

Robust Analog/RF Circuit Design With Projection-Based Performance Modeling

Xin Li, *Member, IEEE*, Padmini Gopalakrishnan, *Student Member, IEEE*,
Yang Xu, *Member, IEEE*, and Lawrence T. Pileggi, *Fellow, IEEE*

Abstract—In this paper, a robust analog design (ROAD) tool for posttuning (i.e., locally optimizing) analog/RF circuits is proposed. Starting from an initial design derived from hand analysis or analog circuit optimization based on simplified models, ROAD extracts accurate performance models via transistor-level simulation and iteratively improves the circuit performance by a sequence of geometric programming steps. Importantly, ROAD sets up all design constraints to include large-scale process and environmental variations, thereby facilitating the tradeoff between yield and performance. A crucial component of ROAD is a novel projection-based scheme for quadratic (both polynomial and posynomial) performance modeling, which allows our approach to scale well to large problem sizes. A key feature of this projection-based scheme is a new implicit power iteration algorithm to find the optimal projection space and extract the unknown model coefficients with robust convergence. The efficacy of ROAD is demonstrated on several circuit examples.

Index Terms—Analog/RF circuit, circuit optimization, performance model, robust design.

I. INTRODUCTION

AS IC technologies are scaled to finer feature sizes and circuit applications move to higher frequency bands, analog/RF circuit design faces several new challenges. First, device models have become increasingly complex in order to characterize the physical behavior of nanoscale transistors at high frequencies. Second, at these frequencies, parasitic couplings become more important and more complex. Finally, but perhaps most importantly, with subwavelength photolithography, process variations become a critical issue and significantly impact the overall circuit performance. It is complex, if not impossible, to handle all these second-order effects via hand analyses. Therefore, manually designing analog/RF circuits is time consuming and requires a lot of design intuition and experience. Today's analog/RF circuits are typically designed and verified through several iterations [1].

Manuscript received November 22, 2005; revised March 10, 2006. This work was supported in part by the Microelectronics Advanced Research Corporation Focus Center for Circuit and System Solutions under Contract 2003-CT-888. This paper was presented in part at the IEEE/ACM International Conference on Computer Aided Design, San Jose, CA, 2004. This paper was recommended by Associate Editor C.-J. R. Shi.

X. Li, P. Gopalakrishnan, and L. T. Pileggi are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: xinli@ece.cmu.edu; pgopalak@ece.cmu.edu; pileggi@ece.cmu.edu).

Y. Xu was with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA. He is now with Qualcomm Inc., San Diego, CA 92122 USA (e-mail: yangxu@qualcomm.com).

Digital Object Identifier 10.1109/TCAD.2006.882513

To address this increasing difficulty of manual design, various approaches have been proposed for parametric analog optimization [1]–[10]. These methods take a fixed circuit topology as input and optimize the device sizes to meet design specifications. For example, DELIGHT.SPICE [2] is an optimization engine that utilizes a gradient-based algorithm to locally optimize the circuit. Advanced stochastic search algorithms such as simulated annealing and genetic programming have also been applied to search the entire design space for a globally optimal solution [3]–[7]. Compared with gradient-based approaches, these stochastic search algorithms typically yield better optimization results at the expense of increased computational cost.

Recent research [8]–[10] has demonstrated that many analog circuit specifications can be cast as posynomial functions. As such, analog circuit sizing can be formulated as a geometric programming problem that can be solved extremely efficiently [11]. However, the traditional geometric programming approach requires the creation of posynomial design equations by hand. Manually derived equations apply various simplifications and ignore many second-order effects that can be important at high operation frequencies and/or in nanoscale technologies.

In addition to modeling inaccuracies, process and environmental variations have an increasingly significant impact on circuit performance, thereby posing additional challenges for analog optimization. For example, given a circuit that is optimized for nominal process and environmental conditions, substantial device resizing may be required to accommodate large-scale process and environmental variations and improve the product yield. Corner enumeration can be used to achieve a robust design [8]–[10], where the analog circuit is analyzed and/or simulated at all process and environmental corners during sizing. However, this approach suffers from the problem that the total number of simulations increases exponentially with the number of independent process/environmental parameters. Furthermore, it is not guaranteed that the worst case design will occur at one of these corners.

During the past two decades, many statistical optimization approaches have been proposed to statistically handle process and environmental variations for analog sizing [12]–[19]. These techniques can be classified into four categories in general, namely: 1) *direct yield optimization* [12]–[14]; 2) *design centering* [15], [16]; 3) *worst case optimization* [17], [18]; and 4) *infinite programming* [19]. Direct yield optimization methods maximize the parametric yield that is estimated by either numerical integration or Monte Carlo analysis. Design centering approaches attempt to find the optimal design that is

farthest away from all constraint boundaries and is therefore least sensitive to process and environmental variations. Worst case optimization techniques optimize worst case circuit performance, instead of nominal performance, over all process and environmental variations. Finally, infinite programming methods formulate the circuit optimization problem as a nonlinear infinite program where the cost function and constraints are defined over an infinite set covering all process and environmental variations.

In this paper, we propose a new approach to posttune (i.e., locally optimize) analog/RF circuits based on accurate transistor-level simulations and with consideration of large-scale process and environmental variations. Our approach is called robust analog design (ROAD). Using ROAD, a robust analog/RF design can be achieved via two steps. First, an initial nominal design is created from either manual analysis or automatic optimization by traditional algorithms [1]–[10]. Simplified device/coupling models could be utilized in this step to ease the manual design or speed up the automatic optimization. This initial optimization provides a rapid but coarse search over the entire design space. Then, in the second step, ROAD is applied with detailed device/coupling/variation models to perform a more fine-grained search and optimizes the worst case circuit performance, considering all process and environmental variations. From this point of view, ROAD is a new worst case optimization tool that is in the same category as that in [17] and [18].

Given an initial design, ROAD first applies quadratic response surface (i.e., polynomial) modeling and statistical analysis to extract the probability distributions of all circuit performances. Next, the worst case circuit performances are accurately approximated as posynomial functions in the local design space. Unlike the existing geometric programming approaches in [8]–[10] where posynomial performance models are derived from hand analysis equations, our posynomial models are fitted using transistor-level simulation data. Finally, ROAD utilizes these preextracted posynomial models and optimizes the circuit by geometric programming. This fitting and optimization procedure is repeatedly applied to successively narrowed local design spaces. Since ROAD sets up all design constraints with process and environmental variations and the posynomial model becomes increasingly accurate in the successively narrowed local design space, ROAD is able to produce an accurate design with high parametric yield.

Compared with other statistical optimization approaches [12]–[19], the novelty of ROAD lies in a projection-based approach for quadratic performance (both polynomial and posynomial) modeling. Instead of fitting a full-rank quadratic model, as is currently done in [20]–[25], ROAD applies a projection operator with the goal of obtaining an optimal low-rank model by minimizing the approximation error. With this novel projection scheme, ROAD is able to dramatically reduce the number of unknown model coefficients and the number of simulation samples, thereby significantly reducing the modeling cost and facilitating scaling to much larger problem sizes than what are feasible with current modeling techniques.

