

Research Article

Graph-Based Symbolic Technique and Its Application in the Frequency Response Bound Analysis of Analog Integrated Circuits

E. Tlelo-Cuautle, S. Rodriguez-Chavez, and A. A. Palma-Rodriguez

INAOE, 72840 Tonantzintla, Puebla, PUE, Mexico

Correspondence should be addressed to E. Tlelo-Cuautle; etlelo@inaoep.mx

Received 21 March 2014; Accepted 2 July 2014; Published 17 July 2014

Academic Editor: Jianshe Lei

Copyright © 2014 E. Tlelo-Cuautle et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new graph-based symbolic technique (GBST) for deriving exact analytical expressions like the transfer function $H(s)$ of an analog integrated circuit (IC), is introduced herein. The derived $H(s)$ of a given analog IC is used to compute the frequency response bounds (maximum and minimum) associated to the magnitude and phase of $H(s)$, subject to some ranges of process variational parameters, and by performing nonlinear constrained optimization. Our simulations demonstrate the usefulness of the new GBST for deriving the exact symbolic expression for $H(s)$, and the last section highlights the good agreement between the frequency response bounds computed by our variational analysis approach versus traditional Monte Carlo simulations. As a conclusion, performing variational analysis using our proposed GBST for computing the frequency response bounds of analog ICs, shows a gain in computing time of 100x for a differential circuit topology and 50x for a 3-stage amplifier, compared to traditional Monte Carlo simulations.

1. Introduction

With the downscaling of the integrated circuit (IC) technology, nanometer circuit designs become more and more sensitive to process variations [1, 2], which are produced by fluctuations at the moment of manufacturing, and have been continuously increasing in relative magnitude as IC technology continues to scale to 45 nm and below. On the one hand, IC designers usually perform repeated Monte Carlo (MC) simulations to predict variations, leading to an expensive computational cost, while the main goal is computing the bounds of a given performance by varying the value of some parameters under certain percentage. On the other hand, to mitigate this drawback on performing repeated MC simulations, performance bound methods emerged as attractive techniques for variational analysis of analog ICs under parameter variations [3–7]. However, although those bound methods are quite efficient computing the lower and upper bounds of the magnitude and phase of a transfer function $H(s)$, systematic methods have not been proposed yet to obtain the variational symbolic expression for $H(s)$. Henceforth, this paper introduces a performance bound

technique that derives the exact analytical $H(s)$ of a linear (linearized) analog IC, as CMOS amplifiers, by applying a new graph-based symbolic technique (GBST), which is described in the next section. Afterwards, the frequency response bounds, for the magnitude and phase for $H(s)$ of three amplifiers, are obtained by applying nonlinear constrained optimization.

2. Graph-Based Symbolic Technique

Symbolic analysis of analog circuits has been demonstrated to be useful for deriving analytical expressions such as $H(s)$ [8–14]. An important contribution on the development of symbolic analysis techniques was the introduction of determinant decision diagrams (DDDs) [8, 9], whose advantage is computing symbolic expressions sharing many subexpressions [10, 11]. However, it was not realized for deriving exact analytical expressions for CMOS amplifiers [15]. That way, we introduce herein a new graph-based symbolic technique (GBST) for deriving exact symbolic expressions of analog

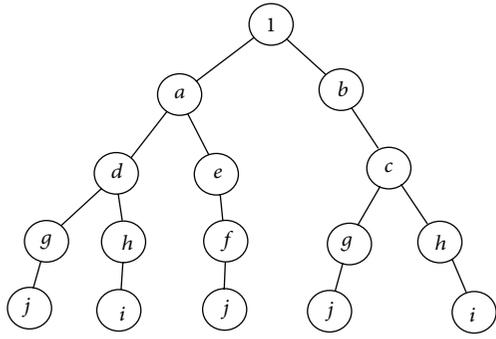


FIGURE 1: Graph representation of (1).

ICs. Our proposed GBST exploits DDD concepts and has the advantage of being compact and unique as well.

2.1. Simple Case: Symbolic Determinant without Node Reuse. Let us consider the determinant given by (1) [9], of size $n \times n$. By applying GBST, the graph representation is built in a depth-first search (DFS) fashion, while one expects having paths of $n + 1$ levels. Every element in the graph corresponds to a nonzero entry in M . In this manner, one obtains the graph shown in Figure 1, where applying the rule of signs from Cramer’s rule (see (2)) does the assignation of signs to each node. A path is eliminated if a zero entry in the nodal admittance matrix is found. Consider

$$|M| = \begin{vmatrix} a & b & 0 & 0 \\ c & d & e & 0 \\ 0 & f & g & h \\ 0 & 0 & i & j \end{vmatrix} = adgj - adhi - aefj - bcgj + bchi, \quad (1)$$

$$\text{sign} = (-1)^{\text{row}+\text{col}}. \quad (2)$$

A tree in which arithmetic operations are encoded in the depth of the tree nodes represents the graph. That is, different depth implies multiplication and equal depth implies addition. This leads us to derive the expression:

$$|M| = a [d (gj - hi) + e (-fj)] + b [c (-gj + hi)]. \quad (3)$$

2.2. Advanced Case: Symbolic Determinant with Node Reuse. In (3), the determinant is expressed as a sequence of products [8, 9], because the graph shown in Figure 1 does not reuse node information, while having many repetitive terms that correspond to the repeated minors gj and hi . For this case, the smallest the matrix minor, the highest the repetition rate. Therefore, our main idea is reusing the information of those repeated nodes. For example: gj , hi , and fj are product terms of 2×2 minors in (1). In this manner, there are five two-nodded subgraphs with vertex sets $V_1 = \{g, j\}$, $V_2 = \{h, i\}$, $V_3 = \{f, j\}$, $V_4 = \{g, j\}$, and $V_5 = \{h, i\}$. As one sees, edges $E_1 = \{g, j\}$ and $E_4 = \{g, j\}$ are equivalent, so that subgraphs $G_1 = \{V_1, E_1\}$ and $G_4 = \{V_4, E_4\}$ carry the same information. In this case, node reuse is possible, and one just needs to identify node ancestors; for example, the subgraphs $G_1 = G_4$ and $G_3 = \{V_3, E_3\}$ have the node ancestors: g for

$G_1 = G_4$ and f for G_3 . Recall that each node is linked to a nonzero matrix entry; that is, $A_{\text{row,col}} \neq 0$. Node f is in turn a representation of $A_{3,2}$ and g of $A_{3,3}$.

