end{bmatrix}, \quad (12)$$

where  $\mathbf{D}_i = [0 \cdots \omega \Delta L \cdots -\omega \Delta L \cdots 0]$ , with non-zero elements in  $k$ th and  $l$ th positions.

### 2.3. Concurrent fault simulation in time domain

For the time domain analysis, dynamic elements are of interest, in particular, inductors and capacitors. If a gyrator is used to formulate the equations for an inductive branch, we only need to consider the case of a capacitive branch. Time domain analysis presumes that the values of voltages and currents in a circuit are dependent on previous time instances. Considering the backward Euler integration rule, a capacitive branch in the time domain is described as

$$i_C = C \frac{dv_C}{dt}, \quad (13)$$

which is discretized as

$$i_C^{n+1} = G_C^n v_C^{n+1} + i_{Cs}^n, \quad (14)$$

where the superscripts  $n$  and  $n+1$  denote the previous and present time points, respectively,  $G_C^n = C/h^n$ ,  $i_{Cs}^n = -Cv_C^n/h^n$ , and  $h^n$  is the time step:  $h^n = t^{n+1} - t^n$ . Eq. (14) expresses the fact that the capacitive branch in the time domain introduces conductance into the  $\mathbf{Y}$  matrix and current source into the right-hand side vector. The system of equations equivalent to Eq. (2) is now

$$\begin{bmatrix} \mathbf{Y}^n & \mathbf{A} \\ \mathbf{B} & -1 \end{bmatrix} \begin{bmatrix} \mathbf{v}^{n+1} \\ i_A \end{bmatrix} = \begin{bmatrix} \mathbf{j}^{n+1} - \hat{\mathbf{j}}_{Cs}^n \\ 0 \end{bmatrix}, \quad (15)$$

where  $(j_{Cs}^k)_k$ , the  $k$ th element of vector  $\hat{\mathbf{j}}_{Cs}^n$ , is equal to the sum of all  $i_{Cs}^n$  leaving the  $k$ th node in  $N$ .  $\mathbf{j}^{n+1}$  is the value of the vector of excitations  $\mathbf{j}$  at  $t = t^{n+1}$ .

If the same time step,  $h^n$ , is used for both the original and the perturbed circuit, the introduction of a perturbation will give the same  $\mathbf{Y}^n$ , but different  $\hat{\mathbf{j}}_{Cs}^n$ . For the perturbed network, the following system of equations is valid:

$$\begin{bmatrix} \mathbf{Y}^n & \mathbf{A} \\ \mathbf{D}^n & -1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{v}}^{n+1} \\ \hat{i}_A^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{j}^{n+1} - \hat{\mathbf{j}}_{Cs}^n \\ -\hat{i}_{\Delta C}^n \end{bmatrix}, \quad (16)$$

where  $\mathbf{D}^n = G_{\Delta C}^n \mathbf{A}^T$ ,  $G_{\Delta C}^n = (\Delta C)/h^n$ , and  $\hat{i}_{\Delta C}^n = -G_{\Delta C}^n (\hat{v}_k^n - \hat{v}_l^n)$  is the current source dependent on the capacitance voltage from the previous time point.

If we introduce  $\hat{\mathbf{v}}^{n+1} = \mathbf{v}^{n+1} + \Delta \mathbf{v}^{n+1}$ , and  $\hat{i}_A^{n+1} = i_A^{n+1} + \Delta i_A^{n+1} = 0 + \Delta i_A^{n+1}$ , the following system of equations in time domain is obtained:

$$\begin{bmatrix} \mathbf{Y}^n & \mathbf{A} \\ \mathbf{D}^n & -1 \end{bmatrix} \begin{bmatrix} \Delta \hat{\mathbf{v}}^{n+1} \\ \Delta \hat{i}_A^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{j}_{Cs}^n - \hat{\mathbf{j}}_{Cs}^n \\ -\hat{i}_{\Delta C}^n - G_{\Delta C}^n (\hat{v}_k^{n+1} - \hat{v}_l^{n+1}) \end{bmatrix}. \quad (17)$$

The choice of the discretization step when applying this method is a particularly important task. In order for the simulation to be correct, it is essential that the time step is of the same order of magnitude as the smallest time constant active in the circuit. On the contrary, simulation time is usually much greater than the smallest time constant. That is, in

a general case, the reason for use of a variable time step determined by the local truncation error in every time instant [20]. This would be done in a Monte Carlo simulator applying a repetitive circuit analysis. When concurrent fault simulation is performed, however, in order that Eq. (17) is to be applicable, all the circuits i.e. original and all the perturbed replicas, must be simulated with the same time advancing scheme. This means that in every time step, the slowest simulation (replica) will dictate the time advancement. Nevertheless, we are speaking about linear circuits whose equations are not as stiff as for the nonlinear ones and the issue of time step estimation is not as crucial. In addition, we suppose that a fault influences only a small number of time constants within the circuit so that the difference between the original and the perturbed circuit, from the time step point of view, is not large. This is why the CONSTELLATION simulator uses a fixed time step for time domain analysis, whereas an algorithm for effective time step evaluation in every instant is under development.

2.4. Modeling various linear elements

In Section 2.3, we have seen how dynamic elements are modeled. Here, we will discuss some other linear elements such as controlled sources. The goal is to develop “stamps” for elements, enabling the automation of an MNA formulation.

