

An Accurate Error Control Mechanism for Simplification Before Generation Algorithms

O. Guerra, J. D. Rodríguez-García, E. Roca, F. V. Fernández and A. Rodríguez-Vázquez
Instituto de Microelectrónica de Sevilla, Centro Nacional de Microelectrónica, Sevilla, SPAIN
Phone: +34 954239923, FAX: +34 954231832, E-mail: pacov@imse.cnm.es

Abstract

The use of simplification before generation techniques to enable the approximate symbolic analysis of large analog circuits is discussed. This paper introduces an error control mechanism to drive the circuit reduction, which overcomes the accuracy problems of previous approaches. The features and efficiency of the new methodology are demonstrated through several practical examples.

1. Introduction

Unlike SPICE-like simulators, in which all circuit parameters are assigned a numerical value, symbolic analyzers handle circuits with symbolic parameters. The result of their analysis task is obviously a symbolic expression of the circuit characteristic at hand. One of the main limitations of symbolic analyzers has been traditionally found in the exponential increase of the expression length with the circuit size [1]. On the one hand, this makes the symbolic results very difficult to interpret or use. On the other, it constitutes a drastic limitation to the maximum circuit size that can be analyzed.

An important advance in the solution of these problems has been achieved by the introduction of simplification before and during generation techniques [1]. The role that these techniques play within the symbolic analysis flow is better understood by looking at Fig. 1. A typical analysis problem starts from the small-signal model of the circuit. Using some analysis technique (i.e., signal flow graphs, MNA, etc.), a set of network equations, either in matrix or graph form, is obtained. Simplification before generation (SBG) techniques either eliminate entries from the circuit matrix, or eliminate graph branches and contract graph nodes, yielding a reduced matrix or graph which is much easier to solve.

Once approximated, the resulting simplified system of equations must be solved. The complete solution usually

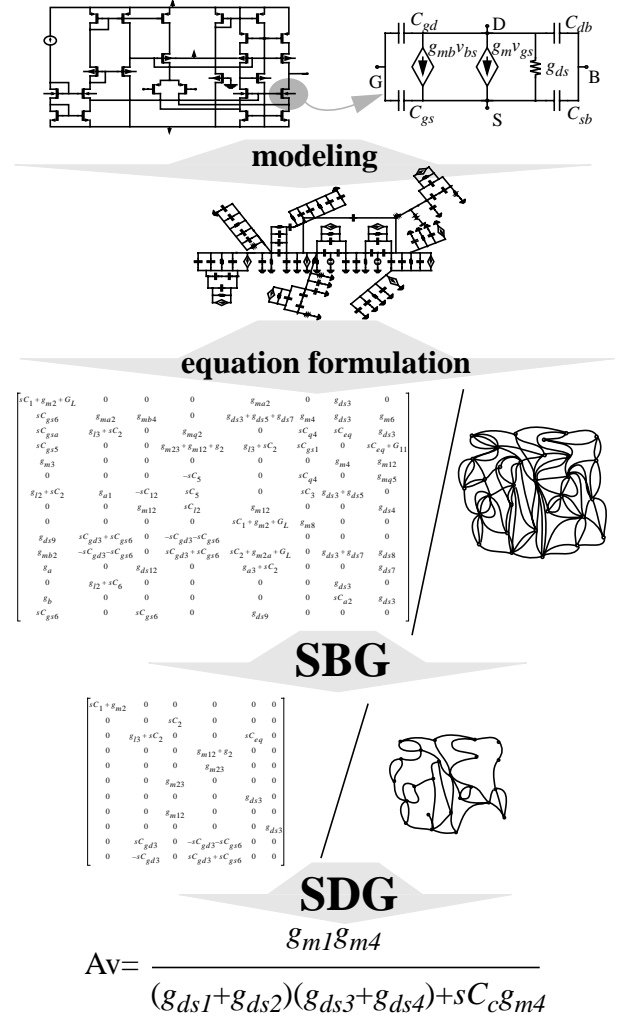


Fig. 1. Combination of simplification strategies in the symbolic analysis flow.

contains a huge number of insignificant contributions. Simplification During Generation (SDG) techniques aim to calculate directly an approximated solution, which contains only the dominant contributions [1]-[3].