In addition, while the traditional projection theory generally trades accuracy for simplicity in terms of the dimension of the

projection space [26], [27], we find that the rank-one projection is especially meaningful in our application. Our theoretical analysis proves that a quadratic polynomial/posynomial is invariant (i.e., remains a polynomial/posynomial) under the rank-one approximation. This allows us to use a one-dimensional projection space to build models of these functions efficiently. Furthermore, our numerical experiments show that the rank-one projection allows us to approximate many analog performances with sufficient accuracy to enable a good local optimization.

In addition to the concept of using subspace projection for quadratic polynomial/posynomial modeling, another important contribution of our work is the algorithm that is used to obtain this projection. As part of ROAD, we have formulated a novel implicit power iteration algorithm that is used to determine the optimal projection space and extract the unknown model coefficients. Essentially, this algorithm solves a sequence of overdetermined linear equations or convex quadratic programming problems and exhibits robust convergence. Using this implicit power iteration technique, ROAD is able to achieve significant speedup for generating rank-one quadratic performance (both polynomial and posynomial) models. As demonstrated by the numerical examples in Section IV, ROAD is able to extract accurate models and reduce the computational cost by up to $5.5\times$ when compared with the traditional full-rank quadratic modeling.

The remainder of this paper is organized as follows: In Section II, we briefly review the background materials in the areas of parametric analog optimization and performance modeling. Then, we propose our ROAD approach, including response surface modeling, statistical analysis, posynomial fitting, and geometric programming in Section III. The efficacy of ROAD is demonstrated by several circuit examples designed in the IBM BiCMOS $0.25\ \mu\text{m}$ or Taiwan Semiconductor Manufacturing Company Ltd. (TSMC) CMOS $0.13\ \mu\text{m}$ process in Section IV. Finally, we draw conclusions in Section V.

II. BACKGROUND

A. Feasible Region in Analog Design Space

It is well known that the entire analog/RF design space is strongly nonlinear and contains many local minima that prevent local optimization from converging to the globally optimal point. The strong nonlinearity of the analog design space makes analog optimization extremely difficult and motivates the application of stochastic search algorithms, e.g., simulated annealing and genetic programming, to find a good optimization solution [3]–[7]. It is theoretically guaranteed that stochastic search algorithms can find the global optimum after they visit a sufficiently large number of design points in the design space.

Recently, Graeb *et al.* [28] and Stehr *et al.* [29] proposed the concept of feasible region that is defined by a set of implicit topology-given specifications—e.g., MOSFETs should stay in the saturation region in order to provide the correct analog functionality. As shown in Fig. 1, the feasible region is a small subset of the entire analog design space. It is observed that in many applications, analog design space is weakly nonlinear in the feasible region [28]. Therefore, starting from an

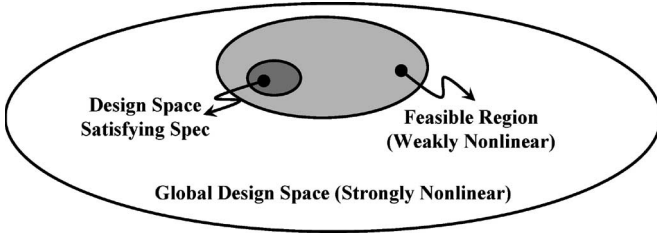


Fig. 1. Feasible region in analog design space.

initial design in the feasible region, it is likely that a good final design can be achieved even by using local optimization algorithms.

Due to this observation, we utilize a two-step approach for parametric analog optimization, which is similar to that used in [29]. In the first step, an initial global search is applied to explore the entire design space and get close to a good initial solution in the feasible region. The initial global search can be done by manual analysis, traditional analog optimization algorithms [1]–[10], or the initial sizing algorithm proposed in [29]. Simplified device/coupling models can be utilized in this step to ease the manual design or speed up the automatic optimization. Then, in the second step, ROAD is used to perform a more fine-grained local posttuning with detailed device/coupling/variation models to improve the design accuracy and the parametric yield.

Since ROAD is a local optimization tool, the initial starting point can have a significant impact on the quality of the optimized design. For example, if a bad initial design (e.g., out of the feasible region) is used, ROAD can even fail to converge. In this paper, we do not attempt to make the optimization result independent of the initial starting point. Instead, we focus on our formulation and evaluation of a novel projection-based performance modeling technique to reduce the computational cost and facilitate the application of posttuning to large problem sizes.

B. Posynomial Function and Geometric Programming

Let $X = [x_1, x_2, \dots, x_M]^T$ be M real and positive design variables (e.g., bias current, transistor sizes). A function f is called posynomial if it has the following form:

$$f(X) = \sum_i c_i x_1^{\alpha_{1i}} x_2^{\alpha_{2i}} \dots x_M^{\alpha_{Mi}} \quad (1)$$

where $c_i \in R_+$, and $\alpha_{ij} \in R$. Note that the coefficients c_i must be nonnegative, but the exponents α_{ij} can be real values.

Several research works have demonstrated that many analog circuit specifications (e.g., gain, bandwidth) can be cast into posynomial functions [8]–[10]. As such, analog circuit sizing can be formulated as a geometric programming problem as follows:

$$\begin{aligned} & \text{minimize} && f_0(X) \\ & \text{subject to} && f_i(X) \leq 1, \quad i = 1, \dots, K \\ & && x_j > 0, \quad j = 1, \dots, M \end{aligned} \quad (2)$$

where f_0, f_1, \dots, f_K are normalized circuit performance matrices, and they are approximated as posynomial functions. The geometric programming problem in (2) attempts to find the optimal value of X that yields minimal f_0 while satisfying all other constraints $\{f_i(X) \leq 1, i = 1, 2, \dots, K\}$ and $\{x_j > 0, j = 1, 2, \dots, M\}$. The optimization in (2) can be converted into a convex programming problem and solved in an extremely efficient way [11].

Geometric programming has been previously applied to parametric analog optimization in [8]–[10]. However, these approaches optimize analog/RF circuits using simplified models, which is similar to using hand analysis equations. In contrast, in ROAD, we use the solution derived from the optimization based on such simplified equations as a starting point. Our objective is to further tune the design for robust performance using detailed device/variation modeling information.

ROAD utilizes posynomial performance models and geometric programming for local optimization. However, linear and/or quadratic polynomial performance models can also be used, where the optimization problem should be solved by linear programming and/or quadratic programming. The detailed comparison between geometric programming, linear programming, and quadratic programming is beyond the scope of this paper and is, therefore, not included.

C. Linear and Quadratic Performance Modeling

Given a circuit topology, the circuit performance (e.g., gain, bandwidth) is a function of the design variables (e.g., bias current, transistor sizes), as well as the process and environmental parameters (e.g., V_{TH} , temperature). If the variations on all design, process, and environmental parameters are sufficiently small, the circuit performance f can be approximated as a linear response surface (i.e., polynomial) model [20], [21], i.e.,

$$f(Y) = B^T Y + C \quad (3)$$

where $Y = [y_1, y_2, \dots, y_N]^T$ contains the design, process, and environmental parameters, $B \in R^N$ and $C \in R$ stand for the model coefficients, and N is the total number of the variational parameters.

