Computer Simulation of Continuous-Time and Switched Circuits: Limitations of SPICE-Family Programs and Pending Issues

Dalibor BIOLEK[1], *Josef DOBEŠ*[2]

[1] Dept. of EE, University of Defence Brno, Kounicova 65, 612 00 Brno, Czech Republic
[2] Dept. of Radioelectronics, Czech Technical University, Technická 2, 166 27 Praha, Czech Republic

dalibor.biolek@unob.cz, dobes@fel.cvut.cz

Abstract. *The contribution deals with some open problems from the area of Transient and AC analyses of large continuous-time and switched circuits with the focus on switched capacitor circuits and switched DC-DC converters. The discussed weak points of SPICE-like and SPICE-compatible simulation programs consist in the absence of finding effectively the steady-state, in an inefficient integration algorithm for the transient analysis of switching phenomena, and in the absence of a direct AC analysis of switched circuits. The paper also describes some new approaches, which could be helpful in overcoming the above limitations.*

Keywords

SPICE, modeling, computer simulation, steady state, switched circuits, Newton-Raphson method, Markowitz criterion, LU factorization, generalized transfer functions.

1. Introduction

Today's well known circuit analysis programs, based on the SPICE2, SPICE3, or PSPICE standards, find application in the simulation of a large scale of phenomena in electrical and other systems. One of the most popular products both in the academic institutions and the industry is OrCADPSpice [1]. Probably the most widespread SPICE3 –based program is HSPICE [2].

The computational core of such programs has practically remained unchanged for years. The basic analyses are available: OP, DC, AC, TRANS, NOISE, FOUR, as well as a set of augmented analyses SENS, TF, and MC. In addition, distortion and pole-zero analyses are available in SPICE3.

Most of these analyses are based on the algorithm of finding DC operating points. That is why in the literature great attention is paid to iterative methods of solving a large set of nonlinear algebraic equations. However, the results of such research are reflected in the internal algorithms of commercial simulating programs only marginally, according to the rule "it is not necessary to change a standard that is working". The well-known unpleasant consequences for program users are error messages due to convergence problems when analyzing some kinds of circuits, as well as cumbersome procedures of avoiding these problems by modifying global settings or already compiled circuit models.

In general, the nonlinear circuit has several DC operating points. Beside the trivial cases of flip-flop circuits, there exist a number of concrete demonstrations of this fact. For example, the output voltage of the Brokaw band-gap reference can settle on two different values, depending on the operation of starting circuitry [3]. As shown in [4], at least one half of all existing operating points are unstable. In other words, such operating points are not sustainable in real circuits. Nowadays, we know algorithms for finding all existing DC operating points [5] as well as algorithms for testing their so-called potential stability [6]. However, they are not implemented in any commercial simulating program. For default settings, SPICE finds maximally one DC operating point. It is rather paradoxical that if 3 operating points exist in a circuit, 2 of them stable and one unstable, SPICE will usually find the unstable one. The user can use the trial and error method of finding other operating points, utilizing the .NODESET command.

Also, the set of built-in mathematical functions for behavioral modeling is not overly extended. A more pronounced progress is associated with passing from PSPICE/AD version 10.5 to version 15.7 with several new mathematical functions, for example, inverse hyperbolic goniometric functions, functions zero, one, and the quite important functions ceil, floor, and intq, which can be used for modeling nonlinear effects, associated for example with quantization, etc. The tools of behavioral modeling are also extended by the new functions delta, time, state, and break [7].

No progress has been achieved in the area of controlling the simulation by using command files. This fact is limiting for such simulation tasks which require subsequent runs of different types of analyses, when the results of one analysis serve as input or controlling data for the subsequent analysis.

An unsatisfactory situation persists in the question of finding periodical steady states. In addition to the common possibility of repeatedly utilized .SAVEBIAS and .LOADBIAS commands (with manually editing the data files), the user can only choose sufficiently large simulation times and modification of this time in the PROBE environment.

The principle of internal algorithms of solving algebraic-differential equations, used in transient analysis, together with the absence of algorithms of finding steady states, complicates the simulation of phenomena in analog circuits with a large spread of time constants, the investigations of internal operation in oscillators with high-Q subcircuits, in RF circuits such as mixers or modulators, and in circuits containing analog switches, where switched capacitor filters, switched DC-DC converters, and sigma-delta modulators are typical representatives. While analyzing such circuits, SPICE has to choose a sufficiently small time step, taking into account the fastest phenomena in the circuit. This strategy often leads to unacceptably long computational times together with an accumulation of rounding errors. Such a method of analysis is not effective, especially from the viewpoint of the user, who is only interested in the low-frequency steady-state envelope of the signal being analyzed.

Results of the above simulations are very sensitive to the quality of algorithms of transient analysis. The user can check this, for instance, on the example of starting oscillations of quartz oscillator from zero initial state. If we model this problem identically, i.e. by the identical SPICE netlists, in PSPICE/AD v. 15.7 and in Micro-Cap 9 [8], we get significantly different results. A more instructive test is represented by programming this task only in PSPICE/AD, but under two different options in global settings, namely .OPTIONS SOLVER=0 and .OPTIONS SOLVER=1. The following brief note about the SOLVER option is mentioned in the reference [7]:

PSPICE now contains two solution algorithms for simulation. Solver 1 increases the simulation speed over Solver 0, particularly for larger circuits with substantial runtimes. Solver 1 has slightly better convergence characteristics than Solver 0. Having both algorithms available improves convergence, since there are two different algorithms that can perform the simulation.

For such types of simulation tasks, the user has no big chance of finding which simulation result is closer to the correct behavior of the mathematical model of the circuit. However, this model is absolutely deterministic, or, to put it succinctly, there exists only one correct solution.

The conventional SPICE analysis can lead to unacceptable errors during the analysis of switched networks [9]. The relevant source of errors is represented by incorrect identification of time instants, when switches change their states. This is especially evident for so-called circuits with internal switching, in which the switch states depend on internal voltages and currents in the circuit. The PWM switch inside the DC-DC converter is a typical case: the time instant of closing the switch depends on the relation between the output signal of the amplifier of control deviation and the auxiliary ramp signal. Another problem is called inconsistent initial conditions [10] of circuit models, containing idealized switches, when the switch can, for instance, make a short connection between two nodes, whose voltages are defined by two grounded capacitors. At this moment – as a consequence of idealized modeling of the switch by zero on-resistance –Kirchhoff's voltage law is formally violated. It implies the current in the form of the Dirac impulse. Conventional numerical methods of transient analysis in SPICE cannot deal with such conditions.