Extending node reuse in the whole graph, the first obvious consequence is that there are no repeated nodes; in other words, for a matrix A with nz nonzero entries, there are nz nodes.

2.3. The Advanced Case in Detail. To derive the symbolic expression with node reuse, three different data structures are required. The first and most obvious is the node structure that contains the following:

- (i) node name: a unique name for each node, assigned as an index number;
- (ii) terms: an array containing the index and sign of the element;
- (iii) column: the column of the nonzero entry where the node belongs to;
- (iv) descendants: an array of node pointers linking to the descendants of the current node in the graph structure.

The second data structure is a graph type with the fields:

- (i) graph name: a unique name for the current graph. It is possible to have many different graphs; for example, to compute the transfer function $H(s)$, two graphs are required: one for deriving the numerator and the other for the denominator [9];
- (ii) matrix size: the size of the square matrix;
- (iii) root node: it is a trivial node with term value equal to 1 (see Figure 1) and row and column equal to zero. When multiple graphs are constructed (during factorization), the root node can be any of the nz nodes;
- (iv) visited columns: when traversing a graph to represent the determinant, the column of a visited node is appended to this array.

The third structure stores matrix elements and the independent vector.

From these structures, $nz + 1$ nodes are created. The row and column fields are used to evaluate the sign by (2) as well as to determine which nodes are to be skipped. The graph is generated starting with a trivial node named 0 with term value of 1; then the multiplication of terms is codified as the depth in the graph, and nodes are linked accordingly. The algorithm to build the graph structure for the representation of $|A|$ is sketched by Algorithm 1. Algorithm 2 computes the expression for the determinant. That way, the graph associated to the determinant in (1) is shown in Figure 2.

2.4. Symbol Factorization. Factorization takes place by executing Algorithm 3. The result is the expression as a polynomial that is represented by an array of sum of products with one entry for each power of symbol w .

```

m ← number of Columns of A
n ← number of Rows of A
D ← set of descendants of Ancestor
Ensure: m > 0 ∧ n = m
function buildGraph(A, Ancestor)
  if m > 1 then
    prune = 1
    for all Columns j that Ai,j ≠ 0 do
      if BUILDGRAPH(Cij, Ai,j) ≠ 0 then
        D = D ∪ Ai,j
        prune = 0
  else
    if Ai,j ≠ 0 then
      prune = 0
  return prune
    
```

ALGORITHM 1: Build graph (A(i, j), Ancestor).

- (1) Read the graph in a DFS fashion:
 - (a) Keep track of which columns have been visited
 - (b) Skip nodes from columns already visited
- (2) The symbolic expression is the sum of products according to the visited descendants

ALGORITHM 2: Symbolic determinant from a graph.

- (1) Expand the nodes in the graph containing the symbol w.
- (2) Read the graph in a DFS fashion and preserve only those routes from root to bottom with at least one occurrence of symbol w.
- (3) The number of occurrences of the symbol is the power of w for a given route.
- (4) Express each route as a sum of products.
- (5) The coefficients are the summation of all appended routes for each power of w.

ALGORITHM 3: Graph to polynomial form.

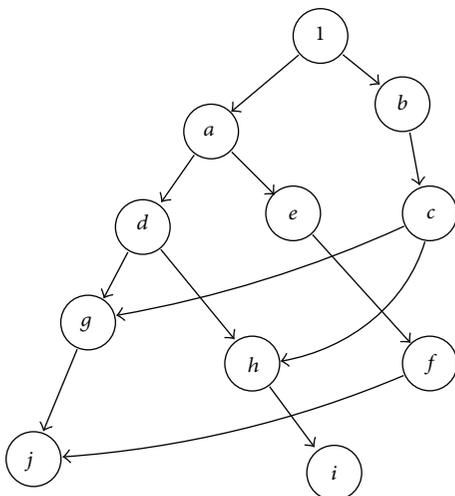


FIGURE 2: Determinant graph with node reuse.

Let us consider matrix M in (4). To derive the polynomial form as powers of w, the associated graph is shown in Figure 3, and the polynomial is given by (5). Consider

$$M = \begin{bmatrix} a + w & b & 0 & 0 \\ c & d & e & 0 \\ 0 & f & g & h + w \\ 0 & 0 & i & j \end{bmatrix}, \tag{4}$$

$$w^0 = adj - adhi - aefj - bcgj + bchi,$$

$$w^1 = adi + dgj - dhi - efj - bci, \tag{5}$$

$$w^2 = di.$$

2.5. Applying the Graph-Based Symbolic Technique. Our proposed GBST starts off with an SPICE-like netlist as input. The allowed circuit elements are R, C, L, V, I, E, G, and

