Rapid Frequency-Domain Analog Fault Simulation Under Parameter Tolerances*

Michael W. Tian and C.-J. Richard Shi

Department of Electrical and Computer Engineering
University of Iowa
Iowa City, Iowa 52242
{wtian, cjshi}@eng.uiowa.edu

Abstract

Fault-driven analog and mixed-signal testing calls for rapid fault simulation techniques. A problem that has not been addressed effectively by existing research is that circuit parameters have tolerance ranges. In this paper, we propose representing parameters under variations as intervals, and present an efficient algorithm — based on interval analysis and Householder's formula — to compute the worst-case response bounds of good and faulty linear(ized) circuits under parameter variations. Our approach takes CPU time comparable to one nominal circuit simulation, and always produces correct and conservative results. The algorithm has been implemented into SPICE3F5. Experimental results show an acceptable accuracy.

1 Introduction

In the past few years, analog and mixed-signal integrated circuits have grown in importance, due to the rapid convergence of computing, consumer electronics, and communication. Since the early 1990s, the average growth rate of the mixed-signal IC market has been between 15% and 20% per year [3]. The ever increasing level of integration complexity and shortening product life cycles of mixed-signal ICs and systems have created many design and test challenges. Among them, testing the correctness of analog circuits, *i.e.*, analog testing, is one of the most important issues. It is recognized that the influence of analog testing on time-to-market and final cost of the circuit is increasingly significant.

Analog fault simulation is a central issue in analog testing, with applications in test selection, fault coverage analysis [22], and design for test [14, 27]. Some research effort has been directed to exploit standard analog/mixed-signal simulators such as SPICE [7], Eldo [19], and Saber [4] to perform fault simulation. This line of research has been plagued by a number of problems. First, analog circuit parameters are associated with tolerance ranges. Only parametric variations which cause the circuit performances to be outside the preset specifications are considered to be faults. As a result, the responses for both good and faulty circuits are *bands*, called *good (response) band* and *fault (response) band* respectively. This is illustrated in Figure 1.

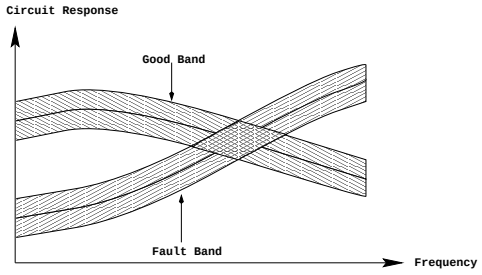

Figure 1: Illustration of good and fault bands.

Another related issue arises in mixed-signal simulation. Unknown states in digital portion give rise to parameter tolerances in analog portion, which cannot be handled directly by circuit simulators. In addition, most conventional circuit simulators encounter the numerical difficulty when simulating circuits that contain opens and shorts. The problem is further compounded by many numbers of faults needed to be simulated.

Methods have been proposed to speed up fault simulation of linear analog circuits some 20 years ago [21, 23] and recently [16, 25]. However, these methods did not address parameter tolerances. To our knowledge, Pahwa and Rohrer were the first to consider efficient fault simulation of linear circuits under parameter tolerances [18]. Nominal sensitivity analysis is used to estimate the parameter worst-case conditions. Fault bands are then approximated from the good band, and the resulting bands are called *band fault*. However, the basic assumption that parameter worst-case conditions remain unchanged even under fault (called the *band fault assumption*) is not hold in general. As a consequence, band fault may deviate from fault bands significantly. Furthermore, no method exists to check whether a given circuit satisfies the band fault assumption.

The most widely used approach to model parameter variations is the Monte Carlo method. With this technique, simulation is repeated for random combinations of values chosen from within the range of each parameter. Unfortunately, accurately determining bounds on the behavior of a circuit requires a large number of simulations to be effective. The method becomes prohibitively expensive for fault simulation, since Monte Carlo method needs to repeat the random sampling process for each fault. Improvements have been suggested to reduce the simulation time by us-

* This work is sponsored by the U.S. Defense Advanced Research Projects Agency (DARPA) under grant number F33615-96-1-5601 from the U.S. Air Force, Wright Laboratory, Manufacturing Technology Directorate, and by the National Science Foundation under grant CDA 96-01503.