We will consider modeling of a voltage controlled voltage source (VCVS). It is one of the elements described by the voltage equation, as described above. In order to model this component in a perturbed circuit without loss of the circuit’s structure, a gyrator should be employed. Fig. 5(b) presents the idea, where an increment current source branch is added to the circuit on the right side of the gyrator that converts current to voltage and vice versa.  $\mu$  is an equivalent transconductance in Fig. 5(b). The MNA formulation is

$$v_k - v_l = \mu V = \mu(v_a - v_b), \tag{18}$$

where  $v_k$  and  $v_l$  are the node voltages at the terminals of the VCVS, while  $V = v_a - v_b$  is the controlling voltage. The procedure of the perturbation introduction is similar as for the inductance. Equation appended in the perturbed circuit is

$$\Delta\mu(\hat{v}_k - \hat{v}_l) - \hat{i}_A = 0. \tag{19}$$

In the same manner, models of other linear elements were elaborated, and their contribution implemented in the simulator CONSTELLATION.

3. Statistical simulator development

The method of concurrent fault simulation can be applied in a statistical circuit simulator. This means that repetitive simulations of toleranced circuits must be performed. In order to provide a valuable demonstrate strength of the new method power, a new simulator, CONSTELLATION, has been developed [21]. It performs frequency and time domain analysis, and is applicable only to linear circuits. It uses Monte Carlo method to achieve multiple circuit simulation results. The element can be given various kinds of distributions, such as uniform, Gaussian, and others including piecewise linear descriptions. One can specify the number of simulations that are going to be taken per each frequency/ time point, as well as the way of analyzing variances of the elements separately, or at the same time.

It employs modern algorithms for manipulating sparse matrices and of solving system equations [22]. Although its purpose was to implement the method of concurrent fault simulation into statistical simulator, there has been an opportunity left to statistically simulate circuits using classical method of repetitive simulations (option `checking`

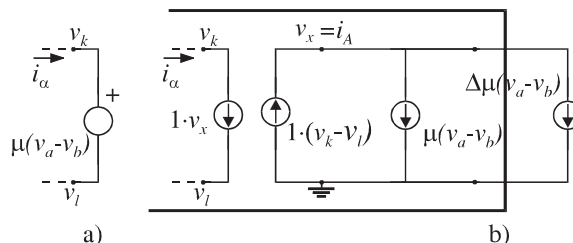


Fig. 5. Tolerancing a voltage controlled voltage source: (a) VCVS symbol, (b) representation with inserted perturbation.

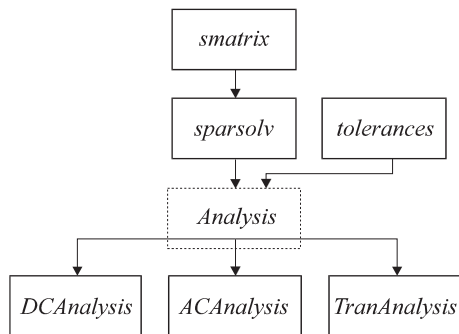


Fig. 6. Class hierarchy in CONSTELLATION.

included), so the simulation times taken by each method could be compared. Keeping in mind that faults can be observed individually, one by one, or all at the same time, there has been an option left (`individual`) of determining the method of fault analysis.

The topology of a circuit is described in a SPICE-like manner, and is retrieved by the simulator from an input file. The statistical-distribution parameter values of elements in a circuit are given in percents. Simulation parameters can be changed through the input mask (windows application). An appropriate graphical postprocessor is incorporated in the simulator for the purpose of observing simulation results. It enables graphical representations of output data, from the raw output characteristics to the graphics of the elaborated statistical properties.

The programming environment chosen for this purpose is Borland c++ Builder [23], a fast rapid application development (RAD) tool. It enables fast design of the simulator interface, taking care of the project speed, with great possibilities even on system level, thus giving an opportunity of high optimization.

Class organization of the simulator is given in Fig. 6. Root class is `smatrix`, dealing with sparse matrices. Its ancestor, class `sparsolv` carries out solving a system of equations. The central class `Analysis` is the base class for all types of analysis (DC, AC, Tran). It contains declarations of the functions used for matrix fill-in, factorization, and solving system of equations. Some of these functions, such as the one that solves system of equations have the same implementation in all analysis domains, and those are defined in this class. Others, such as the function for matrix fill-in, have various implementations, depending on the type of analysis. Those are virtually declared in the `Analysis` class. A part of the `Analysis` class declaration is presented in the following code:

```

class Analysis: public sparsolv, public tolerances {
protected:
    sparsolv Y; // original circuit system of equation entity
    elements* el; // circuit topology pointer
    int toli; // index of currently probed tolerance
    . . .
public:
    Analysis (elements& elem, int is_it_AC);
    ~Analysis();
    // methods for resolving the original circuit system of eqns.
    virtual void FillYMatrix(elements& el, sparsolv& Y); // fill-ins
    void PrepareYMatrix(sparsolv& Y); // prepare matr. for solving
    void SolveYMatrix(sparsolv& Y); // solving system of equations
    // methods for resolving the perturbed circuit system of eqns.
    virtual void FillEntireMatrix(elements& el, int sim_number);
    void PrepareEntireMatrix();
    void SolveEntireMatrix(int sim_number);
    . . .
    virtual void DoAnalysis() { };
};

```



Successor classes of the Analysis class are DCAnalysis, ACAnalysis, and TranAnalysis. These classes inherit some common functions from Analysis class, and introduce a definition of analysis specific functions declared in class Analysis, as it is shown in the next part of the class ACAnalysis declaration

```
class ACAnalysis: public Analysis {
    double frequency; // current frequency of analysis
    void EvaluateNewFrequency(int currpoint); // calc. of new freq.
    . . .
public:
    ACAnalysis (elements&);
    // fill-in original circuit system of equations
    void FillyMatrix(elements&, sparsolv&);
    // fill-in perturbed circuit system of equations
    void FillEntireMatrix(elements& el, int sim_number);
    . . .
    void DoAnalysis();
};
```

Finally, tolerances class performs an automated random-number generation, used throughout the repetitive simulations. It should be stated here that these random numbers are determined at the first frequency/time point and stored in memory. During later simulations, these values are recalled in order to achieve correct results.