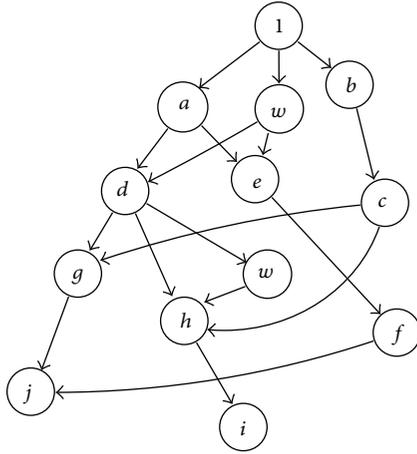


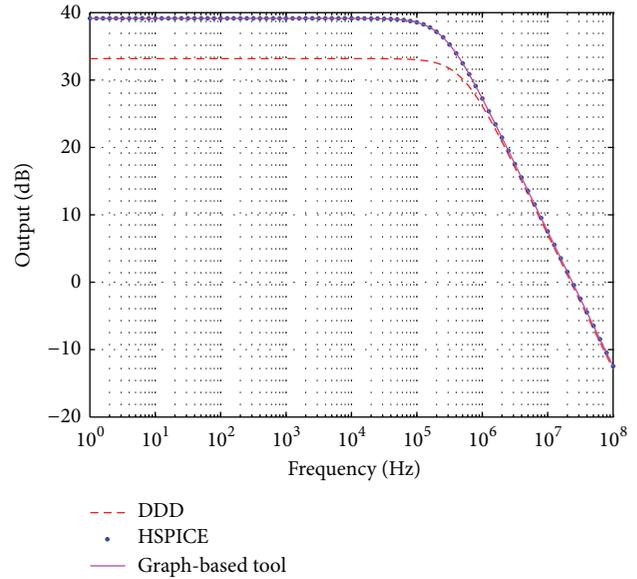
FIGURE 3: Expanded graph.

M, being resistor, capacitor, inductor, independent voltage source, independent current source, voltage-controlled voltage source, voltage-controlled current source, and MOSFET. When formulating the equations, the symbol name is taken exactly as specified in the netlist, for example, R_name, C_name, M_name, and so forth. Their corresponding small-signal model based on controlled sources substitutes all active elements. In turn, controlled sources are modeled with combinations of norator and nullator in order to make use of the extensive studies on analysis of Nullor based circuits given in [9, 11, 14, 15]. The guidelines for obtaining the nodal admittance matrix by applying nodal analysis are summarized in [11]. Afterwards, applying our proposed graph-based symbolic technique derives the solution. In this manner, the computing information on the solutions for the common source amplifier, differential pair amplifier, three stages uncompensated OTA [15], recycled folded cascode OTA [16], and 741 opamp is listed in Table 1. Using the values computed by HSPICE, in order to verify correctness, does the numerical evaluation of the derived symbolic expression.

To demonstrate the suitability of the new graph-based symbolic technique (GBST), we performed a comparison with HSPICE and the DDD symbolic tool [3], as shown in Figure 4. As one sees, GBST is in good agreement with the numerical response computed by SPICE, while the DDD technique has an error around 10%. That way, GBST is applied herein to derive the exact analytical expression $H(s)$ of CMOS amplifiers. The derived $H(s)$ is used to perform variational analysis, in order to compute the frequency response bounds (maximum and minimum) of the magnitude and phase from $H(s)$, subject to some ranges of process variational parameters by performing nonlinear constrained optimization.

3. Constrained Optimization

To highlight the appropriateness of applying the new GBST in variational analysis for finding the lower and upper performance bounds for the magnitude and phase of an amplifier, we formulate a nonlinear constrained optimization problem. That way, we start from a transfer function, whose coefficients

FIGURE 4: $H(s)$ comparison among DDD [3] (dashed), HSPICE (dots), and our graph-based symbolic technique (solid) for the differential pair topology.

(e.g., $a_i(p_1, \dots, p_m)$ and $b_j(p_1, \dots, p_n)$) are obtained by our proposed graph-based symbolic technique in an s -expanded form; that is,

$$H(\omega) = \frac{\sum_{i=0}^m a_i(p_1, \dots, p_m) s^i}{\sum_{j=0}^n b_j(p_1, \dots, p_n) s^j}. \quad (6)$$

Notice that $H(s, p_1, \dots, p_m)$ is a nonlinear function of p_1, \dots, p_m . Furthermore, each parameter p_i is a random variable with a variational range.

For $s = j\omega$, the nominal transfer function becomes

$$H(s) = H(j\omega) = H(j\omega) e^{j\theta(\omega)}. \quad (7)$$

From it, one can obtain a variational transfer function with bounded magnitude and phase regions, which is described by [3]

$$\begin{aligned} H_l(\omega) &\leq H(\omega) \leq H_u(\omega), \\ \theta_l(\omega) &\leq \theta(\omega) \leq \theta_u(\omega), \end{aligned} \quad (8)$$

where $H_l(\omega)$ and $H_u(\omega)$ are the lower and upper bounds of the magnitude, respectively, and $\theta_l(\omega)$ and $\theta_u(\omega)$ are the lower and upper bounds of the phase. The evaluation of (7) gives a complex valued result, where the magnitude $H(\omega) = |H(j\omega)|$ and the phase angle $\theta(\omega) = \angle H(j\omega)$ are real values. The goal is to find the bounds of the magnitude and phase for $H(j\omega)$, such that one can obtain (8). Henceforth, in the presence of process variations, the signal is perturbed from its nominal behavior, and it is usually bounded between its minimum and maximum limits, as sketched in Figure 5.

TABLE 1: Symbolic formulation and numerical evaluation of $D(s)$, $N(s)$, and $H(s)$ by applying the new GBST.

Circuit	Circuit features			Computer time (seconds)		
	Elements	Nodes	Equations	$D(s)$	$N(s)$	$H(s)$
Differential pair	35	26	1.1235	0.122	0.1464	1.4895
RFC OTA [16]	106	56	1.6603	0.201	0.1869	2.2633
LV Amp [15]	33	18	2.35	0.058	0.0464	2.4544
Common source	8	6	0.8581	0.041	0.0205	0.9811
741	112	77	0.5123	1.37	0.822	2.7043

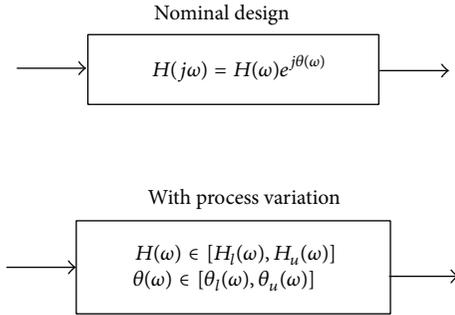


FIGURE 5: Transfer function description including parameter variations.

