Symbolic Analysis of Linear Circuits with Modern Active Elements

ZDENEK KOLKA[1], DALIBOR BIOLEK[2], VIERA BIOLKOVA[1]
[1]Department of Radio Electronics, [2]Department of Microelectronics
Brno University of Technology
Purkyňova 118, 612 00 Brno
CZECH REPUBLIC
kolka@feec.vutbr.cz

Abstract: - The paper deals with the symbolic analysis of linear circuits with modern active elements using the SNAP program. The simulator can perform both classical and approximate analyses. All device models are stored in a text library, which can be easily extended. Device parameters can be defined as simple symbols or as formulae in the netlist. This allows the simulation of electronic circuits both with basic elements and with complex behavioral models.

Key-Words: - Symbolic Analysis, Symbolic Approximation, Linear Circuits, Active Elements

1 Introduction

An important tendency in analog signal processing is linear applications of modern active elements with the aim of implementing low-power and low-voltage circuits, very often with extreme requirements for speed and accuracy [1]. The most frequent application whose models can be studied in the linear domain includes active frequency filters, linear amplifiers, various immittance converters and also oscillators operating in the classical voltage, current, or hybrid mode.

Nowadays, hundreds of linear applications with tens of various modern active elements are described in the literature. Together with the classical VFA (Voltage Feedback Amplifier) and CFA (Current Feedback Amplifier) operational amplifiers, transconductance amplifiers (OTA) and all sorts of current conveyors let us mention at least CDBA (Current Differencing Buffered Amplifier) [2], CDTA (Current Differencing Transconductance Amplifier) [3], and OTRA (Operational Transresistance Amplifier) [4]. For virtually all these elements there are several modifications.

The design of linear applications which include the above mentioned elements usually proceeds according to a scenario involving some of the following steps:

1. Circuit synthesis usually based on intuitive approach and making use of the designer's experience.
2. Symbolic analysis of network functions, i.e. finding transfer or immittance functions whose coefficients are symbols of parameters of active and passive elements. The analysis is always performed manually.
3. Sensitivity analysis. A typical example is the analysis of sensitivity of 2nd-order filter parameters ω_0 and Q to the variations of parameters of passive circuit elements. The analysis is usually performed by hand, using symbolic formulae for the circuit characteristic of interest.
4. Analysis of basic real parameters, e.g. the analysis of influence of parasitic impedances on frequency response. This analysis is usually realized using the numerical SPICE simulators.
5. Design of transistor structure of an active element in available integrated-circuit technology or adoption of already designed structures. Numerical simulation of the whole circuit with transistor-level models of active elements.
6. Optimization of designed application with the help of available simulation software.

Individual steps are mutually coupled so that the design process is not straightforward and its complexity depends on the nature of design task. In particular the first two steps are closely related as the results of symbolic analysis may be of great information value as to the direction the design process should follow.

It is evident that symbolic analysis [5] in particular is a tool that can play a specific role in all the above-mentioned steps. The classical symbolic analysis providing complete symbolic results without any simplifications plays an important role in step 2, while it is practically useless in step 5 because of the complexity of transistor-level models. However, the approximate symbolic analysis [6] can be used here.

It generates approximate symbolic formulae providing acceptable numerical accuracy. If symbolic results are available, it is possible to apply an analytical differentiation algorithm to obtain sensitivity functions in the symbolic form (step 3). Substituting numerical values of network parameters in the symbolic expression we obtain the so-called semisymbolic result [7] with the possibility of analyzing the frequency characteristics and time responses. Steps 4 to 6 can be realized with the help of behavioral models [8].

Behavioral modeling [9] is an important tool with practical consequences in step 4. For example, the influence of parasitic resistance of terminal *x* of a current conveyor on filter attenuation in the stop-band can be studied in SPICE only with difficulty if we have merely the transistor structure of the conveyor. The same analysis using a symbolic program whose library contains a behavioral model of CCII including R_X as a parameter will provide exactly the results that are required.

Although there are several commercial programs for symbolic analysis nowadays [10], [11], [12], their use for the above-mentioned purposes can be problematic for several reasons such as the limited size of circuits being analyzed, limited types of available models or prohibitive price. An interesting program is SIASCA [13], in which however the set of available active elements is limited to OTAs and CCII- current conveyors, and the program requires MAPLE for its operation. Models of other circuit elements can be created using nullors. The SAPWIN program [14] provides a simple symbolic approximation. Models of new elements can be created only in the form of SPICE-like subcircuits.