The global algorithm for the frequency domain simulation is shown in Fig. 7. Three main loops can be sighted: frequency loop, individual tolerance loop, and repetitive simulations loop. The second loop would not exist if all the given faults were analyzed at the same time. Matrix fill-in and factorization for the original circuit are performed once for each frequency point, whereas the rest of the perturbed matrix is filled later, inside the inner loop. At the end of one loop pass, the result of the perturbed circuit has to be added to the ones obtained for the original circuit, in order to obtain voltages and currents, and not their increments. Time domain algorithm would be similar, but with two main differences: the zero time DC analysis must be performed in order to establish the boundary conditions, and previous time point solution has to be memorized for it is used in the current time point simulation.

Some of the method application specifics are as follows: Repetitive solving of the system of equations to obtain statistical characteristics includes LU factorizations. In the previous section, we stated that  $\mathbf{Y}$  matrix needs not to be factorized. Now we can add that for the same structure of the matrix shown in Fig. 2, i.e. when only the elements parameters are changed, we do not need to factorize matrix  $\mathbf{A}$  either. Also, if we divide the added row by the value of added conductance  $g$  from Fig. 1(b), we obtain

$$\hat{v}_A - (1/\Delta G)\hat{i}_A = \hat{v}_k - \hat{v}_l - (1/\Delta G)\hat{i}_A = 0, \quad (20)$$

which means that matrix  $\mathbf{D}$  contains constant values, so the only part of the entire matrix that should be decomposed is the lower-right part of the matrix, containing only the diagonal elements. In the same way as for conductance, new equations introduced by other linear elements can be transformed to have the same form as Eq. (20). Fig. 8(a) shows the order of matrix decomposition for the first statistical simulation of each fault at one frequency/time point. For the first simulation of the first fault at that point, matrix  $\mathbf{Y}$  should also be factorized. In Fig. 8(b), we can see that factorization

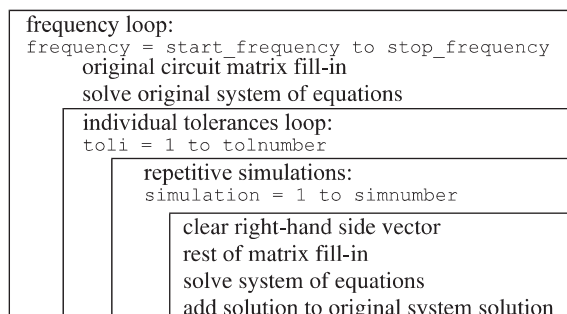


Fig. 7. Global algorithm for frequency domain analysis.

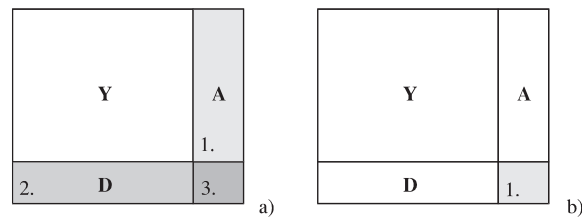


Fig. 8. Parts of the circuit matrix requiring factorization: (a) for the first simulation of the fault, (b) for each next simulation of that fault.

for each statistical simulation of the same fault needs much smaller computational effort than the factorization of the whole matrix. Therefore, time spent in this phase of simulation is significantly reduced.

In the phase of forward and back substitutions, computational effort is slightly greater in the case of the new method. It particularly depends on the number of branches added to the circuit: higher the number of elements with tolerances that are observed at the same time, greater the difference in simulation times. However, if we observe faults in a circuit one at a time, only one factorization of  $Y$  matrix per frequency/time point is going to be necessary for all those faults.

Another method to further reduce frequency domain simulation time is, as mentioned earlier, use of the sparsity of the right-hand side vector for the perturbed circuit. Forward substitution is performed by the next equation

$$b_i = \begin{cases} b_i & \text{for } i = 1 \\ \left( b_i - \sum_{j=1}^{i-1} l_{ij} b_j \right) / l_{ii} & \text{for } = 2, \dots, \text{system\_size} \end{cases} \quad (21)$$

where  $b_i$  is the  $i$ th element of the right-hand side vector, and  $l_{ij}$  is a member of  $L$  submatrix of the system. Considering Eq. (7), we can reach a conclusion that the first  $2n$  elements of the right-hand side vector are equal to zero, where  $n$  is the number of voltages and currents in the original circuit. Clearly, the first  $2n$  forward substitutions need not be performed, i.e. forward substitutions start from  $(2n + 1)$ th element of the right-hand side vector. Keeping in mind that the factorization process is already speeded-up in repetitive simulations using the method of concurrent simulations, the additional reduction of forward substitution time is not negligible. We observed that simulation time savings up to 20% may be reached if this property of the concurrent fault simulation method is exploited. We can benefit from this, only when performing frequency domain analysis, for the reason that in time domain analysis, dynamic elements generate unpredictable fill-ins in the right-hand side vector. Therefore, the first  $n$  elements of the right-hand side vector in the system of equations describing the perturbed circuit do not have to be zeros.