This paper focuses in the most critical part of an SBG algorithm: an efficient and accurate mechanism to control

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

the error induced by the reduction process. Section 2 reviews previous work and introduces a new methodology to solve the problems of reported approaches. The methodology is extensively tested with practical examples in Section 3.

2. Simplification before generation

2.1. Background and previous work

The computational complexity of the solution algorithms for a set of network equations grows exponentially with the circuit size. And so does the complexity of the resulting symbolic expressions.

However, there are usually large differences among the relative contribution of the different circuit parameters to the global circuit behavior. Negligible parameters make computationally more expensive the solution of the set of network equations and more difficult the interpretation of the results.

Reported SBG approaches simplify the system of equations prior to address its solution. In [4],[5], device parameters are eliminated from the nodal admittance matrix while the error induced is below a given threshold. In [6], graph branches are pruned or graph nodes are contracted while their contribution to the network function keeps below some given error. Although these techniques exhibit significant differences, they share a common feature: the error induced by a matrix entry elimination, or by branch pruning or node contraction in a graph, is evaluated at a single or a finite number of frequency points.

Therefore, accuracy is not guaranteed at frequencies different from those in the set of sampling frequencies, as the practical example in Fig. 2 illustrates. This figure shows the magnitude and phase errors at $1\text{MHz} \leq f \leq 100\text{MHz}$, induced when sampling-based SBG algorithms, like those in [4]-[6], are applied to the integrator in Fig. 3. The magnitude and phase error specs were $\Delta|H| \leq \pm 5\text{dB}$ and $\Delta\phi_H \leq \pm 5^\circ$ in the frequency range $1\text{Hz} \leq f \leq 100\text{MHz}$. As Fig. 2 shows, the error specs are met at the sampling frequencies, but exceeded at intermediate ones.

An obvious solution is to use a denser frequency sampling, at least in the neighborhood of the poles and zeros of the system. However, this increases noticeably the computational cost of the algorithm, on the one hand, and there is not a systematic procedure on how dense should be the frequency sampling to guarantee full accuracy in a frequency range, on the other. The methodology presented herein solves this problem by introducing error evaluation mechanisms which guarantee the required accuracy at any frequency within a given range.

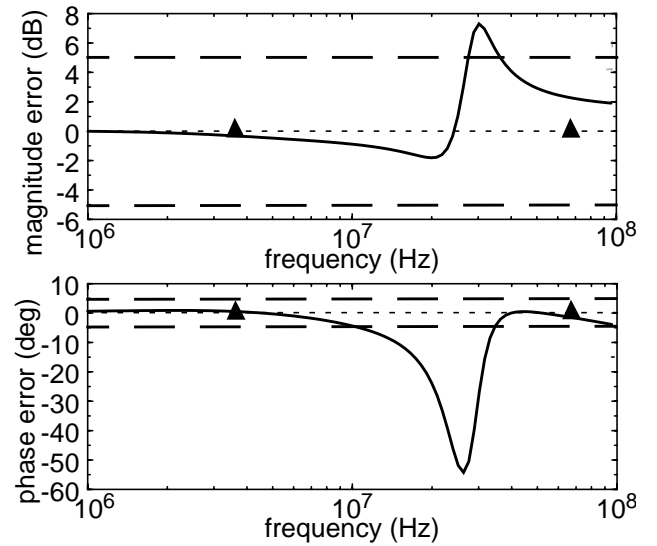


Fig. 2. Magnitude and phase errors of Fig. 3 at $1\text{MHz} \leq f \leq 100\text{MHz}$ due to the application of an SBG algorithm. Solid triangles denote the sampling frequencies

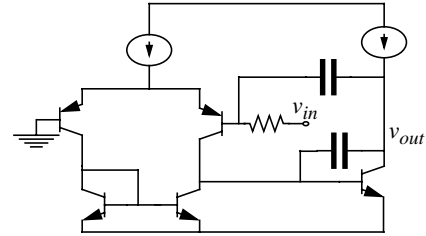


Fig. 3. Integrator.

2.2. New SBG methodology

Our approach performs the approximation by replacing those elements whose contribution (appropriately measured) to the network function is small, with a zero-admittance (device removal) or zero-impedance element (contraction of nodes).

