Variational Analog Integrated Circuit Design via Symbolic Sensitivity Analysis*

Guoyong Shi and Xiaoxuan Meng
School of Microelectronics, Shanghai Jiao Tong University
Shanghai 200030, China
{shiguoyong,mengxiaoxuan}@ic.sjtu.edu.cn

***Abstract*—This paper presents a symbolic AC sensitivity analysis technique using a graph-reduction based symbolic simulator *GRASS*. Symbolic sum-of-products are derived from a graph reduction process and represented by a binary decision diagram. The GRASS simulator maintains a one-to-one correspondence between the circuit parameters and the simulator symbols, with which differentiating the frequency response with respect to any circuit parameter becomes straightforward. The implementation details of symbolic AC sensitivity are presented and the potential applications are demonstrated.**

***Index Terms*—Analog circuit design, binary decision diagram, design optimization, graph reduction, symbolic AC sensitivity.**

I. Introduction

The goal of developing a symbolic circuit simulator is manifold; the main applications cover those tasks demanding repeated computations, such as Monte Carlo verification and design space exploration [1], [2]. A graph reduction analog symbolic simulator (GRASS) recently reported in [3], [4] has the advantages of matrix-free computation, cancellation-free sum-of-product terms, maintaining one-to-one correspondence from the circuit parameters to the simulator symbols, and direct circuit-topology based construction. These advantages are not observed in other existing symbolic simulators. With a binary decision diagram (BDD) based representation of sum-of-products, the circuit size that the GRASS simulator can treat ([3]) is competitive to the latest state-of-the-art DDD simulator ([5], [6]).

An early research on symbolic approach to circuit sensitivity analysis was presented in [7], but was limited to small-scale circuits. Symbolic AC sensitivity analysis with DDD was mentioned in [5] (Section V.B) but not expanded in detail. Although it is in principle possible to carry out symbolic AC sensitivity analysis in the framework of DDD, its implementation in terms of *cofactors* is expected to be involved and possibly lack of efficiency due to the cross dependence of one simulator symbol on multiple circuit parameters, an inherent fact by the MNA formulation. An Element-Coefficient Diagram (ECD) method is proposed in [8], [9] for sensitivity analysis. This method was extended from the DDD work and applications to analog circuit synthesis were reported there.

The basic idea of GRASS AC sensitivity analysis is explained in Section II with a simple example. Presented in Section III is a general algorithm for symbolic AC sensitivity computation implemented in GRASS. As a demonstration, application of the symbolic AC sensitivity to the $\mu A741$ opamp is introduced in Section IV. Section V concludes this paper. A few sensitivity identities are derived in the appendix for reference.

*This research was supported in part by Shanghai Pu Jiang Scholar Funding Project in 2007 (Grant No. 07pj14053), by the National Natural Science Foundation of China (Grant No. 60876089), and by the Initiative Research Fund from the Ministry of Education (2008).

II. An Example

The normalized sensitivity of $H(s)$ with respect to a real parameter p is defined by

$$\text{Sens}(H(s), p) := \frac{p}{H(s)} \frac{\partial H(s)}{\partial p} = \frac{\partial \ln H(s)}{\partial \ln p}. \tag{1}$$

The basic steps of symbolic AC sensitivity analysis in GRASS are explained by the simple RC circuit shown in Fig. 1.

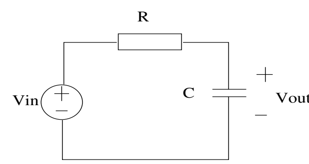

Fig. 1. Circuit example 1.

The transfer function of the RC circuit from V_{in} to V_{out} is

$$H(s) = \frac{1}{1 + RCs}; \tag{2}$$

its normalized sensitivity with respect to (w.r.t.) R is

$$\text{Sens}(H(s), R) = -\frac{RCs}{1 + RCs}. \tag{3}$$

The task of *symbolic AC sensitivity analysis* is to derive an analytical sensitivity expression as in (3), given a netlist with an input-output specification and a sensitivity analysis command. The benefit of symbolic AC sensitivity is the availability of an analytical sensitivity function of all interested circuit parameters. This is certainly advantageous to a numerical simulator in such applications as circuit optimization and Monte Carlo simulation.