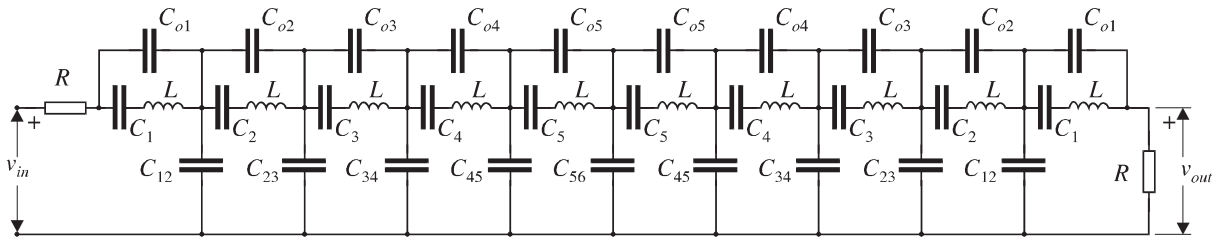
The implementation of the concurrent fault simulation method, and the classical method of repetitive simulations in the statistical simulator enabled a comparison of the simulation times. The results are shown in Section 4.

#### 4. Results and discussion

Due to the fact that Monte Carlo analysis produces a huge amount of data, a designer confronted with the complete results might have difficulty in identifying and extracting the significant values. Therefore, the task of the postprocessor is to organize these data into brief comprehensible forms. This means that a summary of test statistics should be presented to the designer. This includes the yield percentage, average value, standard deviation, maximum and minimum values.

The simulator has been tested on several circuits [24], one of them being the 10-stage bandpass crystal filter shown in Fig. 9 [25]. Nominal values of the parameters are given at the bottom of the figure. This filter is designed to have 1 MHz central frequency and 1 kHz passband. This example was chosen due to its extremely high sensitivity to circuit parameter variations. In addition, representing, in fact, a real crystal filter, sets of the filter elements may be considered as crystal's model and looked upon as separately varied. This gives rise to our method which is intended to be applied for a "small" number of toleranced elements.

We shall consider the frequency domain analysis, and observe first four inductances in the circuit (denoted as  $L_1$ ,  $L_2$ ,  $L_3$  and  $L_4$ ). Let the values of these elements have Gaussian distribution with standard deviations of 0.005%. If we observe changes of each inductance separately (option *individual* included), we obtain particular influences on frequency



$R = 5,878 \text{ k}\Omega$	$L = 1 \text{ H}$	
$C_1 = 0,025342612 \text{ pF}$	$C_{o1} = 10,189 \text{ pF}$	$C_{12} = 15,5 \text{ pF}$
$C_2 = 0,025358482 \text{ pF}$	$C_{o2} = 8,140 \text{ pF}$	$C_{23} = 14,816 \text{ pF}$
$C_3 = 0,025358513 \text{ pF}$	$C_{o3} = 8,137 \text{ pF}$	$C_{34} = 15,463 \text{ pF}$
$C_4 = 0,025357569 \text{ pF}$	$C_{o4} = 8,235 \text{ pF}$	$C_{45} = 15,898 \text{ pF}$
$C_5 = 0,025357089 \text{ pF}$	$C_{o5} = 8,286 \text{ pF}$	$C_{56} = 16,035 \text{ pF}$

Fig. 9. Ten-stage crystal band-pass filter.

characteristic. Results of the simulation shown as families of amplitude characteristics when the value of  $L_1$  is toleranced are shown in Fig. 10(a).

The simulation included 1000 analyses of the circuit for each of the mentioned inductances for the frequencies between 999.00 and 1000.54 kHz using 200 frequency points. Altogether 200,000 circuit analyses were performed. For that simulation, CONSTELLATION consumed 547.7 s if the concurrent fault simulation method was used, and 2637.4 s when the method of repetitive simulations was applied. Part of the simulation results is shown in Fig. 11(a), where one can see that the number of frequency points used is not appropriate because the effects of the variations of elements are not scanned precisely, especially by the upper bound of the passband, on frequencies near 1000.5 kHz. Fig. 11(b) shows frequency characteristics for the frequencies in the interval from 1000.4 to 1000.5 kHz, obtained when one uses 1000 frequency points. Therefore, 1,000,000 analyses were performed for the whole Monte Carlo simulation. Comparing these two figures, one concludes that small number of samples leads to an obscure characteristic, and to uncertainties concerning the real properties of the observed filter. For that reason, simulation in extended number of frequency points is necessary, which consequently increases simulation time, taking 2740.2 s with the concurrent fault simulation method, and 13,198.5 s using the classical method of repetitive simulations. Now, we see that the use of the concurrent fault simulation method reduces the simulation time nearly five times, and that the savings in the simulation time are significant.

If we use this method to observe the variations of elements simultaneously, the perturbed system of equations grows bigger, increasing simulation time. To obtain results in the case when all four inductances are toleranced at the same time, concurrent fault simulation method consumed 2614.2 s, whereas the repetitive simulations method needed 3350.7 s. Obviously, the difference is significantly decreased, and will get even smaller if the number of toleranced elements is increased.