Let us consider the general lack of suitable and easily available programs for symbolic analysis and the extent and variety of active circuit elements whose behavioral models should be included in the libraries of these simulation programs. Let us also compare these facts with the current practice where the designer performs by hand the symbolic calculations and for verification simulations uses numerical simulators with limited possibilities of behavioral modeling. Thus it seems useful to create a software tool based on symbolic calculations for the support of the above-mentioned applications. With respect to the large number of variants and modifications of present-day active elements and with respect to future developments in this area the libraries of such simulators should be open and give the possibility of easy definition of behavioral models for new circuit elements. At the same time the program should allow the use of complex SPICE transistor-level models and be able to extract linear models at a given operational point for subsequent approximate symbolic analysis.

The paper describes SNAP (Symbolic and Numerical Analysis Program), which was developed to comply with the above-mentioned requirements. Its potential for both classical and approximate symbolic analysis is demonstrated below.

2 Architecture of SNAP

The SNAP symbolic simulator reads the netlist as its input. It can be used with any schematic editor that can generate the appropriate netlist format.

The circuit to be analyzed symbolically can be set up directly, using linear or linearized components. If numerical analyses are also required, the user has to supply all small-signal parameters. The other option is an automated calculation of small-signal parameters by means of the SPICE simulator. The input is then a standard circuit, which is normally used for numerical simulation.

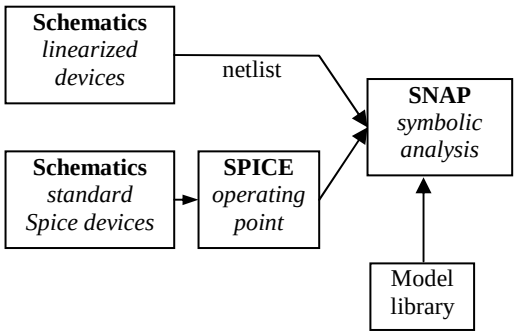


Fig. 1 Simulator input data.

The SNAP netlist is similar to the SPICE netlist but it offers more ways of device parameter specification, see Table 1.

Table 1 Device parameter specification in netlist.

	syntax	example
1	<*symbol*>	R1
2	<*symbolic expression*>	R1+s*L1
3	<*numerical value*>	10k
4	<*numerical expression*>	{R1/2+1k}
5	<*symbol*> = <*value*>	R2=10k R2={R1/2+1k}

Symbolic expressions are constrained to rational functions. The symbolic name of a network parameter is not necessarily related to the device name. For example, resistances of several resistors can be denoted by the symbol *R*. These elements are then considered identical. The numerical value can be defined as a constant or as an expression containing other symbols.

The symbolic analysis is always performed. Moreover, if all parameters are assigned a numerical value or expression, then it is possible to run all numeric-based analyses. The combined parameter definition (case 5) allows using the symbol for symbolic processing and the numerical value for numerical processing. It is always used in the case of automatic small-signal parameter extraction.

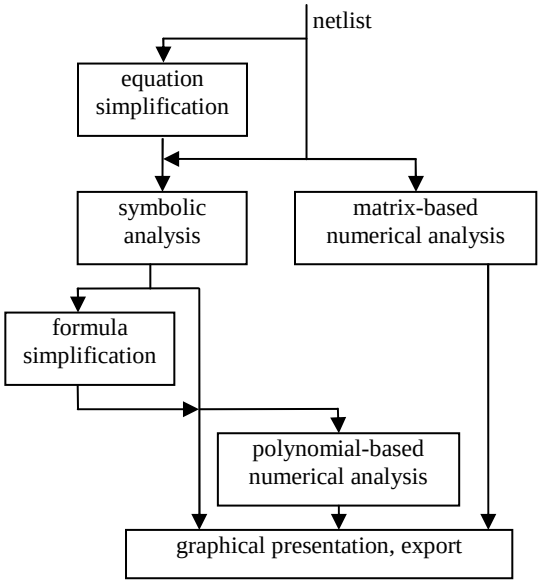


Fig. 2 SNAP analysis flowchart.

The circuit is always regarded as a two-port. Input and output ports are defined by means of special symbols in the schematic editor (or by special components in the netlist). The actual type of network function for analysis is chosen in SNAP. Fig. 2 shows the analysis flowchart of SNAP.

If numerical values of all device parameters are available, then it is possible to display the frequency response and to perform the pole/zero analysis. In addition to graphical presentation the results can be exported to Matlab, MathCad and Maple for further processing.

Two algorithms for approximate symbolic analysis are implemented in SNAP. First, the network equations are simplified. This step is usually very effective, so that it is possible to generate simplified network functions for circuits containing tens of transistors. In the next step the network function is further simplified in the postprocessor.

SNAP is based on the modified nodal analysis [16]. Each network element is defined by its "stamp" in a library, which can be easily extended. The stamp describes how the parameters of an element appear in the hybrid network matrix. This allows defining symbolic behavioral models.