The linear model in (3) is not sufficiently accurate for modeling large-scale variations. It, in turn, suggests that applying quadratic response surface (i.e., polynomial) model is required to improve the modeling accuracy [20], [21], i.e.,

$$f(Y) = Y^T A Y + B^T Y + C \quad (4)$$

where $C \in R$ is the constant term, $B \in R^N$ represents the linear coefficients, and $A \in R^{N \times N}$ denotes the quadratic coefficients. The unknown model coefficients A , B , and C can be determined by solving the overdetermined linear equation [20], [21] described as follows:

$$Y_i^T A Y_i + B^T Y_i + C = \tilde{f}_i, \quad i = 1, 2, \dots, S \quad (5)$$

where Y_i and \tilde{f}_i are the value of Y and the exact value of the performance f for the i th simulated sample, respectively, and S is the total number of the sampling points.

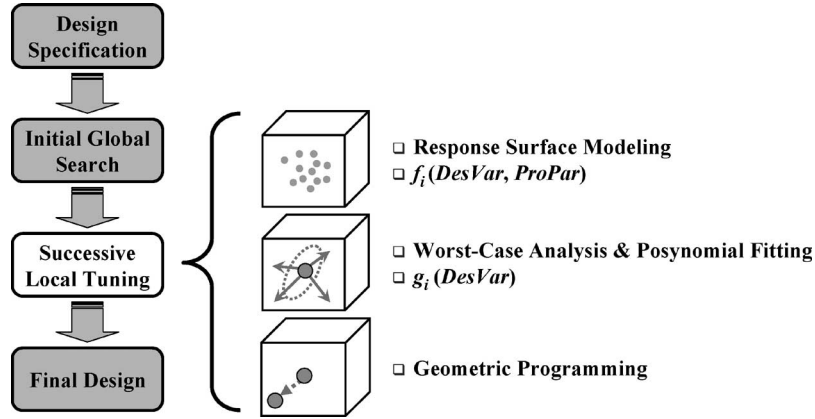


Fig. 2. ROAD optimization flow.

Another important category of performance model is the posynomial function shown in (1). Several previous techniques have been proposed for quadratic posynomial model generation [22]–[25]. Considering the tradeoff between modeling accuracy and computational cost, direct fitting [24] is one of the most efficient approaches. Direct fitting approximates a posynomial function by the following quadratic form:

$$f(X) = \tilde{X}^T \tilde{A} \tilde{X} + \tilde{B}^T \tilde{X} + \tilde{C} \quad (6)$$

where $X = [x_1, x_2, \dots, x_M]^T$ contains M design variables, $\tilde{X} = [x_1^{-1}, x_2^{-1}, \dots, x_M^{-1}, x_1, x_2, \dots, x_M]^T$ is a column vector, and $\tilde{A} \in R_+^{2M \times 2M}$, $\tilde{B} \in R_+^{2M}$, and $\tilde{C} \in R_+$ are the unknown posynomial coefficients that can be determined by the following optimization [24]:

$$\begin{aligned} \text{minimize} \quad & \psi(\tilde{A}, \tilde{B}, \tilde{C}) = \sum_{i=1}^T \left(\tilde{X}_i^T \tilde{A} \tilde{X}_i + \tilde{B}^T \tilde{X}_i + \tilde{C} - \tilde{f}_i \right)^2 \\ \text{subject to} \quad & \tilde{A} \in R_+^{2M \times 2M}, \tilde{B} \in R_+^{2M}, \text{ and } \tilde{C} \in R_+ \end{aligned} \quad (7)$$

where \tilde{X}_i and \tilde{f}_i are the value of \tilde{X} and the exact value of the function f for the i th simulated sample, respectively, and T is the total number of the sampling points. Note that unlike the quadratic polynomial fitting, the coefficient matrices \tilde{A} , \tilde{B} , and \tilde{C} in (7) must be nonnegative so that the approximated function in (6) is a posynomial. The cost function in (7) is a positive semidefinite quadratic function restricted to a convex constraint set [24]. Therefore, the optimization problem in (7) is convex and is guaranteed to find a globally optimal solution.

In addition, the authors in [24] proposed a template estimation technique to reduce the posynomial modeling cost. The template estimation algorithm first fits the sampling points to a quadratic polynomial function. Then, in the second step, dominant posynomial terms are selected based on the polynomial coefficients, and only these dominant terms are put into the cost function in (7) for final optimization. As such, the computational cost can be significantly reduced.

It is straightforward to verify that the numbers of the unknown coefficients in (5) and (7) are $O(N^2)$ and $O(M^2)$, respectively. The overall computational cost for determining all these coefficients consists of two portions.

- 1) *Simulation cost*: the cost for running a simulator to determine the performance value \tilde{f}_i at each sampling point. The number of simulation samples should be no less than the number of unknown coefficients, in order to uniquely solve the linear equation in (5) or the convex optimization in (7). Therefore, at least $O(N^2)$ and $O(M^2)$ sampling points are required for fitting the quadratic polynomial model in (4) and the quadratic posynomial model in (6), respectively. In practical applications, the number of simulation samples is generally selected to be significantly larger than the unknown coefficient number to avoid overfitting. Typically, the simulation cost is the dominant portion of the overall computational cost.
- 2) *Fitting cost*: the cost for solving the overdetermined linear equation in (5) or the convex optimization in (7). The fitting costs are of the order of $O(N^6)$ and $O(M^6)$ for the quadratic polynomial model in (4) and the quadratic posynomial model in (6), respectively.

The aforementioned high computational cost limits the traditional quadratic polynomial and posynomial modeling approaches to small- or medium-size applications. This observation, therefore, motivates us to propose a novel projection-based performance modeling algorithm in ROAD, which can significantly reduce both simulation cost and fitting cost.

III. ROAD METHODOLOGY

Our proposed ROAD tool uses a combination of response surface modeling, statistical analysis, posynomial fitting, and geometric programming. Fig. 2 outlines the ROAD optimization flow that consists of three individual steps.

- Step 1) Run transistor-level simulation and fit the quadratic response surface (i.e., polynomial) models $f_i(DesVar, ProVar)$ for all circuit performances in the local design space, where f_i stands for the i th circuit performance, $DesVar$ contains the design variables, and $ProVar$ contains the process and environmental parameters.
- Step 2) Based on the response surface models $f_i(DesVar, ProVar)$, apply statistical analysis to process and environmental variations and extract the probability distribution of the circuit performances

f_i . Fit the worst case performances $g_i(DesVar)$ as the posynomial functions of the design variables $DesVar$.

Step 3) Using these preextracted posynomial models, optimize the circuit by geometric programming.

The fitting and optimization procedure in Fig. 2 is repeatedly applied to successively narrowed local design spaces during the ROAD iterations. Since ROAD sets up all design constraints with process and environmental variations and the performance model becomes increasingly accurate in the successively narrowed local design space, ROAD can converge to an accurate design with high parametric yield. In this section, we will describe the key algorithms used in ROAD and highlight their novelties.