Symbolic computation of the sensitivity $\text{Sens}(H(s), p)$ depends on the computation of the derivative $\partial H(s)/\partial p$ w.r.t. a parameter p, which in turn depends on how the frequency response $H(s)$ is represented in a symbolic simulator. In GRASS, the transfer function $H(s)$ is derived as follows: The input-output relation is modeled as a voltage-controlled voltage-source (VCVS), i.e., $V_{in}(s) = X(s)V_{out}(s)$. The gain $X(s)$ is treated as a symbol in GRASS and satisfies the following sum-of-products equation

$$G + Cs - X(s)G = 0, \tag{4}$$

where $G = R^{-1}$ and the three product terms are derived from the circuit topology by a graph reduction process [4]. Once a sum-of-products (SOP) expression as (4) is established, both the frequency response $H(s) = V_{out}(s)/V_{in}(s) = 1/X(s)$ and its derivative $\partial H(s)/\partial p$ can be derived analytically. Therefore, deriving a sum-of-products (SOP) expression like (4) is all we need for both symbolic AC analysis and AC sensitivity analysis. It is easy to see that computing the derivative of $X(s)$ w.r.t. any circuit parameter is

978-1-4244-3828-0/09/$25.00 ©2009 IEEE

especially easy for SOP expressions such as (4), because taking the derivative of a product term w.r.t. one parameter is a straightforward matter.

Concerning implementation, we recall that in GRASS each *signed* product term is stored as a *one-path* in a BDD as shown in Fig. 2(a) [4]. This decision diagram is called a *Symbol Decision Diagram* (SDD), where all edges are *signed*. Each one-path together with the edge-signs along the path determines one of the three *signed* product terms in (4).

Generally speaking, let $H(s) = 1/X(s) = N(s)/D(s)$ be a rational transfer function expressed in fractional form. It corresponds to the following SOP equation

$$D(s) - N(s)X(s) = 0. \tag{5}$$

In the GRASS implementation, the solid arrow from the SDD root node points to the expression $-N(s)$ (i.e., the coefficient of $X(s)$), while the dashed arrow from the root points to the expression $D(s)$ (i.e., the partial SOP not involving $X(s)$.) By Identity 4 in the appendix, $\mathrm{Sens}(X(s), G) = \mathrm{Sens}(D(s), G) - \mathrm{Sens}(N(s), G)$, which implies that it suffices to compute the sensitivities of $D(s)$ and $N(s)$ w.r.t. G in order to compute $\mathrm{Sens}(X(s), G)$.

By the definition of sensitivity, in addition to the construction of an SDD for $H(s)$, we need to compute two extra terms, $\partial D(s)/\partial G$ and $\partial N(s)/\partial G$, which can be obtained by simply modifying the SDD. Shown in Fig. 2(b) is the result, where the SDD nodes of symbol G have been set to *one* while all the *dashed* arrows issued from G have been pointed to the terminating node "0" (meaning that those product terms not involving the symbol G vanish after taking the derivative.) The resulting SDD is called a *differentiated SDD*.

For the current example, one can read from Fig. 2(b) that $D(s) = G + Cs$ and $N(s) = G$, which give rise to $\mathrm{Sens}(D(s), G) = G/(G + Cs)$ and $\mathrm{Sens}(N(s), G) = 1$, by which $\mathrm{Sens}(X(s), G) = \mathrm{Sens}(D(s), G) - \mathrm{Sens}(N(s), G) = -Cs/(G + Cs)$. Furthermore, by Identity 3 in the appendix, $\mathrm{Sens}(H, R) = \mathrm{Sens}(X, G) = -Cs/(G + Cs)$, which agrees with (3).

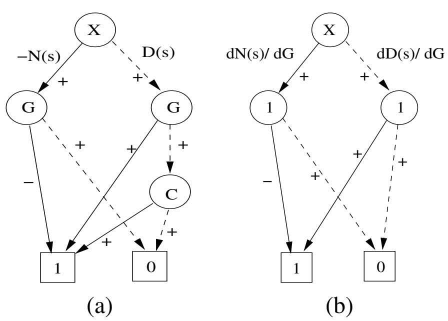

Fig. 2. (a) The SDD for example 1. (b) The differentiated SDD.

