Analog Integrated Circuits and Signal Processing, 31, 131–145, 2002

Approximate Symbolic Analysis of Hierarchically Decomposed Analog Circuits

O. GUERRA,[1] E. ROCA,[1] F. V. FERNÁNDEZ[1,2] AND A. RODRÍGUEZ-VÁZQUEZ[1,2]

[1]Instituto de Microelectrónica de Sevilla, Centro Nacional de Microelectrónica, Edificio CNM, Avda. Reina Mercedes s/n., E-41012-Sevilla, Spain, Tel.: 34 955056666, Fax: 34 955056686

[2]Área de Electrónica, Escuela Superior de Ingenieros, Edificio Plaza de América, Isla de la Cartuja s/n., E-41092-Sevilla, Spain, Tel.: 34 954487378, Fax: 34 954487379

E-mail: pacov@imse.cnm.es

Received June 15, 1999; Revised September 25, 2000; Accepted May 1, 2001

Abstract. This paper presents a methodology for the symbolic analysis of large analog integrated circuits using a hierarchical approach. The drawbacks of previous approaches are solved by the introduction of error-controlled approximation strategies. A proper modeling methodology through the different hierarchical levels allows to combine the optimum techniques for generation of the symbolic expressions and the most efficient numerical techniques for error control. These approximation strategies together with mechanisms for partitioning and union of blocks through the hierarchy yield optimum results in terms of speed, accuracy and complexity of the symbolic results.

Key Words: analog circuits, symbolic analysis, simplification techniques, hierarchical approximation

1. Introduction

Symbolic analysis is devoted to the analysis of integrated circuits in which part or all their elements and even the complex frequency variable are represented by symbols. Its use to provide insight into circuit operation, generate behavioral models, design equations, etc., is well-known [1].

The application of symbolic analyzers has traditionally suffered from the exponential growth of the complexity of the symbolic results with the circuit size. This has motivated the introduction of approximation techniques [1]. Their objective is to provide usable analytical models by generating the simplest expression, that keeps the dominant behavior of a circuit, in the shortest time.

Approximations are based on the elimination of the least significant parts of the symbolic solutions, based upon the large differences between the numerical values of the different circuit parameters. Modern symbolic analyzers incorporate approximation techniques which fall into one or more of the following categories: (1) Simplification After Generation (SAG) techniques prune the least significant symbolic terms or subexpressions after the complete expression has been computed; (2) Simplification Before Generation (SBG) techniques directly simplify the system of circuit equations; (3) Simplification During Generation (SDG) techniques calculate only the dominant contributions of the system solution.

There is not a clear limit for the applicability of these approximation techniques for non-decomposed circuit analysis,[1] because this depends, not only on the circuit size (number of devices/nodes, connectivity, complexity of the small-signal model for the transistors, amplifiers, etc.), but also on the frequency range in which the analysis has to be performed and the specified error constraints. However, it can be roughly estimated as graphically displayed in Fig. 1.[2]

When the circuit size grows beyond the limits of non-decomposed analysis capabilities, new techniques that apply different methodologies for the analysis and generation of symbolic expressions are needed. Divide-and-conquer approaches are commonly used by designers when facing the design of complex functionalities. Therefore, hierarchical symbolic analysis techniques, that can get benefit from the inherent hierarchy present in large circuits, may enable the symbolic analysis of very large integrated circuits.

However, previously reported approaches for hierarchical symbolic analysis [2–7] are not appropriate enough for the analysis of practical large circuits. No approximation is applied and, consequently, the expression complexities grow exponentially with the

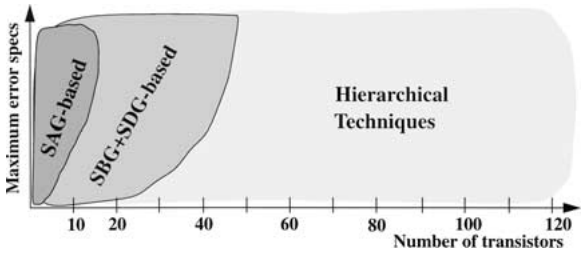

Fig. 1. Illustrating the range of applicability for different symbolic analysis approaches.

circuit size, although, obviously, at a much smaller pace than with non-decomposed analysis approaches. Then, interpretability is drastically reduced and, more importantly, analyzable circuit sizes are strongly limited unless extremely simple models are used or a high number of hierarchical levels is chosen. This is corroborated by the fact that reported experimental results use extremely simple block models; i.e., filters using ideal models for the operational amplifiers.

This paper overcomes this problem by introducing a new hierarchical analysis methodology for analog circuits arbitrarily described at any description level, which incorporates error-controlled approximation techniques. The introduction of a modeling strategy in terms of (trans)admittances at each hierarchical level allows to combine very efficient graph methods for the generation of the symbolic expressions and numerical solution of system matrices for error evaluation and control.

Section 2 presents the basics of hierarchical analysis techniques and reviews the most important reported approaches. Their limitations motivate the introduction of a new strategy for symbolic analysis of hierarchically-decomposed circuits in Section 3. The main parts of this methodology are presented in detail in Sections 4 to 6. Finally, the quality of the proposed methodology is demonstrated in Section 7 through the analysis of some practical large circuits.

2. Hierarchical Analysis Techniques

Reported hierarchical analysis approaches divide the analysis process into three parts [8], which are illustrated in Fig. 2:

- **Circuit Partitioning,** where the circuit is divided into smaller blocks, preferably in such a way that devices in the same block are strongly interconnected while those in different blocks are weakly interconnected. Depending on the developed approach, this partitioning step can be performed at the circuit level or at the graph level. This is not an intrinsic step to any hierarchical analysis methodology because,

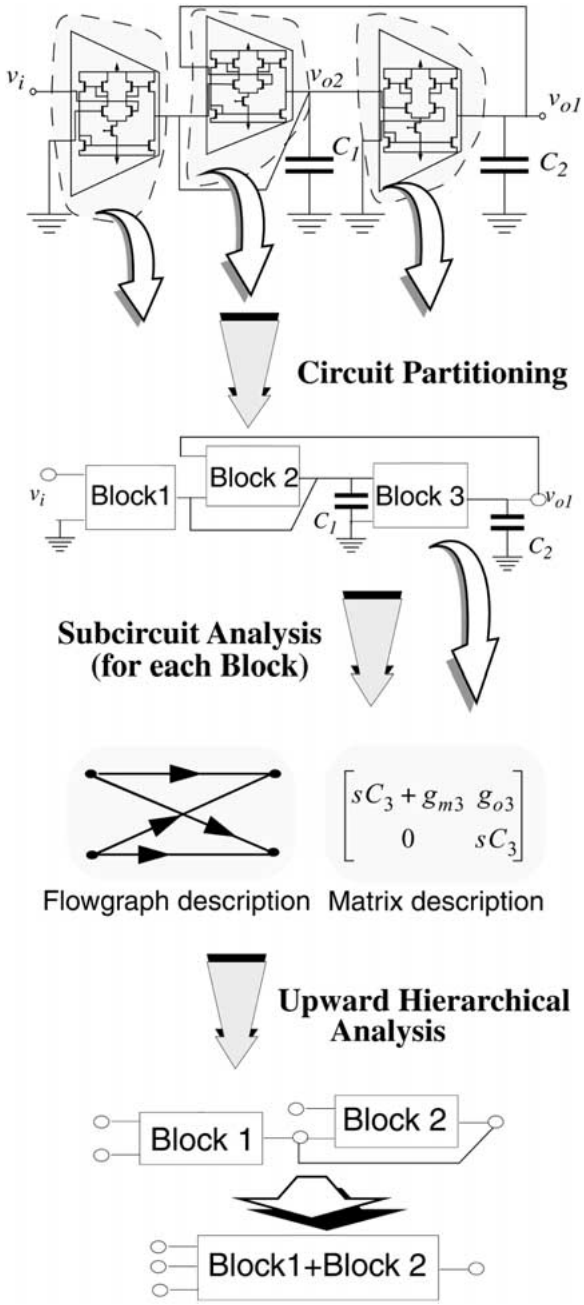

Fig. 2. Hierarchical analysis main steps.

in most cases, as the one depicted in the example of Fig. 2, the hierarchical partitioning arises naturally, as the designer usually benefits from a constructive process which builds complex circuits from the interconnection of simpler ones.

