Error control in simplification before generation algorithms for symbolic analysis of large analogue circuits

J.D. Rodríguez-García, O. Guerra, E. Roca, F.V. Fernández and A. Rodríguez-Vázquez

Circuit reduction is a fundamental first step in addressing the symbolic analysis of large analogue circuits. A new algorithm for simplification before generation is presented which is very efficient in terms of speed and the amount of circuit reduction, and solves the accuracy problems of previously reported approaches.

Introduction: One of the main drawbacks to symbolic analysers concerns the exponential increase in expression complexity with circuit size. On the one hand, this makes the symbolic results very difficult to use. On the other hand, it imposes a drastic limitation on the maximum size of circuit that can be analysed. In recent years, this drawback has been partially overcome through the use of techniques for simplification before (SBG) and during (SDG) generation [1, 2]. SBG techniques simplify the system of equations (either in matrix or graph form) before solving it [3 – 5]. Although the reported SBG approaches exhibit significant differences, they share the feature that the error induced by the elimination of an entry in a matrix, or by the removal of a branch or the contraction of a node in a graph, is evaluated at either one single frequency or a finite number of frequencies. Therefore, accuracy is not guaranteed at frequencies other than the sampling frequencies, as Fig. 1 illustrates. This Figure displays the magnitude and phase errors induced by the application of SBG to the integrator of Fig. 2*a*. The magnitude and phase error specifications were $\Delta|H| \leq \pm 5$dB and $\Delta\phi_H \leq \pm 5°$ in the frequency range $1\,\text{Hz} \leq f \leq 100\,\text{MHz}$. As Fig. 1 shows, the error specifications are met at the sampling frequencies, but exceeded at intermediate frequencies. The methodology in this Letter solves this problem by introducing error evaluation mechanisms which guarantee the required accuracy at any frequency within a given range.

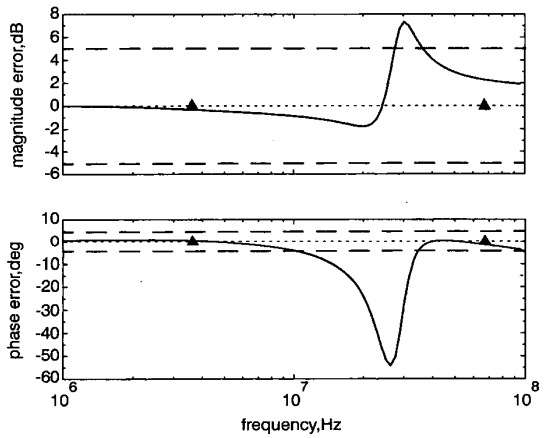

Fig. 1 *Magnitude and phase errors of Fig. 2a at $1\,MHz \leq f \leq 100\,MHz$ due to application of SBG algorithm*

▲ sampling frequencies

New SBG methodology: Our approach involves replacing those elements whose contribution to the network function is small by either an open circuit (device removal) or a short circuit (node contraction). The objective is to find the sequence of node contractions and device removals yielding the simplest circuit in which errors are kept below some threshold.

In our methodology, node contractions are given priority over device removals as the computational complexity of the posterior (SDG-based) solution algorithms increases much more quickly with the number of circuit nodes than with the number of devices. First of all, the algorithm calculates the contribution to the transfer function of the contraction of the terminal nodes of each individual device. The least significant contraction is chosen and the induced magnitude and phase errors are evaluated. If the allowed error is not exceeded, node contraction is carried out and all devices connected in parallel are removed. The contraction process continues iteratively with the next least-significant one while the accumulated magnitude and phase error does not exceed the specified maximum error. When the contraction process is finished a similar operation for device removal is performed.

As explained above, both the node contraction and device removal processes start with an evaluation of the contribution of each possible contraction or removal. This means that the maximum difference in magnitude and phase 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, after each node contraction is performed, a check must be made to determine if the maximum difference in magnitude and phase between the original circuit and the reduced circuit, in which the contraction at hand, together with all previously accepted contractions, has been performed, exceeds the error specifications. A similar test must be performed when each device removal is attempted.

$H_{ex}(s)$ denotes 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 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

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

Therefore, evaluation of the maximum magnitude and phase errors requires the calculation of: (*a*) the network functions $H_{ex}(s)$ and $H_{ap}(s)$ of (usually large) analogue circuits; and (*b*) the maxima of eqn. 1 when ω varies within a given interval. The first problem can be solved by means of numerical interpolation techniques. An efficient interpolation technique able to handle large analogue circuits can be found in [6].