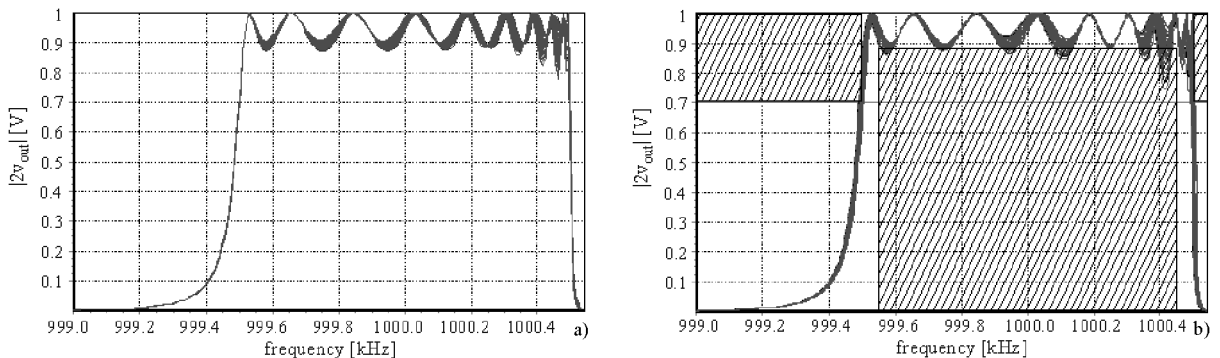


Fig. 10. Family of amplitude characteristics with toleranced inductance: (a)  $L_1$ , (b)  $L_4$ . Yield constraints are depicted.

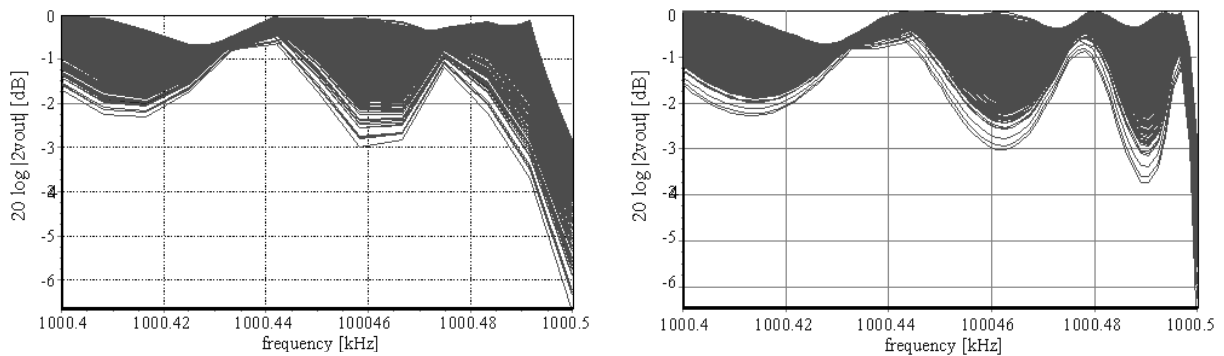


Fig. 11. Part of the amplitude characteristics response, obtained with: (a) 200, (b) 1000 sample points.

When the yield estimation was of interest, we were looking for the circuits obeying the requirement for the passband width to be exactly 1 kHz, between 999.5 and 1000.5 kHz. In the simulated example, when  $L_4$  was tolerated, the lower cutoff frequency condition was satisfied by 70.6% of the circuits, while the upper cutoff frequency condition was satisfied by 71.8% of the circuits. Both the conditions were satisfied by 51.5% of the circuits which means that even this small standard deviations of observed elements (0.005%) significantly influences the filter performances. The yield becomes much smaller when the passband insertion loss of the filter is limited too. If, for example, in addition to the previous requirements, the maximum insertion loss is supposed to be 1 dB, the yield value falls down to 23.8%. The yield constraints discussed here are depicted in Fig. 10(b). Standard deviation as a function of frequency of the output voltage is given in Fig. 12. In that simulation, all four inductances were tolerated simultaneously.

The full power of the concurrent fault simulation method application is recognized when complex circuits are statistically simulated. One such circuit is the two-stage active lowpass filter [26] shown in Fig. 13. No time domain tolerance analysis of such circuit may be found in the literature, as to our knowledge. Values of the elements are also noted in the same figure. The method of concurrent fault simulation carried out 1000 simulations of this circuit in 378.5 s, whereas the classical method of repetitive simulations required 1591.52 s. The simulation results in the form of the amplitude characteristics are given in Fig. 14, when tolerances of the resistor  $R_8$  and capacitor  $C_2$  are 5%, the values of which have uniform distribution. The same figure contains response requirements (graphically presented), which are to be satisfied by the circuit. These requirements are summarized in Table 1.

The ratio of the number of characteristics satisfying these conditions and the complete number of characteristics, versus frequency is given in Fig. 15. The overall yield of the circuit for the above requirements is 52.4%. One can notice that the most sensible frequency range is between 1 and 4 kHz.

The step response of the same circuit will be used in order to demonstrate the effectiveness of the method. The rise time of the response is defined as the time needed for the output to go from 10% to 90% of the nominal value. The following requirements were imposed: the output voltage should reach 0.094 V before 30  $\mu$ s, and 0.846 V before 0.11 ms, meaning that the maximum allowable rise time was 80  $\mu$ s. In addition, the value of the overshoot should not be

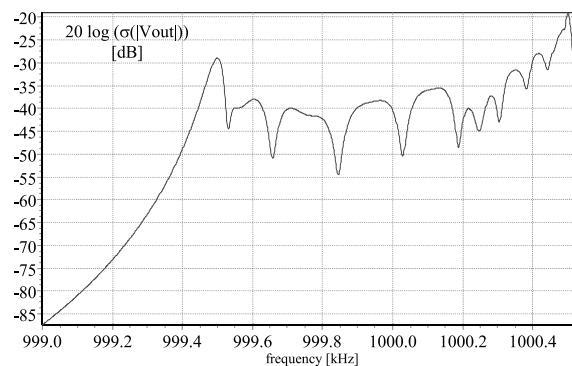


Fig. 12. Frequency dependence of the output voltage standard deviation.