- **Subcircuit Analysis,** commonly known as **Terminal Block Analysis.** At this step, each subcircuit is analyzed symbolically, providing descriptions that can be formulated either in matrix form or in graph form. The goal of this step is to model each block only in terms of its inputs and outputs; therefore, nodes that are internal to each block mean variables that are eliminated from the equations, yielding a simpler description for the entire circuit. For instance, the description involving Block 3 in Fig. 2 yields a flowgraph with two external nodes or a second-order matrix, once the appropriate eliminations of internal variables are performed.
- **Upward hierarchical analysis,** also known as **Middle Block Analysis,** where the solutions for the blocks are combined to find the symbolic solution corresponding to the entire circuit. The procedure is to combine descriptions of blocks that have one or more *terminal nodes* in common; these nodes may become internal, providing new variable eliminations. This recombination is performed iteratively while ascending levels in the hierarchy until the complete circuit, described in terms of the global inputs and outputs, is reached.

Instead of the conventional output of non-decomposed symbolic analysis techniques in the form of a single symbolic network function, the natural symbolic result of a hierarchical analysis methodology is a *Sequence of Expressions* (SOE), which contains multiple expressions with a hierarchical dependence on each other [8]. An example of a Sequence of Expressions can be found in Section 7.1.

This paper focuses on the development of Terminal and Middle Block analysis techniques capable of incorporating approximation strategies. Therefore, circuits will be considered to be previously partitioned.

For Middle and Terminal Block analysis there exist basically three approaches (see [8] for an excellent review):

- Based on *Coates flowgraph analysis* [2,3].
  Although the Coates flowgraph method is not widely extended in electrical applications, the first reported hierarchical symbolic analysis approach was based on it. The method is based on the enumeration of the multiconnections in a block and the successive elimination of some nodes [3].
- Based on a *Mason flowgraph analysis* of subcircuits partitioned at the circuit level [4,5].
  Mason signal flowgraph techniques [9] are used to analyze each partition yielding a description in terms of Mason's graphs. This leads to a signal flowgraph description based only on input and output nodes for each block by means of successive reductions of internal nodes according to some rules. Finally, a recombination of the blocks is made applying, if necessary, the same reduction techniques, until a reduced description for the entire circuit is reached.
- Based on *Modified Nodal Analysis* [6,7].
  The method is called Direct Network because it operates directly on the network matrix level rather than on the flowgraph level. This method operates by reducing the *Modified Nodal Admittance* (MNA) *Matrix* representing each subblock, into a *Reduced Modified Nodal Admittance* (RMNA) *Matrix,* which only depends on terminal nodes of the subblock. This is made by progressive recombination of some elements associated to internal nodes of the subblocks. A similar process is repeated to combine different subblocks when progressing up in the hierarchy, until an RMNA matrix defining the entire circuit is obtained [6,7].

3. Hierarchical Analysis Approach and Approximation Strategy

None of the analysis methods in Section 2 incorporate approximation techniques. This exceedingly hampers their application to large practical circuits. Moreover, the result of node reductions in the Mason and Coates flowgraph methods and matrix operations in the Direct Network method is usually a large sequence of small expressions. This makes the approximation of an expression of the sequence by pruning its least significant parts more difficult, especially if the error introduced has to be controlled. Given the capabilities of existing approximation techniques to extract only significant portions of large expressions, on the one hand, and the difficulties associated to propagation of errors introduced by approximation through lengthy sequences of expressions, on the other, it would be more convenient to operate on smaller sequences of formally larger expressions.

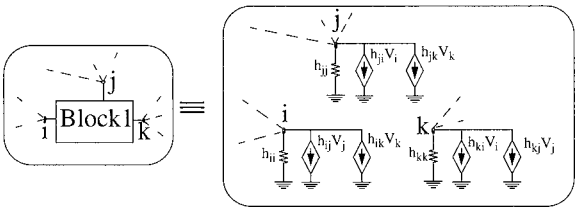

Fig. 3. Modeling each block during hierarchical analysis.

A characteristic of the hierarchical analysis techniques in Section 2 is that, after some transformations, blocks at any hierarchical level are described using terminal node variables. For instance, column and row operations are performed on the MNA matrix, which describes a block at any hierarchical level in the Direct Network method, to obtain an RMNA matrix which describes such block only in terms of the variables associated to terminal nodes. In our methodology, blocks at each hierarchical level will also be modeled in terms of their terminal nodes. This means that, if an admittance description is selected, a model containing admittances and transadmittances will be formulated, as Fig. 3 shows for a three-terminal block.

The techniques by which the Direct Network Method obtains each element of the RMNA matrix starting from the complete MNA matrix are probably the optimum ones to get a complete symbolic expression of each element of the RMNA matrix as a function of the elements of the MNA matrix. But they are not suitable to incorporate approximation techniques. However, many results from the mature approximation techniques existing for non-decomposed circuit analysis can be applied to obtain approximate symbolic expressions for each (trans)admittance of a given block [10–12] in Fig. 3 as a function of the devices composing such block.

When symbolically analyzing a circuit, a set of network equations (topological and constitutive relationships) has to be solved. For the approximate symbolic solution of a set of linear equations, graph methods have proven to be superior [10]. At any hierarchical level, the circuit is composed of an interconnection of devices and/or blocks which are modeled as in Fig. 3. Conventional graph-based non-decomposed analysis techniques, like the two-graph method [13], can be used at this hierarchical level because once the corresponding (trans)admittances replace each block, the circuit is a flat interconnection of basic circuit elements, but with a much smaller number of nodes and devices than a non-decomposed approach.

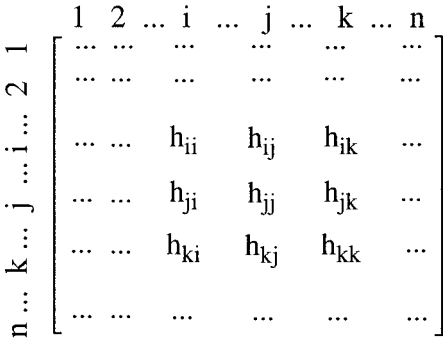

Fig. 4. Matrix with previous block information.