A. Projection-Based Response Surface Modeling

As shown in Fig. 2, the first step in ROAD is to extract the quadratic response surface (i.e., polynomial) models based on transistor-level simulation. Although high-order polynomial functions can be utilized to improve the modeling accuracy, ROAD uses quadratic models because they are easy to fit and facilitate an efficient probability extraction using the asymptotic probability extraction (APEX) algorithm [30]. Our experimental evaluation also shows that the accuracy provided by quadratic models is sufficient to converge to a good design solution.

The key disadvantage of the traditional quadratic response surface modeling algorithms [20], [21] is the need to compute all elements of the matrix A in (4). This matrix is often sparse and rank deficient in many practical problems. Therefore, instead of finding the full matrix A , ROAD approximates A by another low-rank matrix A_L . Such a low-rank approximation problem can be stated as follows: Given a matrix A , find another matrix A_L with rank $p < rank(A)$ such that their difference $\|A_L - A\|_F$ is minimized. Here, $\|\cdot\|_F$ denotes the Frobenius norm, which is the square root of the sum of the squares of all matrix elements. For simplicity, we assume that A is symmetric in this paper. Any asymmetric quadratic form can be easily converted to the equivalent symmetric form $0.5 \cdot X^T(A + A^T)X$ [27].

From the matrix theory [26], for any symmetric matrix $A \in R^{N \times N}$, the optimal rank- p approximation with the least Frobenius norm error is

$$A_L = \sum_{i=1}^p \lambda_i P_i P_i^T \quad (8)$$

where λ_i is the i th dominant eigenvalue, and $P_i \in R^N$ is the i th dominant eigenvector. The eigenvectors in (8) define an orthogonal projector $P_1 P_1^T + \dots + P_p P_p^T$, and every column in A_L is the *projection* of every column in A onto the subspace $span\{P_1, \dots, P_p\}$. We use this orthogonal projector for quadratic response surface modeling in this paper.

Although it is possible to increase the dimension of the projection space to achieve higher modeling accuracy, ROAD utilizes the rank-one projection $A_L = \lambda_1 P_1 P_1^T$ in order to achieve the best tradeoff between accuracy and complexity. As

we will demonstrate via numerical examples in Section IV, the rank-one projection is much easier to compute than higher order projections, whereas it provides sufficient accuracy for approximating many circuit performances in the local/small design space. The main advantage of such a rank-one projection is that for approximating the matrix $A \in R^{N \times N}$ in (4), only $\lambda_1 \in R$ and $P_1 \in R_N$ need to be determined, thus reducing the number of problem unknowns to $O(N)$. Compared with the problem size $O(N^2)$ in traditional approaches, our proposed projection-based method is more efficient and can be applied to larger size problems.

B. Coefficient Fitting via Implicit Power Iteration

Since the matrix A in (4) is not known in advance, we cannot use standard matrix techniques to compute its dominant eigenvalue λ_1 and eigenvector P_1 for a rank-one approximation. One approach for finding the optimal rank-one model is to solve the following optimization problem:

$$\begin{aligned} & \text{minimize} \quad \psi(\lambda_1, P_1, B, C) \\ & = \sum_{i=1}^S \left(\lambda_1 Y_i^T P_1 P_1^T Y_i + B^T Y_i + C - \tilde{f}_i \right)^2 \\ & \text{subject to} \quad \|P_1\|_2 = 1 \end{aligned} \quad (9)$$

where $\|\cdot\|_2$ denotes the two-norm of a vector.

Compared with (4), (9) approximates the matrix A by $\lambda_1 P_1 P_1^T$. Therefore, we expect that minimizing the cost function $\Psi(\lambda_1, P_1, B, C)$ in (9) will cause λ_1 and P_1 to converge to the dominant eigenvalue and eigenvector, respectively, of the original matrix A . Unfortunately, $\Psi(\lambda_1, P_1, B, C)$ is a sixth-order polynomial and might not be convex. In addition, the constraint set in (9) is specified by a quadratic equation and is not convex either. Therefore, the optimization in (9) is not a convex programming problem, and there is no efficient optimization algorithm that can guarantee to find the globally optimal solution.

Instead of solving the nonconvex optimization problem in (9), we propose a novel implicit power iteration method to efficiently extract the unknown coefficients λ_1 and P_1 . This implicit power iteration solves a sequence of overdetermined linear equations and exhibits robust convergence. An outline of this algorithm is shown in Fig. 3.

Next, we explain why the implicit power iteration yields the optimal rank-one approximation $A_L = \lambda_1 P_1 P_1^T$. Note that Step 3 in Fig. 3 approximates the matrix A by $Q_k Q_{k-1}^T$, where Q_{k-1} is determined in the previous iteration step. Finding such an optimal approximation is equivalent to solving the following overdetermined linear equation:

$$Q_k Q_{k-1}^T = A. \quad (10)$$

The least square error solution for (10) is given by [26]

$$Q_k = A Q_{k-1} \cdot (Q_{k-1}^T Q_{k-1})^{-1} = A Q_{k-1}. \quad (11)$$

In (11), $Q_{k-1}^T Q_{k-1} = \|Q_{k-1}\|_2^2 = 1$, since Q_{k-1} is normalized in Step 2 in Fig. 3. Equation (11) reveals an interesting fact that

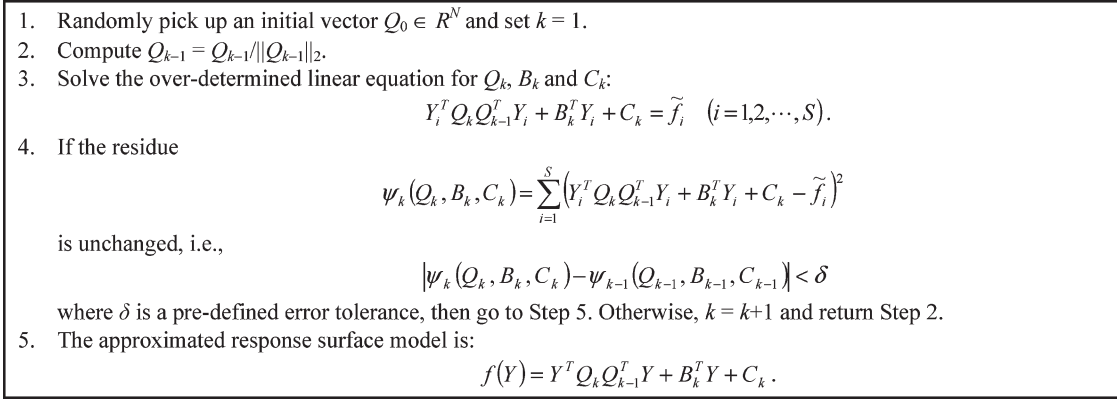


Fig. 3. Implicit power iteration algorithm for quadratic response surface modeling.