The objective is to find the sequence of node contractions and device removals yielding the simplest circuit (smallest number of nodes and branches) and whose induced error keeps below some given threshold. First, it must be decided if node contractions must be prioritized over device removals or viceversa. After the SBG process, the resulting simplified graph must be solved, commonly by the application of a SDG process. The most efficient SDG algorithms reported are based on the two-graph method [7] and their computational complexity grows much faster with the number of circuit nodes than with the number of devices. Therefore, node contractions are prioritized in our algorithm.

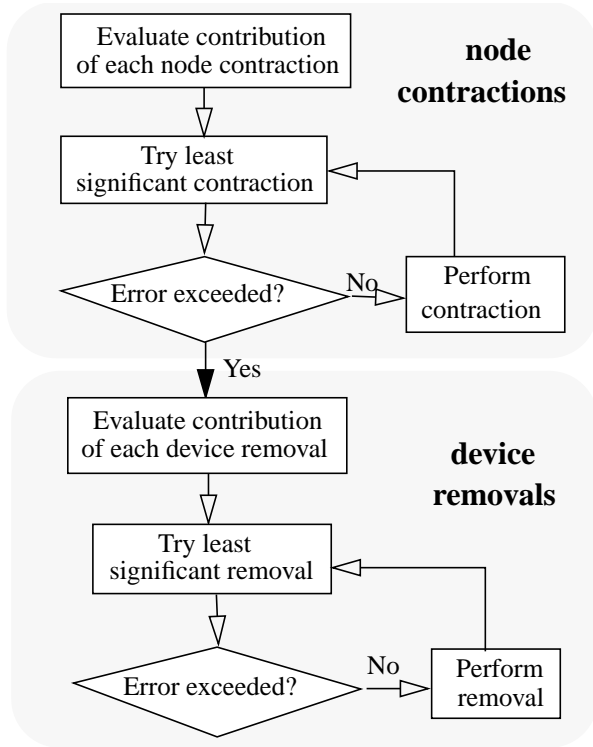


Fig. 4. Flow diagram of the SBG methodology.

The different steps of the simplification algorithm are graphically shown in Fig. 4. First, the contribution to the transfer function of the contraction of the terminal nodes of each device individually is computed individually and a sorted list is built. The least significant contraction from the list is picked and the induced magnitude and phase errors are evaluated. If the allowed error is not exceeded the node contraction is performed and all devices connected in parallel are removed. The contraction process continues iteratively with the following one in the sorted list while the accumulated error in magnitude and phase does not exceed the specified maximum errors. When the contraction process is finished an analogous operation with device removals is performed.

2.3. Error evaluation

As shown in Fig. 4, both, the node contraction and the device removal processes start with an evaluation of the contribution of each possible contraction or removal. That means that the difference in magnitude and phase behavior between the original circuit and a modified circuit in which a pair of nodes have been contracted or a device has been removed must be evaluated. Also, when each node contraction is tried it must be checked if the difference in magnitude and phase behavior between the original circuit and the reduced circuit, in which the contraction at hand,

together with all previously accepted contractions have been performed, exceeds the error specifications. A similar test must be performed when each device removal is tried. Our objective is to evaluate the maximum magnitude and phase deviations for any frequency in a given range in all error checking steps described above.

Let us denote $H_{ex}(s) = N_{ex}(s)/D_{ex}(s)$ the network function of the complete circuit with only the complex frequency s as symbolic parameter, and $H_{ap}(s) = N_{ap}(s)/D_{ap}(s)$ the analogous network function of a simplified circuit in which the appropriate node contraction(s) and/or device removal(s) have been performed. The magnitude and phase errors are given by:

$$\Delta|H| = \frac{|H_{ex}(j\omega)| - |H_{ap}(j\omega)|}{|H_{ex}(j\omega)|} = 1 - \frac{\sqrt{\frac{N_{apr}^2 + N_{api}^2}{D_{apr}^2 + D_{api}^2}}}{\sqrt{\frac{N_{exr}^2 + N_{exi}^2}{D_{exr}^2 + D_{exi}^2}}} \quad (1)$$

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

$$= \text{atan} \frac{N_{exi}}{N_{exr}} - \text{atan} \frac{D_{exi}}{D_{exr}} - \text{atan} \frac{N_{api}}{N_{apr}} + \text{atan} \frac{D_{api}}{D_{apr}}$$

where subscripts r and i denote real and imaginary parts.