Let us consider an idealized single-pole model of operational amplifier ($Z_{in} = \infty$, $Z_{out} = 0$) with parameters A_0 and τ_1, whose transfer function is

$$A(s) = \frac{v_{out}}{v_{in}} = \frac{A_0}{1 + s\tau_1}. \quad (1)$$

The standard approach used for programs with a fixed library is based on the technique of macromodels, i.e. the circuit in Fig. 3 will be used as the model. Its transfer function is

$$A(s) = \frac{GAR_1}{1 + sR_1C_1}. \quad (2)$$

Product R_1C_1 will appear in the resulting symbolic formula instead of the symbol τ_1.

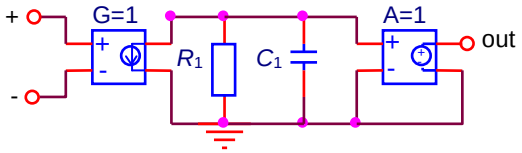
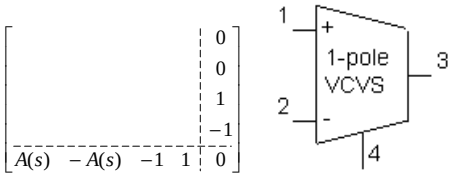


Fig. 3 Macromodel of operational amplifier.

Each matrix entry can be a rational fraction function containing element parameters, constants, and complex frequency s. It is easy to model even special elements like FNDR ($Y = D\,s^2$). Fig. 4 shows the stamp and the library model for the single-pole amplifier.

a)




```
[Etau]          ;model name
nodes  = 4      ;number of nodes
params = 2      ;number of parameters
names  = A0, tau1;parameter names
add    = 1      ;number of additional
                  variables
mat1= -1  1  0  2 : A0/(1+s*tau1)
mat2= -1  3  0  4 :-1
mat3=  3 -1  4  0 : 1
```

b)

Fig. 4 Single-pole amplifier model in SNAP library.

The stamp element P_i (from $mat_i = a\ b\ c\ d : P_i$) appears with the positive sign in positions (a,b) and (c,d), and with the negative sign in (a,d) and (c,b). The coordinates are numbered according to the local numbers of the device nodes. The first additional row and column are denoted "-1" while "0" means "no entry" to the matrix.

3 Symbolic Algorithms

3.1 Exact Symbolic Analysis

Let us consider a linear circuit without independent sources described by the Generalized Modified Nodal Analysis in the form

$$\mathbf{H}\,\mathbf{x} = \mathbf{0}\ , \tag{3}$$

where **H** is the indefinite hybrid circuit matrix and **x** is the column vector of unknown voltages and currents. The indefinite matrix is singular (det(**H**) = 0) as no element is connected to the virtual reference node [17]. This is to avoid using different computational procedures for floating and grounded input and output ports.

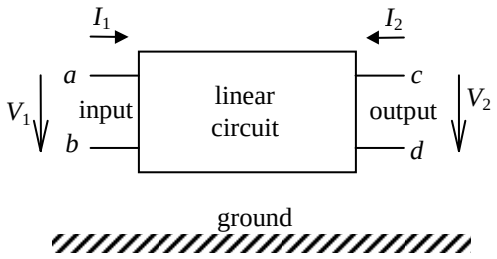


Fig. 5 Analyzed circuit as floating two-port network.

The circuit is always considered a two-port, Fig. 5. All network functions can be expressed symbolically by means of algebraic cofactors of **H** [17].

Table 2 summarizes standard algebraic cofactors. The operation denoted $x > 0$ means the deletion of column or row x. The operation $x > y$ means the addition of column or row x to column or row y before the deletion of x. The calculation of basic network functions can be found in Table 3.

Table 2 Standard algebraic cofactors [17].

cofactor	row operations	column operations
A	$a > 0, b > 0$	$a > 0, b > 0$
B	$a > b, c > 0, d > 0$	$a > b, c > 0, d > 0$
C	$i > 0$ i is arbitrary	$j > 0$ j is arbitrary
D	$c > 0, d > 0$	$c > 0, d > 0$
M	$a > 0, b > 0$	$c > 0, d > 0$
W	$c > 0, d > 0$	$a > 0, b > 0$

The Modified Nodal Analysis leads to sparse network matrix **H**. The calculation of determinants based on the Laplace recursive expansion is inconvenient since the majority of particular expansions ends with zero result.