However, an approximate solution must be accompanied of an evaluation of the committed errors, and error evaluation must be performed numerically. But for a numerical evaluation of the errors, the equations must also be solved and this is performed much more efficiently using sparse matrix techniques [14]. An MNA matrix description of the circuit at any hierarchical level can be easily formulated using stamps, as Fig. 4 illustrates for a circuit containing the block in Fig. 3. Therefore, our hierarchical analysis strategy will be based on graph techniques for the symbolic solution of circuit equations and matrix techniques for the numerical solution of circuit equations.

The practical implementation of the methodology is composed of several modules. Its flow diagram is shown in Fig. 5.

The needed data at the input is a hierarchical circuit description, network function to calculate, error constraints and frequency ranges. The input circuit is first checked for connectivity, appropriate excitations, consistency of output variables, etc. Then, a circuit reduction is carried out, consisting in performing node contractions and device removals (illustrated in Fig. 6) whose contribution to the global circuit behavior (network function at hand) is negligible. This procedure

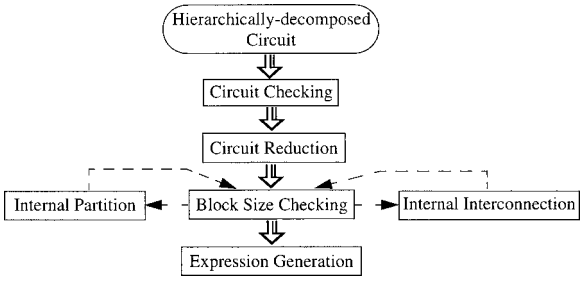

Fig. 5. Module structure.

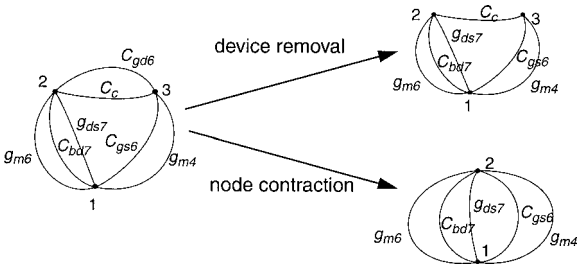

Fig. 6. Examples of branch removal and node contraction operations on the graph level.

imitates the simplifications commonly performed by designers when facing a complex analysis task. Obviously, the error introduced by such circuit transformations must be carefully controlled.

Then, the appropriate network functions must be generated at each hierarchical level. But prior to this, the sizes of the resulting blocks after the circuit reduction process are checked. The motivation for this is that some blocks may contain such a small number of elements that their independent analysis is very inefficient. By the contrary, some blocks may be still too large for the subsequent analysis algorithms and a further partitioning may dramatically increase the efficiency. According to this size checking, blocks are further partitioned or several ones are joined together into a single block.

Now, the hierarchically decomposed circuits are ready for symbolic generation of the network functions (admittances, transadmittances) of each block as a function of the composing elements of that block. Such network functions are not fully generated but only the dominant part of them. Therefore, a term generation in decreasing order of magnitude is needed for each network function at each hierarchical level, and an appropriate error control mechanism to decide the number of terms that must be generated for each network function to satisfy the error constraints on the global circuit behavior.

4. Circuit Reduction

The objective of the circuit reduction process is to find the sequence of node contractions and device removals yielding the simplest circuit (smallest number of nodes and devices) and whose induced error keeps below some given threshold [12]. First, it must be discussed if this circuit reduction mechanism should be applied hierarchically to the different subcircuits, or to the complete expanded circuit.

If performed hierarchically, two basic approaches can be conceived:

- **Based on error propagation.** Allowed errors are propagated to each block and circuit reduction is applied to each one separately. Assuming that the transfer function v_{o1}/v_i of the hierarchical circuit in Fig. 2 must be calculated, the error constraints must be propagated to each of the 17 (trans)admittances which model the subblocks (n^2 (trans)admittances for each block with n terminal nodes). Then, a circuit reduction step for each input/output configuration, using the previously propagated errors, must be performed. This necessarily yields more conservative results (more complex circuits) and, consequently, has a negative impact on the global performance of the analysis methodology. Therefore, this approach will not be considered in our discussion.
- **Based on error contribution.** A mechanism to avoid a conservative error propagation is to perform circuit transformations (node contractions and device removals) at each block according to the error introduced in the complete system. The error evaluation of each circuit transformation implies the extra cost of solving the MNA matrices at each hierarchical level.

On the other hand, a non-decomposed circuit reduction is not against the hierarchical analysis philosophy. Notice that from the two main analysis tasks in Fig. 5: Circuit Reduction and Term Generation, the latter explodes much faster with the number of devices and nodes in the circuit, and, therefore, it is this task which promptly needs a hierarchical decomposition.

Error evaluation in the non-decomposed approach is performed by numerically solving the MNA matrix representing the circuit, in this case, the complete expanded circuit. Such matrix is solved using very efficient sparse matrix techniques. This means that, except for very high order matrices, the solution of this single matrix is more efficient than the solution of a relatively large number of smaller matrices (corresponding to a hierarchical circuit reduction based on error propagation) [15]. Therefore, a non-decomposed approach is the most efficient one for the usual sizes involved and only for extremely large circuits a hierarchical approach based on error contributions becomes advantageous. Although the circuit reduction mechanism is applied to the non-decomposed circuit, the original

hierarchical structure is formally kept, so that when the reduction process is finished, nodes and devices can be identified and the hierarchical structure is rebuilt. Obviously, the resulting blocks are simpler than the original ones and some of them may even have disappeared.

After the circuit reduction process, the resulting simplified circuit must be analyzed. The most efficient approximate symbolic analysis algorithms are based on matroid theory and their computational complexity grows much faster with the number of circuit nodes than with the number of devices [11]. For this reason, node contractions are given priority in our circuit reduction algorithm.

The following steps summarize the algorithm operation:

1. Compute the contribution to the network function of the contraction of the terminal nodes of each device individually and build a ranked list.
2. Pick the less significant contraction from the list and compute the error committed.
3. If the error has not been exceeded, perform the node contraction, remove from the contraction list all devices connected between that pair of nodes, reorder the contraction list and go to step 2; otherwise continue with step 4.
4. Compute the contribution to the network function of the removal of each branch and build a ranked list.
5. Pick the less significant device removal from the list and compute the error committed.
6. If the error has not been exceeded, perform the device removal, reorder the removal list and go to step 4; otherwise finish.

As shown, (magnitude/phase) errors introduced by node contractions and device removals must be evaluated. Errors are not numbers but functions of the frequency. The solution in previous approaches has been to evaluate the errors at a set of sample frequency points [16,17]. A major drawback is that accuracy is not guaranteed at frequencies different from those in the set of sampling frequencies. The accuracy problem could be partially palliated with a dense sampling, but this increases noticeably the computational cost of the algorithm. Moreover, experience shows that many contractions and removals introduce very steep slopes in the magnitude and phase error curves, and, therefore, an error excess may keep hidden even in a dense sampling.

To achieve optimal efficiency of the algorithm without any possibility of sacrifying accuracy, a new strategy has been developed [12]. Initially, a reduced set of frequency samples is selected. The algorithm steps above are executed for each frequency sample. At each one, error evaluations are performed using very efficient sparse matrix techniques. If a node contraction or device removal can be performed for all frequency samples, it is executed. It is clear that the error constraints are met at the frequency samples but may be violated between them. This possible error excess is evaluated using the algorithm in Section 4.1 and, if the error constraints are violated, the frequency $\omega_{\max}$ at which the maximum error occurs is determined and the possible contractions and removals are evaluated at a set of frequency samples in which $\omega_{\max}$ has been added. This procedure is recursively repeated until the error constraints are not violated at any frequency in the desired range.