III. The Symbolic AC Sensitivity Algorithm

Following the steps outlined in the previous section for the simple RC circuit, a general algorithm for symbolic AC sensitivity is formalized here.

Any linear network satisfying minor conditions specified in [4] can be symbolically analyzed by GRASS in which a frequency response function is derived from the following algebraic SOP representation:

$$X(s)\left(\sum_{i=1}^{m_1} t_i\right) + \left(\sum_{j=1}^{m_2} T_j\right) = 0, \tag{6}$$

where $X(s)$ identifies the input-output response, t_i's are those signed product terms multiplied by $X(s)$, and T_j's are the rest of the signed product terms without $X(s)$.

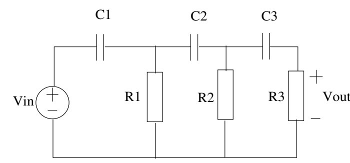

Fig. 3. Circuit example 2.

The GRASS simulator never explicitly enumerates all the signed product terms, which would be an astronomical number for relatively large circuits, rather it enumerates *implicitly* all the product terms by a graph reduction process integrated with a BDD construction process. The BDD hashing mechanism effectively avoids enumerating those sub-SOP expressions already existing, thus eliminating the exponential growth problem [4].

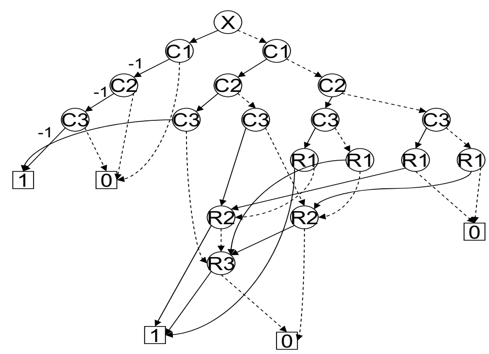

Fig. 4. The SDD for example 2.

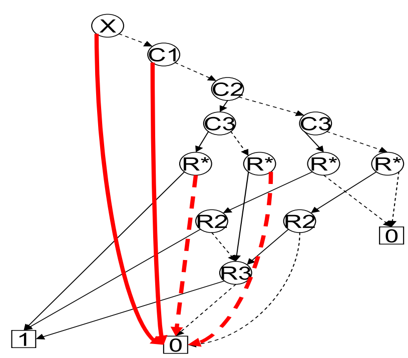

Fig. 5. The differentiated SDD for example 2. The nodes with the sensitivity symbol (marked $R*$) are all set to "1".

Define $N(s) = -\sum_{i=1}^{m_1} t_i$ and $D(s) = \sum_{j=1}^{m_2} T_j$. Note that both $N(s)$ and $D(s)$ are in the form of SOP, hence can be represented by sub-BDDs. In GRASS, the partial sum $-N(s)$ is available at the *1-edge* (solid arrow) of the SDD root node $X(s)$, while the partial sum $D(s)$ is available at the *0-edge* (dashed arrow) of the SDD root $X(s)$. At each specified frequency point $s = j\omega$, dividing each other the two values available at the two pointers from the root node $X(s)$ gives rise to the frequency response of $H(s)$ at ω, i.e., $H(j\omega) = N(j\omega)/D(j\omega)$.

The symbolic AC sensitivity algorithm can be formalized by an inspection of the SDD. Firstly, it is clear from the identities 4, 5, and 6 in the appendix that the sensitivities of the magnitude and phase of $H(s)$ can be derived directly from the sensitivities of $N(s)$ and $D(s)$ w.r.t. any selected parameter. Secondly, as noted earlier, $\mathrm{Sens}(N(s), p)$ and $\mathrm{Sens}(D(s), p)$ can be constructed directly by modifying the SDD within one partial traversal.

It is more instrumental to use another slightly larger circuit example to illustrate this SDD modification process. We shall derive the symbolic sensitivity of $H(s)$ w.r.t. the parameter R_1 for the circuit in Fig. 3.

The SDD for this circuit is constructed in Fig. 4, where the sensitivity symbol R_1 appears in four SDD nodes. We also see that some one-paths involve the sensitivity symbol R_1 while others do not. Intuitively, differentiation requires that those one-paths not involving the symbol R_1 be deleted while the others remain unaltered except for setting the value of R_1 to one. Formally, this can be accomplished by the next algorithm (recall that all symbols are indexed during the BDD construction.)