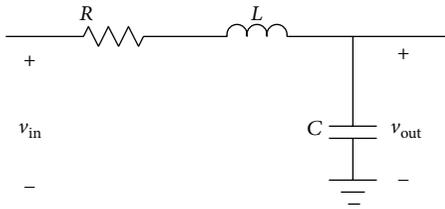


FIGURE 6: RLC circuit.

For instance, using the lower bound of the magnitude response $|H(j\omega)|$ at frequency ω , then the objective function is minimized and described by

$$\begin{aligned} & \text{minimize} && |H(j\omega, x)| \\ & \text{subject to} && x_l \leq x \leq x_u. \end{aligned} \quad (9)$$

In (9) $x = [p_1, \dots, p_m]$ represents the circuit parameters variable vector, subject to the optimization constraints $[x_l, x_u]$. In IC design foundries and cell library vendors supply the constraints. Algorithm 4 summarizes this process.

3.1. A Simple Example in MATLAB. Let us consider Figure 6. In the frequency domain, capacitors and inductors are analyzed as complex impedances, and then the transfer function is given by

$$H(j\omega) = \frac{V_{\text{out}}}{V_{\text{in}}} = \frac{1/j\omega C}{R + j\omega L + (1/j\omega C)}. \quad (10)$$

Assuming that C and L have variations of 20% from their nominal values $C = 1 \mu\text{F}$ and $L = 1 \mu\text{H}$, then $C \in$

$[0.8, 1.2] \mu\text{F}$ and $L \in [0.8, 1.2] \mu\text{H}$. For this example, the iterative method called active set was used. As a result, three snapshots at different frequency points are shown in Figure 7.

3.2. Line Search Algorithm. A line search algorithm searches for a decrease in f in a descent direction using the Armijo rule [17, 18] for stepsize control. The steplength is chosen to minimize f along the ray $\{x + \lambda\Delta x \mid \lambda \geq 0\}$, where Δx is called the *descent direction* and the point $x + \lambda\Delta x$ is called x_+ , or even to just reduce f “enough” [19]. That way, given the current point x_c and descent direction Δx , it looks for λ , such that

$$f(x_c + \lambda\Delta x) \leq f(x_c). \quad (11)$$

However, if the decreasing achieved by this inequality for some λ is too small, it is not possible to guarantee convergence to a local minimum. So in order to avoid this issue, λ must satisfy *Armijo rule (sufficient decrease)* given by

$$f(x_c + \lambda\Delta x) \leq f(x_c) + \lambda\gamma\nabla f(x_c)^T \Delta x, \quad (12)$$

where $\gamma \in (0, 1)$ [20]. This is shown in Figure 8.

In Figure 8, the interval where (12) is accomplished is $[0, \lambda_{\text{max}}]$. We can rewrite this by (13) where the Armijo condition [18] to accept a trial point x_+ is given by (14), and M is an integer greater than zero. If x_+ is rejected, the steplength is redefined by (15), where λ^* minimizes a quadratic model. This strategy of repeatedly testing sufficient decrease and reducing the stepsize if the test fails is called backtracking. The projected gradient (PGRAD) and spectral projected gradient (SPG) methods used herein for variational analysis of CMOS amplifiers are based on backtracking line search. Consider

$$f(x_+) \leq f(x_c) + \gamma\nabla f(x_c)^T (x_+ - x_c), \quad (13)$$

$$f(x_+) \leq \max_{0 \leq j \leq \min\{c, M-1\}} f(x_{c-j}) + \gamma\nabla f(x_c)^T (x_+ - x_c), \quad (14)$$

$$\lambda = \max\{\sigma_1\lambda, \min\{\sigma_2\lambda, \lambda^*\}\}. \quad (15)$$

3.3. Projected Gradient Method. The PGRAD method is described in [21]. It is the natural extension of the *steepest descent algorithm* [20], used in unconstrained optimization to bound constrained problems.

Given a current iterate x_c , the new iterate is

$$x_+ = P(x_c - \lambda\nabla f(x_c)), \quad (16)$$

- (1) Parse circuit netlist.
- (2) Set bounds on parameters affected by process variations.
- (3) Derive the symbolic expression, for example, the transfer function.
- (4) **for** All frequencies ω_i **do**
- (5) Constrained optimization to find magnitude bounds on ω_i .

ALGORITHM 4: Calculation of frequency response bounds using our graph-based symbolic technique from Section 2 and constrained optimization.

- (1) **if** $\|P(x_c - \nabla f(x_c)) - x_c\| = 0$, that is, x_c is stationary **then**
- (2) Stop
- (3) Compute $d_c = P(x_c - \alpha_c \nabla f(x_c)) - x_c$
- (4) Set $\lambda \leftarrow 1$
- (5) Set $x_+ = x_c + \lambda x_c$
- (6) **if** $f(x_+) \leq f(x_c) + \gamma \langle x_+ - x_c, \nabla f(x_c) \rangle$ **then**
- (7) $\lambda_c = \lambda$, $x_{c+1} = x_+$, $s_k = x_{c+1} - x_c$, $y_k = \nabla f(x_{c+1})$
- (8) Go to Step 12
- (9) **else**
- (10) Define $\lambda_{\text{new}} \in [\sigma_1 \lambda, \sigma_2 \lambda]$ and $\lambda \leftarrow \lambda_{\text{new}}$
- (11) Go to Step 5
- (12) Compute the trial step length $\alpha_{c+1} = \alpha_c \|f(x_c)\|^2 / \|f(x_{c+1})\|^2$

ALGORITHM 5: Projected gradient method.

where the gradient $\nabla f(x)$ is defined by

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \dots, \frac{\partial f(x)}{\partial x_n} \right) \quad (17)$$

for $x = (x_1, \dots, x_n)$.