4.1. Algorithm for Maximum Error Detection

The proposed methodology requires the evaluation of the magnitude and phase errors introduced by some circuit transformation in some given frequency range. This means that the difference between the magnitude and phase behaviors of the original circuit and those of a simplified circuit, obtained from the original one by performing some node contractions and/or device removals, has to be evaluated.

Let us denote $H_{ex}(s)$ the network function of the complete circuit with only the complex frequency s as symbolic parameter, and $H_{ap}(s)$ the corresponding network function of the simplified circuit. The magnitude and phase errors are given by:

$$\Delta|H| = \frac{|H_{ex}(j\omega)| - |H_{ap}(j\omega)|}{|H_{ex}(j\omega)|} \qquad (1)$$

$$\Delta\phi_H = \angle H_{ex}(j\omega) - \angle H_{ap}(j\omega)$$

A direct method is to evaluate equation (1) at a number of frequency samples but errors then may be largely underestimated because the behavior between a pair of frequency samples remain unknown.

A two-step methodology has been developed for the evaluation of the maximum magnitude and phase errors. This methodology consists in the calculation of: (a) the network functions $H_{ex}(s)$ and $H_{ap}(s)$ of (usually large) analog circuits; and, (b) the maxima of equation (1) when ω varies within a given range.

4.1.1. Network Function Calculation. The first problem is a symbolic network function calculation with only the complex frequency s as symbolic variable. Among the reported symbolic analysis techniques, the numerical interpolation method is acknowledged to be the most efficient one when the complex frequency is the only symbolic parameter [13]. However, this method is known to suffer from numerical problems due to the finite number of bits representing each real number in computers. Although this limits the number of network function coefficients that can be determined accurately, [18] has introduced an adaptive scaling method that allows to calculate network functions with a large number of coefficients (corresponding to large circuits) using the smallest number of iterations of the numerical interpolation method.

4.1.2. Error Maxima in a Frequency Range. This problem can be easily modeled using interval analysis techniques [19]. In interval analysis, the basic element is the interval: a closed bounded set of real numbers, which can be basically written as:

$$[a, b] = \{x | (a \leq x \leq b)\} \tag{2}$$

Interval operators can be easily defined as an extension of real arithmetic operators, yielding the sum of intervals, product of intervals, etc.

Detecting the error maxima in a frequency range $\omega \in [\omega_L, \omega_U]$ reduces to determining the interval of all values of the functions $\Delta|H|(\omega)$ and $\Delta\phi_H(\omega)$ in equation (1) when ω varies in $[\omega_L, \omega_U]$. The problem is solved if sharp bounds of the range of the functions: $\Delta|H|$ and $\Delta\phi_H$, can be calculated.

Substitution of the real variable ω in equation (1) by the corresponding interval variable and real operators (addition, product, quotient, etc.) by the corresponding interval operators yields the so-called natural interval extension [19]. Unfortunately, computing this interval extension for a given interval variable, $[\omega_L, \omega_U]$, usually overestimates the interval bounds of $\Delta|H|$ and $\Delta\phi_H$ (and, therefore, the errors) [20].

To solve this problem, the proposed methodology uses the derivatives of equation (1). The calculation of accurate estimates of the interval bounds of $\Delta|H|$ and $\Delta\phi_H$ reduces to obtaining the values of ω which make the derivatives equal to zero (real roots of the derivatives), and evaluate $\Delta|H|$ and $\Delta\phi_H$ in equation (1) at those values of ω. The calculation of such frequency values is done by a proper combination of:

- An interval extension of the derivatives to delimit frequency subranges in which the maximum magnitude and phase errors occur. Although the interval width obtained from such interval extension is also overestimated, this is not a major problem because the only information needed is to know if zero is included in the interval.
- A Newton-Raphson technique to determine the exact frequency values with maximum magnitude/phase errors. The use of interval analysis techniques to delimit subranges containing the critical frequencies is necessary to obtain proper convergence of the Newton-Raphson technique.

5. Internal Partition/Interconnection

After the circuit reduction process, the hierarchical structure is reconstructed if such reduction was performed on the non-decomposed circuit. Then, all blocks are checked to detect too simple or too complex ones.

If a simplified block contains a reduced set of devices or internal nodes, keeping that subcircuit as an isolated block becomes disadvantageous in terms of efficiency in the computations and complexity of the final sequence of expressions. This typically occurs when biasing subcircuits are included in the netlist. In this case, this block is joined to its best neighbor or incorporated into its immediately upper hierarchical level.

After this step, the different subblocks must be analyzed using some SDG algorithm. It is well-known that the complexity of these algorithms grows exponentially with the circuit size. Therefore, if the subcircuit to analyze is too complex, it is convenient to internally partition this subcircuit before proceeding to its analysis. To decide if an internal partitioning is needed, three characteristics of the subcircuit after the circuit reduction are taken into account:

- The number of devices.
- The number of nodes.
- The number of capacitors, which is related with the highest power of s in the coefficients of the corresponding network function.

Both processes, union of blocks and block partition, are not usually critical, that is, the analysis can proceed with the predefined partitions although with smaller efficiency. Taking into account that the user, who

partitioned the circuit, might not agree with these transformations, they are not performed automatically but subject to the user approval.

6. Expression Generation

Once the hierarchical block structure has been rebuilt and checked, a circuit where blocks are modeled in terms of its (trans)admittances is built up. Then, appropriate analysis algorithms generate approximate expressions for each (trans)admittance of each block in the hierarchical structure as a function of the composing devices of that block. Analogous analysis algorithms are applied to obtain the desired network function in terms of the (trans)admittances modeling the blocks at the uppermost hierarchical level. The result of this step is the required network function in SOE format.

6.1. Term Generation

As an approximate network function is desired, analysis techniques able to provide only the dominant part of the symbolic solution are needed. Efficient techniques capable of doing this are available for non-decomposed analysis, in which the circuit to analyze is modeled in terms of two-terminal elements, independent and controlled sources [11]. These same techniques can be applied to our hierarchical approach because each terminal block is a non-decomposed circuit and each middle block is composed of an interconnection of (trans)admittances that model the lower level blocks.

The two-graph tree enumeration method has been shown to be the most efficient technique to implement a term generator in decreasing order of magnitude [10]. The two-graph method makes use of a voltage graph G_V and a current graph G_I, both easily built from the original network [13]. The term generation problem reduces to the *enumeration of spanning trees common to both, the voltage and the current graph, in decreasing order of weight*.

Very efficient algorithms can be applied by modeling the two-graph method in terms of matroid theory. A *matroid* $M = (E, J)$ is a structure in which E is a finite set of *elements* and J is a family of subsets of E, which satisfy some axioms [21]. A subset I in J is an *independent set* of the matroid $M = (E, J)$. A maximal independent set is a *base* of the matroid. A matroid $M = (E, J)$ is said to be the *graphic matroid* of the graph G if E is the set of arcs of G and a subset $I \subseteq E$ is in J if and only if I is a cycle-free subset of arcs. In a connected graph, a base of its graphic matroid corresponds to a spanning tree. A graphic matroid can be defined on the voltage graph G_V, so that each spanning tree in G_V is a base. Another graphic matroid is defined analogously on G_I. The tree enumeration problem corresponds to the enumeration of bases common to both matroids in decreasing order of weight. There are very efficient polynomial-time algorithms for this weighted intersection of two matroids [22].