In another annoying position are users of the SPICE-family programs who need to analyze small-signal frequency responses of switched circuits, e.g. switched capacitor filters. The convenient AC analysis cannot be used because it starts from the circuit linearization around the DC operating point. However, the operating point of switched circuit is periodically swept by virtue of the switching signal. The only universal method, which is very labored and time consuming indeed, consists in the steady state transient analysis with the input signal swept by a sufficiently small harmonic signal of a given frequency, the selection of AC component of output signal, and its comparison with the input. This procedure must be repeated for more frequencies. A certain degree of automation of such actions in WinSpice (SPICE3) program is described in [11].

A number of algorithms for gaining the above frequency responses directly and with much less computational effort are described in the literature [12-16]. However, these procedures are not compatible with mathematical algorithms, implemented in the SPICE-family programs.

Switched DC-DC converters are a special case of switched circuits. For their effective analysis, the so-called average models have been developed [17]. They can be easily implemented in SPICE by using the methods of behavioral modeling. Then all the basic analyses, i.e. OP, TRAN, DC a AC can be applied. However, average models cannot describe the switching effects, and their validity is limited to relatively low frequencies in comparison to the switching frequency. They offer no information about the aliasing phenomena. For AC analysis, they model only the spectral component of output signal on the frequency of input AC term.

Other known limitations of the SPICE-family programs occur in noise analysis, which is performed on the level of linearized small-signal circuit model. That is why we cannot model noise relations in most nonlinear circuits such as mixers. The spectrum of noise power is shifted along the frequency axis. This is beyond the scope of conventional SPICE noise analysis [18, 19].

2. Special software tools

It can be concluded from Chapter 1 that there is a long-standing discrepancy between the practical needs of simulation of special networks on the one hand and the limitations of SPICE-family programs on the other hand. This contradiction is solved by two different approaches: **1** Special single-purpose programs are developed, based mostly on the behavioral modeling. They include new algorithms, which are not compatible with the SPICE computation core. **2** Development of SPICE-like programs [20], i.e. programs that offer basic types of SPICE analyses, utilizing the standard SPICE models of circuit components. However, their internal algorithms are built differently from SPICE.

From the first program group, let us name the Swann [21], SwitCap [22], and SPIN [15] programs for the analysis of switched circuits. The first two programs are based on principles described in the monograph [16]. The important internal algorithm is here the numerical Laplace inversion (NLI), enabling the identification of Dirac impulses inside the signals analyzed. It enables an easy analysis of circuits with inconsistent initial conditions [9, 10, 12]. The SPIN program also uses NLI. This algorithm serves for building special circuit matrices for AC analysis based on the so-called generalized transfer functions [15]. A drawback of such programs consists in utilizing behavioral models, not models on the transistor level. Such simple models cannot include all significant parasitic effects.

The second group contains several SPICE-like programs. Their developers flexibly respond to new requirements on simulation problems. Let us name the SpectreRF program from Cadence Design Systems [23], Advanced Design System (ADS) from Agilent EEsoft [24], SP/XL-RF from Avista Design Systems [25], SmartSpice-RF from SILVACO [26], SSpice (public domain program, Univ. of California at Berkeley) [27], and CIA from CVUT Prague [28]. The survey paper [18] describes special methods of analysis, developed especially for the simulation of RF networks: the Newton shooting method and the harmonic balance method for finding periodic and quasiperiodic steady states, the Mixed Frequency-Time Methods for multitone-excited circuits, the Envelope Methods for fast analysis of circuits where the aim of the analysis is LF envelope of HF oscillations. The so-called Linear Time-Varying Analysis [29] represents a generalization of conventional small-signal AC analysis in the SPICE-family programs. It is used, for example, for frequency analysis of switched capacitor filters. The periodical steady state is first found without applying the AC component of input signal. The resulting state is called the periodic operating point. A linearization around this point is done, with the small-signal AC excitation being considered. An implementation of this method in the SpectreRF program for frequency analysis of circuits with periodically controlled switches is described in [30].

Three interesting programs, designed especially for the simulation of switched-mode power supplies, lie on the boundary of the above groups of programs: SwitcherCADIII from Linear Technology [31], POWER 4-5-6 from RidleyEngineering [32], and PSIM from Powersim Technologies [33]. SwitcherCAD is denoted as a „new SPICE“. It serves for a fast simulation of switched power supplies. The analysis is based on fundamentally new but unpublished algorithms. The steady states are reached by repetitive transient analysis, which is automatically run from the final circuit state of the preceding analysis, until the internal algorithm identifies the steady state. The POWER 4-5-6 program serves for a fast design and simulation of selected topologies of DC-DC converters in the time and the frequency domains. Internal methods of analysis have not been published. The PSIM program works on quite a different principle than SPICE. A fixed time step is used in the transient analysis. Capacitors and inductors are modeled by equivalent time-varying resistors on the principle of the discretization of differential equations. Any nonlinearities are approximated by piecewise models. Simulation times are by several orders shorter than those for conventional SPICE transient analysis [34].

The development of unconventional methods of computer simulation is going on. Next Chapters describe some novel procedures and algorithms. Some of them are continuously implemented in the Spice-like program CIA [28].

3. Improvements of Transient Analysis

The majority of SPICE-like programs use the classical Gear algorithm [35] for implicit numerical integration. In the following subsection, the main features of a more flexible algorithm are described, which is based on a sophisticated form of the Newton interpolation polynomial. Moreover, a new reliable method is described, which efficiently suppresses possible divergence in direct-current and transient analyses.

To enhance the efficiency of repeated solutions of linear systems necessary for the Newton-Raphson method, a new modification of the Markowitz criterion is also suggested, which is compatible with the fast modes of the LU factorization.

Finding the steady-state seems to be a fundamental problem for the analysis of large continuous-time and switched circuits. For this reason, an efficient and relatively robust algorithm is described in the next subsection, which is based on the idea of extrapolation [36].

3.1 Algorithm for Implicit Integration

In this subsection, a robust and efficient algorithm for the implicit integration of a system of algebraic-differential equations is described, which is based on backward scaled differences. In fact, the formulae represent a sophisticated form of the Newton interpolation polynomial, and use efficient recurrent forms for enhancing the execution.