Our solution for the second problem is based on the use of interval analysis techniques [7]. $\Delta|H|$ and $\Delta\phi_H$ in eqn. 1 are functions in ω, which can take any value within the frequency interval $[\omega_L, \omega_U]$. The problem is solved if accurate estimates of the interval 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 eqn. 1 and real operators (addition, product, quotient, etc.) by the corresponding interval variables and operators yields the so-called natural interval extension [7]. Unfortunately, this computation usually overestimates maximum errors [8].

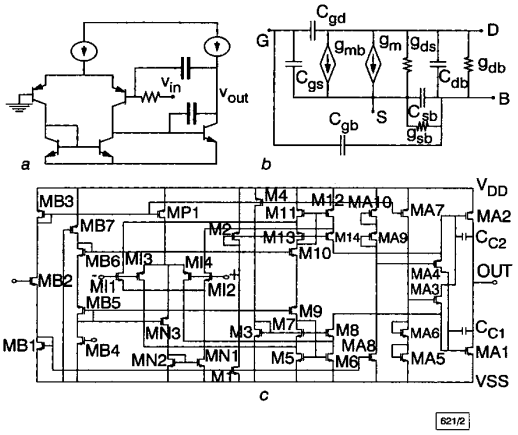

Fig. 2 *Integrator, transistor model and CMOS opamp*

a Integrator
b Transistor model
c CMOS opamp

To solve this problem, the natural interval extension is applied to the derivatives of eqn. 1. Although the interval widths of the derivatives are also overestimated, the zero inclusion in the resulting interval extension is sufficient for delimiting the 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.

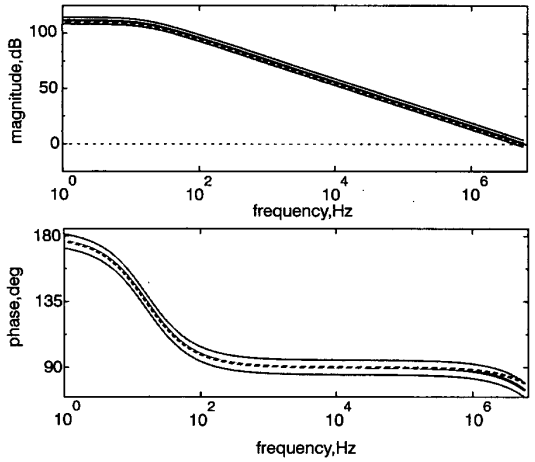

Fig. 3 *Magnitude and phase diagrams of Fig. 2c before and after application of SBG algorithm*

Experimental results: The accuracy and complexity reduction capabilities of the proposed methodology are illustrated through the circuit of Fig. 2*c* with the transistor model of Fig. 2*b*. By applying our SBG technique, the number of nodes in the small-signal model was reduced from 26 to 8 and the number of devices from 253 to 26 in only 11s of CPU time. Magnitude and phase deviations were also kept within the prescribed margins: $\Delta|H| \leq \pm 3$dB and $\Delta\phi_H \leq \pm 5°$, as Fig. 3 shows. Because of the significant complexity reduction achieved, the circuit was manageable using our SDG algorithm, which provided the following approximated expressions for the transfer function:

$$A_v = \frac{v_{OUT}}{v_-}$$
$$= [-G_{m14}G_{m8}G_{m13}(G_{m6}+G_{m12})(G_{mA1}+G_{mA2}) - G_{m14}G_{m8}G_{m6}G_{mA2}G_{m11}]/[G_{ds12}G_{ds14}G_{m8}(G_{m5}+G_{m11})(G_{dsA1}+G_{dsA2}) + G_{m14}G_{ds6}G_{ds8}(G_{m5}+G_{m11})G_{dsA2} + G_{m14}G_{m5}G_{ds8}G_{ds14}G_{dsA2} + s(C_{c1}+C_{c2})G_{m14}G_{m8}(G_{m5}+G_{m11})(G_{mA1}+G_{mA2})]$$

Conclusions: In this Letter we have has demonstrated the possibility of generating very compact, interpretable expressions for the main behaviour characteristics of even large building blocks, while maintaining accuracy.

Acknowledgments: This work was supported by the EEC ESPRIT Project #21812 (AMADEUS) and the Spanish C.I.C.Y.T. under contract TIC97-0580.