Table 3 Basic network functions [17].

function	definition	formula
voltage transfer	$V_2/V_1,\ I_2 = 0$	M/A
current transfer	$(-I_2)/I_1,\ V_2 = 0$	M/D
input impedance, output open	$V_1/I_1,\ I_2 = 0$	A/C
input impedance, output shorted	$V_1/I_1,\ V_2 = 0$	B/D
transimpedance	$V_2/I_1,\ I_2 = 0$	M/C
transadmittance	$I_2/V_1,\ V_2 = 0$	-M/B

The circuit matrix is represented internally as a list of stamp elements P_i of all parts in the form $\boxed{P_i}\boxed{a}\boxed{b}\boxed{c}\boxed{d}$, where a, b, c, d are the real coordinates in matrix **H**. The determinant expansion procedure is based on the recursive formula [17]

$$\Delta = \tilde{\Delta} + P_i\,\tilde{\Delta}^{[a>c:b>d]}\ , \tag{4}$$

where Δ is the determinant of the matrix, while $\tilde{\Delta}$ and $\tilde{\Delta}^{[row\_op:col\_op]}$ are the determinant and algebraic cofactor of the matrix without $\boxed{P_i}\boxed{a}\boxed{b}\boxed{c}\boxed{d}$. The same rule is applied to $\tilde{\Delta}$ and $\tilde{\Delta}^{[a>c:b>d]}$.

The expansion of a particular branch stops successfully when the determinant of a zero-dimension matrix is to be computed. If it stops for another reason (empty matrix, nonexistence of cofactor), then the branch is discarded.

As the size of resulting symbolic formula depends exponentially on the number of network elements, this form of symbolic analysis is practically constrained to very small idealized circuits [5].

3.2 Approximate Symbolic Analysis

Symbolic analyses can also be performed for larger circuits but at the expense of approximation. The expression generated loses its universality [5].

The basic principle of approximate symbolic analysis consists in identifying and removing numerically negligible symbolic terms. Thus it is necessary to know the numerical values of all parameters of network elements.

The approximation error should be guaranteed on interval $F \times D$, where $F = \langle f_1, f_2 \rangle$ is the frequency interval of interest, and $D \subset R^r$ is the interval of r parameters of network elements. Due to computational complexity the error is usually checked at several points of $F \times D$.

Since we work with frequency-domain characteristics, it is reasonable to observe the error in the Bode diagram, i.e. to observe the phase and

amplitude errors. Let us define the complex error function as

$$E(f,\mathbf{p}) = \frac{F_A(f,\mathbf{p})}{F_N(f,\mathbf{p})}\ , \tag{5}$$

where **p** is the vector of network parameters, $F_N(f, \mathbf{p})$ is the original (exact) network function, and $F_A(f, \mathbf{p})$ is the approximated function obtained by omitting some elements of F_N. For $F_A = F_N$ we obtain $E = 1$. The real-valued error e for a particular point $(f_i,\mathbf{p}_i) \in F \times D$ is defined as a weighted sum of magnitude and phase errors

$$e(f_i,\mathbf{p}_i) = \frac{\left|20\log\left|E(f_i,\mathbf{p}_i)\right|\right|}{\Delta M_i} + \frac{\left|\arg(E(f_i,\mathbf{p}_i))\right|}{\Delta P_i}\ , \tag{6}$$

where ΔM_i and ΔP_i define indirectly the weight for each point. For $F_A = F_N$ we obtain $e = 0$.

The goal of the approximate symbolic analysis is to find F_A with least complexity such that

$$\max_{i=1..m}\left(e(f_i,\mathbf{p}_i)\right) \le 1 \tag{7}$$

holds for the weighted error observed at m points.

The simplification can be done at different stages of the computation of network function. The technique of Simplification Before Generation (SBG) and the technique of Simplification After Generation (SAG) are implemented in SNAP [5].

3.2.1 SBG Method

The SBG technique is based on a simplification of circuit model that consists in removing negligible stamp elements P_i. The removal of any element P_i always leads to the simplification of resulting symbolic formula in case that all parameters have unique names.

```
compute reference solution;
while e < e_max {
  generate prospective operations;
  rank the operations;
  execute selected operations;
  update numerical solution and e;
}
undo last operation;
```

Fig. 6 Main loop of SBG method.

The process of simplification of network equations of the SBG method can be seen as a sequence of elementary operations. Each operation corresponds to the removal of one stamp element P_i. The simplification is carried out in a loop, Fig. 6. First, all prospective operations are ranked according to the error the removal would cause. One or more operations with the lowest error are actually performed. The simplification stops when the maximum error is reached, i.e. when condition (7) is violated. The result is a simplified network model, which is then analyzed symbolically.

Let us consider network equations (3)

$$\mathrm{H}\mathbf{x} = \mathbf{0}\ ,$$

which were formulated using the generalized modified nodal method. Any coefficient of **H** is the sum of stamp elements.