The Symbolic AC Sensitivity Analysis Algorithm:

Step 1. Specify a sensitivity symbol p, the frequency range, and the number of frequency points requested. (It could be a Spice-like netlist command *.sens*.)

Step 2. Construct an SDD. Then construct a differentiated SDD by modifying the SDD while traversing all one-paths: If the index of the sensitivity symbol p is passed without seeing the symbol p, then redirect the current solid arrow to the terminating node *zero*. If the sensitivity symbol p is encountered, then set the symbol value to $p = 1$ while pointing the dashed arrow from p to the terminating node *zero*. Keep all unaltered signed edges intact. Run *Reduce* to reduce the differentiated SDD.

Step 3. Evaluate the sensitivity from the differentiated SDD in s-expanded form for the selected nominal parameter values at the specified frequency points. Plot the sensitivity curve.

The differentiated SDD w.r.t. R_1 is shown in Fig. 5.

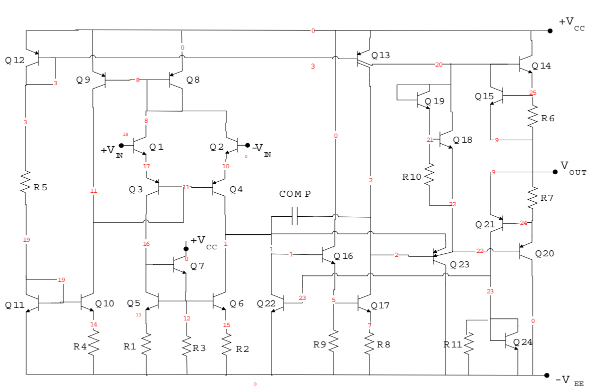

Fig. 6. Schematic of $\mu A741$.

IV. Application Example

Sensitivity analysis is a useful tool for analog circuit designers. Spice simulators provide DC and AC sensitivity analyses. The Spice AC sensitivity analysis is different from the symbolic AC sensitivity addressed in this paper. The Spice AC sensitivity analysis provides a measure of the output signal variation to the input signal variation, rather than to the circuit parameter variation, in the frequency-domain. The GRASS symbolic AC sensitivity provides the analog designer a new way of tuning circuit parameters in the frequency domain. In this section, the $\mu A741$ opamp (Fig. 6) is used as an example to demonstrate this application.

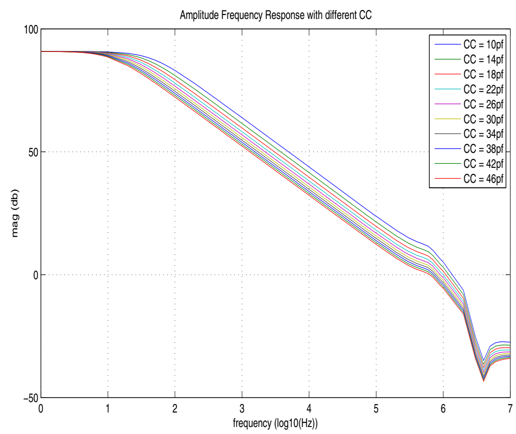

Fig. 7. The variational magnitude versus *COMP* (CC).

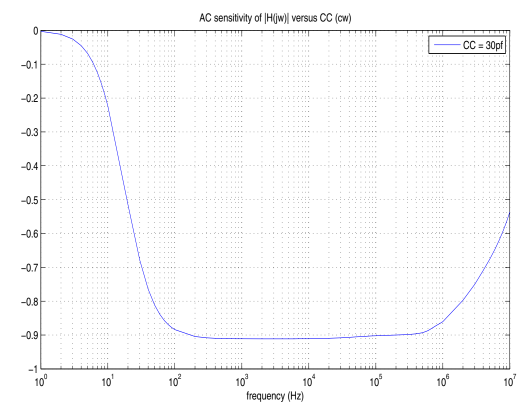

Fig. 8. The sensitivity of magnitude versus *COMP* (CC).