In (17), λ is a steplength parameter given by the Armijo rule, where one searches on a ray from x_c in a direction where f is locally decreasing [18]. Besides, in order to implement a line search scheme, one must specify what sufficient decrease means. Therefore, for bound constrained problems the *sufficient decrease* condition for line search is defined by

$$f(x(\lambda)) - f(x) \leq \frac{-\alpha}{\lambda} \|x - x(\lambda)\|^2, \quad (18)$$

where

$$x(\lambda) = P(x - \lambda \nabla f(x)) \quad \text{for } \lambda \geq 0. \quad (19)$$

In this case $\lambda = \beta^m$, where $\beta \in (0, 1)$ and $m \geq 0$, which is the smallest nonnegative integer such that there is *sufficient decrease* in f . For the end condition, it is necessary to define the active and inactive sets. The set of constraints is called *feasible set* (Ω), and a point in this set is a *feasible point*. Because the *feasible set* is compact, there is always a solution for this minimization problem. The i th constraint is *active* at $x \in \Omega$ if either $x_i = L_i$ or $x_i = U_i$. If the i th constraint is not *active*, it is called *inactive*. Therefore, one can write $A(x)$ and $I(x)$ for the active and inactive sets, respectively. An *active/inactive set* is the set of indexes i such that the i th constraint is active/inactive. In this manner,

$$r_0 = \|x_0 - x_0(1)\|. \quad (20)$$

The stop criterion is given by (21), where τ_r and τ_a are relative and absolute tolerances. Consider

$$\|x - x(1)\| \leq \tau_a + \tau_r r_0. \quad (21)$$

This optimization method is shown in Algorithm 5. It starts with $x_0 \in \Omega$ and uses a sufficient decrease parameter $\gamma \in (0, 1)$ and safeguarding parameters $0 \leq \sigma_1 \leq \sigma_2 \leq 1$. Initially, $\alpha_0 = 1/\|f(x_0)\|^2$. Given x_c and $\alpha_c \geq 0$ the algorithm shows how to obtain x_{c+1} and α_{c+1} and when to stop the optimization engine.

3.4. Spectral Projected Gradient Method. The SPG method improves choosing the steplength, which greatly speeds up the convergence of gradient methods [22]. Unlike the projected gradient method, the spectral projected gradient one [21] is more related to the quasi-Newton family [17]. The main idea behind the spectral choice of steplength is that the steepest descent method is very slow but it can be accelerated taking, instead of the stepsize that comes from the minimization of the function along the gradient of the current iteration, the one that comes from the one-dimensional minimization at the previous step.

The point in the first iteration of this method should be a feasible point; that is, the algorithm starts with $x_0 \in \Omega$ and uses an integer $M \geq 1$, a small parameter $\alpha_{\min} \geq 0$, a large parameter $\alpha_{\max} \geq \alpha_{\min}$, a sufficient decrease parameter $\gamma \in (0, 1)$, and safeguarding parameters $0 \leq \sigma_1 \leq \sigma_2 \leq 1$. Consider $\|P(x_c - \nabla f(x_c)) - x_c\| = 0$.

```

(1) if  $\|P(x_c - \nabla f(x_c)) - x_c\| = 0$ , that is,  $x_c$  is stationary then
(2) Stop
(3) Compute  $d_c = P(x_c - \alpha_c g(x_c)) - x_c$ .
(4) Set  $\lambda \leftarrow 1$ 
(5) Set  $x_+ = x_c + \lambda \Delta f(x_c)$ 
(6) if  $f(x_+) \leq \max_{0 \leq j \leq \min\{c, M-1\}} f(x_{c-j}) + \gamma \lambda \langle d_c, \nabla f(x_c) \rangle$  then
(7) Define  $\lambda_c = \lambda$ ,  $x_{c+1} = x_+$ ,  $s_c = x_{c+1} - x_c$ ,  $y_c = \nabla f(x_{c+1}) - \nabla f(x_c)$ 
(8) Got to Step 11
(9) else
(10) Set  $\lambda_{\text{new}} \in [\sigma_1 \lambda, \sigma_2 \lambda]$ ,  $\lambda \leftarrow \lambda_{\text{new}}$  and go to Step 5
(11) Compute  $b_c = \langle s_c, y_c \rangle$ .
(12) if  $b_k \leq 0$  then
(13)  $\alpha_{c+1} = \alpha_{\text{max}}$ .
(14) else
(15)  $\alpha_c = \langle s_c, s_c \rangle$  and  $\alpha_{c+1} = \min\{\alpha_{\text{max}}, \max\{\alpha_{\text{min}}, \alpha_c/b_c\}\}$ 
    
```

ALGORITHM 6: Spectral projected gradient method.