6.2. Error Control

A term generation process is formally started to generate symbolic expressions for the modeling (trans)admittances of each block as a function of the component devices at the next lower hierarchical level. The only exception is the top level in the hierarchy in which terms are generated for the network function defined by the input and output signals of the entire circuit. The error control mechanism controls which term generators must become active and in which sequence. The operation of the error-controlled term generation is illustrated in Fig. 7.

Initially, a representative frequency value is chosen. Each element has its admittance (purely imaginary in case of reactive elements, and real in resistive ones) as associated weight. The weight associated to each modeling (trans)admittance of any intermediate block is a complex number because functionally depends on all devices composing such block.

The contribution of each (trans)admittance to the global circuit behavior is then numerically evaluated. This is efficiently done using the hierarchical MNA formulation sketched in Section 3 and sparse techniques to solve the MNA matrices. The dominant term of the (trans)admittance with the largest contribution is generated using the techniques in [11]. Contributions of each (trans)admittance are then updated. This allows to decide which is the term generator (corresponding to some (trans)admittance) which must become active.

The error at each step is a combination of the error in each (trans)admittance (only part of it has been generated) and the contribution of such (trans)admittance to the global behavior. The generation process continues iteratively until the error criterion is met. Obvi-

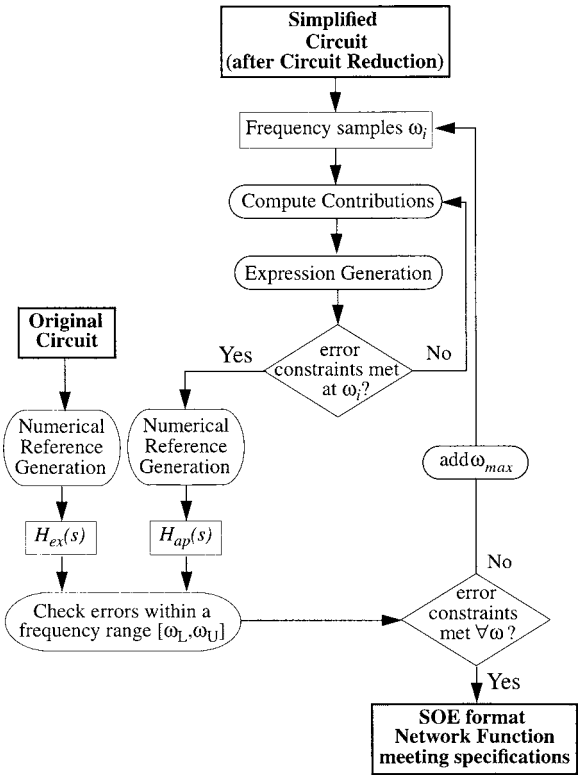

Fig. 7. Error-controlled term generation.

ously, meeting the error specs is guaranteed only at the selected frequency sample. The algorithm for maximum error detection using interval analysis techniques in Section 4.1 is used to detect frequency values where the errors are exceeded. Then, the frequency $\omega_{\max}$ at which the maximum error occurs is determined and the term generation process started again with all devices and (trans)admittances evaluated at this frequency. The resulting sequence of expressions is joined to that obtained at the previous frequency samples. Detection of violations of error constraints and generation at new frequency samples is repeated until the error criteria are met in the required frequency range.

7. Experimental Results

In this section, two examples are analyzed using the proposed hierarchical technique. The first one is a filter, inherently composed of blocks, described at the transistor level, and the second one is the μA741 opamp, which has been partitioned according to functional stages.

7.1. A Band-Pass Filter

The first example is the analysis of the bandpass filter in Fig. 8(a), where the transconductance amplifiers are described at the transistor level, as shown in Figs. 8(b) and 8(c), giving a total number of 83 MOS transistors. The transistors are modeled using the small-signal model in Fig. 8(d). The hierarchical structure in this case has two levels. A symbolic expression for the voltage gain is desired valid in the frequency range 10^4 Hz $\leq f \leq 10^7$ Hz and with the following error constraints: $|\Delta_{\text{mag}}| \leq 1$ dB and $|\Delta_{\text{phs}}| \leq 5$ degrees.

The small-signal model expansion of the circuit yields a circuit model with 618 devices and 45 nodes. The circuit reduction process is performed on the complete expanded circuit, with no hierarchical decomposition. After the circuit reduction, the expanded model is considerably simpler: 67 devices and 26 nodes, that once the circuit hierarchy is reconstructed, are distributed according to Table 1. The accurate error control mechanism introduced in Section 4.1 is crucial here due to the fast magnitude and phase excursions around the central frequency of the filter, as the magnitude and phase plots in Fig. 9 show.

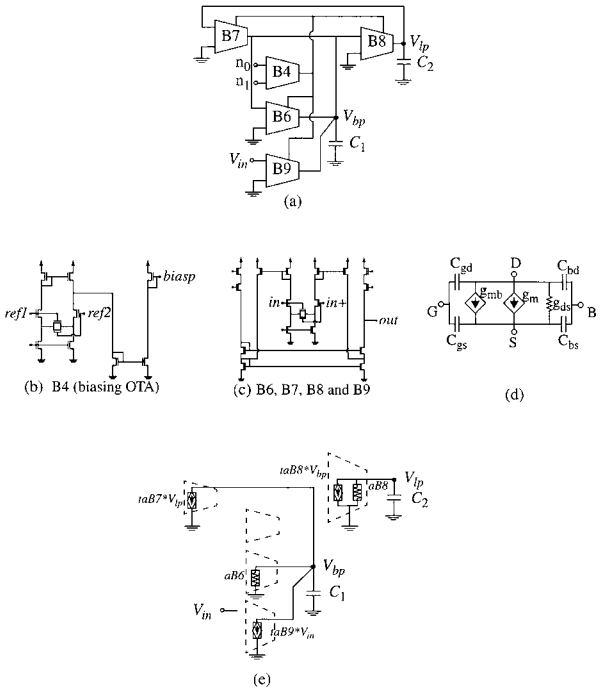

Fig. 8. (a) Bandpass filter; (b) Bias OTA; (c) OTA schematics; (d) small-signal model for the transistors; (e) hierarchical model after the circuit reduction step.

Table 1. Block complexity after the circuit reduction step.

Block	# Devices	# Nodes	# Terminal Nodes
B6	14	7	4
B9	14	7	2
B7	19	8	2
B8	17	8	2
B4	0	0	0

It is interesting to notice that the effect of the circuit reduction process at intermediate levels of the hierarchy is the elimination of many of the terminal (trans)admittances, as the top level model in Fig. 8(e) shows.[3] This already can give extra insight because the terminal characteristics of subblocks which mostly influence the filter behavior can be easily identified.