Design Automation Conference ®

0-89791-847-9/97/0006/$3.50 DAC 97 - 06/97 Anaheim, CA, USA

ing inductive fault analysis and behavioral modeling [6]. However, no systematic methods exist for extracting behavioral (fault) models from transistor-level (fault) models. Recently, Spinks and Bell proposed the use of the Monte Carlo method to estimate the parameter worst-case conditions, and then approximate the faulty responses by performing circuit simulation on faulty circuits using the same parameter worst-case conditions [22]. This method suffers the same shortcoming as the band fault approach.

In this paper, we present a rapid, correct and conservative approach for frequency-domain fault simulation of linear(ized) analog circuits and systems under parameter variations. A large class of circuits and systems widely used in video and image processing, digital signal processing, control, communications, and many other applications fall into this category. Further, recent studies have revealed that faults which shift the operating point of a transistor-level analog circuit can be easily detected by inexpensive DC testing or power supply current monitoring, whereas the most difficult faults to detect are those that only cause performance deviations [5]. It is this class of linearized analog circuits that our research is targeted at.

This paper details our approach and presents a prototype analog fault simulator utilizing the proposed technique. An interval based framework for handling parameter variations is introduced in Section 2. In Section 3, using this framework, circuit equations are formulated via Generalized MNA formulation (*GNA*) approach as a set of linear interval equations, and analog faults are modeled as changes of intervals. Computing the good bands amounts to solve a set of linear interval equations. A recent algorithm for this purpose is adopted in Section 4. Section 5 shows how fault bands can be obtained with a very minor computational cost from the good band using Householder's formula. Implementation and experimental results are described in Section 6. Section 7 concludes the paper.

2 Notations from Interval Mathematics

Let $p \in \mathcal{R}$ be a real number whose value may not be precisely known. Instead, we are often given a range and p is uncertain within this range. This can be represented by an *interval number* p^I, with *lower (left) bound* p^L and *upper (right) bound* p^R, denoted by $p^I = [p^L, p^R]$. The *midpoint* $mid(p^I)$ of an interval number p^I is defined as:

$$mid(p^I) = \frac{1}{2}(p^R + p^L),$$

and the *radius* $rad(p^I)$ of p^I is defined as:

$$rad(p^I) = \frac{1}{2}(p^R - p^L).$$

Given two interval numbers a^I and b^I, the following interval arithmetic operations are defined as follows [1]:

$$\begin{aligned} a^I + b^I &= [a^L + b^L, a^R + b^R] \\ a^I - b^I &= [a^L - b^R, a^R - b^L] \end{aligned}$$

$$\begin{aligned} a^I \times b^I &= [min(a^L b^L, a^L b^R, a^R b^L, a^R b^R), \\ &\qquad max(a^L b^L, a^L b^R, a^R b^L, a^R b^R)] \end{aligned}$$

$$\begin{aligned} a^I / b^I &= [min(a^L / b^L, a^L / b^R, a^R / b^L, a^R / b^R), \\ &\qquad max(a^L / b^L, a^L / b^R, a^R / b^L, a^R / b^R)] \end{aligned}$$

For example, let $x^I = [0, 2]$, then $1 - x^I = [-1, 1]$, $x^I * x^I = [0, 4]$, and $1 - x^I + x^I * x^I = [-1, 1] + [0, 4] = [-1, 5]$, where actually the value set $\{1 - x + x * x \mid x \in x^I\} = [3/4, 3]$. This demonstrates a peculiar characteristic of interval operation: the value set may be *expanded* (*overestimated*) due to the fact that correlations among the values represented by intervals (e.g., $1 - x^I = [-1, 1]$ and $x^I * x^I = [0, 4]$) are ignored by interval arithmetic.

An *interval vector* $\mathbf{x}^I$ is a vector whose elements are interval numbers, and we write an interval vector as $\mathbf{x}^I = [\mathbf{x}^L, \mathbf{x}^R]$. An *interval matrix* $\mathbf{A}^I$ is a matrix whose elements are interval numbers and we write an interval matrix as $\mathbf{A}^I = [\mathbf{A}^L, \mathbf{A}^R]$.