The system of nonlinear algebraic-differential equations of a circuit is generally defined in the implicit form

$$\mathbf{f}[\mathbf{x}(t), \dot{\mathbf{x}}(t), t] = \mathbf{0}\,. \tag{1}$$

Let us now assume that the first n steps of a numerical integration of (1) have finished. Let us mark $\mathbf{x}(t_n)$ by $\mathbf{x}_n$ and define the backward scaled differences by the formulae

$$\begin{aligned} &\delta^{(0)}\mathbf{x}_n = \mathbf{x}_n, \\ &\delta^{(k)}\mathbf{x}_n = \delta^{(k-1)}\mathbf{x}_n - \alpha_n^{(k-1)}\delta^{(k-1)}\mathbf{x}_{n-1}, \\ &\quad n = 1,\ldots, \quad k = 1,\ldots,k_n + 2, \end{aligned} \tag{2}$$

(a conception of the backward scaled differences with a comprehensive stability analysis can be found in [37]), where k_n is the order of the polynomial used in the last integration step, and the $\alpha_n^{(\cdots)}$ multipliers are also determined in the recurrent way

$$\begin{aligned} &\alpha_n^{(0)} = 1, \\ &\alpha_n^{(k)} = \alpha_n^{(k-1)} \frac{t_n - t_{n-k}}{t_{n-1} - t_{n-1-k}}, \quad k = 1,\ldots,k_n + 1. \end{aligned} \tag{3}$$

The **predictor** of the variables for the next chosen time (i.e., for t_{n+1}) marked by $\mathbf{x}_{n+1}^{(0)}$ is determined by the polynomial extrapolation using scaled differences (2)

$$\mathbf{x}_{n+1}^{(0)} = \sum_{k=0}^{k_{n+1}} \alpha_{n+1}^{(k)}\, \delta^{(k)}\mathbf{x}_n\,, \tag{4}$$

which is a more sophisticated form of the Newton interpolation polynomial – see the proof of Theorem 1 in [28]. The predictor of the derivatives of the variables with respect to time is also determined in the recurrent way

$$\dot{\mathbf{x}}_{n+1}^{(0)} = \sum_{k=0}^{k_{n+1}} \beta_{n+1}^{(k)}\, \delta^{(k)}\mathbf{x}_n\,, \tag{5}$$

where the β multipliers can easily be derived differentiating (3) (the highest order is not necessary in this case)

$$\begin{aligned} &\beta_n^{(0)} = 0, \\ &\beta_n^{(k)} = \frac{\alpha_n^{(k-1)} + (t_n - t_{n-k})\beta_n^{(k-1)}}{t_{n-1} - t_{n-1-k}}, \quad k = 1,\ldots,k_n. \end{aligned} \tag{6}$$

The **corrector** of the variables $\mathbf{x}_{n+1} := \mathbf{x}_{n+1}^{(j_{\max_{n+1}})}$ for t_{n+1} is determined by the modified Newton iterations

$$\begin{aligned} &\left[\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)_{n+1}^{(j)} + \gamma_{n+1}\left(\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{x}}}\right)_{n+1}^{(j)}\right]\Delta\mathbf{x}_{n+1}^{(j)} = -\mathbf{f}_{n+1}^{(j)}, \\ &\quad n = 0,\ldots, \quad j = 0,\ldots,j_{\max_{n+1}}, \end{aligned} \tag{7}$$

i.e., by repeatedly solving linear system (7) via applying the γ_{n+1} factor approximated by (see the proof in [28])

$$\gamma_{n+1} = \sum_{k=1}^{k_{n+1}} \frac{1}{t_{n+1} - t_{n+1-k}}\,, \tag{8}$$

which gives the classic formula $\gamma_{n+1} = 1/(t_{n+1} - t_n) = 1/\Delta t_{n+1}$ if the first-order (i.e., Euler's) method is used.

After resolving linear system (7), the vectors $\mathbf{x}_{n+1}^{(j)}$ and $\dot{\mathbf{x}}_{n+1}^{(j)}$ are updated in the following standard way:

$$\mathbf{x}_{n+1}^{(j+1)} = \mathbf{x}_{n+1}^{(j)} + \Delta\mathbf{x}_{n+1}^{(j)}, \quad \dot{\mathbf{x}}_{n+1}^{(j+1)} = \dot{\mathbf{x}}_{n+1}^{(j)} + \gamma_{n+1}\Delta\mathbf{x}_{n+1}^{(j)}\,. \tag{9}$$

However, the convergence of the corrector (7)–(9) is often problematic, especially in the operating-point analysis (in the transient analysis, the values of the previous step serve as a good estimation for the following step). To avoid possible divergence in both static and dynamic domains, a new control method has been developed for handling the differences during iterations:

if $j = 0$ then
  $\mathbf{x}' := \mathbf{x}_{n+1}^{(0)}$,
  $\dot{\mathbf{x}}' := \dot{\mathbf{x}}_{n+1}^{(0)}$,
  $\Delta\mathbf{x}' := \Delta\mathbf{x}_{n+1}^{(0)}$,
  $\mathbf{f}' := \mathbf{f}_{n+1}^{(0)}$,
  iteration is accepted,
else
  if $\displaystyle \frac{1}{n_x}\sum_{i=1}^{n_x} \frac{\left|\mathrm{f}_{n+1_i}^{(j)}\right|}{\left|\mathrm{f}_i'\right| + \mathrm{f}_{\mathrm{null}_i}} < 1$ then
    $\mathbf{x}' := \mathbf{x}_{n+1}^{(j)}$,
    $\dot{\mathbf{x}}' := \dot{\mathbf{x}}_{n+1}^{(j)}$,
    $\Delta\mathbf{x}' := \Delta\mathbf{x}_{n+1}^{(j)}$,
    $\mathbf{f}' := \mathbf{f}_{n+1}^{(j)}$,
    iteration is accepted,
  else
    $\Delta\mathbf{x}' := \dfrac{\Delta\mathbf{x}'}{2}$,
    $\mathbf{x}_{n+1}^{(j)} := \mathbf{x}'$,
    $\dot{\mathbf{x}}_{n+1}^{(j)} := \dot{\mathbf{x}}'$,
    $\Delta\mathbf{x}_{n+1}^{(j)} := \Delta\mathbf{x}'$,
    iteration is rejected. (10)

The algorithm checks the residual values (i.e. each element of $\mathbf{f}_{n+1}^{(j)}$) after each iteration. The basic idea of handling the differences $\Delta\mathbf{x}_{n+1}^{(j)}$ in the above way relates to the fundamental property of the Newton-Raphson method. If the average value of the residual values does not decrease, the difference is halved and the iteration is repeated. The halving terminates when the average residual value decreases. It is *certain* that the occurrence of the decreased average residue will be found – the algorithm does not even contain a check for possible infinite loop (!). As a result, only such $\Delta\mathbf{x}_{n+1}^{(j)}$ can be used for updating the vector $\mathbf{x}_{n+1}^{(j)}$ which ensures the decrease of the average value of the residual values.

The algorithm needs four auxiliary vectors $\mathbf{x}'$, $\dot{\mathbf{x}}'$, $\Delta\mathbf{x}'$ and $\mathbf{f}'$, the parameters $\mathrm{f}_{\mathrm{null}_i} \to 0+$ prevent potential division by zero, and n_x is the dimension of all the vectors.

3.2 Modified Markowitz Criterion

The LU factorization of the system matrix of (7) can be considered the most critical part of the circuit analyses from the computational point of view, especially for very large scale circuits. As known, the elimination process during the LU factorization creates non-zero entries at positions which correspond to zeros in the original matrix. The fill-in is a set of all the entries which were originally zeros and took on non-zero values at any step of the elimination. For large circuits, the fill-in can greatly increase the memory requirement, and the total computational time correspondingly.