Then, approximate symbolic expressions are generated at each hierarchical level. The network function corresponding to the top level in the hierarchy is:

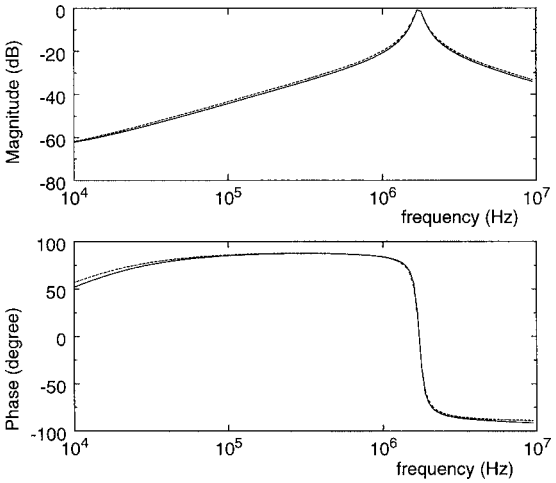

Fig. 9. Comparison between magnitude and phase behavior of the bandpass filter and plots obtained by evaluation of the symbolic expressions.

$$H(s) = \frac{taB9.\text{bp.in}[aB8.\text{lp.lp} + C_2 s]}{-taB7.\text{bp.lp}taB8.\text{lp.bp} + (aB6.\text{bp.bp}C_2 + aB8.\text{lp.lp}C_1)s + C_1 C_2 s^2} \tag{3}$$

The symbolic expressions of the needed (trans)admittances are:

$$\begin{aligned} taB9.\text{bp.in} = {} & gm_0 gmb_{19}\{(gds_2 + gds_3)(gm_{11}gm_{14}gm_{17} + gm_{12}gm_{13}gm_{18}) \\ & + (gm_{14}(gm_{17}Cbd_0(gds_2 + gds_3) + gds_2 gm_{11}Cbd_{17}))s\} \\ & /(gmb_{19}gm_{17}gm_{11}[gm_0 gm_{12} + (gm_0 + gm_{12})(gds_2 + gds_3)] \\ & + gm_{11}gm_{17}Cbs_{19}(gds_2 + gds_3)(gm_0 + gm_{12})s \\ & + gm_0 gm_{12}[gm_{17}(gm_{11}Cbs_{19} + gmb_{19}Cbd_0) + gm_{11}gmb_{19}Cbd_{17}]s) \end{aligned} \tag{4}$$

$$aB8.\text{lp.lp} = gds_{42} \tag{5}$$

$$\begin{aligned} taB7.\text{bp.lp} = {} & gm_{19}gm_0 gm_1(gds_2 + gds_3)[gm_{17}gm_{11}gm_{14} + gm_{12}gm_{13}(gm_{18} + gds_{18})] \\ & /(gm_{17}gm_{19}gm_{11}gm_{12}[gmb_1(gds_2 + gds_3)] \\ & + gm_{17}(gds_{18} + gm_{19})gm_{11}gm_{12}[gm_0(gm_1 + gmb_1) + (gm_0 + gm_1)(gds_2 + gds_3)] \\ & + gm_{17}gm_{19}gm_{20}gm_1(gm_{11}Cbd_1 + gm_{12}Cbd_0)s) \end{aligned} \tag{6}$$

$$taB8.\text{lp.bp} = \frac{-gds_2(gm_{11}gm_{14}gm_{17} + gm_{12}gm_{13}gm_{18})}{gm_{17}gm_1 gm_{12}} \tag{7}$$

$$\begin{aligned} aB6.\text{bp.bp} = {} & gm_1 gmb_{19}(gm_{11}gm_{14}gm_{17} + gm_{12}gm_{13}gm_{18})(gds_2 + gds_3) \\ & + gm_1 gmb_{19}(gm_{13}gm_{18}Cbd_1(gds_2 + gds_3) + gm_{11}gm_{14}Cbd_{17}gds_2)s \\ & /(gm_{17}gmb_{19}gm_{12}[gm_1(gm_{11} + gds_2 + gds_3) + gm_{11}(gds_2 + gds_3)] \\ & + [gm_{17}gm_{12}Cbs_{19}(gm_1 + gm_{11})(gds_2 + gds_3) \\ & + gm_1 gm_{11}gmb_{19}(gm_{11}Cbd_1 + gm_{12}Cbd_{17})]s) \end{aligned} \tag{8}$$

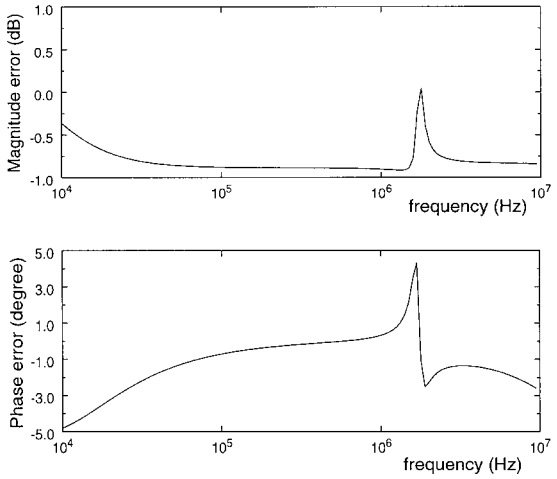

Fig. 10. Magnitude and phase errors between the filter behavior and that predicted by the approximate symbolic expressions.

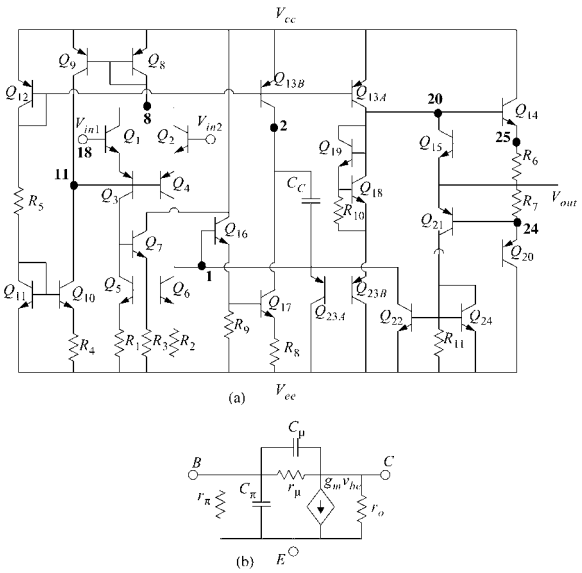

Fig. 11. (a) μA741 schematic and (b) small-signal model used for bipolar transistors.

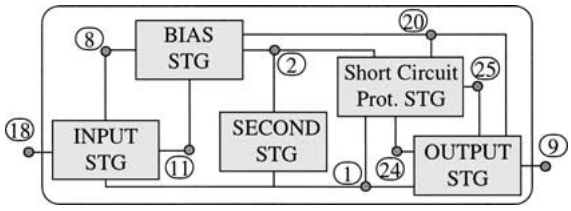

Fig. 12. Partitioning structure for the μA741 opamp.

For the complete analysis methodology 105 seconds of CPU time were necessary.[4]

Figure 9 compares the magnitude and phase behavior obtained by direct evaluation of the symbolic expressions in equations (3) through equation (8) with the behavior of the original circuit obtained by numerical simulation. The accuracy of the symbolic expressions is better seen in the error plots in Fig. 10, where it can be seen that the magnitude and phase error constraints are met in the specified frequency range.

It is important to notice that hierarchical analysis results for circuits like the filter in this example, in which a transistor level description of the blocks has been used, have never been reported before, on the one hand, and that conventional non-decomposed symbolic analyzers have never addressed such complex circuits like that in this example, on the other.