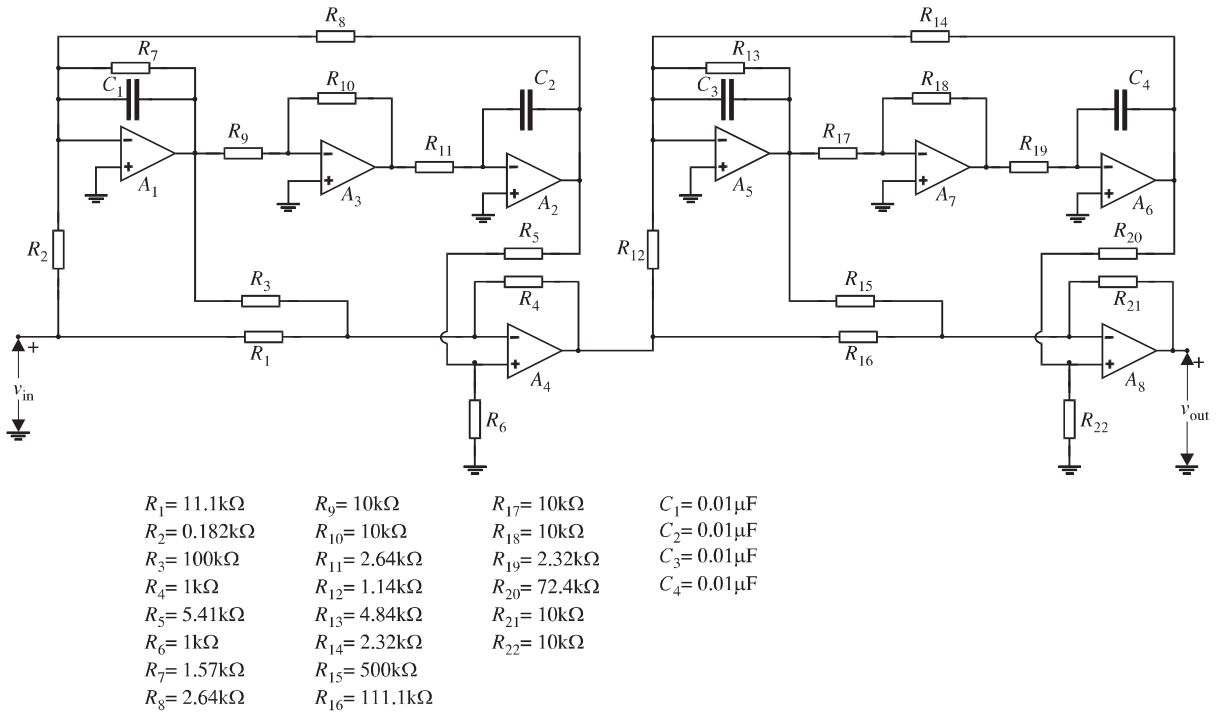


Fig. 13. Two-stage low-pass filter.

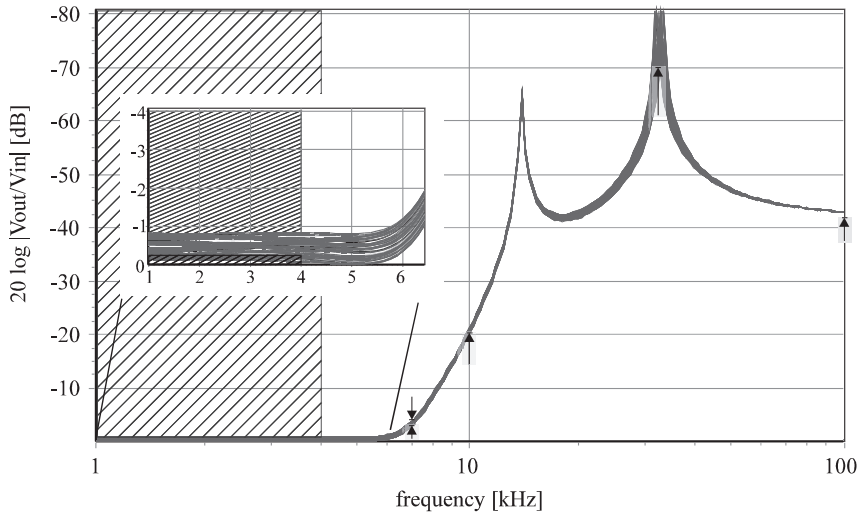


Fig. 14. Frequency dependence of the filter response.

Table 1

Frequency (kHz)	Upper bound of the gain (dB)	Lower bound of the gain (dB)
1–4	-0.25	-0.85
7	-3	-4
10	-20.4	-
32	-70	-
100	-42	-

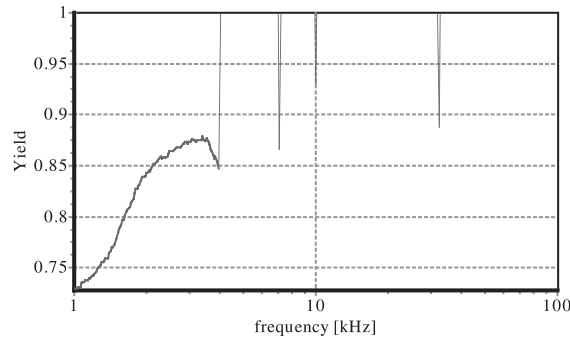


Fig. 15. Ratio of the number of satisfactory circuits and the entire number of the circuits.

greater than 20%, i.e. the maximum value of the output have to be less than 1.128 V. The circuit depicted in Fig. 13 was simulated with standard deviations for the resistor  $R_1$  of 10%, and capacitance  $C_2$  of 20%. Simulation was performed for one toleranced element at a time. One thousand analyses were performed by CONSTELLATION, taking 317.63 s for the concurrent fault simulation method, and 1082.97 s when the classical method of repetitive simulations was applied. Families of the responses when capacitance  $C_2$  is toleranced are shown in Fig. 16, along with the described requirements. The time dependence of the yield for the responses from Fig. 16 is given in Fig. 17. The overall yield, i.e. the ratio of the number of satisfactory responses, no matter the time versus the total number of responses, occurs to be 73.8%.