3 Generalized MNA Formulation of Circuit Equations and Fault Models

Consider a linear time-invariant circuit, where some circuit parameters are under variations, and represented by interval numbers. The *Modified Nodal Analysis* (*MNA*) method developed by Ho, *et. al.* [13] and popularized by SPICE [15], is adopted with slight modification to formulate the circuit equations. Basically, for all the components which do not have interval parameters, the rules of MNA are followed. For every component with interval parameters, an additional variable (branch current) and an additional equation (branch equation) are introduced for every interval parameter. This is called *generalized MNA formulation*, or simply *Generalized Nodal Analysis) (GNA)*. Note that the GNA formulation for a circuit without parameter variations degenerates to the original MNA formulation. In general, the GNA formulation results in a set of complex linear interval equations, represented as:

$$\mathbf{T}^I \mathbf{x}^I = \mathbf{w}^I \tag{1}$$

Consider a circuit shown in Figure 2, where r_1 is under parameter variation.

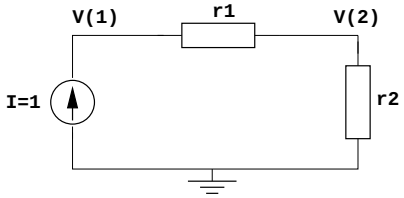

Figure 2: A simple two-resistor circuit.

The GNA formulation is as follow:

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1/r_2 & -1 \\ 1 & -1 & -r_1^I \end{pmatrix} \begin{pmatrix} v(1) \\ v(2) \\ i(r_1) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Note that GNA formulation may have more variables and equations than the original MNA formulation. Similar to the sparse tableau formulation [11], it takes approximately the same computational effort as the MNA by using the sparse matrix technique [26]. However, the proposed GNA formulation has following "nice" properties:

- At most one interval matrix entry appears in any given row or any given column of $\mathbf{T}^I$.
- Each interval matrix entry is contributed by a unique circuit parameter under variation and *vice versa*.
- If all interval circuit parameters are independent with each other, then all the matrix entries in $\mathbf{T}^I$ are independent with each other.

These properties help to find tight solution bounds.

The system of complex interval equations in Eqs. (1) can be transformed to a system of real interval equations as follows:

$$\begin{pmatrix} \mathbf{T}^I_{\mathcal{R}} & -\mathbf{T}^I_{\mathcal{I}} \\ \mathbf{T}^I_{\mathcal{I}} & \mathbf{T}^I_{\mathcal{R}} \end{pmatrix} \begin{pmatrix} \mathbf{x}^I_{\mathcal{R}} \\ \mathbf{x}^I_{\mathcal{I}} \end{pmatrix} = \begin{pmatrix} \mathbf{w}^I_{\mathcal{R}} \\ \mathbf{w}^I_{\mathcal{I}} \end{pmatrix} \tag{2}$$

where subscripts $\mathcal{R}$ and $\mathcal{I}$ denote, respectively, the real part and the imaginary part of a complex matrix, vector, or number. Therefore, under our formulation, frequency-domain circuit simulation under parameter variations amounts to solve Eqs. (2) — a set of linear interval equations — for a given set of frequency points.

A fault associated with circuit component can be represented by a change of component parameter from an original interval, say p^I, to a new interval, say q^I. This model captures various analog faults due to fluctuations in the IC manufacturing process. In particular,

- Parametric faults. Parametric faults are excessive statistical variations in manufacturing process conditions which cause circuit performances to be outside of some preset specification tolerance. There are two special cases: nominal value shifting, i.e.,
$$T^I_{ij} \Longrightarrow T^I_{ij} + \Delta t$$
and variation expansion, i.e.,
$$T^I_{ij} \Longrightarrow [T^L_{ij} - \Delta t, \quad T^R_{ij} + \Delta t]$$
Here Δt is a scalar quantity, not an interval.
- Structural faults. Structural faults are random defects which cause opens and/or shorts of circuit components, *i.e.*, changes of circuit topology. Examples of such faults are lithography errors and oxide pinholes. We distinguish two types of structure faults. Type I refers to resistive shorts and conductive opens, which forces the circuit matrix entry to zero, and can be modeled as:
$$T^I_{ij} \Longrightarrow 0$$
Type II refers to resistive opens and conductive shorts, which forces the circuit matrix entry to infinity, and can be modeled as
$$T^I_{ij} \Longrightarrow \infty$$

4 Solving Systems of Linear Interval Equations

In this section, we consider how to solve a system of real linear interval equations in the following form

$$\mathbf{A}^I \mathbf{x}^I = \mathbf{b}^I \tag{3}$$

The tightest bounds $\mathbf{x}^I$ can be obtained by solving the set of equations over all the possible combinations of parameter values within the given interval ranges and taking the union of the solutions. This is known as *United Extension*, but it is not computationally feasible [1]. *Gaussian Elimination* and *LU Decomposition* with interval arithmetic can be applied [17]. However, these approaches have been known to produce too loose solution bounds, and were never applied to practical problems.