7.2. The μA741 opamp

In order to demonstrate the capabilities of the methodology in case of densely interconnected circuits, the analysis of the μA741 opamp, depicted in Fig. 11(a), using the small-signal model shown in Fig. 11(b), is presented. The partitioning has been performed looking for a unique behavior within each stage, yielding the structure presented in Fig. 12 [23]. Terminal node numbers allow an easy identification of the blocks.

Now, the error constraints are a magnitude error $|\Delta_{\text{mag}}| \leq 3$ dB and a phase error $|\Delta_{\text{phs}}| \leq 6$ degrees in the frequency range 1 Hz $\leq f \leq 10^6$ Hz (thus including the complete gain-bandwidth product).

The new hierarchical analysis technique generates a symbolic expression for the high-level network function containing 22 terms (9 in the numerator and 13 in the denominator), which depend on the block (trans)admittances and for each of them an approximate symbolic expression is generated containing the number of terms indicated in Table 2.[5] The complete result is generated in 43 seconds of CPU time.

It can be seen in Table 2 that a new block, *Input2,* has appeared in the expression. This block is a result of the Block checking step, that has found that the *Input* Block is too large to be adequately handled, thus partitions this block internally. On the other hand, *Bias* block has been joined to the upper level because of the small number of devices that had once simplified. If we do not make use of this internal checking, the resulting expression

is significantly more complex and the CPU time grows to 192 seconds.

The magnitude and phase behavior of the original circuit and the magnitude and phase plots obtained by evaluating the simplified symbolic expressions provided by the hierarchical analysis methodology are shown in Fig. 13. Magnitude and phase deviations are better visualized in the error plots in Fig. 14.

Table 2. Complexity of the expressions generated for the blocks.

(Trans)admittance	# Terms
taInput.8.18	13
taInput. 8.11	39
aInput. 11.11	22
taInput.11.18	8
taInput. 1.11	9
taInput.1.12	9
aInput. 12.12	2
taInput.16.18	4
taInput.16.11	4
aInput.16.16	2
aSecondStg.1.1	26
taSecondStg.1.2	40
aSecondStg.2.2	8
taSecondStg.2.1	6
aOutput.9.9	4
taOutput.9.2	19
aInput2.12.12	5
aInput2.12.16	3
aInput2.16.16	4
taInput2.16.12	4

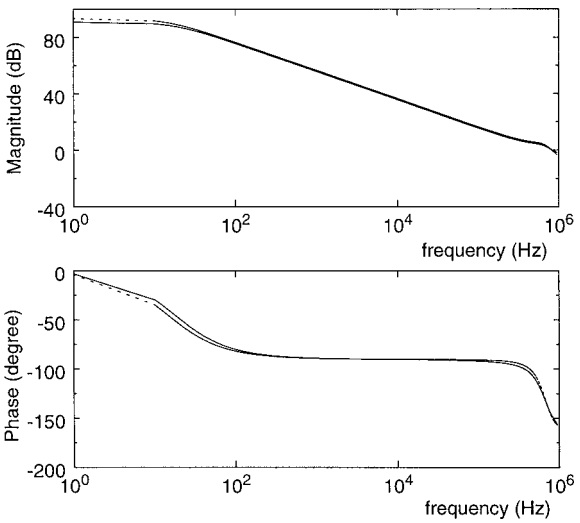

Fig. 13. Magnitude and phase plots of the original circuit and the symbolic expressions.

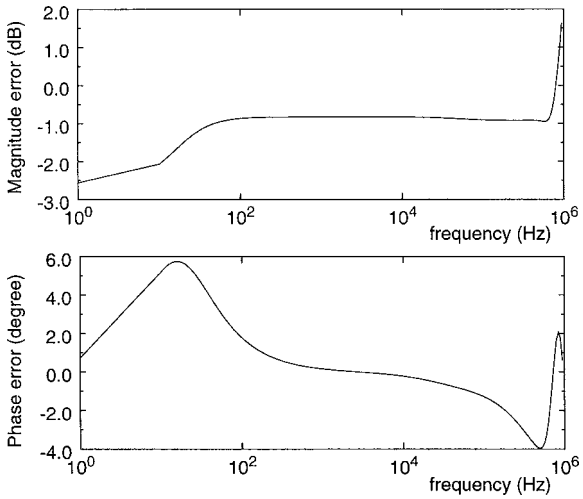

Fig. 14. Magnitude and phase errors between the μA741 opamp and the approximated symbolic expressions for its voltage gain.

8. Conclusions

This paper has introduced a methodology for the incorporation of approximation strategies into a hierarchical analysis method. On the one hand, this overcomes the problems of approximate non-decomposed analysis techniques when facing very large circuits. On the other, the inherent hierarchy of large circuits is kept, making the results more interpretable and more efficiently evaluated than if a non-decomposed approach were used.

The hierarchical analysis methodology incorporates a modeling strategy that fits the approximation needs. Systems of equations are formulated and solved at the different hierarchical levels in graph form for symbolic computations and in matrix form for numerical error control. In this way, the best trade-off between speed, accuracy and complexity is achieved.

The error-controlled symbolic results can be efficiently used to get insight into circuit operation and for repetitive evaluations, for instance, for behavioral modeling or synthesis applications. Even in the case that the approximated symbolic results for a given error spec are too complex for gaining insight, the circuit reduction step gives useful information on the terminal characteristics which mostly influence the complete circuit behavior.

Acknowledgments

This work has been supported by the ESPRIT Program of the European Commission in the Framework of the Project #21812 (AMADEUS), the Spanish C.I.C.Y.T. under contract TIC97-0580, and the Spanish C.I.C.Y.T. under contract TIC2001-0929.

Notes

1. By non-decomposed analysis we will refer to analysis of circuits with no hierarchical decomposition, neither provided by the user, nor performed by the analysis methodology.
2. It must be pointed out that these estimations have been done taking into account complete small-signal models for the transistors and supposing an adequate biasing point for the circuits. Otherwise, the capabilities can be *dramatically* (but also *artificially*) increased.
3. Dashed blocks represent the original blocks for easy identification of the remaining (trans)admittances after the circuit reduction step.
4. All CPU times in this paper have been measured on a SUN Ultra 1 at 140 MHz with 64MB RAM running under Solaris 2.6.
5. The symbolic expressions are not shown here for space reasons.