Having all these examples in mind, one may consider the applicability and computational efficiency of the method introduced here. First, there is no doubt that the extreme simplicity in the circuit equation formulation phase may be attributed to this method. Neither the difference form of Tellegen’s theorem nor the Householder transformations are needed. The second and equally important issue is related to the computational complexity in the circuit analysis part. To get a picture related to this aspect, the efforts for matrix factorization for the new method and the classical (analysis repetition) method will be compared. To do this, we will assume that the number of long operations (multiplications and divisions) needed for matrix factorization is proportional to  $n^k$ , say  $\alpha n^k$ , where  $\alpha$  and  $k$  are constant. The value of  $k$  is dependent on how the factorization is programmed, on whether the matrix is sparse and, in that context, how large  $n$  is. Namely, for a small  $n$ , no advantages over the sparsity can be taken.

If sparsity is not exploited,  $k = 3$  is obtained. For very large and very sparse systems, one claims that  $k = 1.2$  may be achieved [1]. In the kind of circuits we are analyzing here, one may expect  $2 < k < 2.5$ . If so, the classical method will need  $2(\alpha n^k)$  long operations for factorization, whereas our method needs  $\alpha(n+m)^k$ . According to this, our method will be preferable if  $2n^k > (n+m)^k$ . From the matrix factorization point of view these two methods will be equally complex if  $m = (2^{1/k} - 1)n$ . For  $k = 2$  and  $n = 20$ , this results in  $m = 8.2$ . This is almost valid for comparison of the overall

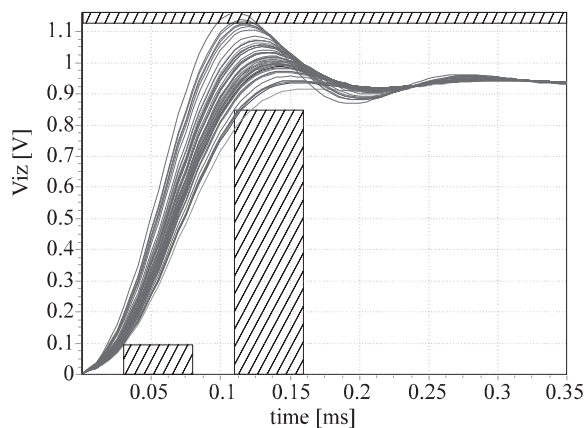


Fig. 16. Step response of the filter. Yield constraints are denoted.

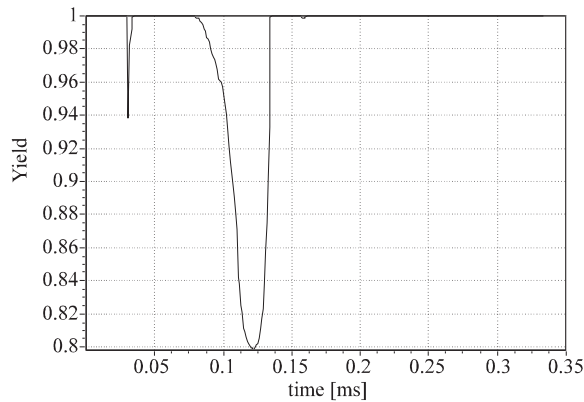


Fig. 17. Time domain yield.

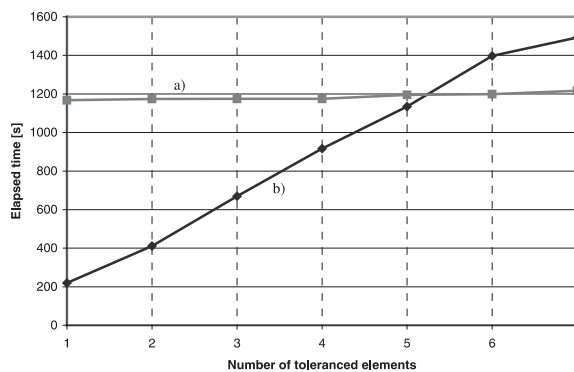


Fig. 18. Elapsed time versus number of tolerated elements: (a) repetitive and (b) concurrent simulation in the frequency domain of the circuit of Fig. 9.

complexity as shown in Fig. 18. Here, (a) represents repetitive (one for original and second for the perturbed circuit) whereas (b) represents concurrent simulation elapsed time as a function of the number of tolerated elements, for the example of Fig. 9. One may observe the linear dependence of the elapsed time on the number of tolerated elements for the new method which is favorably compared with the  $m^3$  curve obtained by application of the Householder formula [13]. The linearity of this curve may be estimated by the following: The difference of the number of long operations during factorization may be approximated by  $\alpha[(2^{1/k}n)^k - (m+n)^k] = \alpha[2^{1/k}n - n - m][2^{1/k}n + n + m]$  for  $k = 2$ . If  $(2^{1/k} + 1)n \gg m$ , then this difference may be approximated by  $\alpha[(2^{1/2} - 1)n - m](2^{1/2} + 1)n$  which is linear with respect to  $m$ .