Therefore, the evaluation of the maximum magnitude and phase errors requires:

- a technique to obtain the network functions $H_{ex}(s)$ and $H_{ap}(s)$ of (usually large) analog circuits, and
- an efficient technique to obtain the maxima of the functions in (1) when ω varies within a given range.

The first problem can be solved by means of numerical interpolation techniques [7]. An efficient interpolation technique based on adaptive scaling able to handle large analog circuits can be found in [8],[9].

Our solution for the second problem is based on the use of interval analysis techniques [10]. $\Delta|H|$ and $\Delta\phi_H$ in (1) are univariate functions in ω , which can take any value within the frequency interval $[\omega_L, \omega_U]$, where ω_L and ω_U are the lower and upper bounds of the interval, respectively. The problem is solved if accurate estimates of the lower and upper bounds of $\Delta|H|$ and $\Delta\phi_H$, when $\omega \in [\omega_L, \omega_U]$, can be calculated. This computation, commonly known as the interval extension of $\Delta|H|$ and $\Delta\phi_H$, can make use of interval arithmetic operators. Substitution of the real variable ω in (1) and real operators (addition, product, quotient, etc.) by the corresponding interval variables and operators yields the so-called natural interval

extension. Unfortunately, this computation usually yields too conservative estimates of the maximum errors [11].

To solve this problem, the natural interval extension is applied to the derivatives of (1). Although, the estimates of the derivatives are also very conservative, the zero inclusion in the resulting interval extension is enough to delimit frequency subranges in which the maximum magnitude and phase errors occur. Then, the exact frequency points for which the maximum magnitude or phase errors occur in those frequency subranges are easily calculated using the Newton-Raphson method.

3. Experimental results

The efficiency and the complexity reduction capabilities of the proposed SBG methodology are illustrated with the circuits in Fig. 5(a)-(d) where the transistor models in Fig. 5(e)-(f) were used. The maximum magnitude and phase deviations allowed in the voltage gain were $\Delta_{\text{mag}} \leq \pm 3\text{dB}$ and $\Delta_{\text{phase}} \leq \pm 5^\circ$ in the frequency range $f \in [1\text{Hz}, 1\text{MHz}]$. The complexity reduction achieved (measured as the number of devices and nodes in the orig-

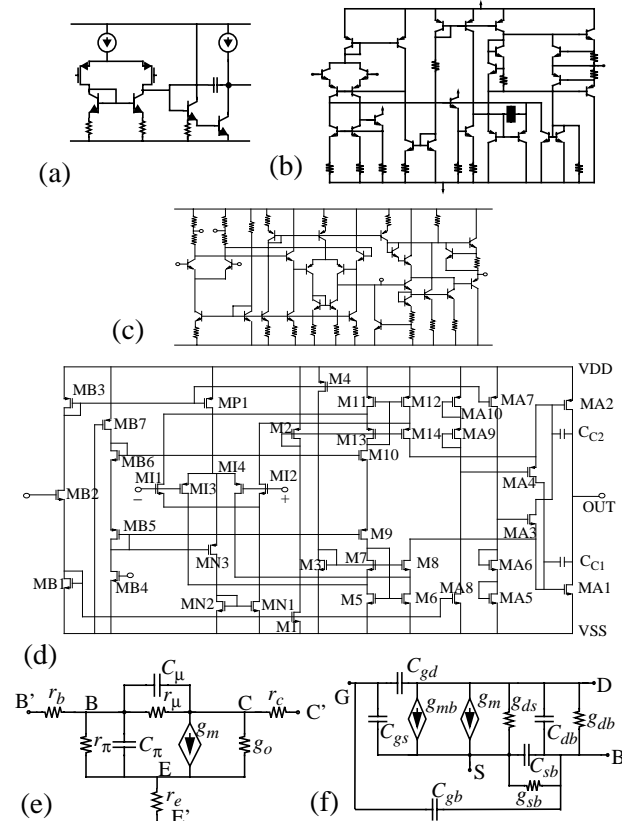


Fig. 5. (a) Simple BiCMOS opamp; (b) μA741 opamp; (c) μA725 opamp; (d) CMOS opamp; (e) bipolar transistor model; and (f) MOS transistor model.

inal circuit versus those in the simplified circuit) and the computation time are listed in Table 1.

A large, hierarchical example is the bandpass filter in Fig. 6a. It is composed of four OTAs, whose transistor-level schematic is shown in Fig. 6b, and one biasing OTA, shown in Fig. 6c. When expanding the small-signal models, the resulting circuit model contains 45 nodes and 619 basic devices.