4 November 1998
Electronics Letters Online No: 19990235
DOI: 10.1049/el:19990235

J.D. Rodríguez-García, O. Guerra, E. Roca, F.V. Fernández and A. Rodríguez-Vázquez (*Instituto de Microelectrónica de Sevilla, Centro Nacional de Microelectrónica, Avda. Reina Mercedes s/n, E-41012 Sevilla, Spain*)

References

1 FERNÁNDEZ E.V., RODRÍGUEZ-VÁZQUEZ A., HUERTAS J.L., and GIELEN G.: 'Symbolic analysis techniques. Applications to analog design automation' (IEEE Press, Piscataway, NJ, 1998)

2 WAMBACQ, P., FERNÁNDEZ, F.V., GIELEN, G., SANSEN, W., and RODRÍGUEZ-VÁZQUEZ, A.: 'Algorithm for efficient symbolic analysis of large analogue circuits', *Electron. Lett.*, 1994, **30**, (14), pp. 1108–1109

3 HSU, J.-J., and SECHEN, C.: 'Fully symbolic analysis of large analog integrated circuits'. IEEE Custom Int. Circuits Conf. Proc., San Diego, CA, 1994, pp. 21.4.1–21.4.4

4 SOMMER, R., HENNIG, E., DROGE, G., and HORNEBER, E.H.: 'Equation-based symbolic approximation by matrix reduction with quantitative error prediction', *Alta Frequenza*, 1993, **5**, (6), pp. 317–325

5 YU, Q., and SECHEN, C.: 'A unified approach to the approximate symbolic analysis of large analog integrated circuits', *IEEE Trans.*, 1996, **CASI-43**, (8), pp. 656–669

6 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', to be published in *IEEE Trans. Circuits Syst. II: Analog Digital Signal Process.*, 1998, **45**, (10), pp. 1351–1361

7 MOORE, R.E.: 'Methods and applications of interval analysis'. SIAM, Philadelphia, 1979

8 FERNÁNDEZ, F.V., RODRÍGUEZ-VÁZQUEZ, A., MARTÍN, J.D., and HUERTAS, J.L.: 'Formula approximation for flat and hierarchical symbolic analysis', *Analog Integrated Circuits Signal Proc.*, 1993, **3**, pp. 43–58

Improvements in hierarchical symbolic tolerance and sensitivity analysis

F. Eberhardt, W. Tenten and P.R. Shepherd

Sensitivity and tolerance analyses are important for circuit optimisation, but unfortunately very time consuming. In recent years new hierarchical symbolic methods have been developed for large linear circuits. These approaches make symbolic techniques a worthwhile alternative to classical numerical methods. Using the hierarchical symbolic approach, a new strategy is introduced which significantly reduces the computational expense of sensitivity and tolerance analyses compared to previous procedures.

Introduction: Let $H(s, X)$ denote the transfer function of the circuit where s is the Laplace variable and X is the set of circuit parameters. The tolerances of H caused by variations of the parameters can be measured by determining the small change sensitivity (gradient) and large change sensitivity (step changes). Classical numerical methods for determining the sensitivities [1, 2] are only valid for small change sensitivity and require a complete new solution of the system matrix at each frequency point.

Symbolic analysis provides a good alternative for tolerance investigations, especially when the number of fiequency points is large [3]. In recent years hierarchical symbolic analysis methods [4, 5] have been developed. The benefit of the hierarchical approach is that the computational expense has a linear up to quadratic dependence on the circuit complexity as opposed to the exponential nature of classical symbolic analysis. Consequently, much larger circuits can be described.

The purpose of this Letter is to introduce a hierarchical symbolic sensitivity analysis method which significantly minimises the computation time compared to previous procedures, especially when the influence of many or all parameters is investigated.

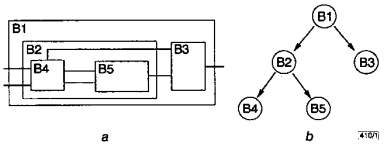

Fig. 1 *Hierarchical network partitioning and binary partition tree*

a Hierarchical network partitioning
b Binary partition tree

Hierarchical symbolic analysis: The symbolic analysis in [5] is based on a hierarchical partitioning of the circuit. This partitioning process is modelled by a binary partition tree (Fig. 1). The system matrices of the circuit blocks are calculated starting with the leaves of the tree and proceeding from the bottom up. The result is a sequence of expressions (SOE) for the network function H:

$$H_1 = f(s, X), \quad H_2 = f(s, X, H_1), ...,$$
$$H = H_k = f(s, X, H_1, ..., H_{k-1}) \qquad (1)$$