To solve this problem, the heuristic Markowitz criterion is still widely used [38]–[40]. The Markowitz criterion prevents expressions from swelling in a sparse matrix by selecting a pivot such that the number of entries that stay zero during the elimination is maximal, or, to put it the other way round, such that the fill-in created by the operation is minimal. However, the Markowitz criterion is not compatible with some fast algorithm modes for the LU factorization because a selection of the pivot from a submatrix is not allowed (the algorithm modifications for the fast LU factorization only allow the pivot selection from a row or column). Therefore, the Markowitz criterion must be modified correspondingly for an implementation in the fast LU factorization algorithm.

The system of linear equations (7) generally written $\mathbf{Ax} = \mathbf{z}$ is frequently solved with the same matrix $\mathbf{A}$ and different vectors $\mathbf{z}$ – consider the class of sensitivity analyses as a typical example. Therefore, the LU factorization has an advantage compared with the classical Gaussian elimination because the matrix operations can efficiently be separated from the operations on the right side. Hence, the matrix of the system is first factorized to a product of the lower and the upper triangular matrices $\mathbf{L}$ and $\mathbf{U}$, respectively, i.e., $\mathbf{A} = \mathbf{LU}$, and after that the two tasks with the triangular matrices of the system are solved successively, i.e., $\mathbf{Ly} = \mathbf{z}$ and $\mathbf{Ux} = \mathbf{y}$.

The LU factorization can be efficiently carried out using the following recurrent scheme:

$$
\begin{aligned}
&U_{11} = 1,\\
&L_{i1} = A_{i1},\ i = 1,\ldots,n_x,\\
&U_{12} = \frac{A_{12}}{L_{11}},\\
&U_{22} = 1,\\
&L_{i2} = A_{i2} - L_{i1}U_{12},\ i = 2,\ldots,n_x,\\
&U_{1j} = \frac{A_{1j}}{L_{11}},\\
&U_{ij} = \frac{A_{ij} - \sum_{k=1}^{i-1} L_{ik}U_{kj}}{L_{ii}},\ i = 2,\ldots,j-1,\\
&U_{jj} = 1,\\
&L_{ij} = A_{ij} - \sum_{k=1}^{j-1} L_{ik}U_{kj},\ i = j,\ldots,n_x,\\
&\qquad j = 3,\ldots,n_x,
\end{aligned} \tag{11}
$$

where n_x marks the dimension of the $\mathbf{A}$, $\mathbf{L}$, and $\mathbf{U}$ matrices.

Although the procedures for LU factorization (11) are relatively simple even for such a type of programming which utilizes the sparsity of matrices, the retrieval of an optimal system of pivots represents a relatively complicated problem, because many standard criteria are incompatible with fast modes of the LU factorization [41].

A novel criterion can be based on an appropriate modification of the famous Markowitz one. The Markowitz criterion itself is not too complicated. Suppose that the LU factorization has proceeded through the first k stages. For each i^{th} row in the active $(n_x - k)(n_x - k)$ submatrix, let $r_i^{(k)}$ denote the number of entries. Similarly, let $c_j^{(k)}$ denote the number of entries in the j^{th} column. The Markowitz criterion is to select as pivot the entry $A_{ij}^{(k)}$ from the $(n_x - k)(n_x - k)$ submatrix that satisfies

$$\min_{i,j}\left(r_i^{(k)} - 1\right)\left(c_j^{(k)} - 1\right). \tag{12}$$

Using this entry as the pivot causes $\left(r_i^{(k)} - 1\right)\left(c_j^{(k)} - 1\right)$ entry modifications at the k^{th} step. Not all these modifications will result in a fill-in; therefore, the Markowitz criterion is actually an approximation to the choice of pivot which introduces the least fill-in. However, the Markowitz weights (12) must be used in another way, compatible with the fast scheme of the LU factorization [41].

Tested circuit	Total number of nonzero matrix elements			
	Before LU fact.	After LU fact.		
		Unarranged	Numb. in columns	Modif. Markowitz
JFET preamplifier	19	21	19	19
Two-way rectifier	43	58	62	56
Astable multivibrator	45	59	58	55
Colpitts oscillator	47	68	62	58
Low-noise amplifier	79	137	93	89
Logical circuit (GaAs)	89	140	113	93
Logical circuit (TTL)	105	175	151	146
Oscillator (GaAs)	107	220	162	157
RIAA preamplifier	129	301	211	199
Voltage source (CMOS)	177	510	259	250
Flip-flop circuit (CMOS)	203	338	270	283
Mixer (CMOS)	232	550	265	266
Logical circuit (ECL)	264	708	668	619
Power amplifier	342	2284	1030	1035
Operational amplifier	531	1929	656	651
Microwave oscillator	1485	4047	2816	2770

Tab. 1. Comparison of the total number of nonzero matrix elements before and after the LU factorization for various circuits.

A modification compatible with the fast modes of the LU factorization is performed in the following way. To each potentially nonzero element located in i^{th} row and j^{th} column, the weight is assigned equal to the product of the numbers of other potentially nonzero elements in the i^{th} row $r_i - 1$ and j^{th} column $c_j - 1$ (r_i and c_j are the numbers of the nonzero elements in the i^{th} row and j^{th} column)

$$W_{ij} = \begin{cases} (r_i - 1)(s_j - 1) \text{ for potentially nonzero element,} \\ 0 \text{ for element which is always zero,} \end{cases} \qquad (13)$$
$$i = 1, \ldots, n_x,\ j = 1, \ldots, n_x.$$

The weight W_{ij} represents an estimation of the fill-in addition that can arise if the (i, j) element becomes a pivot. The next part of the procedure differs from the original one because the column orientation of LU factorization (11) must be held. Instead of an immediate reordering in the matrix, the average weights are determined for all columns using the element weights (13)

$$w_j = \frac{1}{c_j} \sum_{i=1,\ldots,n_x \wedge A_{ij} \neq 0} W_{ij},\ j = 1, \ldots, n_x \qquad (14)$$

– finally, the columns are reordered according to the weights (14) – the first and the last should be those with the minimum and the maximum weights, respectively.

A comparison of the efficiency of the modified Markowitz criterion is shown in Table 1. It is clear that both the criterion based only on numbers of nonzero elements and the modified Markowitz criterion give substantially better results compared with those for unarranged matrices.

3.3 Identification of Steady-State

The problem of computing the periodic steady-state response can be formulated as a solution of a nonlinear symbolic equation of the form

$$\mathbf{x}_{ss} = \boldsymbol{I}\left(\mathbf{x}_{ss}, t_0\right), \qquad (15)$$

where $\boldsymbol{I}(\mathbf{x}_{ss}, t_0)$ marks the solution vector after integrating the system of algebraic-differential equations of a circuit in the implicit form (1) on the interval of one period.