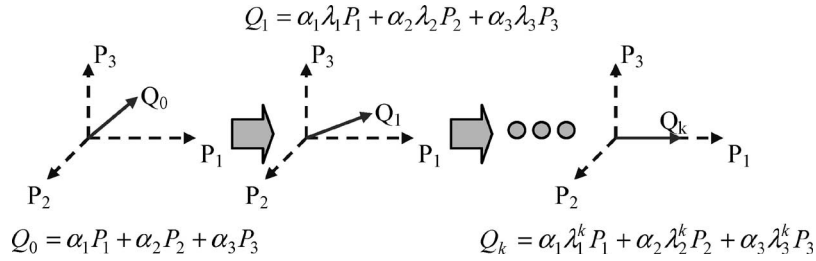


Fig. 4. Convergence of the implicit power iteration in a three-dimensional space.

solving the overdetermined linear equation in Step 3 “implicitly” computes the matrix–vector product AQ_{k-1} , which is the basic operation required in the traditional power iteration for dominant eigenvector computation [26].

Given the initial vector

$$Q_0 = \alpha_1 P_1 + \alpha_2 P_2 + \dots \quad (12)$$

where Q_0 is represented as the linear combination of all eigenvectors of A , the k th iteration step yields

$$Q_k = A^k Q_0 = \alpha_1 \lambda_1^k P_1 + \alpha_2 \lambda_2^k P_2 + \dots \quad (13)$$

In (13), we ignore the normalization $Q_{k-1} = Q_{k-1} / \|Q_{k-1}\|_2$, which is nothing else but a scaling factor. This scaling factor will not change the direction of Q_k . As long as $\alpha_1 \neq 0$ in (12), i.e., P_1 is not orthogonal to the initial vector Q_0 , $\alpha_1 \lambda_1^k P_1$ (with $|\lambda_1| > |\lambda_2| > |\lambda_3| > \dots$) will become increasingly dominant over other terms. Thus, Q_k will asymptotically approach the direction of P_1 , as shown in Fig. 4.

After the iteration in Fig. 3 converges, we have $Q_{k-1} = Q_{k-1} / \|Q_{k-1}\|_2 = P_1$ and $Q_k = AQ_{k-1} = \lambda_1 P_1$. $Q_k Q_{k-1}^T$ is the optimal rank-one approximation $A_L = \lambda_1 P_1 P_1^T$. Thus, the proposed implicit power iteration extracts the unknown coefficients λ_1 and P_1 with guaranteed convergence, but in an implicit way (i.e., without knowing the full matrix A). This “implicit” property is the key difference between the proposed algorithm and the traditional power iteration in [26].

The implicit power iteration in Fig. 3 needs to solve $2N + 1$ unknown coefficients, for which the required number of simulation samples is of the order of $O(N)$, and solving the overdetermined linear equation in Step 3 in Fig. 3 has a complexity of $O(N^3)$. Therefore, the proposed projection-based modeling

is much more efficient than the traditional full-rank quadratic modeling, which requires $O(N^2)$ simulation samplings and has a fitting cost of $O(N^6)$ for solving the overdetermined linear equation.

The aforementioned implicit power iteration is used for rank-one quadratic response surface (i.e., polynomial) modeling. If a quadratic coefficient matrix contains p ($p \geq 2$) dominant eigenvalues and, therefore, a rank- p approximation is required, the implicit power iteration can be repeatedly applied to extract the first p dominant eigenvalues and eigenvectors. More details on this rank- p approximation are beyond the scope of this paper and can be found in [31].

Finally, it is worth mentioning that the implicit power iteration is probably convergent if A is symmetric. For an asymmetric A , we can show that Q_{k-1} and Q_k should iteratively converge to the directions of the dominant left and right singular vectors of A in order to achieve the optimal rank-one approximation. However, the global convergence of the implicit power iteration is difficult to prove in that case.

C. Extension to Projection-Based Posynomial Modeling

The projection-based technique described in Sections III-A and B can be extended to quadratic posynomial modeling, which is required in the second step of the ROAD optimization flow in Fig. 2. Most of the projection theories described in Section III-A can be directly applied to quadratic posynomial modeling. For example, given the symmetric square matrix \tilde{A} in (6), the optimal rank- p approximation is still given by (8), where λ_i and P_i in (8) now represent the i th dominant eigenvalue and eigenvector of \tilde{A} , respectively. The difference between polynomial and posynomial fitting is that for posynomial fitting, we need to further prove that $A_L \in R_+^{2M \times 2M}$,

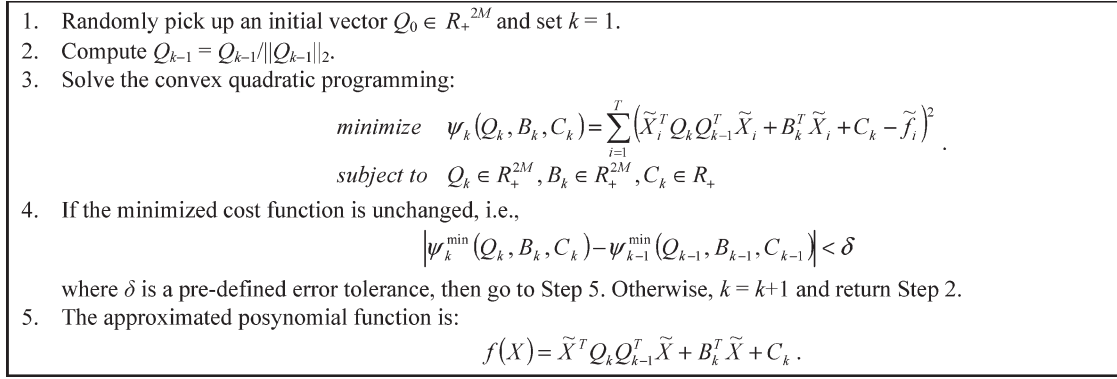


Fig. 5. Implicit power iteration algorithm for quadratic posynomial modeling.

i.e., that the low-rank approximation A_L is nonnegative, if $\tilde{A} \in R_+^{2M \times 2M}$. In order to prove this, we need the following theorem.

Perron–Frobenius Theorem [27]: Let A be a real nonnegative matrix, i.e., $A \in R_+^{n \times n}$. Then, $\lambda_1 = \rho(A)$, where the spectral radius of A is a simple eigenvalue of A . Moreover, there exists an eigenvector P_1 with nonnegative elements associated with this eigenvalue.

One conclusion from the Perron–Frobenius theorem is that since the first dominant eigenvalue λ_1 and eigenvector P_1 for the nonnegative matrix \tilde{A} in (6) are both nonnegative, the rank-one approximation $A_L = \lambda_1 P_1 P_1^T$ is also nonnegative. In other words, a quadratic posynomial is invariant (i.e., remains a posynomial) under the rank-one projection.

On the other hand, the eigenvectors P_i are mutually orthogonal for the symmetric matrix \tilde{A} . Since P_1 is nonnegative, P_2, P_3, \dots must contain nonpositive elements. This implies that any rank- p projection with $p \geq 2$ might convert a posynomial to a signomial with negative coefficients.