We have developed an efficient and elegant algorithm, which is described in pseudo-code in Fig. 3. The major computation cost comes from computing the inverse of two matrices ($\mathbf{C}^{-1}$ and $(\mathbf{M}^L)^{-1}$). The complexity of the algorithm is $O(n^3)$ — the same as nominal circuit simulation. We have proved that the computed bounds always contain the exact solution bounds [20]. Further, the expansion of solution bounds is relatively small, as we have observed in our experiments.

```
INTERVAL_SYSTEM_SOLVE(A^I, b^I)
 1  C_ij ← (A^L_ij + A^R_ij)/2   i, j = 1, 2, ..., n
 2  M^I ← C^{-1} A^I   r^I ← C^{-1} b^I
 3  P ← (M^L)^{-1}
 4  for i = 1 to n do
 5       s_i ← |mid(r^I_i)| + rad(r^I_i)
 6  for i = 1 to n do
 7       f_i ← Σ_{j=1}^{n} P_ij s_j
 8       g_i ← f_i − 2P_ii |mid(r^I_i)|
 9       if g_i ≥ 0
10            if mid(r^I_i) > 0
11                 x^I_i ← [−g_i, f_i]
12            else
13                 x^I_i ← [−f_i, g_i]
14       else
15            if mid(r^I_i) > 0
16                 x^I_i ← [−g_i/(2P_ii − 1), f_i]
17            else
18                 x^I_i ← [−f_i, g_i/(2P_ii − 1)]
19  return x^I
```

Figure 3: An algorithm for solving linear interval systems.

The algorithm is based on an original idea of Hansen [10], and exploited later in [24]. Lines 1 and 2 transform Eqs.(3) into the following system:

$$\mathbf{M}^I \mathbf{x}^I = \mathbf{r}^I \tag{4}$$

This is known as *preconditioning*. Preconditioning may expand the solution bounds, however, the *preconditioned* system has a special property that matrix M^I is centered around the identity matrix $\mathbf{I}$, *i.e.*:

$$1 - M^L_{ii} = M^R_{ii} - 1 \quad \text{and} \quad -M^L_{ij} = M^R_{ij} \quad i \neq j$$

As shown in [20], the rest of the algorithm computes the exact solution bounds of the preconditioned system. Note that Hansen's original method requires the interval matrix $\mathbf{A}^I$ to be *strongly diagonal dominant*, and is thus not applicable directly to our application.

5 An Efficient Fault Simulation Algorithm

In this section, we show how fault simulation can be performed using a small amount of computation effort from the simulation results of the good circuit. The method is based on Householder's inverse matrix formula and interval operations. The key is to exploit the fact that the circuit equation for a faulty circuit differs slightly from that of the good circuit.

Let us assume that matrix entry T_{ij} is shifted by an interval Δt^I_{ij}. Then the system of faulty circuit equations

$$\mathbf{T}^I_f \mathbf{x}^I_f = \mathbf{w}^I \tag{5}$$

can be written as

$$(\mathbf{T}^I + \Delta t^I_{ij} \mathbf{e}_i \mathbf{e}_j^T) \mathbf{x}^I_f = \mathbf{w}^I \tag{6}$$

According to Householder's inverse matrix formula [12], we have

$$(\mathbf{T}^I_f)^{-1} = (\mathbf{T}^I)^{-1} - \frac{(\mathbf{T}^I)^{-1} e_i e_j^T}{(\Delta t^I_{ij})^{-1} + ((\mathbf{T}^I)^{-1})_{ji}} (\mathbf{T}^I)^{-1} \tag{7}$$

Note that

$$\mathbf{x}^I_f = (\mathbf{T}^I_f)^{-1} \mathbf{w}^I \quad \text{and} \quad \mathbf{x}^I = (\mathbf{T}^I)^{-1} \mathbf{w}^I \tag{8}$$

then we have

$$x^I_{fk} = x^I_k - m^I_k x^I_j \qquad k = 1, 2, \ldots, n \tag{9}$$

where

$$m^I_k = \frac{((\mathbf{T}^I)^{-1})_{ki}}{(\Delta t^I_{ij})^{-1} + ((\mathbf{T}^I)^{-1})_{ji}} \tag{10}$$

The equation above can be simplified for each specific type of faults:

- Parametric faults. Let Δt^I_{ij} denotes the difference of matrix entry T_{ij} between the good circuit and faulty circuit, then

$$m^I_k = \frac{((\mathbf{T}^I)^{-1})_{ki}}{(\Delta t^I_{ij})^{-1} + ((\mathbf{T}^I)^{-1})_{ji}} \tag{11}$$

- Structural fault — type I. Suppose a fault forces the value of matrix entry T_{ij} to be zero, for example, short fault of a resistive component and open fault of a conductive component, then we have

$$m^I_k = \frac{((\mathbf{T}^I)^{-1})_{ki}}{-T^{-1}_{ij} + ((\mathbf{T}^I)^{-1})_{ji}} \tag{12}$$

  Note that T_{ij} here is the nominal value of T^I_{ij} instead to be an interval number.

- Structural fault — type II. Suppose that a fault forces the value of matrix entry T_{ij} to be infinity, for example, open fault of a resistive component and short fault of a conductive component, then we have

$$m^I_k = \frac{((\mathbf{T}^I)^{-1})_{ki}}{((\mathbf{T}^I)^{-1})_{ji}} \tag{13}$$

Fault simulation consists of computing (9), (11), (12), and (13).

We observe that the ith column of $(\mathbf{T}^I)^{-1}$ is the solution of the following system of linear interval equations:

$$\mathbf{T}^I \mathbf{x}^I = \mathbf{e}_i \tag{14}$$

where $\mathbf{e}_i$ is a vector with ith component equal to 1, and the rest of components being 0. Eqs. (14) can be solved very efficiently, since Eqs. (14) has the same coefficient matrix as Eqs. (1) for the good circuit, and $\mathbf{C}^{-1}$ and $\mathbf{P}^{-1}$ are available already from good circuit simulation.

We note that all quantities in Eqs (9), (11), (12), and (13) are *interval complex quantities*, and are computed using interval operations. This method is rapid. However, because interval operations ignore the correlation among interval quantities x^I_k, x^I_j, $((\mathbf{T}^I)^{-1})_{ki}$ and $((\mathbf{T}^I)^{-1})_{ji}$ in Eqs. (9), (11), (12) and (13), certain bound expansion may occur. Nevertheless, the results are still correct—computed bounds always contain the exact bounds. Further, as validated by our experiments, the bound expansion is usually small for not substantially large parameter variations. As an alternative, we can perform fault simulation by applying directly the algorithm in Fig. 3 to solve Eqs. (5).

6 Experimental Results

The proposed algorithms have been implemented into a computer program SIVA-AC (*S*PICE3 *I*ncorporating *V*HDL-*A*). SIVA-AC is based on SPICE3F5, and extends the sparse matrix package in SPICE3F5 to implement the proposed algorithms described in Sections 4-5. A number of experiments have been conducted mainly to test the validity of analog fault simulation by our interval analysis based algorithms. The CPU time, which is collected on a SPARC-Ultra 1 workstation, is also given for comparison.

6.1 State Variable Filter

We test our program on the state variable filter circuit [9]. Its schematic is shown in Figure 4. The nominal parameter values are given as

$$R_1 = 10k, R_2 = 20k, R_3 = 10k, R_4 = 10k$$

$$R_5 = 100k, R_6 = 10k, R_7 = 30k, R_8 = 45.5k$$

$$R_9 = 2.2k, R_{10} = 10k, C_1 = 1.6nf, C_2 = 1.6nf$$

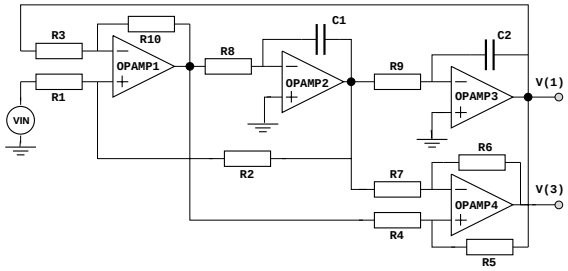

Figure 4: State variable filter circuit.

We perform parametric fault simulation. Since the outputs of the state variable filter are very sensitive to parameters, for the clarity of comparison, we assume that all parameters have $\pm 0.1\%$ statistical variations. Two parametric faults considered are $R1$ with 50% and -50% changes, respectively. The simulated fault bands and good band are shown in Figs (5)-(6), where the good band is computed by the algorithm in Figure 3, and fault bands are computed from Eqs. (9) and (11).