Instead of the usual integration of system (1), a much shorter integration is performed. Samples of the solution are immediately collected after each of the periods. The samples become the input for the scalar ε-algorithm [36], which is able to estimate the state of the system in future. An outcome of the algorithm becomes a new initial condition of system (1) and the process is repeated until the steady-state is detected.

The number of periods needed for an extrapolation loop depends on the number of slowly decaying transients. This number can be reduced by a low-pass filtering performed by a numerical integration of a suitable length

$${}_j\mathbf{x}_0 = {}_j\mathbf{x}\left(t_0 + \Delta t_{\text{extpol}}\right) = \int_{t_0}^{t_0+\Delta t_{\text{extpol}}} \boldsymbol{F}\left(\mathbf{x}, t\right) \mathrm{d}t \qquad (16)$$

for $j = 1, \ldots, j_{\max}$, where j marks the iteration index of the ε-algorithm, and $\dot{\mathbf{x}} = \boldsymbol{F}(\mathbf{x}, t)$ indicates the vector of functions to be integrated in a numerical way. This is only a *symbolic* notation because the original system (1) is always implicit and cannot be transferred to an explicit form. The parameter Δt_{extpol} is chosen such that the rapidly decaying transients are insignificant at $t_0 + \Delta t_{\text{extpol}}$. After the low-pass filtering, the extrapolation is assumed to be only concerned with the slowly decaying transients and the order of the extrapolation algorithm marked below by k_{extpol} can be decreased correspondingly.

The vector ${}_j\mathbf{x}_0$ represents the first sample for the extrapolation algorithm. The entire sequence of samples is also obtained by the implicit numerical integration of system (1)

$$
\begin{aligned}
{}_j\mathbf{x}_k &= {}_j\mathbf{x}\left(t_0 + \Delta t_{\text{extpol}} + \sum_{i=1}^{k} {}_jT_i\right) = \\
&\int_{t_0+\Delta t_{\text{extpol}}}^{t_0+\Delta t_{\text{extpol}}+\sum_{i=1}^{k} {}_jT_i} \boldsymbol{F}(\mathbf{x},t)\,\mathrm{d}t, \\
k &= 1,\dots,2k_{\text{extpol}},\ j = 1,\dots,j_{\max},
\end{aligned}
\tag{17}
$$

where ${}_jT_i$ are the periods that must be detected for autonomous circuits automatically during the integration (they are known for non-autonomous circuits, of course).

When sampling is finished after the integration, the scalar ε-algorithm starts after the initialization (n_x marks the number of elements of $\mathbf{x}$)

$$
\begin{aligned}
&\varepsilon_{-1_i}^{(k)} := 0,\ k = 1,\dots,2k_{\text{extpol}}, \\
&\varepsilon_{0_i}^{(k)} := {}_jx_{k_i},\ k = 0,\dots,2k_{\text{extpol}}, \\
&\quad i = 1,\dots,n_x
\end{aligned}
\tag{18}
$$

by the extrapolation formulae [36]

$$
\begin{aligned}
&\varepsilon_{l+1_i}^{(k)} := \varepsilon_{l-1_i}^{(k+1)} + \frac{1}{\varepsilon_{l_i}^{(k+1)} - \varepsilon_{l_i}^{(k)}}, \\
&\quad l = 0,\dots,2k_{\text{extpol}} - 1, \\
&\quad k = 0,\dots,2k_{\text{extpol}} - 1 - l
\end{aligned}
\tag{19}
$$

for all elements $\varepsilon_{l+1_i}^{(k)}, i = 1,\dots,n_x$ of the vector $\boldsymbol{\varepsilon}_{l+1}^{(k)}$. The outcome of the ε-algorithm becomes the new initial condition of system (2), i.e.

$$
{}_{j+1}x(t_0)_i := \varepsilon_{2k_{\text{extpol}_i}}^{(0)},\ i = 1,\dots,n_x,
\tag{20}
$$

and the procedure is repeated until there is a stop due to detected convergence (the parameter $\varepsilon_{\text{extpol}}$ is allowable extrapolation error)

$$
\begin{aligned}
&\text{if } \max_{i=1}^{n_x} \frac{\left| {}_jx_{k_i} - {}_jx_{k-1_i} \right|}{\left| {}_jx_{k_i} \right| + x_{\text{null}_i}} \le \varepsilon_{\text{extpol}} \text{ then} \\
&\quad \mathbf{x}_{\text{ss}} := {}_j\mathbf{x}_k
\end{aligned}
\tag{21}
$$

or overstepping the maximum allowable number of the extrapolations $j_{\text{extpol}_{\max}}$

$$
\begin{aligned}
&\text{if } j > j_{\text{extpol}_{\max}} \text{ then} \\
&\quad \mathbf{x}_{\text{ss}} := {}_{j_{\text{extpol}_{\max}}+1}\mathbf{x}(t_0).
\end{aligned}
\tag{22}
$$

The progression of the ε-algorithm (18) and (19) is represented by the following diagram – the values at the corners of the triangles create new values sequentially in the directions of the arrows

$$
\begin{array}{ccccc}
 & \varepsilon_{0_i}^{(0)} & & & \\
\varepsilon_{-1_i}^{(1)} & \lhd\!\!\rightarrow & \varepsilon_{1_i}^{(0)} & & \\
 & \varepsilon_{0_i}^{(1)} & \lhd\!\!\rightarrow & \varepsilon_{2_i}^{(0)} & \\
\varepsilon_{-1_i}^{(2)} & \lhd\!\!\rightarrow & \varepsilon_{1_i}^{(1)} & & \\
 & \varepsilon_{0_i}^{(2)} & & \vdots & \\
\vdots & & \vdots & & \\
 & \vdots & & &
\end{array}
\tag{23}
$$

4. Improvements of AC Analysis of Switched Circuits

An unconventional approach of AC analysis of circuits, employing periodically controlled switches, is briefly described in this Chapter. The symptoms of both continuous-time and discrete-time circuits are included in their behavior. The method demonstrated is trying to model both these features simultaneously with the utilization of the so-called generalized transfer functions (GTF) [15]. These functions model the behavior of switched circuits in their (quasi) periodic steady states, utilizing the well-known operators *s* (Laplace transform) and *z* (Z-transform) simultaneously, thus $K(s, z)$. This general approach has its boundary forms: the s-domain transfer function of continuous-time system $K(s)$, and the z-domain transfer function of discrete-time system $K(z)$, respectively. The GTF is passing to the above conventional transfer functions when no switches are present in the circuit or when the switches compensate mutually their operation ($K(s,z)\to K(s)$), or when – for instance – as a consequence of ideal switches – the circuit operates as a discrete-time system ($K(s,z) \to K(z)$). The typical case is an idealized switch-capacitor filter.