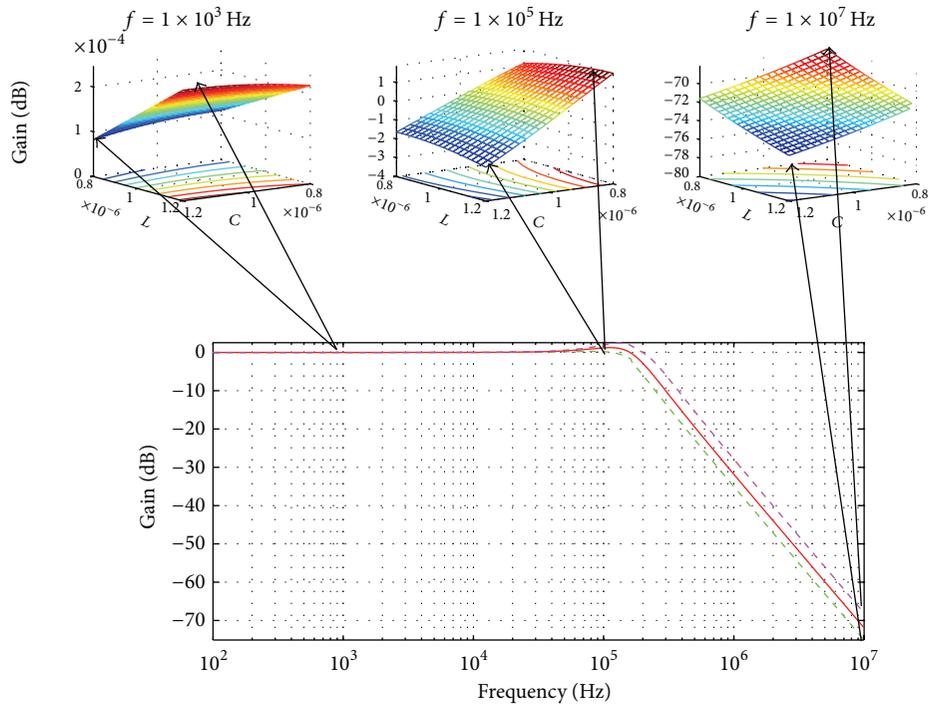


FIGURE 7: Frequency response of the RLC circuit [3]. Solid curve is the magnitude response with nominal parameters. The dashed curves are the lower and upper bounds due to variations. The three surfaces at the top have L and C as x -axis and y -axis, respectively, and z -axis shows the magnitude variations at three sampling frequencies.

The method is shown in Algorithm 6, where λ_{new} uses one-dimensional quadratic interpolation and it is safeguarded taking $\lambda \leftarrow \lambda/2$ when the minimum of the one-dimensional quadratic lies outside $[0.1, 0.9\lambda]$. The line search condition in step 6 guarantees that the sequence $\{x_c\}$ remains in $\Omega_0 = \{x \in \Omega : f(x) \leq f(x_0)\}$.

3.5. Remarks. The projected gradient and spectral projected gradient methods start at $x_0 \in \Omega$ and use as search direction the internal projected gradient direction. In case of rejection of the first trial point, the next ones are computed along

the same line. Also, for both methods, the calculation of λ_{new} uses a one-dimensional quadratic interpolation. Both algorithms involve a projection on the convex set Ω , a function evaluation $f(x)$, and a gradient evaluation per iteration $\nabla f(x)$.

4. Simulation Results

The analog ICs for testing the variational methods PGRAD and SPG that use $H(s)$ derived by the new GBST are shown in Figure 9. That way, after deriving the transfer function

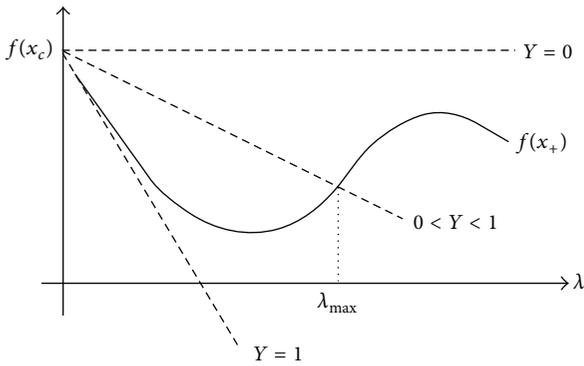


FIGURE 8: Line search representation.

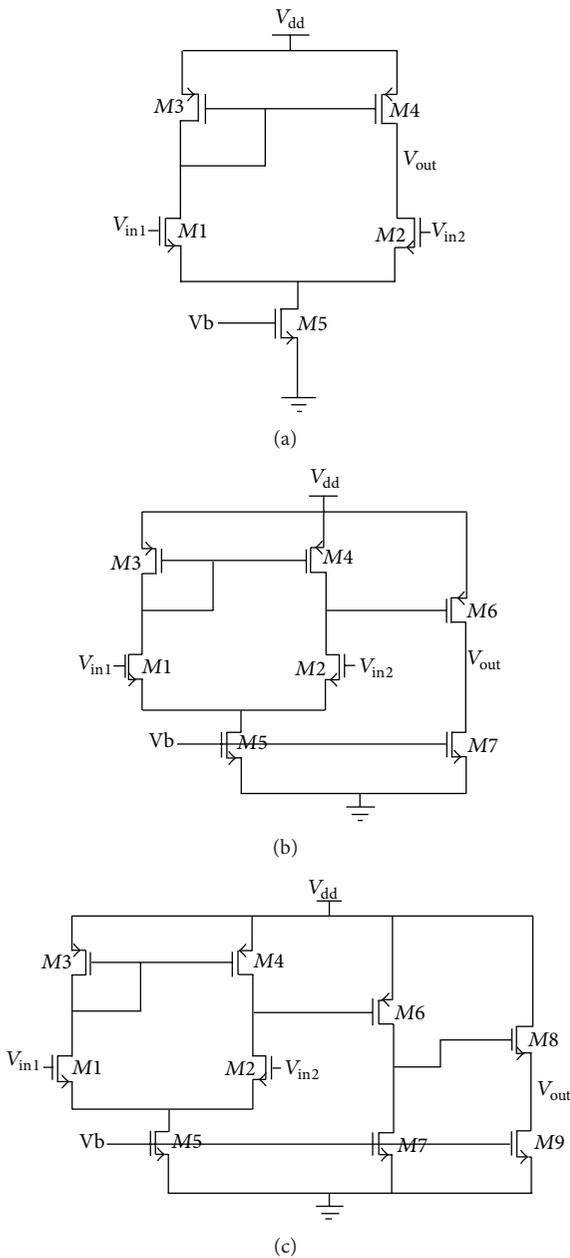
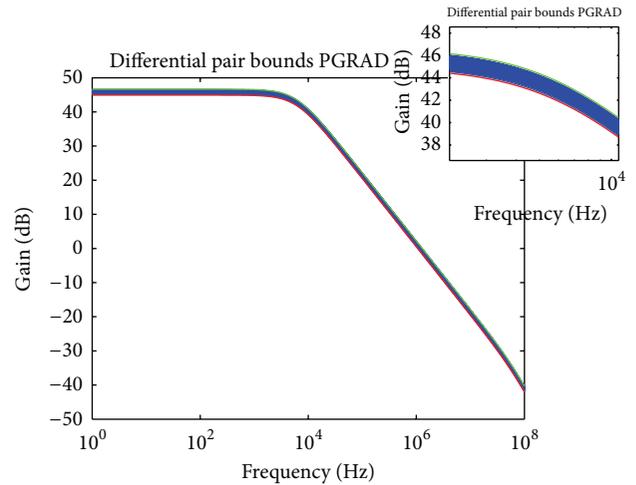
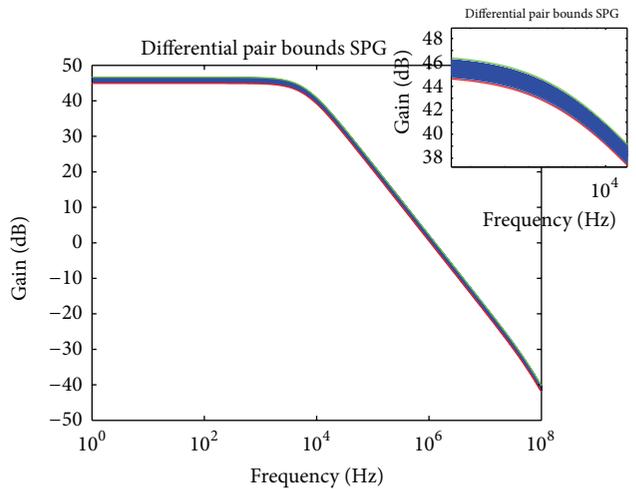