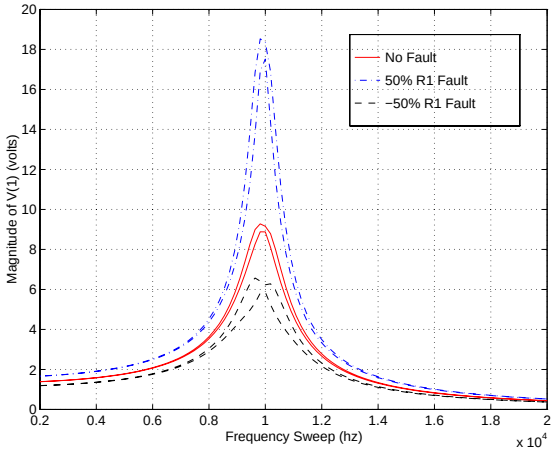

Figure 5: Parametric fault simulation: Magnitude of $V(1)$.

The CPU time taken for good band computation over 100 frequency points is 16.78 seconds, and 0.93 seconds for one fault simulation. An interesting observation is that the magnitude fault bands exhibit significant difference from that of the good band and also between each other over a wide range of frequency, while the phase fault bands do not. This implies that the magnitude can be used as fault detectability measure and diagnosibility measures.

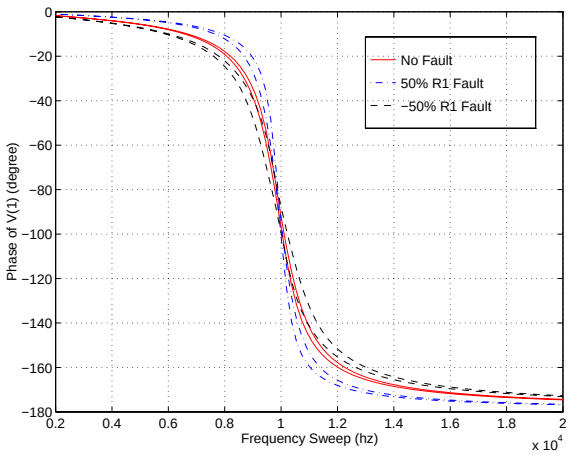

Figure 6: Parametric fault simulation: Phase of $V(1)$.

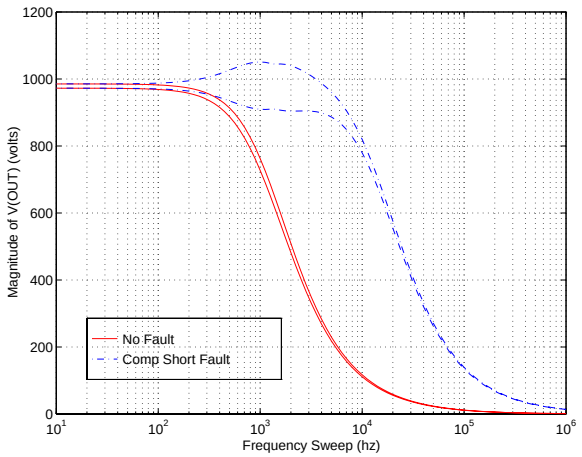

Figure 8: Structural fault simulation: Magnitude of $V(OUT)$.

6.2 μA 741 Operational Amplifier

We further test our program on the the more complex μA 741 operational amplifier. Figure 7 shows its schematic. The nominal values of circuit parameters are

$$R_1 = 1k, R_2 = 50k, R_3 = 1k, R_4 = 3k$$

$$R_5 = 39k, R_6 = 50, R_7 = 25, R_8 = 100$$

$$R_9 = 50k, R_{10} = 40k, R_{11} = 50k, COMP = 30pf$$

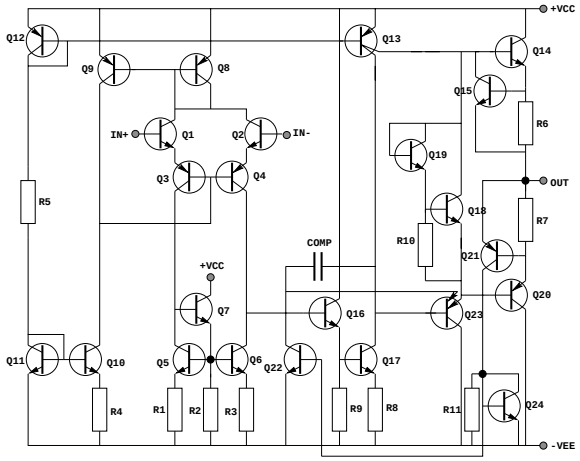

Figure 7: μA741 operational amplifier circuit.