4.1 Generalized transfer functions

Consider a simple model of the Sample-Hold (S-H) circuit in Fig. 1 (a) and the corresponding waveforms in Fig. (b). At any moment, the circuit operates in one of two possible phases, denoted in Fig. 1 as phase 1 (the clock signal ϕ is HIGH and the switch is closed), and phase 2 (the clock signal ϕ is LOW and the switch is open). The ratio ε of the duration of phases 1 and 2 can be from 0 to 1. In real S-H circuits, ε is below ½, typically $\varepsilon = 0.1$.

Due to the nonzero time constant of charging the capacitor through the nonzero switch on-resistance, the output voltage at time instant $kT+\varepsilon T$ of finalizing the sample selection will not be exactly equal to the input voltage. Analyzing this dynamical error of sample selection, we are interested not in the entire output signal, but only in its discrete values at the end of switching phases 1. The corresponding dots are marked in Fig. 1 (b) by unfilled rings. The dependence of the "quality" of the sample selection on the frequency of sampled signal can be evaluated as follows:

1. We excite the S-H circuit by a harmonic voltage of a certain frequency. This signal will gradually set the circuit in the (quasi)periodic steady state.

2. In the output signal we identify points which we are interested in. In our case, these points lie at the ends of switching phases 1.

3. We interleave the points from step 2 with a harmonic signal of a frequency equal to that of the input signal. This task is unambiguous [15]. We obtain a signal called equivalent signal v^{1e} to signal v [15].

4. Comparing the two harmonic signals v^{1e} and v yields values of generalized amplitude and phase frequency responses of the circuit on the concrete frequency of input signal.

5. We repeat the above steps for various frequencies of excited signal, taking them from required limits on the frequency axis in order to obtain generalized frequency responses.

Note that just the GTF $K(s,z)$ is a compact mathematical description of the generalized frequency response. The frequency responses can be obtained from GTF after the well-known substitutions $s=j\omega$ and $z=exp(j\omega T)$, where T is the switching period.

It is obvious from the above that frequency responses will depend on the choice of the "points of interest" on the output waveform. As an example, the second equivalent signal v^{2e} is shown in Fig. 1 (b), which is interleaved with crosses at the ends of phase 2. Another point, marked by filled circle T_p seconds after the end of phase 1, contains information about the value of output voltage at the time of finishing AD conversion by the converter attached to the S-H output. Another equivalent signal and frequency response will be assigned to these points. Comparing it with the signal v^{1e} and the corresponding frequency response, one can deduce the measure of voltage drop in the output voltage during the AD conversion in the HOLD state.

Let us conclude that an infinite set of equivalent signals can be assigned to a single signal of switched circuit, and one GTF to any relation of input-equivalent output signal. In this way, one can model the switched circuit from more points of view, depending on the circuit features we are interested in.

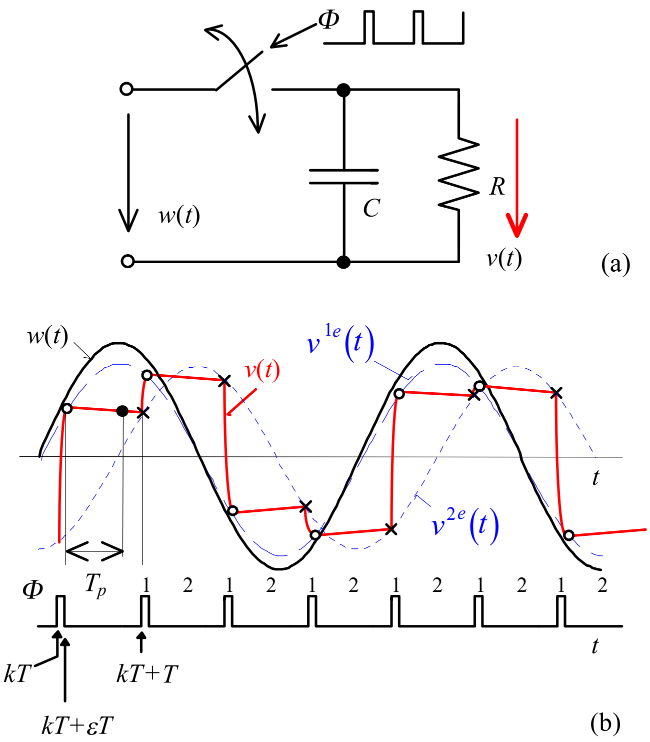

Fig. 1. (a) Model of the Sample-Hold circuit, (b) the corresponding waveforms.

4.2 Example of GTF evaluation for S-H circuit

Consider the model of S-H circuit in Fig. 1 (a) with nonzero and signal independent on-resistance R_{on} and infinite off-resistance of the switch. Then the circuit is described by linear equations in each switching phase. The following equations evaluate the output voltage at the ends of given phases. The index of this phase is marked by a superscript.

Phase 1: $kT < t \le kT+\varepsilon T$

$$v^1(kT+\varepsilon T) = v^2(kT)e^{-\frac{\varepsilon T}{\tau_1}} + \int_0^{\varepsilon T} g_1(\xi) w(kT+\varepsilon T-\xi)d\xi \qquad (24)$$

Phase 2: $kT+\varepsilon T < t \le kT+T$

$$v^2(kT+T) = v^1(kT+\varepsilon T)e^{-\frac{(1-\varepsilon)T}{\tau_2}}, \qquad (25)$$

where

$$\tau_1 = (R_{on} \,\&\, R)C, \tau_2 = RC, g_1(t) = \frac{1}{R_{on}C}e^{-\frac{t}{\tau_1}}. \qquad (26)$$

Now consider the harmonic input signal described by a complex phasor

$$w(t) = \hat{W}e^{st},\ s = j\omega\,. \tag{27}$$

The left sides of (24) and (25) describe output voltage samples at the ends of switching phases 1 and 2. In other words, they are samples of equivalent signals $v^{1e}(t)$ and $v^{2e}(t)$ from 1 (b), whose nature is also harmonic. Let us describe these signals by the complex phasors

$$v^{1e}(t) = \hat{V}^{1e}e^{st},\ v^{2e}(t) = \hat{V}^{2e}e^{st}\,. \tag{28}$$

Substituting (27) and (28) into (24) and (25) and arranging yield the resulting forms of Eqs. (24) and (25):

$$\hat{V}^{1e} = e^{-\frac{\varepsilon T}{\tau_1}}\hat{V}^{2e}z^{-\varepsilon} + G_1(s,z)\hat{W}\,, \tag{29}$$

$$\hat{V}^{2e} = e^{-\frac{(1-\varepsilon)T}{\tau_2}}\hat{V}^{1e}z^{-(1-\varepsilon)}\,, \tag{30}$$

where

$$z = e^{sT}\,, \tag{31}$$

$$G_1(s,z) = \int_0^{\varepsilon T} g_1(\xi)e^{-s\xi}d\xi = \frac{R}{R+R_{on}}\frac{1-e^{-\frac{\varepsilon T}{\tau_1}}z^{-\varepsilon}}{1+\tau_1 s}\,. \tag{32}$$