Each network function can be expressed as a ratio of two algebraic cofactors, i.e. as a ratio of two subdeterminants of **H** with a suitable sign

$$H = (-1)^{\alpha}\frac{\det(\mathbf{H}_1)}{\det(\mathbf{H}_2)}\ . \tag{8}$$

Matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ are derived from **H** by means of adding or deleting some rows and columns, see Tables 2 and 3, [17]. Considering a stamp element P on position (i, j) in matrix **H** we can write

$$\det(\mathbf{H}_1) = \Delta' + P\Delta^{[i>0:\,j>0]}\ , \tag{9}$$

where Δ' is the determinant without P, and $\Delta^{[i>0:\,j>0]}$ is a cofactor. The same holds for $\mathbf{H}_2$. Generally, the element P appears on different positions in $\mathbf{H}_1$ and $\mathbf{H}_2$. The stamp elements are considered different for the simplification process despite the fact that they can be marked with the same symbol. Then, according to (9), any network function can be written as

$$F = \frac{aP + b}{cP + d}\ , \tag{10}$$

where a, b, c, d are complex numbers. For example, if P represents an admittance, then shorting the element corresponds to $P \to \infty$ and removing the element corresponds to $P \to 0$. From (10) we can easily derive

$$\lim_{P\to\infty} F = \frac{a}{c}\ , \quad \lim_{P\to 0} F = \frac{b}{d}\ . \tag{11a,b}$$

The new value of network function for operation ranking can be computed effectively using (11). To do so we need all numerical cofactors of $\mathbf{H}_1$ and $\mathbf{H}_2$.

A matrix inversion can be formally expressed using the adjoint matrix as

$$\mathbf{A}^{-1} = \frac{1}{\Delta}\begin{bmatrix} \Delta^{[1>0:1>0]} & \cdots & \Delta^{[n>0:1>0]} \\ \vdots & & \vdots \\ \Delta^{[1>0:n>0]} & \cdots & \Delta^{[n>0:n>0]} \end{bmatrix} = \frac{1}{\Delta}\left(\mathbf{A}_{\Delta}\right)^T, \tag{12}$$

where $\mathbf{A}_\Delta$ is the matrix of algebraic cofactors, which can be expressed as

$$\mathbf{A}_\Delta = \det(\mathbf{A})\left(\mathbf{A}^{-1}\right)^T . \quad (13)$$

As the determinant is a byproduct of matrix inversion, the computation of (13) requires approximately $(n-1)^3/3$ long operations, where n is the dimension of original matrix.

3.2.2 SAG Method

The SAG method removes, step by step, the negligible terms from the full formula as long as condition (7) is met. The term to be removed is selected in such way that its deletion causes the lowest error. It is a standard simplification method. Details can be found in [5].

4 Examples

4.1 Exact Symbolic Analysis

Let us check that the circuit in Fig. 7 [15] containing the negative current conveyor CCII- is a low-pass filter with the parameters defined in the figure description. First of all, we verify the circuit principle, avoiding any non-idealities. Then the modeling of the main real influences will be performed along with the analysis of their mechanisms, using the symbolic and semisymbolic results.

The input/output gates are marked with the components In (input)/Out (output) in the schematic editor. The symbolic results and the pole/zero analysis of the transfer function I_{out}/I_{in} by the SNAP program are summarized in Table 4. It is confirmed in the column "ideal" that the circuit is a second-order low-pass filter with a given transfer function. The corresponding frequency response in Fig. 8 shows the resonance ripple according to $Q = 10$ on the frequency $f_0 = 1$MHz.

In the column R_0, the analysis is put more precisely by taking into consideration the parasitic resistance R_0 of the x terminal of the current conveyor. As known, this resistance implies the undesirable distortion of the frequency response in the transition and stop bands. This phenomenon is confirmed in Fig. 8. The symbolic result in Table 4 offers an explanation: Owing to the resistance R_0, two real zeros have appeared in the transfer function. Additionally, the original complex poles are modified. Zeros represent two additional folds in the Bode plots on the 1.75 MHz and 18 MHz frequencies, and together with the poles they are responsible for the finite stop-band attenuation $1 + R_1/R_2 + R_1/R_0$. To eliminate this phenomenon in a passive way (i.e. without modifying the topology), the condition $R_1 >> R_0$ has to be fulfilled.

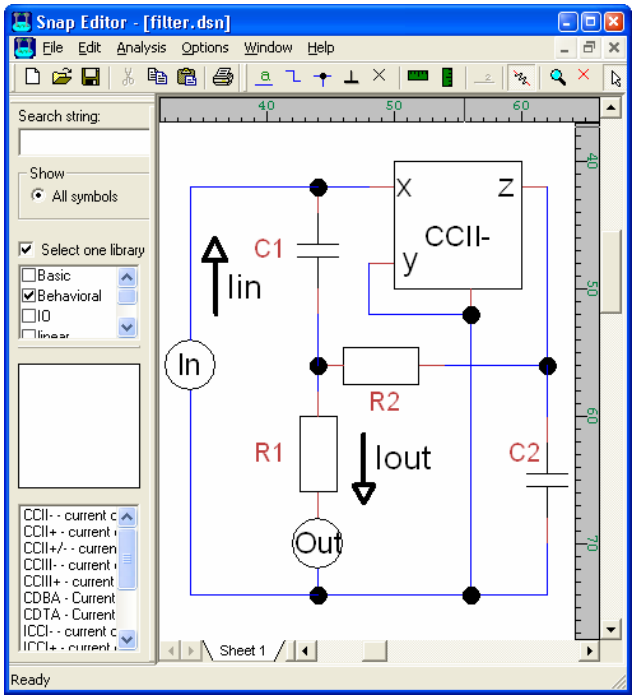


Fig. 7 Second-order current-mode low-pass filter [15], f_0 = 1MHz, Q = 10; $R_1 = R_2$ = 160Ω, C_1 = 20nF, C_2 = 50pF.