All the parameters are assigned $\pm 2\%$ statistical variations. We consider a structural fault: the compensation capacitor $Comp$ is shorted. Figure 8 and Figure 9 show both the magnitude and phase fault bands, as well as the good bands.

Both the magnitude and phase fault bands show a significant difference from their corresponding good band over the entire interested frequency range. It can be seen that the $3DB$ frequency changing point increases about 10 times if the compensation capacitor is shorted. For 100 frequency points, the CPU time taken is 90.44 seconds for good band computation, and 6.49 seconds for each fault simulation.

7 Conclusions

An interval-based framework was presented for frequency-domain fault simulation of linear(ized) analog circuits and systems under parameter variations. Our approach has several major features that differentiate it from previous approaches.

1. Our method is extremely fast, and comparable to regular nominal circuit simulation. Suppose that there are m-trial Monte Carlo simulation for n faults, the proposed fault simulation method (using Householder's formula) gains a theoretical speed-up of $O(mn)$ over the Monte Carlo method.
2. Our method is accurate and conservative. The fault bands computed by our method always contain accurate fault bands. This property ensures the correctness and robustness of the use of our fault simulation method in test selection, fault coverage analysis and design for test. Further, if the ranges of parameter tolerances are relatively narrow, our method produces very tight bounds. For those faults that cause a large change of circuit performance, we note that they can be easily detected by inexpensive DC testing and power supply current monitoring (no AC testing is needed).
3. Since fault simulation is performed by Householder's formula, the numerical singularity problem associated with the direct use of conventional circuit simulators do not exist.
4. Because fault simulation is performed at the circuit level, no mapping of faults to the behavioral level, as required by DRAFTS [16], nor behavioral fault modeling [6, 8], is needed. This ensures fault coverage unchanged.

REFERENCES

[1] G. Alefeld, *Introduction to Interval Computations*, New York: Academic Press, 1983.

[2] I.M. Bell, K.R. Eckersall, S.J. Spinks and G.E. Taylor, Fault oriented test and fault simulation of mixed signal integrated circuits, pp. 389–392 in *Proc. IEEE International Symp. Circuits and Systems*, 1995.

[3] L.R. Carley, G.E. Gielen, R.A. Rutenbar and W.M.C. Sansen, Synthesis tools for mixed-signal ICs: progress on frontend and backend strategies, pp. 298–303 in *IEEE/ACM Design Automation Conference*, 1996.

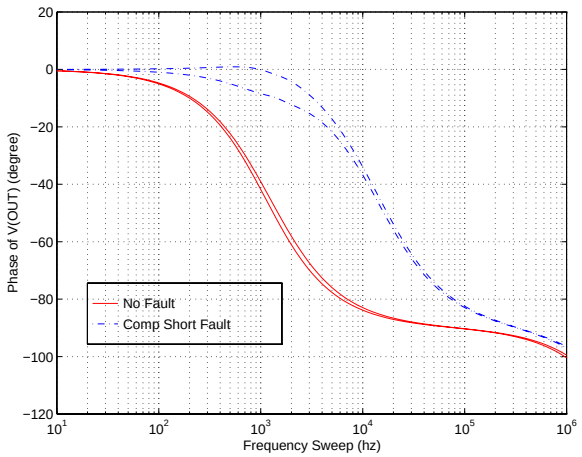

Figure 9: Structural fault simulation: Phase of $V(OUT)$.

[4] P. Caunegre and C. Abraham, Fault simulation capacities for mixed-signal systems, pp. 469–477 in *Proc. European Design and Test Conf.*, 1995.

[5] C.Y. Chao, H.J. Lin and L. Milor, Optimal testing of VLSI analog circuits, to appear in *IEEE Trans. Computer-Aided Design*.

[6] G. Devarayanadurg, P. Goteti and M. Soma, Hierarchy based statistical fault simulation of mixed-signal ICs, pp. 521-527 in *Proc. IEEE International Test Conference*, Oct. 1996.

[7] B.R. Epstein, M. Czigler and S.R. Miller, Fault detection and classification in linear integrated circuits: an application of discrimination analysis and hypothesis testing, *IEEE Trans. Computer-Aided Design*, vol. 12, no. 1, pp. 102–113, Jan. 1993.