Eqs. (29) and (30) can be used to derive formulas for the phasors of both equivalent signals:

$$\hat{V}^{1e} = K^1(s,z)\hat{W},\ \hat{V}^{2e} = K^2(s,z)\hat{W}, \tag{33}$$

where

$$K^1(s,z) = \frac{G_1(s,z)}{1-e^{-\frac{\varepsilon T}{\tau_1}}e^{-\frac{(1-\varepsilon)T}{\tau_2}}z^{-1}},$$

$$K^2(s,z) = K^1(s,z)e^{-\frac{(1-\varepsilon)T}{\tau_2}}z^{-(1-\varepsilon)}$$

are generalized transfer functions of the S-H circuit, corresponding to the above equivalent signals $v^{1e}(t)$ and $v^{2e}(t)$.

4.3 GTF implementation in PSPICE

Equations (29) and (30) can be easily implemented in PSPICE program for a direct AC analysis of S-H circuits. Below is a complete list of the circuit file for PSPICE analysis:

```
S-H circuit, sampling frequency 100kHz, GTF modeling
*
.param C=10nF R=100kOhm Ron=50Ohm fs=100kHz ep=0.1
+Ts={1/fs} RonR={Ron*R/(Ron+R)}
+tau1={C*RonR} tau2={C*R}
+exp1={exp(-ep*Ts/tau1)} exp2={exp((ep-1)*Ts/tau2)}
*
Vin in 0 AC=1
Eout1 out1 x LAPLACE=
+{V(in)} {R/(R+Ron)*(1-exp1*exp(-s*ep*Ts))/(1+tau1*s)}
Ex x 0 LAPLACE={V(out2)} {exp1*exp(-s*ep*Ts)}
Eout2 out2 0 LAPLACE={V(out1)} {exp2*exp((ep-1)*s*Ts)}
*
.step param Ron list 10 20 50 100
.AC dec 100 100 500k
.probe
.end
```

Results of the AC analysis for four different switch on-resistances are in Fig. 2.

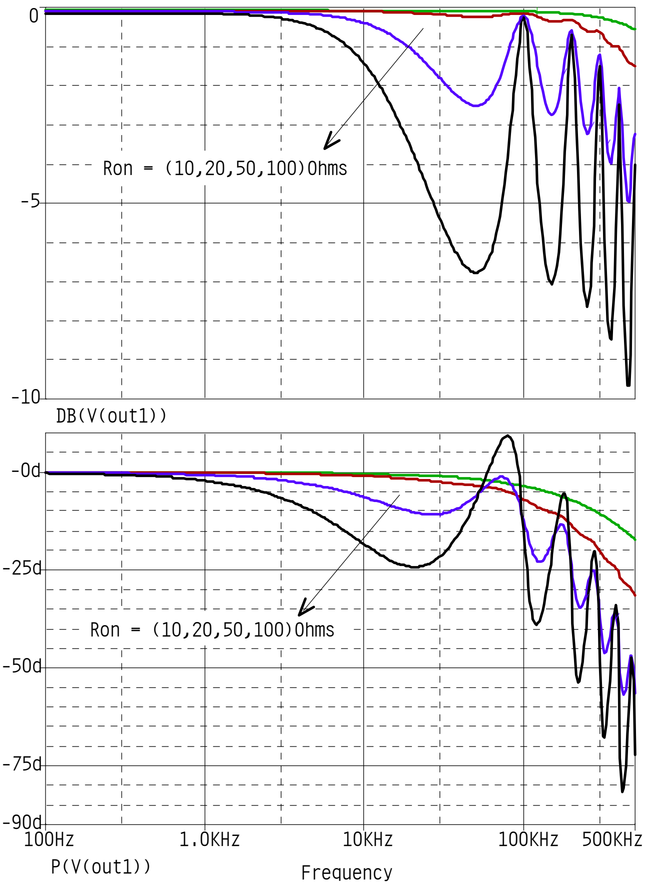

Fig. 2. Amplitude and phase frequency responses of S-H circuit for equivalent signal v^{1e}.

5. Conclusions

A number of fundamental limitations of the SPICE-family programs are described in the introduction of this paper. These limitations are particularly significant for circuits with simultaneously existing slow and fast phenomena, as well as for circuits whose models cause convergence problems.

Chapter 2 provides a summary of special programs, which respond to some limitations of the SPICE-family programs. Among them is the CIA program, whose new specific algorithms are introduced in Chapter 3. An advanced integration algorithm convenient for time-domain analysis is proposed, which suppresses possible physically incorrect or unstable results of the traditional methods. A novel method suppressing a potential divergence of the Newton-Raphson method is also suggested, which is used for solving the algebraic-differential equations of nonlinear circuits. To enhance the efficiency of repeated solutions of linear systems, necessary in the Newton-Raphson method, a new modification of the Markowitz criterion is suggested, which is compatible with the fast modes of the LU factorization. An algorithm for determining the steady-state, which is based on the idea of extrapolation and which uses the epsilon-algorithm as an efficient tool for acceleration, is described. For the autonomous circuits, a reliable detection of unknown period is implemented using derivatives of circuit variables, which are available as one of the outcomes of the integration algorithm. The algorithms of determining the steady-state are useful especially for RF circuits and switched-capacitor or switched DC-DC converters with long transients. For such circuits, a novel method of AC analysis is also described. This method utilizes the so-called generalized *s*-*z* transfer functions. In comparison with the classical methods based on averaged modeling, the advantages consist in a more faithful modeling of circuit behavior, particularly in the frequency range around $f_{switch}/2$, as well as in the ability to model correctly the transfers above this border frequency. A drawback consists in the necessity of analytical preprocessing of circuit models prior to their implementation in SPICE. An algorithm of automated building up the corresponding circuit matrices is published in [15]. However, it is incompatible with the internal SPICE algorithms. That is why it can be implemented only in special-purpose programs.

Acknowledgements

Research described in the paper was financially supported by the Czech Grant Agency of the Czech Republic under grant No. 102/05/0277.

References

[1] OrCADPSpice program, http://www.cadence.com/products/orcad/index.aspx

[2] HSPICE program, http://www.synopsys.com/products/mixedsignal/hspice/hspice.html

[3] BIOLEK, Z., BIOLEK, D. DC Stability of Nonlinear Systems. Int. Conference MIXDES'00, Gdynia, Poland, 2000, p. 127-132.