ROAD utilizes the rank-one projection, which is theoretically guaranteed by the earlier discussion to map a posynomial to another posynomial. In addition, the unknown dominant eigenvalue λ_1 and eigenvector P_1 can be extracted by a similar implicit power iteration, as summarized in Fig. 5.

In Fig. 5, the optimization in Step 3 requires all coefficients $Q_k, B_k,$ and C_k to be nonnegative so that the approximated function is a posynomial. In Step 3, the cost function $\Psi_k(Q_k, B_k, C_k)$ is a positive semidefinite quadratic function, and the constraint is a convex set. Therefore, the optimization in Step 3 is a convex programming problem. The overall implicit power iteration in Fig. 5 consists of a sequence of such convex programming steps and will robustly converge to the optimal rank-one posynomial model.

D. Robust Analog Optimization With Process/Environmental Variations

Given a circuit topology, the circuit performance (e.g., gain, bandwidth) is a function of design variables (e.g., bias current, transistor sizes) and process and environmental parameters (e.g., V_{TH} , temperature). The process and environmental parameters can be modeled as random variables to capture their variations. In such cases, the circuit performance $f(X)$ with the fixed design variables $X = [x_1, x_2, \dots, x_M]^T$ is also a random

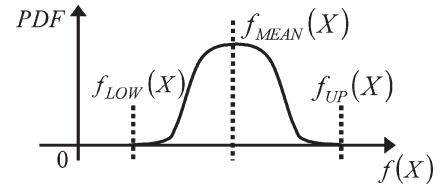


Fig. 6. Probability density function of the circuit performance $f(X)$.

TABLE I
ROAD FORMULATION FOR GEOMETRIC PROGRAMMING¹

Specification	Cost Function/Constraint
$f_{MEAN}(X) \leq Spec$	$f_{MEAN}(X)/Spec \leq 1$ and $Spec/f_{MEAN}(X) \leq 1$
$Var[f(X)] \leq Spec$	$[f_{UP}(X) - f_{LOW}(X)]/Spec \leq 1$
$f(X) \leq Spec$	$f_{UP}(X)/Spec \leq 1$
$f(X) \geq Spec$	$Spec/f_{LOW}(X) \leq 1$

¹The table only considers positive *Spec* values. Negative *Spec* values can be normalized to positive ones through proper scaling/shifting [24].

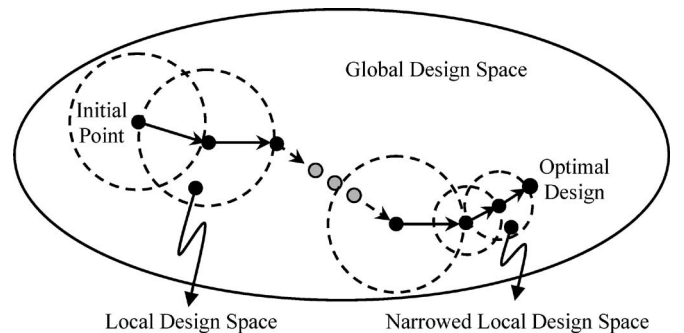


Fig. 7. Illustration of the ROAD iterations.

variable that can be characterized by a probability density function. In ROAD, instead of handling the complete probability density function, we define three important metrics for each circuit performance, namely: 1) the mean value $f_{MEAN}(X)$; 2) the lower bound $f_{LOW}(X)$; and 3) the upper bound $f_{UP}(X)$, as shown in Fig. 6. The lower bound $f_{LOW}(X)$ and the upper bound $f_{UP}(X)$ in Fig. 6 are defined as two specific points (e.g., the 1% point and the 99% point, respectively) on the cumulative distribution function.

In order to achieve a robust design, ROAD incorporates both process and environmental variations into the cost function and/or constraints during optimization. For example, the design

1. Start from an initial design D_0 and perturbation size $\pm\epsilon\%$. Set iteration index $k = 1$.
2. Make a perturbation of $\pm\epsilon\%$ on all design variables of D_{k-1} . This perturbation defines a local design space S_k for the k -th post-tuning. Using orthogonal arrays [32], generate a set of samples $\{Y_i\}_k$ in S_k , where Y contains all design variables and process/environmental parameters. The perturbation on design variables is bounded by the local design space S_k and the variation on process/environmental parameters is determined by the manufacture/environmental conditions.
3. For every sampling point in $\{Y_i\}_k$, run transistor-level simulation to compute the circuit performance $f(Y_i)$. Apply the projection-based technique to fit the response surface model $f(Y)$.
4. Using orthogonal arrays [32], generate a set of samples $\{X_i\}_k$ in S_k , where X contains all design variables. The perturbation on design variables is bounded by the local design space S_k .
5. For every sampling point in $\{X_i\}_k$, apply the APEX algorithm [30] to compute $f_{MEAN}(X_i)$, $f_{LOW}(X_i)$ and $f_{UP}(X_i)$.
6. Fit the cost function and constraints (e.g. $f_{MEAN}(X)/Spec$, $[f_{UP}(X) - f_{LOW}(X)]/Spec$, etc.) as posynomial functions using the proposed projection-based technique.
7. Run geometric programming and find the optimal design D_k in the design space S_k .
8. If the difference between D_k and D_{k-1} is smaller than a pre-defined error tolerance, then go Step 9. Otherwise, $k = k+1$ and return Step 2.
9. If $\epsilon\%$ is smaller than the pre-defined minimal design space size, then stop iteration. Otherwise, $\epsilon = \epsilon/2$, $k = k+1$ and return Step 2.

Fig. 8. Outline of the overall ROAD iteration scheme.

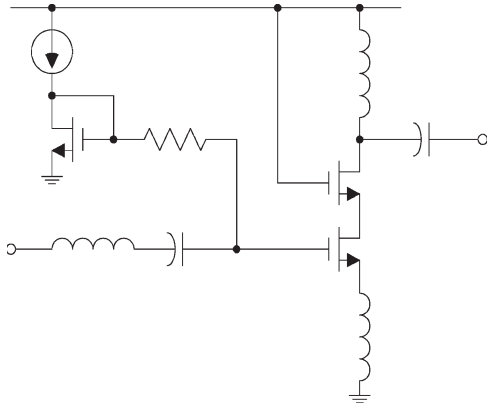


Fig. 9. Circuit schematic of an LNA.