The magnitude and phase plots of the voltage transfer function of this filter are shown in Fig. 7. A maximum circuit reduction is required with maximum magnitude and phase deviations:

$$\Delta_{\text{mag}} \leq \pm 3\text{dB} \quad \Delta_{\text{phase}} \leq \pm 5^\circ \quad (2)$$

in $f \in [100\text{Hz}, 100\text{MHz}]$. These magnitude and phase constraints are shown together with the magnitude and phase plots in Fig. 7.

The application of the SBG algorithm yields a reduced circuit model containing only 31 nodes and 161 devices,

Table 1. Statistics for the circuits in Fig. 5.

Circuits in Fig.		5(a)	5(b)	5(c)	5(d)
# nodes in small-signal model	original	21	76	85	26
	simplified	6	16	18	7
# devices in small-signal model	original	50	221	229	253
	simplified	11	48	52	22
CPU time (s.)		0.8	84	114	11

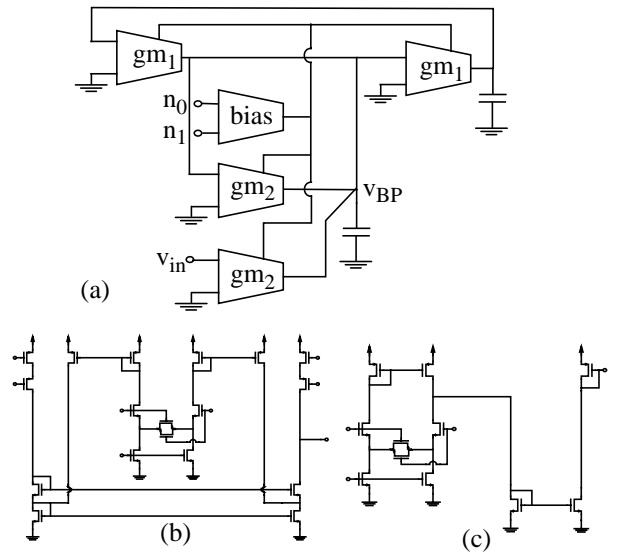


Fig. 6. (a) Filter; (b) OTA schematic; and (c) biasing OTA.

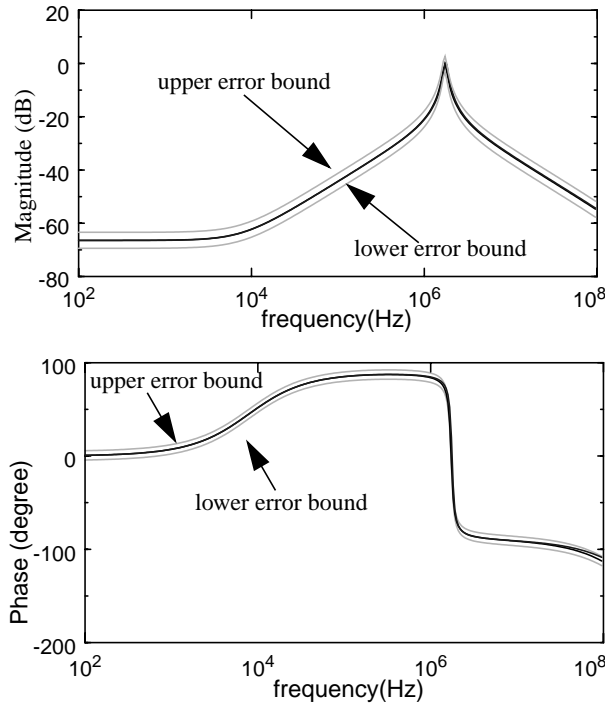


Fig. 7. Bode plots before and after the application of the SBG algorithm to the filter in Fig. 6.

while the magnitude and phase plots of the simplified circuit keep within the specified error limits, as shown in Fig. 7. One important feature is that all devices of the biasing OTA are eliminated in the simplification process. This fact demonstrates the capability of the SBG algorithm to detect and eliminate subcircuits which do not belong to the signal path and, therefore, do not affect the network function.

As illustration of the symbolic expression calculation after the SBG step, Fig. 8 shows the voltage gain provided by the complete SBG+SDG methodology applied to the CMOS opamp in Fig. 5(d). This example shows the possibility of generating very compact, interpretable expressions for the main behavior characteristics of even large building blocks.