## 5. Conclusions

A new method for statistical simulation of linear electronic circuits was presented, based on the concurrent fault simulation method. The new method is more effective than the known ones, for it maintains the circuit structure and, consequently, reduces the number of complete circuit matrix factorization down to one per frequency/time point for all faults. In this way, simulation speed is increased.

As a means for evaluating the usefulness of the concurrent fault simulation method in statistical simulation, a simulator called CONSTELLATION was developed. It uses a fast techniques for circuit analysis, implementing the concurrent fault simulation method.

A number of examples were executed, two of which were discussed in this article. Significant advantages of the statistical simulation based on the concurrent fault simulation method compared to the classical method of repetitive

simulations were discovered. This stands equally for the time domain statistical analysis too. This should be especially recognized when time domain optimization methods are to be considered.

## References

- [1] Brayton RK, Spence R. Sensitivity and optimization. Amsterdam: Elsevier, 1980.
- [2] Statistical circuit design [special issue]. *Bell System Technical Journal*. April 1971;50(4).
- [3] Antognetti P, Massobrio G. Semiconductor modeling with SPICE. New York: McGraw-Hill, 1988.
- [4] Tahim KS, Spence R. An integrated approach to manufacturing yield estimation and design centering. *Proc European Conf Circuit Theory Design, Lausanne 1978*. p. 573–7.
- [5] Bolt M, Rocci M, Engel J. Realistic statistical worst-case simulation of VLSI circuits. *IEEE Trans Semiconductor Manufacturing* 1991;4:193–8.
- [6] Lin Y, Foo SW. Fast search algorithm for tolerance design. *IEE Proc Circuits Dev Syst* 1998;145(1):19–23.
- [7] Leung K-H, Spence R. Efficient statistical circuit analysis. *Electron Lett* 1974;10(17):360–2.
- [8] Director SW, Rohrer RA. The generalized adjoint network and network sensitivities. *IEEE Trans Circuit Theory* 1969;CT-16(3):318–23.
- [9] Tahim KS, Spence R. A radial exploration approach to manufacturing yield estimation and design centering. *IEEE Trans Circuits Syst* 1979;CAS-26(9):768–74.
- [10] Leung K-H, Spence R. Idealized statistical models for low-cost linear circuit yield analysis. *IEEE Trans Circuits Systems* 1977;CAS-24(2):62–6.
- [11] Tellegen BDH. A general network theorem with applications. *Philips Res Rept* 1952;7:259–69.
- [12] Gadenz RN, Rezaei-Fakhr MG, Temes GC. A method for the computation of large tolerance effects. *IEEE Trans Circuit Theory* 1973;CT-20:704–8.
- [13] Householder AS. A survey of some closed methods for inverting matrices. *SIAM J Appl Math* 1957;5(3):155–69.
- [14] Hou J, Chatterjee A. CONCERT: A concurrent fault simulator for analog circuits. *Proc Fourth IEEE/ACM Int Mixed-Signal Testing Workshop, den Hag, June 1998*. p. 3–8.
- [15] Yang ZR, Zwolinski M. Fast, robust DC and transient fault simulation for nonlinear analogue circuits. *Proc Design Automation Test Europe Conf Exhibition 1999, Munich, Germany, March 1999*. p. 244–8.
- [16] Litovski VB, Zwolinski M, Litovski IV. Large change sensitivity computation in linear electronic circuits. *Proc 18th Int Spring Seminar Semiconductor Hybrid Technol Sozopol, Bulgaria, 1996*. p. 268–74.
- [17] Ho CW, Ruehli AE, Brennan PA. The modified nodal approach to network analysis. *IEEE Trans Circuits Syst* 1975;22:504–9.
- [18] Jou JY, Abraham JA. Fault-tolerant matrix arithmetic and signal processing on highly concurrent computing structures. *IEEE Proc* 1986;74(5):732–41.
- [19] Kundert KS. Sparse matrix techniques. In: Ruehli AE, editor. *Circuit analysis, simulation and design*. Amsterdam: North Holland, 1986.
- [20] Mrčarica Ž, Glozić D, Litovski VB, Maksimović D, Ilić T, Gavrilović D. Alecsis, the Simulator for Circuits and Systems. *User's Manual*. Laboratory for Electronic Design Automation, Faculty of Electronic Engineering, Niš, Yugoslavia. 1998.
- [21] Ilić T. A new method for estimation of linear electronic circuits statistical properties in frequency and time domain. *MS Thesis*. Faculty of Electronic Engineering, Niš, Yugoslavia. 1999 (in Serbian).
- [22] Mrčarica Ž. The new hybrid electronic circuit simulator and programming realization of the analog part of the simulator. *MS Thesis*. Faculty of Electronic Engineering, Niš, Yugoslavia. 1992 (in Serbian).
- [23] Calvert C. *Borland C++ builder unleashed*. New York: Macmillan Computer Publishing, 1997.
- [24] Ilić T, Litovski VB. Computationally efficient parametric yield estimation of linear electronic circuits. *The 22nd Int Conf Microelectronics MIEL99, Niš, Yugoslavia, 1999*, accepted for presentation.
- [25] Humpherys DS. *The analysis, design and synthesis of electrical filters*. New York: Prentice-Hall, 1970.
- [26] Semmelman CL, Walsh ED, Daryanani GT. Linear circuits and statistical design. *The Bell System Technical Journal* 1971;50(4):1149–71.