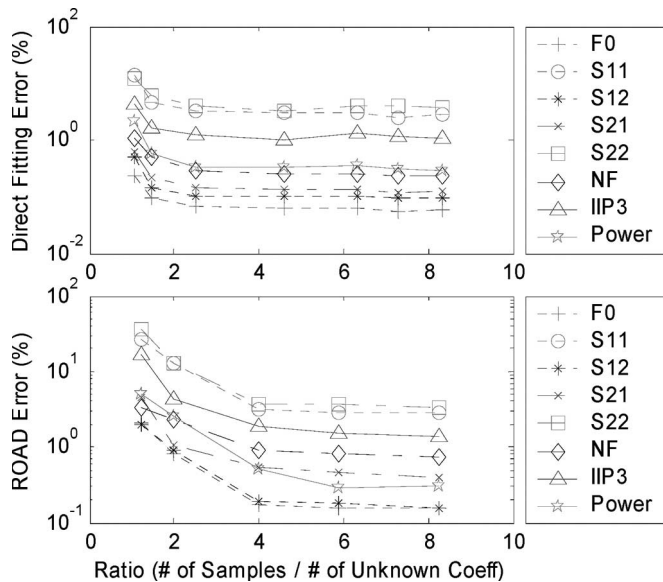


Fig. 10. Effect of the training set size for response surface modeling.

specification $\text{Var}[f(X)] \leq Spec$ is translated to $[f_{UP}(X) - f_{LOW}(X)]/Spec \leq 1$. In other words, we force the distance between the upper bound and the lower bound to be no greater than $Spec$ under process and environmental variations. Table I

TABLE II
RESPONSE SURFACE MODELING ERROR FOR LNA ($\epsilon\% = 5\%$)

Performance	Direct Fitting [20]-[21]	ROAD
F0	0.07%	0.18%
S11	3.25%	3.20%
S12	0.10%	0.19%
S21	0.15%	0.54%
S22	3.93%	3.64%
NF	0.29%	0.89%
IIP3	1.25%	1.84%
Power	0.33%	0.52%

TABLE III
RESPONSE SURFACE MODELING COST FOR LNA ($\epsilon\% = 5\%$)

	Direct Fitting [20]-[21]	ROAD
# of Unknown Coefficients	231	41
# of Samples in Training Set	578	164
Spectre Simulation (Sec.)	17909	5076
Coefficient Fitting (Sec.)	13.38	6.65

summarizes all geometric programming cost functions and constraints utilized in ROAD.

As shown in Fig. 2, after the response surface models are extracted for the circuit performance f and after we know the probability distributions of the process and environmental parameters (e.g., Normal distribution), the probability density function of the circuit performance f can be extracted for any fixed design variables $X = [x_1, x_2, \dots, x_M]^T$ by using the APEX algorithm [30]. It follows that the mean value $f_{MEAN}(X)$, the lower bound $f_{LOW}(X)$, and the upper bound $f_{UP}(X)$ are computed at one sampling point $X = [x_1, x_2, \dots, x_M]^T$. This probability extraction procedure is repeated for a number of sampling points $\{X_i\}$. Then, depending on the design specifications, the cost function and constraints (e.g., $f_{MEAN}(X)/Spec$, $[f_{UP}(X) - f_{LOW}(X)]/Spec$) are approximated as posynomial functions using the projection-based technique proposed in Section III-C. Finally, based on these extracted posynomial

performance models, geometric programming is applied to optimize the design variables and improve the circuit performance and the parametric yield.

E. Road Iteration Scheme

As shown in Fig. 7 and summarized in Fig. 8, starting from an initial design, ROAD iteratively improves the circuit performance and the parametric yield by successive performance modeling, statistical analysis, and geometric programming. In each iteration, ROAD fits posynomial models in a local design space to approximate the worst case circuit performance with consideration of process and environmental variations. Then, geometric programming is utilized to find the optimal design in that local design space.

The overall iteration is performed at two levels, namely: 1) inner loop iteration and 2) outer loop iteration. The inner loop iteration searches for the optimal solution in the local design spaces that have the same size but are centered at different expansion points. During the k th iteration, the k th local design space is defined by a perturbation of $\pm\epsilon\%$ on all design variables of the previous iteration result D_{k-1} . Then, after the optimal design is found under the current perturbation size $\pm\epsilon\%$, ROAD reduces ϵ by a factor of 2 and repeats the inner loop iteration for a finer-grained search. The iteration is stopped if no further improvement is identified between two successive steps and if the current $\epsilon\%$ is smaller than the predefined minimal design space size. Since the performance modeling error is successively reduced due to the iteratively narrowed design space, ROAD can converge to a final design with high accuracy.

From our experience, the final design accuracy of the aforementioned ROAD iteration is mainly determined by the final value of $\epsilon\%$, i.e., the predefined minimal design space size in Step 9 in Fig. 8. This final $\epsilon\%$ determines the performance modeling accuracy at the end of the iteration. Higher design accuracy can be achieved if a smaller final $\epsilon\%$ is utilized. On the other hand, the final ROAD design is not sensitive to the initial value of $\epsilon\%$. In many applications, the initial and final $\epsilon\%$ can be typically selected as 5%–10% and 1%–2%, respectively.

IV. NUMERICAL EXAMPLES

We demonstrate the ROAD flow as applied to circuit examples designed in the IBM BiCMOS 0.25 μm or TSMC CMOS 0.13 μm process. All the numerical experiments are performed on a Sun SPARC 1-GHz server.

A. Low-Noise Amplifier (LNA)

Fig. 9 shows a LNA designed in the IBM BiCMOS 0.25- μm process. The LNA circuit includes 12 design variables and 8 design specifications. Principal component analysis (PCA) [33] is applied to extract the principal factors of the random process variations based on the probability distributions and the correlation information obtained from the IBM design kit. In addition, variable screening [34] is further utilized to identify a subset of the random process parameters that have much greater influence on circuit performance than the others. After PCA and

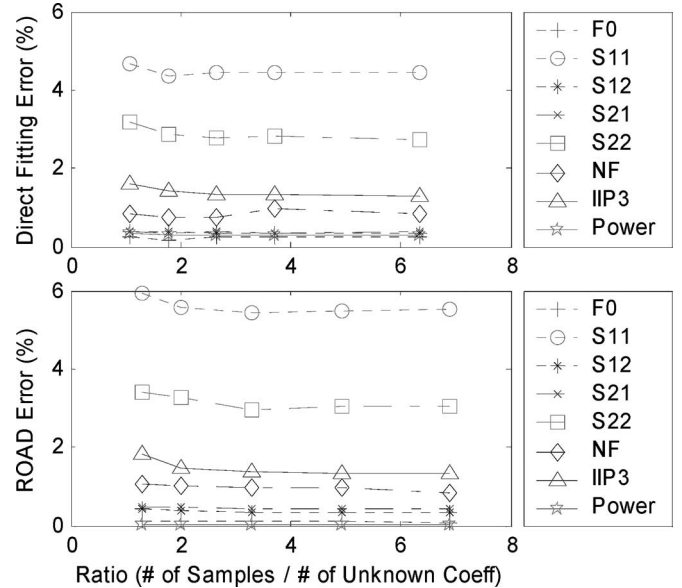


Fig. 11. Effect of the training set size for posynomial modeling.

TABLE IV
POSYNOMIAL MODELING ERROR FOR LNA ($\epsilon\% = 5\%$)

Performance	Direct Fitting w/o Template [24]	Direct Fitting /w Template [24]	ROAD
F0	0.09%	0.25%	0.10%
S11	3.92%	4.45%	5.44%
S12	0.33%	0.35%	0.36%
S21	0.28%	0.40%	0.42%
S22	2.49%	2.78%	2.97%
NF	0.65%	0.74%	0.96%
IIP3	1.31%	1.32%	1.39%
Power	0.02%	0.30%	0.03%

TABLE V
POSYNOMIAL MODELING COST FOR LNA ($\epsilon\% = 5\%$)

	Direct Fitting w/o Template [24]	Direct Fitting /w Template [24]	ROAD
# of Unknown Coefficients	313	91	49
# of Samples in Training Set	722	242	162
Statistical Analysis [30] (Sec.)	924.16	271.04	220.32
Coefficient Fitting (Sec.)	190.68	7.19	16.33

variable screening, eight random factors are left to model the process variations for both active devices (i.e., MOSFETs) and passive components (i.e., resistors, inductors, and capacitors).