Conclusions

This paper has introduced an accurate, but efficient, simplification before generation methodology. Based on its cooperative work with simplification during generation techniques, very readable and interpretable symbolic analysis is achievable.

Its extremely good behavior allows to address its combination with hierarchical decomposition strategies for very large circuit analysis. It is also being used to develop new methodologies for symbolic pole/zero extraction, with the objective to provide additional insight into the circuit behavior.

$$\begin{aligned}
 A_v = \frac{v_{OUT}}{v_-} \Big|_{v_+ = 0} = & \\
 & -G_{m14} \cdot G_{m8} \cdot G_{m13} \cdot (G_{m6} + G_{m12}) \cdot (G_{mA1} + G_{mA2}) \\
 & -G_{m14} \cdot G_{m8} \cdot G_{m6} \cdot G_{mA2} \cdot G_{m11} \\
 = & \\
 & G_{ds12} \cdot G_{ds14} \cdot G_{m8} \cdot (G_{m5} + G_{m11}) \cdot (G_{dsA1} + G_{dsA2}) \\
 & + G_{m14} \cdot G_{ds6} \cdot G_{ds8} \cdot (G_{m5} + G_{m11}) \cdot G_{dsA2} \\
 & + G_{m14} \cdot G_{m5} \cdot G_{ds8} \cdot G_{dsA2} \cdot G_{ds14} \\
 & + s \cdot ((C_{c1} + C_{c2}) \cdot G_{m14} \cdot G_{m8} \cdot (G_{m5} + G_{m11}) \cdot (G_{mA1} + G_{mA2}))
 \end{aligned}$$

Fig. 8. Symbolic expression for the voltage gain of Fig. 5(d).

References

- [1] F.V. Fernández, A. Rodríguez-Vázquez, J.L. Huertas and G. Gielen, *Symbolic Analysis Techniques. Applications to Analog Design Automation*, IEEE Press, 1997.
- [2] F.V. Fernández et al.: "Symbolic analysis of large analog integrated circuits by approximation during expression generation," *Proc. IEEE ISCAS*, V. CAD, pp. 25-28, 1994.
- [3] P. Wambacq, F.V. Fernández, G. Gielen, W. Sansen and A. Rodríguez-Vázquez, "Efficient symbolic computation of approximated small-signal characteristics of analog integrated circuits," *IEEE J. Solid-State Circuits*, Vol. 30, No. 3, pp. 327-330, March 1995.
- [4] Jer-Jaw Hsu and C. Sechen, "Fully symbolic analysis of large analog integrated circuits," *Proc. IEEE Custom Int. Circuits Conf.*, pp. 21.4.1-21.4.4, San Diego, CA, 1994.
- [5] R. Sommer, E. Hennig, G. Droge and E. H. Horneber, "Equation-based symbolic approximation by matrix reduction with quantitative error prediction," *Alta Frequenza*, vol. 5, no. 6, pp. 317-325, November 1993.
- [6] Q. Yu and C. Sechen, "A unified approach to the approximate symbolic analysis of large analog integrated circuits," *IEEE Trans. Circuits and Systems I*, Vol. 43, No. 8, pp. 656-669, Aug. 1996.
- [7] P.M. Lin, *Symbolic Network Analysis*. Elsevier, 1991.
- [8] I. García-Vargas, M. Galán, F.V. Fernández and A. Rodríguez-Vázquez, "New algorithms for reference generation in symbolic analysis of large analog circuits", *Proc. European Design & Test Conf.*, pp. 395-399, 1997.
- [9] F. V. Fernández, O. Guerra, J. D. Rodríguez-García and A. Rodríguez-Vázquez, "Symbolic analysis of large analog integrated circuits: the numerical reference generation problem," *IEEE Trans. Circuits and Systems-II*, vol. 45, no. 10, pp. 1351-1361, Oct. 1998.
- [10] R. E. Moore, *Methods and Applications of Interval Analysis*. SIAM, 1979.
- [11] F.V. Fernández, A. Rodríguez-Vázquez, J.D. Martín and J.L. Huertas, "Formula approximation for flat and hierarchical symbolic analysis," *Analog Integrated Circuits and Signal Proc.*, Vol. 3, No. 1, pp. 43-58, 1993.