FIGURE 9: Circuits used for the variational analysis: (a) differential pair, (b) 2-stage OTA, and (c) 3-stage OTA.



— Lower bound
— Upper bound

(a)



— Lower bound
— Upper bound

(b)

FIGURE 10: Bounds for the differential pair applying (a) PGRAD and (b) SPG.

by applying the new GBST, and after setting the desired frequencies, the PGRAD and SPG methods are applied to find the magnitude and phase bounds for $H(s)$.

Both the PGRAD and SPG methods were programmed in C and compiled in an Ubuntu Linux environment with the GNU C compiler gcc-4.6.1, with 4 GB RAM in an Intel Core i3. The performance bounds results for the differential pair are shown in Figure 10 and for the 3-stage OTA in Figure 11. The lines in blue are those corresponding to Monte Carlo simulations, which are well bounded by the bounds computed by the variational methods PGRAD and SPG. Table 2 summarizes the results where it can be appreciated that both variational methods show better times than by performing repeated Monte Carlo simulations when using HSPICE.

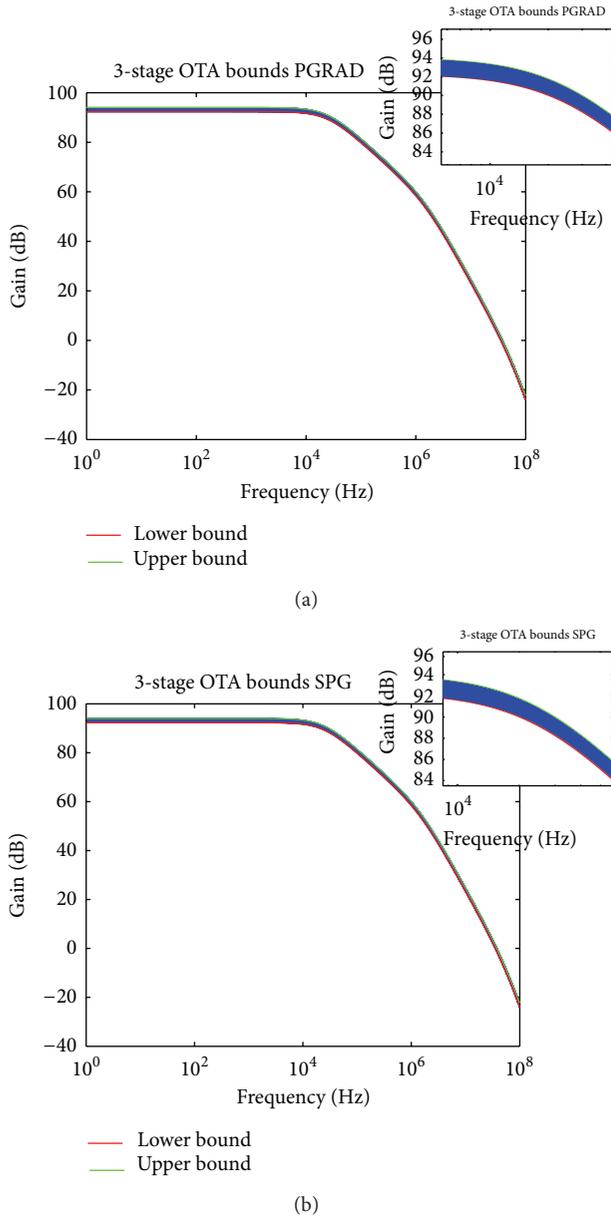


FIGURE 11: Bounds for the 3-stage OTA applying (a) PGRAD and (b) SPG.

TABLE 2: Time comparison between the variational methods and HSPICE.

Circuit	PGRAD	SPG	HSPICE
Differential pair	15.312 ms	16.363 ms	1.716 s
2-stage OTA	44.559 ms	91.936 ms	81.65 ms
3-stage OTA	38.315 ms	39.233 ms	2.163 s

5. Conclusions

A new graph-based symbolic technique (GBST) for deriving exact analytical expressions for analog ICs, such as $H(s)$, and whose symbolic expressions are used to perform variational analysis to obtain performance frequency bounds has been

introduced. The variational analysis needs as input the $H(s)$ computed by applying our proposed GBST, and it was based on nonlinear constrained optimization using two line search methods, namely, projected gradient (PGRAD) and spectral projected gradient (SPG).