Table 4 Analysis results for various models of the CCII: ideal – ideal CCII-; R_0 – resistance of the x terminal; R_t, C_t –components of the parasitic transadmittance of the z terminal.

I_{out}/I_{in}	***ideal***	***R_0 = 5Ω***	***R_0 = 5Ω, R_t=3MΩ, C_t = 4.5pF***
symbol	$\frac{1}{s^2R_1R_2C_1C_2 + sC_2(R_1+R_2)+1}$	$\frac{s^2R_0R_2C_1C_2 + sR_0C_1 + 1}{s^2C_1C_2(R_0R_2 + R_0R_1 + R_1R_2) + s(R_1C_2 + R_2C_2 + R_0C_1)+1}$	*complicated and not relevant*
zeros	-	-1.09612E7 -1.14039E8	-1.10675E7 -1.03617E8
poles	-3.12500E5±j6.24218E6	-2.13235E6 ±j5.67607E6	-1.98363E6 ±j5.45874E6

Analyzing the symbolic equations yields another piece of knowledge: Due to R_0, the absolute values of the complex poles (i.e. frequency $\omega_0 = 2\pi f_0$) are decreased in relation to the squared root of the expression $1 + R_0/R_1 + R_0/R_2$. For our filter, this relation represents falling from 1 MHz to 970 kHz. However, the most important is the drop in Q: In the course of a moderate decrease of ω_0, the real parts of poles were increased significantly. As a result, the quality factor is reduced from 10 to 1.42 (see Fig. 8).

The last column in Table 4 shows the analysis results, taking into account not only the parasitic resistance R_0, but also the output transadmittance of the CCII. It should be noted that this influence is negligible compared to the R_0 effect, which is dominant.

The given analysis indicates that this topology is not suited for filters with higher quality factors in the frequency range near 1 MHz.

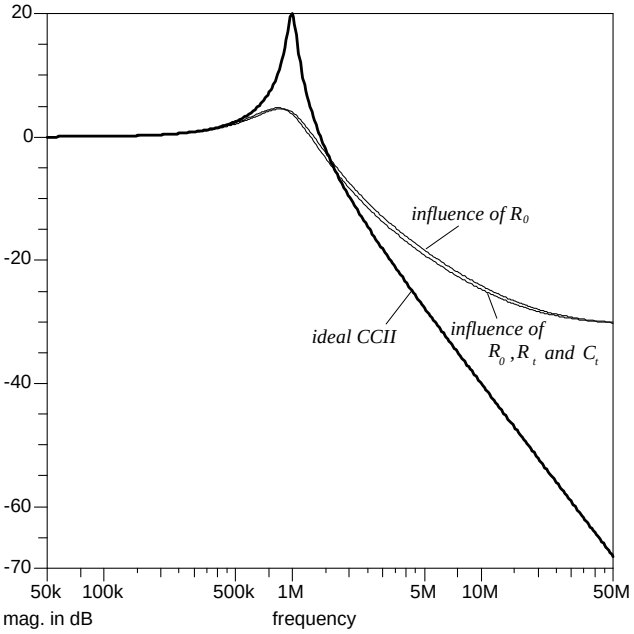


Fig. 8 SNAP outputs: frequency responses for various models of CCII-.

4.2 Approximate Symbolic Analysis

The approximate symbolic analysis is a tool that allows analyzing large transistor-level circuits. The knowledge of all parameters of network elements is a necessary condition for the analysis. A combination of SBG and SAG techniques together with computational power of modern PCs allow analyzing circuits with tens of transistors.

Fig. 9 shows the internal structure of a new network element CDTA implemented in the TSMC 0.35um CMOS technology [18]. CDTA contains an input Current Differencing Unit followed by a multiple-output OTA. The ideal behavior is described by the equations

$$V_p = V_n = 0,\ I_z = I_p - I_n,\quad I_{x+} = g_m V_z,\ I_{x-} = -g_m V_z \qquad (14)$$

The aim of the approximate symbolic analysis is to express the parasitic properties of CDTA by means of small-signal parameters of transistors.