[4] GREEN, M. M. (Almost) Half of Any Circuit's Operating Points are Unstable. *IEEE Transactions on CAS-I*, 1994, vol. 41, no.4, p. 286-293.

[5] LEE, J., CHIANG, H.D. Constructive homotopy methods for finding all or multiple DC operating points of nonlinear circuits and systems. *IEEE Transactions on CAS-I,* 2001, vol. 48, no. 1, p. 35-50.

[6] FOX, R. M. A General Operating-Point Instability Test Based on Feedback Analysis. *Int. Symposium ISCAS'98*, Monterey, CA, USA, 1998.

[7] PSpice A/D Reference Guide. Includes PSpice A/D, PSpice A/D Basics, and PSpice. Product Version 15.7, Cadence, July 2006.

[8] Micro-Cap 9 simulation program, http://www.spectrum-soft.com.

[9] BEDROSIAN, D., VLACH, J. Time-Domain Analysis of Networks with Internally Controlled Switches. *IEEE Transactions on CAS-I*, 1992, vol. 39, no. 3, p. 199-212.

[10] WOJCZIECHOWSKI, J., VLACH, J., OPAL, A. Analysis of Nonlinear Networks with Inconsistent Initial Conditions. *IEEE Transactions on CAS-I*, 1995, vol. 42, no. 4, p. 195-200.

[11] BIČÁK, J, HOSPODKA, J. Frequency Response of Switched Circuits in SPICE. *Int. Conference ECCTD03*, Krakov, 2003, pp. I-333-336.

[12] VLACH, J., SINGHAL, K. Computer Methods for Circuit Analysis and Design. *Van Nostrand Reinhold*, New York, 2nd edition, 1994.

[13] SINGHAL, K., VLACH, J., VLACH, M. Computer Oriented Formulation of Equations and Analysis of Switched Capacitor Networks. *IEEE Transactions on Circuits and Systems*, 1984, vol. 31, no. 9, p. 753-765.

[14] VLACH, J., OPAL, A. Analysis and Sensitivity of Periodically Switched Linear Networks. *IEEE Transactions on Circuits and Systems*, 1989, vol. 36, no. 4, p. 522-532.

[15] BIOLEK, D. Modeling of Periodically Switched Networks by Mixed s-z Description. *IEEE Transactions on CAS-I*, 1997, vol. 44, no. 8, p. 750-758.

[16] YUAN, F., OPAL, A. *Computer Methods for Analysis of Mixed-Mode Switching Circuits*. Kluwer Academic Publishers, 2004.

[17] VORPÉRIAN, V. Fast Analytical Techniques for Electrical and Electronic Circuits. *Cambridge University Press*, 2002.

[18] MAYARAM, K., LEE, D.C., MOINIAN, S., RICH, D.A., ROYCHOWDHURY, J. Computer-Aided Circuit Analysis Tools for RFIC Simulation: Algorithms, Features, and Limitations. *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing,* 2000, vol. 47, no. 4, p. 274 - 286.

[19] KUNDERT, K.S. Introduction to FR Simulation and its Application. http://www.designers-guide.org/Analysis/rf-sim.pdf.

[20] KIELKOWSKI, R. Inside SPICE. McGraw-Hill, Inc., 1994.

[21] VALSA, J., VLACH, J. Swann-a Programme for Analysis of Switched Analogue Nonlinear Networks. *International Journal of Circuit Theory and Applications*, 1995, vol. 23, p. 369-379.

[22] FANG, S. C., TSIVIDIS, Y. P., WING, O. SWITCAP: A switched-capacitor network analysis program. Part I: basic features. *IEEE Circuits Systems Magazine*, 1983, vol. 5, no. 3, p. 4-10. http://www.cisl.columbia.edu/projects/switcap/

[23] Cadence Design Systems. (2000) Affirma Analog Artist circuit design environment. [Online]. Available:

http://www.cadence.com/technology/custom/products/analog_artist.html

[24] Agilent EEsof EDA. (2000) Advanced Design System overview. [Online]. Available:

http://www.tmo.hp.com/tmo/hpeesof/products/ads/adsoview.html

[25] Avista Design Systems. (1999) SP/XL for RF and microwave design. [Online]. Available:

http://www.avista.com/SPXL/index1.html

[26] SmartSpice, http://www.silvaco.com

[27] ASHAR, P. Implementation of algorithms for the periodic-steady-state analysis of nonlinear circuits. *Electronics Research Lab., Univ. California at Berkeley,* UCB/ERL M89/31, 1989.

[28] DOBEŠ, J. Reliable CAD analyses of CMOS radio frequency and microwave circuits using smoothed gate capacitance models. *International Journal of Electronics and Communications* (AEÜ), 2003, vol. 57, no. 6, p. 372-380.

[29] OKUMURA, M., SUGAWARA, T., TANIMOTO, H. An efficient small-signal frequency analysis method for nonlinear circuits with two frequency excitations. *IEEE Trans. Computer-Aided Design,* 1990, vol. 9, no. 3, p. 225–235.

[30] KUNDERT, K. S. Simulating Switched-Capacitor Filters with SpectreRF. http://www.designers-guide.org/Analysis/sc-filters.pdf.

[31] SwitcherCADIII, http://www.analog.com

[32] POWER 4-5-6, http://www.ridleyengineering.com

[33] PSIM program, http://www.powersimtech.com/

[34] BASSO, C.P. Switch-Mode Power Supply SPICE Cookbook. McGraw-Hill, 2001.

[35] VLADIMIRESCU, A. *The Spice Book*. 1st ed. New York: John Wiley & Sons, 1994.

[36] SKELBOE, S. Computation of the periodic steady-state response of nonlinear networks by extrapolation methods. *IEEE Transactions on Circuits and Systems*, 1980, vol. 27, p. 161-175.

[37] PETRENKO, A. I., VLASOV, A. I., TIMTSCHENKO, A. P. *Tabular Methods of Computer-Aided Modeling*. (In Russian.) 1st ed. Kiyv: Higher School, 1977.

[38] BALLARIN, C., KAUERS, M. Solving parametric linear systems: an experiment with constraint algebraic programming. *ACM SIGSAM Bulletin*, 2004, vol. 38, no. 2, p. 33-46.

[39] OSTIGUY, J.-F., NG, K. Y. Optimized wakefield computations using a network model. *Proc. XX Int. Linac Conference*, Monterey, California, 2000, p. 275-277.

[40] DONGARRA, J. J., DUFF, I. S., SORENSEN, D. C., VAN DER VORST, H. A. Numerical Linear Algebra for High-Performance Computers. *SIAM Press*, 1999.

[41] DOBEŠ, J. A modified Markowitz criterion for the fast modes of the LU factorization. *Proc. IEEE Midwest Symposium on Circuits and Systems*, Cincinnati, Ohio, 2005, p. 955-959.