In the symbolic framework, studying the AC sensitivity w.r.t. *any* circuit parameter is equally feasible. For demonstration purpose, we chose to investigate the frequency response sensitivity w.r.t. the compensating capacitor (*COMP*). At an initial design stage, a relatively simple small signal model (refer to Fig. 7 in [4]) would be sufficient for design exploration. This small signal model was used by the GRASS simulator to construct an SDD representation of the symbolic frequency response $H(s)$. After that, a differentiated SDD w.r.t. *COMP* is constructed using the *Symbolic AC Sensitivity Algorithm*. Evaluation of a symbolic sensitivity $\mathrm{Sens}(H(s), p)$ takes time proportional to the number of frequency points requested.

Shown in Fig. 7 is the variational frequency response magnitude $|H(s)|$ versus the compensating capacitor (*COMP*) with all other circuit parameters fixed at their nominal values. We see that in the band of $10Hz$ to $10MHz$ the magnitude varies more appreciably with the parameter *COMP*. The normalized sensitivity $\mathrm{Sens}(|H(s)|, p)$ is plotted in Fig. 8 which shows the relative variation rate across the frequency range. The variational phase plot in Fig. 9 and the phase sensitivity plot in Fig. 10 reveal information on the phase variation. Both the magnitude and phase sensitivities provide the designer a quantitative visualization on the gain/phase margin sensitivities and the design measures on how to improve.

Listed in Table I are the running times measured for two circuit examples of practical size; all data were collected on a personal computer with $2.3GHz$ clock and $2G$ memory. The frequency range was sampled by $1,000$ points. Most of the simulation time was spent on the SDD construction. Once an SDD is constructed, all other repeated analyses took only negligible time.

TABLE I
GRASS SIMULATION TIME FOR $1,000$ FREQUENCY POINTS.

	$\mu A741$	$\mu A725$
Construction of SDD	5.9s	29.25s
s-expansion	2.04s	6.26s
Evaluation of $H(s)$	0.03s	0.05s
Differentiate SDD	0.09s	0.25s
Evaluation of s-coeff.	0.25s	1.3s
Evaluation of $\text{Sens}(H(s),p)$	0.03s	0.03s

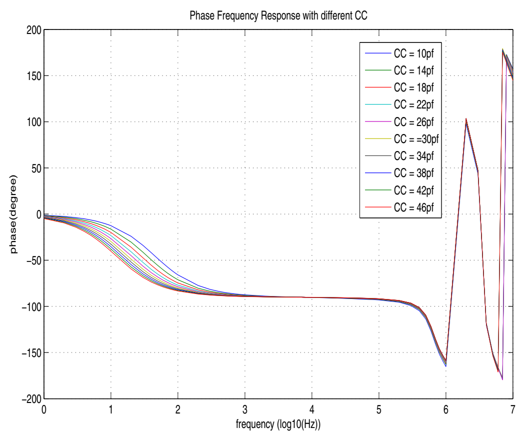

Fig. 9. The variational phase versus *COMP* (CC).

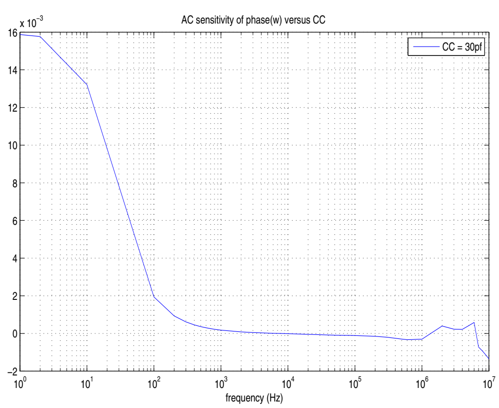

Fig. 10. The sensitivity of phase versus *COMP* (CC).

V. Conclusion

This paper has presented a technique of implementing a symbolic AC sensitivity analysis in the symbolic simulator GRASS. It is stressed that the one-to-one correspondence between circuit parameters and simulator symbols plays a crucial role for an easy and efficient implementation of symbolic AC sensitivity. More applications and design cases are to be explored.