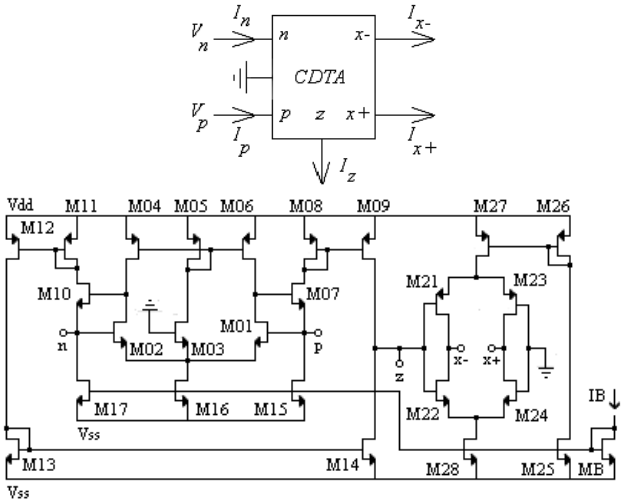


Fig. 9 Symbol of CDTA and its implementation in TSMC 0.35um CMOS technology [18].

Small-signal parameters of all transistors were obtained by means of SPICE operational point analysis. All transistors were modeled in SNAP using a standard SPICE small-signal model, Fig. 10.

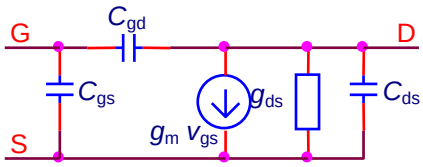


Fig. 10 Small-signal transistor model used for symbolic analysis.

The number N of symbolic terms of the nodal matrix determinant, i.e. the number of terms in the denominator of network functions, can be estimated from

$$\left|\det\left(\mathbf{A}_I \mathbf{A}_V^T\right)\right| \le N \le \min\left(\det\left(\mathbf{A}_V \mathbf{A}_V^T\right), \det\left(\mathbf{A}_I \mathbf{A}_I^T\right)\right) \qquad (15)$$

where $\mathbf{A}_I$ is the incidence matrix of current graph and $\mathbf{A}_V$ is the incidence matrix of voltage graph [19]. For the circuit from Fig. 9 we obtain

$$1.22\ 10^{15} \le N \le \min\left(3.75\ 10^{15}, 5.86\ 10^{15}\right).$$

Thus the number of symbolic terms can be estimated at 10^{15}.

Fig. 11 shows the magnitude of the input impedance of current terminal *p* while the other terminals are grounded. The impedance of terminal *n* is similar.

The following control points for the analysis of input impedance were used in SNAP:

f_1 = 1Hz ΔM_1 = 1dB ΔP_1 = 10°
f_2 = 100MHz ΔM_2 = 3dB ΔP_2 = 20°

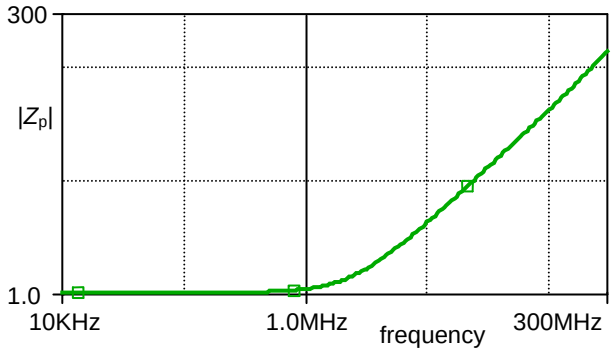


Fig. 11 Magnitude of input impedance of terminal *p*.

The simplification took 2 seconds on a 2GHz Dual-Core Pentium processor. The SBG method left the following parameters (stamp elements) in the simplified circuit: g_{m01}, g_{ds01}, g_{m03}, g_{m05}, g_{m06}, g_{ds06}, g_{m07}, C_{gs07}, C_{gd07}, g_{m08}.

The resulting simplified formula for the input impedance at terminal *p* represents a series connection of low-frequency resistance R_p and equivalent inductance L_p

$$Z_p = R_p + sL_p \ , \tag{16}$$

where

$$R_p = \frac{g_{ds01}}{g_{m01}g_{m07}} + \frac{g_{ds06}g_{m05}(g_{m01}+g_{m03})}{g_{m01}g_{m03}g_{m07}(g_{m05}+g_{m06})} \approx 1\ \Omega,$$

and

$$L_p = \frac{g_{m05}(g_{m01}+g_{m03})(C_{gs07}g_{m08}+C_{gd07}g_{m07})}{g_{m01}g_{m03}g_{m07}g_{m08}(g_{m05}+g_{m06})} \approx 65\text{nH}$$