It was demonstrated that the bound analysis based on GBST is suitable for analog ICs. In addition, the time computation of PGRAD and SPG for obtaining the frequency response bounds of analog ICs was improved with respect to HSPICE, when performing repeated Monte Carlo simulations. As shown in the last section, computing the frequency response bounds by using GBST showed a gain of 100x for the differential pair and 50x for the 3-stage amplifier, compared to repeated Monte Carlo simulations, thus justifying the usefulness of using GBST.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

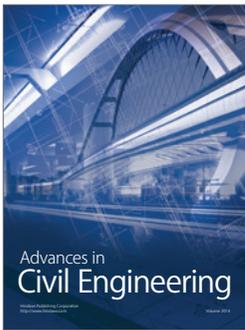
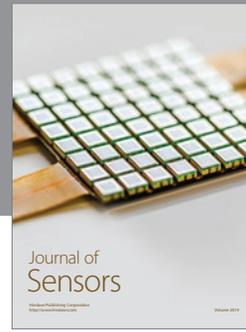
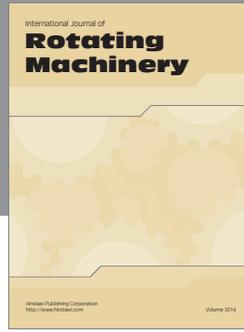
Acknowledgments

This project is partially supported by CONACyT and UC-MEXUS-CONACYT under projects 131839-Y and CN-11-575.

References

- [1] T. McConaghy, K. Breen, J. Dyck, and A. Gupta, *Variation-Aware Design of Custom Integrated Circuits: A Hands-on Field Guide: A Hands-on Field Guide*, Springer, New York, NY, USA, 2013.
- [2] M. Webba and H. Tangb, "Hierarchical statistical analysis of complex analog and mixed-signal systems," *International Journal of Electronics*, 2014.
- [3] X.-X. Liu, S. X.-D. Tan, A. Palma-Rodriguez, E. Tlelo-Cuautle, and G. Shi, "Performance bound analysis of analog circuits in frequency- and time-domain considering process variations," *ACM Transactions on Design Automation of Electronic Systems*, vol. 19, no. 1, article 6, 2013.
- [4] P.-Y. Kuo, S. Saibua, G. Huang, and D. Zhou, "An efficient method for evaluating analog circuit performance bounds under process variations," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 6, pp. 351-355, 2012.
- [5] H. Filiol, I. O'Connor, and D. Morche, "Analog IC variability bound estimation using the cornish-fisher expansion," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 9, pp. 1457-1461, 2012.
- [6] L. V. Kolev, V. M. Mladenov, and S. Vladov, "Interval mathematics algorithms for tolerance analysis," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 8, pp. 967-975, 1988.
- [7] L. Qian, D. Zhou, S. G. Wang, and X. Zeng, "Worst case analysis of linear analog circuit performance based on Kharitonov's rectangle," in *Proceedings of the 10th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT '10)*, pp. 800-802, Shanghai, China, November 2010.
- [8] W. Verhaegen and G. G. E. Gielen, "Symbolic determinant decision diagrams and their use for symbolic modeling of linear

- analog integrated circuits,” *Analog Integrated Circuits and Signal Processing*, vol. 31, no. 2, pp. 119–130, 2002.
- [9] G. Shi, S. X.-D. Tan, and E. Tlelo-Cuautle, *Advanced Symbolic Analysis for VLSI Systems-Methods and Applications*, Springer, New York, NY, USA, 2014.
- [10] Z. Hao, G. Shi, S. X.-D. Tan, and E. Tlelo-Cuautle, “Symbolic moment computation for statistical analysis of large interconnect networks,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 5, pp. 944–957, 2013.
- [11] C. Sanchez-Lopez, F. V. Fernandez, E. Tlelo-Cuautle, and S. X. Tan, “Pathological element-based active device models and their application to symbolic analysis,” *IEEE Transactions on Circuits and Systems. I. Regular Papers*, vol. 58, no. 6, pp. 1382–1395, 2011.
- [12] M. Pierzchala and M. Fakhfakh, “Symbolic analysis of Nullor-based circuits with the two-graph technique,” *Circuits, Systems, and Signal Processing*, vol. 33, no. 4, pp. 1053–1066, 2014.
- [13] W. Tangsrirat, “SFG synthesis of general high-order all-pass and all-pole current transfer functions using CFTAs,” *The Scientific World Journal*, vol. 2014, Article ID 271926, 8 pages, 2014.
- [14] C. Sanchez-Lopez, “Pathological equivalents of fully-differential active devices for symbolic nodal analysis,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 3, pp. 603–615, 2013.
- [15] S. Rodriguez-Chavez, A. A. Palma-Rodriguez, E. Tlelo-Cuautle, and S. X.-D. Tan, “Graph-based symbolic and symbolic sensitivity analysis of analog integrated circuits,” in *Analog/RF and Mixed-Signal Circuit Systematic Design*, vol. 233 of *Lecture Notes in Electrical Engineering*, pp. 101–122, Springer, 2013.
- [16] R. S. Assaad and J. Silva-Martinez, “The recycling folded cascode: a general enhancement of the folded cascode amplifier,” *IEEE Journal of Solid-State Circuits*, vol. 44, no. 9, pp. 2535–2542, 2009.
- [17] J. F. Bonnans, J. C. Gilbert, C. Lemarechal, and C. A. Sagastizabal, *Numerical Optimization, Theoretical and Practical Aspects*, Springer, New York, NY, USA, 2000.
- [18] L. Armijo, “Minimization of functions having lipschitz continuous first partial derivatives,” *Pacific Journal of Mathematics*, vol. 16, no. 1, pp. 1–3, 1966.
- [19] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2009.
- [20] C. T. Kelley, *Iterative Methods for Optimization*, Frontiers in Applied Mathematics, SIAM, 1999.
- [21] L. Bello and M. Raydan, “Convex constrained optimization for the seismic reflection tomography problem,” *Journal of Applied Geophysics*, vol. 62, no. 2, pp. 158–166, 2007.
- [22] Y. H. Xiao, Q. J. Hu, and Z. W. Wei, “Modified active set projected spectral gradient method for bound constrained optimization,” *Applied Mathematical Modelling*, vol. 35, no. 7, pp. 3117–3127, 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