[8] E. Felt and A.L. Sangiovanni-Vincentelli, Testing of analog systems using behavioral models and optimal experimental design techniques, pp. 672–678 in *Proc. International Conf. Computer-Aided Design*, Nov. 1994.

[9] N.B. Hamida, K. Saab, D. Marche, B. Kaminska and G. Quesnel, LIMSoft: Automated tool for design and test integration, pp. 56-71 in *Proc. 2rd IEEE International Mixed Signal Testing Workshop*, Qubéc City, Canada, May 1996.

[10] E.R. Hansen, Bounding the solution of interval linear equations, *SIAM J. Numerical Analysis*, vol. 29, no. 5, pp. 1493–1503, Oct. 1992.

[11] G.D. Hachtel, R.K. Brayton and F.G. Gustavson, The sparse tableau approach to network analysis and design, *IEEE Trans. on Circuit Theory*, vol. 18, pp. 101-113, Jan. 1971.

[12] A.S. Householder, A survey of some closed methods for inverting matrices, *SIAM J. Appl. Math.*, vol. 5, pp. 155–169, 1957.

[13] C.W. Ho, A.E. Ruehli and P.A. Brennan, The modified nodal approach to network analysis, *IEEE Trans. Circuits and Systems*, vol. 22, pp. 504-509, June 1975.

[14] S. Mir, V. Kolarik, M. Lubaszewski, C. Nielsen and B. Courtois, Built-in self-test and fault diagnosis of fully differential analogue circuits, pp. 486–490 in *IEEE International Conf. on Computer-Aided Design*, Nov. 1994.

[15] L.W. Nagel, *SPICE2: A Computer Program to Simulate Semiconductor Circuits*, ERL Memorandum M520, Ph.D. Dissertation, Department of Electrical and Computer Engineering, University of California, Berkeley CA, May 1975.

[16] N. Nagi, A. Chatterjee and J.A. Abraham, DRAFTS: Discretized analog circuit fault simulator, pp. 509–514 in *Proc. IEEE/ACM Design Automation Conference*, 1993.

[17] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge, U.K.: Cambridge University Press, 1990.

[18] A. Pahwa and R.A. Rohrer, Band-faults: Efficient approximations to fault bands for the simulation before fault diagnosis of linear circuits, *IEEE Trans. Circuits and Systems*, vol. 29, no. 2, pp. 81-88, Feb. 1982.

[19] C. Sebeke, J.P. Teixeira and M.J. Ohletz, Automatic fault extraction and simulation of layout realistic faults for integrated analogue circuits, pp. 464–468 in *Proc. European Test Conf.*, 1995.

[20] C.J.R. Shi and M.W. Tian, Simulation and sensitivity of linear(ized) analog circuits under parameter variations by robust interval analysis, submitted to *ACM Trans. Design Automation of Electronic Systems*.

[21] K. Singhal and J. Vlach, Solution of modified systems and applications, pp. 620–623 in *Proc. IEEE International Symp. Circuits and Systems*, 1981.

[22] S.J. Spinks and I.M. Bell, A comparison of relative accuracy of fault coverage analysis techniques based on analogue fault simulation, pp. 17-22 in *Proc. 2rd IEEE International Mixed Signal Testing Workshop*, Qubéc City, Canada, May 1996.

[23] G.C. Temes, Efficient methods of fault simulation, pp. 191–194 in *Proc. 20th Midwest Symp. Circuits and Systems*, 1977.

[24] W. Tian, X.T. Ling and R.W. Liu, Novel methods for circuit worst-case tolerance analysis, *IEEE Trans. Circuits and Systems Part I*, pp. 272–278, 1996.

[25] P.N. Variyam and A. Chatterjee, Fast fault simulation of linear analog systems using polynomial waveform representations, pp. 11–16 in *Proc. 2rd IEEE International Mixed Signal Testing Workshop*, Qubéc City, Canada, May 1996.

[26] J. Vlach and K. Singhal, *Computer Methods for Circuits Analysis and Design*, Van Nostrand Reinhold Press, pp. 214-222, 1994.

[27] B. Vinnakota, R. Harjani and N.J. Stessman, System-level design for test of fully differential analog circuits, pp. 450–454 in *Proc. IEEE/ACM Design Automation Conf.*, June 1995.