References

1. Fernández, F. V., Rodríguez-Vázquez, A., Huertas, J. L. and Gielen, G. (eds.), *Symbolic Analysis Techniques. Applications to Analog Design Automation*. IEEE Press, 1998.
2. Starzyk, J. and Sliwa, E., "Hierarchic decomposition method for the topological analysis of electronic networks." *Int. J. Circuit Theory and Applications* 8, pp. 407–417, 1980.
3. Starzyk, J. A. and Konczykowska, A., "Flowgraph analysis of large electronic networks." *IEEE Trans. Circuit and Systems* CAS-33(3), pp. 302–315, March 1986.
4. Hassoun, M. M., "Hierarchical symbolic analysis of large-scale systems using a Mason's signal flowgraph model," in *Proc. IEEE Int. Symp. Circuits and Syst.,* pp. 801–805, 1991.
5. Hassoun, M. M. and McCarville, K. S., "Symbolic analysis of large-scale networks using a hierarchical signal flowgraph approach." *Analog Int. Circuits and Signal Proc.* 3, pp. 31–42, January 1993.
6. Hassoun, M. M. and Lin, P. M., "A new network approach to symbolic simulation of large-scale circuits," in *Proc. IEEE Int. Symp. Circuits and Syst.,* pp. 806–809, May 1989.
7. Hassoun, M. M. and Lin, P. M., "A hierarchical network approach to symbolic analysis of large-scale networks." *IEEE Trans. Circuits and Syst.-I* 42(4), pp. 201–211, April 1995.
8. Hassoun, M. M., "Hierarchical symbolic analysis of large analog circuits." In: F. V. Fernández, A. Rodríguez-Vázquez, J. L. Huertas and G. E. Gielen (eds.). *Symbolic Analysis Techniques. Applications to Analog Design Automation,* Chapter 5. IEEE Press, 1998.
9. Mason, S. J., "Feedback theory, some properties of signal flow graph," in *Proc. of IRE* 41, pp. 1144–1156, September 1953.
10. Wambacq, P., Fernández, F. V., Gielen, G., Sansen, W. and Rodríguez-Vázquez, A., "A family of matroid intersection algorithms for the computation of approximated symbolic network functions," in *Proc. IEEE Int. Symp. Circuits and Syst.,* Atlanta, pp. 806–809, 1996.
11. Guerra, O., Rodríguez-García, J. D., Roca, E., Fernández, F. V. and Rodríguez-Vázquez, A., "A simplification before and during generation methodology for symbolic large-circuits analysis," in *Proc IEEE Int. Conf. Electronic Circuits and Systems* 3, pp. 81–84, September 1998.
12. Rodríguez-García, J. D., Guerra, O., Roca, E., Fernández, F. V. and Rodríguez-Vázquez, A., "Error control in simplification before generation algorithms for symbolic analysis of large analogue circuits." *IEE Electronic Letters* 35, pp. 260–261, February 1999.
13. Lin, P. M., *Symbolic Network Analysis*. Elsevier, 1991.
14. Vlach, J. and Singhal, K., *Computer Methods for Circuit Analysis and Design*. Van Nostrand Reinhold, 1994.
15. Nitescu, M. and Constantinescu, F., "Symbolic vs. numeric hierarchical methods for solving repetitive computation problems," in *Proc. Int. Workshop on Symbolic Methods and Applications to Circuit Design,* October 1996.
16. Hsu, J. and Sechen, C., "Fully symbolic analysis of large analog integrated circuits," in *Proc. IEEE Custom Integrated Circuits Conf.,* pp. 21.4.1–21.4.4, 1994.
17. Yu, Q. and Sechen, C., "A unified approach to the approximated symbolic analysis of large analog integrated circuits." *IEEE Trans. Circuits and Syst.-I* 43(8), pp. 656–669, 1996.
18. Fernández, F. V., Guerra, O., Rodríguez-García, J. D. and Rodríguez-Vázquez, A., "Symbolic analysis of large analog integrated circuits: The numerical reference generation problem." *IEEE Trans. Circuits and Syst.-II* 45(10), pp. 1351–1361, October 1998.
19. Moore, R. E., *Methods and Applications of Interval Analysis*. Studies in Applied Mathematics, 1979.
20. Kolev, L. V., *Interval Methods for Circuit Analysis*. World Scientific, 1993.
21. Lawler, E. L., *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, 1976.
22. Camerini, P. M. and Hamacher, H. W., "Intersection of two matroids: (Condensed) border graph and ranking." *SIAM J. Disc. Math.* 2, pp. 16–27, February 1989.
23. Sedra, A. S. and Smith, K. C., *Microelectronic Circuits*. Holt Rinehart and Winston, 1987.

Oscar Guerra Vinuesa received the Physics degree in Electronics from the University of Sevilla, Spain, in

1995. He is currently working towards the Ph.D. degree in the field of analog design automation. Since 1995 he has been working at the Department of Analog Circuit Design of the *Instituto Nacional de microelectrónica, Centro Nacional de Microelectrónica* (IMSE-CNM) and since 2000 he is also an assistant professor at the University of Sevilla. His research interests are in the fields of computer-aided design, programming techniques, design and modeling of analog integrated circuits and analog design automation.

Elisenda Roca Moreno received her Physics degree in 1990 and her PhD degree in 1995, both from the University of Barcelona (Spain). From November 1990 to April 1995 she worked at IMEC vzw (Leuven, Belgium) in the field of infrared detection aiming to obtain large arrays of of CMOS compatible silicide Schottky diodes. Since 1995 she is with the Institute for Microelectronics of Seville (IMSE-CNM) where she holds the position of scientific tenure. Her research interests are: design of CMOS analog focal plane array processors for vision applications, CMOS imagers for the visible spectrum, and modeling and design methodologies for analog integrated circuits. Dr. Roca has been involved in several research projects with different institutions: Commission of the EU through the ESPRIT program, ESA, ONR-NICOP, etc. Dr. Roca has also coauthored more than 40 papers in international journals, books and conference proceedings.

Francisco V. Fernández received the Physics degree in Electronics in 1988 and the Ph.D. degree in 1992, both from the University of Sevilla, Spain. In 1988 he joined the Department of Electronics and Electromagnetism at the University of Sevilla and in 1991 he was appointed as an assistant professor. During 1993 he stayed at the ESAT laboratory of the Katholieke Universiteit Leuven, Belgium, as a senior researcher. Since 1995 he is an associate professor at the University of Sevilla. He is also at the Institute of Microelectronics of Sevilla, which is part of the National Microelectronics Center, where he is conducting research on analog CAD. He has been editor of one book and authored more than 60 papers in books, journals and conference proceedings. His research interests include design and modeling of analog integrated circuits and analog design automation.

Angel Rodríguez-Vázquez, *IEEE Fellow,* is a professor of electronics at the Department of Electronics and Electromagnetism (University of Seville). He is also a member of the research staff of the Institute of Microelectronics of Seville—Centro Nacional de Microelectrónica (IMSE-CNM)—where he is heading a research group on Analog and Mixed-Signal Integrated Circuits. His research interests are in the design of analog interfaces for mixed-signal circuits, CMOS imagers

and vision chips, telecom circuits, neuro-fuzzy controllers, symbolic analysis of analog integrated circuits and optimization of analog integrated circuits. In these topics, Dr. Rodríguez-Vázquez has published 5 books, 23 book chapters in other books, around 100 journal papers, and more that 250 conference papers.

Dr. Rodríguez-Vázquez served as an associate editor of the *IEEE Transactions on Circuits and Systems—I* from 1993 to 1995, as guest editor of the IEEE TCAS-I special issue on *Low-Voltage and Low-Power Analog and Mixed-Signal Circuits and Systems* (1995), as guest editor of the IEEE TCAS-II special issue on *Advances in Nonlinear Electronic Circuits* (1999), as guest editor of the IEEE TCAS-I special issue on *Bio-Inspired Processors and Cellular Neural Networks for Vision* (1999), and as chair of the IEEE-CAS Analog Signal Processing Committee (1996). Currently he is an associate editor for IEEE TCAS-II.

He was co-recipient of the 1995 Guillemin-Cauer award of the IEEE Circuits and Systems Society, the best paper award of the 1995 European Conference on Circuit Theory and Design, and the 1999 best paper award of the International Journal on *Circuit Theory and Applications*. In 1992 he received also the young scientist award of the Seville Academy of Science. In 1996 he was elected to the degree of Fellow of the IEEE for "contributions to the design and applications of analog/digital nonlinear ICs."