Appendix

A. Some sensitivity identities

Identity 1: Let $G = 1/R$. Then

$$\text{Sens}(H(s), R) = -\,\text{Sens}(H(s), G). \tag{7}$$

Identity 2: Let $H(s) = 1/X(s)$. Then

$$\text{Sens}(H(s), p) = -\,\text{Sens}(X(s), p). \tag{8}$$

Identity 3: Let $H(s) = 1/X(s)$ and $G = 1/R$. Then

$$\text{Sens}(H, R) = \text{Sens}(X, G). \tag{9}$$

Identity 4: Let $H(s) = N(s)/D(s)$

$$\text{Sens}(H(s), p) = \text{Sens}(N(s), p) - \text{Sens}(D(s), p). \tag{10}$$

Proof: This follows from the definition of normalized sensitivity: $\text{Sens}(H(s), p) = \frac{\partial \ln H(s)}{\partial \ln p} = \frac{\partial \ln N(s)}{\partial \ln p} - \frac{\partial \ln D(s)}{\partial \ln p}$. ■

Identity 5:

$$\text{Sens}(|H(s)|, p) = \text{Re}\left\{\text{Sens}(H(s), p)\right\}. \tag{11}$$

Identity 6:

$$\text{Sens}(\angle H(s), p) = \frac{1}{\angle H(s)}\,\text{Im}\left\{\text{Sens}(H(s), p)\right\}. \tag{12}$$

Proof: Knowing $H(s) = |H(s)|e^{j\angle H(s)}$, we get

$$\begin{aligned} & \text{Sens}(H(s), p) \\ = & \frac{\partial \ln\left(|H(s)|e^{j\angle H(s)}\right)}{\partial \ln p} = \frac{\partial \ln |H(s)|}{\partial \ln p} + j\frac{\partial \angle H(s)}{\partial \ln p} \\ = & \text{Sens}(|H(s)|, p) + j\angle H(s)\,\text{Sens}(\angle H(s), p). \end{aligned}$$

■

Identity 7:

$$\text{Sens}(H(s), C) = \text{Sens}(H(s), Cs). \tag{13}$$

Proof: This is a direct result from the Chain Rule. For any fixed s, $\text{Sens}(H(s), C) = \frac{C}{H(s)}\frac{\partial H(s)}{\partial C} = \frac{Cs}{H(s)}\frac{\partial H(s)}{\partial Cs} = \text{Sens}(H(s), Cs)$. ■

References

[1] F. Fernández, A. Rodríguez-Vázquez, J. Huertas, and G. Gielen, *Symbolic Analysis Techniques – Applications to Analog Design Automation*. New York: IEEE Press, 1998.
[2] P. Wambacq, G. Gielen, and W. Sansen, "Symbolic network analysis methods for practical analog integrated circuits: a survey," *IEEE Trans. on Circuits and Systems – II: Analog and Digital Signal Processing*, vol. 45, no. 10, pp. 1331–1341, 1998.
[3] W. Chen and G. Shi, "Implementation of a symbolic circuit simulator for topological network analysis," in *Proc. Asia Pacific Conference on Circuits and Systems (APCCAS)*, Singapore, Dec. 2006, pp. 1327–1331.
[4] G. Shi, W. Chen, and C.-J. R. Shi, "A graph reduction approach to symbolic circuit analysis," in *Proc. Asia South-Pacific Design Automation Conference (ASPDAC)*, Yokohama, Japan, Jan. 2007, pp. 197–202.
[5] C.-J. Shi and X.-D. Tan, "Canonical symbolic analysis of large analog circuits with determinant decision diagrams," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 1, pp. 1–18, January 2000.
[6] S.-D. Tan and C.-J. Shi, "Efficient approximation of symbolic expressions for analog behavioral modeling and analysis," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 6, pp. 907–918, June 2004.
[7] S.-D. Shieu and S.-P. Chan, "Topological formulation of symbolic network functions and sensitivity analysis of active networks," *IEEE Trans. on Circuits and Systems*, vol. CAS-21, no. 1, pp. 39–45, 1974.
[8] H. Yang, M. Ranjan, W. Verhaegen, M. Ding, R. Vemuri, and G. Gielen, "Efficient symbolic sensitivity analysis of analog circuits using element-coefficient diagrams," in *Proc. Asia South-Pacific Design Automation Conference (ASPDAC)*, Yokohama, Japan, Jan. 2005, pp. 230–235.
[9] H. Yang, A. Agarwal, and R. Vemuri, "Fast analog circuit synthesis using multiparameter sensitivity analysis based on element-coefficient diagrams," in *Proc. IEEE Computer Society Annual Symposium on VLSI*, 2005.