The analysis of input impedance of terminal *z* with the other terminals grounded yielded

$$Z_z = \frac{1}{g_{ds09}+g_{ds14}+s(C_{gs21}+C_{gs22})}, \tag{17}$$

representing 73kOhm resistor in parallel with 87fF capacitor, with the following control frequencies taken into consideration:

f_1 = 100kHz ΔM_1 = 1dB ΔP_1 = 5°
f_2 = 700MHz ΔM_2 = 2dB ΔP_2 = 20°

5 Conclusions

The paper describes the theoretical principles of a program that allows the modeling and analysis of linearized circuits with modern active elements on the basis of symbolic calculations. The program usability was demonstrated for behavioral modeling of idealized circuits used to verify their basic functions and estimate the influence of dominant real parameters, on the basis of exact symbolic analysis. Furthermore, it was shown that the algorithms of approximate symbolic analysis allow analyzing large circuits whose linearized models can be extracted directly from SPICE models.

SNAP is available at
http://www.urel.feec.vutbr.cz/~kolka/snap/

6 Acknowledgements

This work is supported by the Grant Agency of the Czech Republic under grant No. 102/08/0784, by the research programmes of BUT MSM0021630503, MSM0021630513, and by the research programme of UD Brno MO FVT0000403.

References:

[1] FERRI, G., GUERRINI, N.C. *Low-Voltage Low-Power CMOS Current Conveyors*. Cluwer Academic Publishers, 2003.

[2] ACAR, C., OZOGUZ, S. A New Versatile Building Block: Current Differencing Buffered Amplifier. *Microelectronics Journal*, 30 (1999), pp. 157-160.

[3] BIOLEK, D. CDTA – Building Block for Current-Mode Analog Signal Processing. In: *Proceedings of the ECCTD03*, Krakow, Poland, Vol. III, pp. 397-400.

[4] CHEN, J.J., TSAO, H.W., CHEN, C. Operational transresistance amplifier using CMOS technology. *Electronics Letters*, Vol. 28, No. 22, pp. 2087-2088, 1992.

[5] FERNÁNDEZ, F.V., RODRIGUES-V.A., HUERTAS, J.L., GIELEN, G.G.E. *Symbolic Analysis Techniques*. IEEE Press, New York, 1998.

[6] GIELEN, G., WAMBACQ, E, SANSEN, W. Symbolic analysis methods and applications for analog circuits: a tutorial overview. *Proc. IEEE*, Vol. 82, pp. 287-304, Feb. 1994.

[7] BIOLEK, D., BIOLKOVÁ, V., DOBEŠ, J. (Semi) Symbolic Modeling of Large Linear Systems: Pending Issues. In: *Intern. Conference ISSSE2001*, Tokyo, Japan, pp. 397-399.

[8] SANCHES-LOPEZ, C., TLELO-CUAUTLE, E., Behavioral Model Generation for Symbolic Analysis of Analog Integrated Circuits, *Proc. IEEE ISSCS*, Iasi, Romania, July 14-15, vol. 1, pp. 327-330, 2005.

[9] POLDERMAN, J.W., WILLEMS, J.C. *Introduction to Mathematical Systems Theory: A Behavioral Approach*, 424 pages, Springer, New York, 1998.

[10] www.analog-insydes.com

[11] www.tina.com

[12] GIELEN, G., WAMBACQ, E, SANSEN, W. ISAAC: a symbolic simulator for analog integrated circuits. *IEEE J. Solid-State Circuits*, Vol. SC- 24, pp. 1587-1597, Dec. 1989.

[13] TLELO-C.E., QUINTANAR-R. A., GUTIÉRREZ-P.G., GONZÁLEZ, M. SIASCA: Interactive System for the Symbolic Analysis of Analog Circuits. *IEICE Electronics Express*, Vol. 1, No. 1, 2004, pp. 19-23.

[14] LUCHETTA, A., MANETTI, S. SAPWIN – A Symbolic Simulator as a Support in Electrical Engineering Education. *IEEE Trans. on Education*, Vol. 44, No. 2, May 2001, p. 213.

[15] FABRE, A., DAYOUB, F., DURUISSEAU,L., KAMOUN, M. High input Impedance Insensitive Second-Order Filters Implemented from Current Conveyors. *IEEE Trans. on CAS-I*, Vol.41, No.12, 1994, pp. 918-921.

[16] VLACH, J., SINGHAL, K. *Computer Methods for Circuit Analysis and Design*. New York: Van Nostrad Reinhold, 1994.

[17] ČAJKA, J., KVASIL, J. *Theory of linear circuits*. SNTL Prague, 1979.

[18] BIOLEK, D., HANCIOGLU, E., KESKIN, A.U. High-performance current differencing transconductance amplifier and its application in precision current-mode rectification. *AEU - International Journal of Electronics and Communications*, Vol. 61, available online at www.sciencedirect.com.

[19] CHEN, W.K. *Applied Graph Theory*. North Holland Publishing Co., Amsterdam, 1971.