For each posttuning iteration, we sample the local design space by a perturbation of $\pm\epsilon\%$ on all design variables. $\epsilon\%$ is initially set to 5% and then successively reduced to 1% during the iterations. Two independent sampling sets are generated, which are called the training set and the testing set, respectively. The training set is created by orthogonal arrays [32], which pick up the most important samples based on statistical analysis; this is used for polynomial/posynomial coefficient fitting. For testing and comparison, we collect 500 random samples as the testing set (independent of the training set) and use them

TABLE VI
CIRCUIT PERFORMANCE AND OPTIMIZATION COST FOR LNA

	Specification	Initial Design	Nominal ROAD Optimization		Robust ROAD Optimization	
		Nominal	Nominal	Worst-Case	Nominal	Worst-Case
F0 (GHz)	= 2.1	2.07	2.10	—	2.10	—
S11 (dB)	≤ -10	-13.98	-12.92	-11.82	-11.92	-10.04
S12 (dB)	≤ -25	-23.89	-26.23	-25.90	-26.26	-25.73
S21 (dB)	≥ 15	14.60	15.00	14.52	15.46	15.05
S22 (dB)	≤ -10	-14.47	-24.37	-14.91	-17.27	-11.00
NF (dB)	≤ 1.5	1.19	1.30	1.38	1.42	1.50
IIP3 (dBm)	≥ 5	4.97	4.99	3.24	5.51	5.02
Power (mW)	Minimize	13.23	7.47	8.96	9.34	10.58
Cost (# of Spectre Simulations)	—	—	1980	—	2640	—

to verify the performance modeling accuracy for all modeling techniques.

1) *Robust Convergence of Implicit Power Iteration*: We test the convergence of the proposed implicit power iteration for both quadratic response surface (i.e., polynomial) modeling and quadratic posynomial modeling. One hundred different initial vectors are randomly selected and used for running the implicit power iteration. We find that all 100 experiments reliably converge without a single failure.

2) *Response Surface Modeling*: The purpose of response surface modeling is to extract the quadratic polynomial $f_i(Y)$, where f_i stands for the i th circuit performance, and Y contains 12 design variables and 8 process parameters in this example. For both direct fitting [20], [21] and ROAD, Fig. 10 shows the relation between the response surface modeling error and the training set size, where the perturbation $\varepsilon\%$ is set to 5%. The modeling error is measured using the same testing set that includes 500 random sampling points. Studying Fig. 10, we find that the number of samples in the training set should be three to four times greater than the number of unknown coefficients. Further increasing the number of training samples does not have a significant impact on reducing the modeling error. This observation implies that the required number of training samples depends on the number of unknown coefficients. As the unknown coefficient number is reduced in ROAD due to projection, not only can we decrease the computational time for coefficient fitting, but we can also significantly reduce the circuit simulation cost due to the smaller training set.

Table II summarizes the response surface modeling accuracy for both direct fitting [20], [21] and ROAD. The modeling error in Table II is measured using the same testing set that contains 500 random sampling points. Studying Table II, we find that no significant accuracy is surrendered by using the proposed projection-based technique in ROAD. In addition, it is interesting to note that for several circuit performances (e.g., S11 and S22), the ROAD modeling error is even smaller than the direct fitting error. Theoretically, the direct fitting approach should be more accurate, since it takes into account all quadratic polynomial terms in (4). However, this is not necessarily true in many practical applications, where the training set only contains a limited number of sampling points. As shown in Fig. 10, both direct fitting and ROAD require more training samples in order to achieve smaller fitting error. Therefore, given a limited number of samples in the training set (see Table III), ROAD might produce a more accurate response surface model than direct fitting.

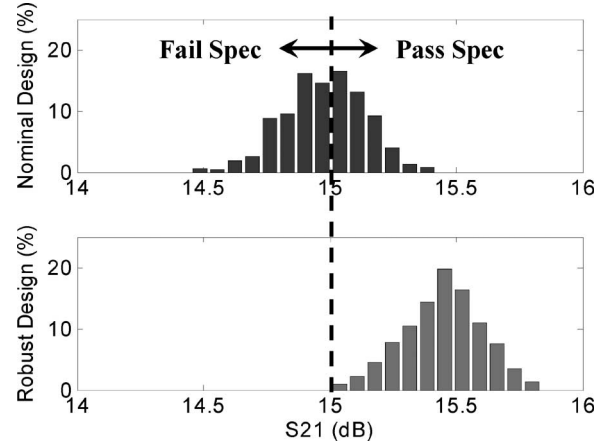


Fig. 12. Probability distribution of S21 achieved by the nominal and robust ROAD optimizations.

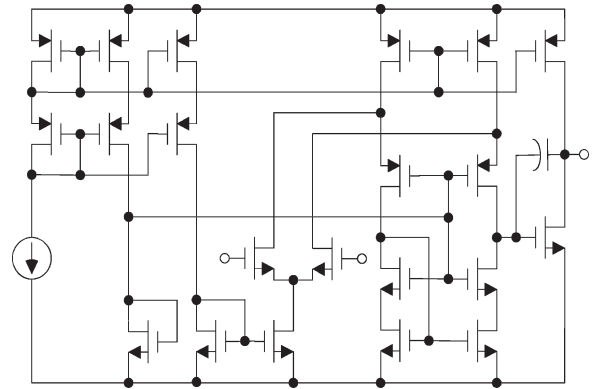


Fig. 13. Circuit schematic of a two-stage folded-cascode operational amplifier.

Table III further shows the number of training samples used for coefficient fitting and the computational cost for generating the response surface models. Even for this small example, ROAD achieves $3.5\times$ speedup for Spectre simulation and $2\times$ speedup for coefficient fitting. As discussed in Section III-A, ROAD reduces the number of unknown coefficients from $O(N^2)$ to $O(N)$, where N is the total number of design variables and process/environmental parameters. Therefore, we expect that as the problem size increases further, ROAD would achieve additional speedup over the direct fitting method.

3) *Posynomial Modeling*: Based on the preextracted response surface models $f_i(Y)$, statistical analysis [30] is applied to evaluate the worst case circuit performances, which are

then approximated as the posynomial functions of the design variables. For both direct fitting [24] and ROAD, Fig. 11 shows the relation between the posynomial modeling error and the training set size, where the perturbation $\varepsilon\%$ is set to 5%. The modeling error is measured using the same testing set that includes 500 random sampling points. As shown in Fig. 11, in order to achieve accurate posynomial models, the number of samples in the training set should be two to three times greater than the number of unknown coefficients. This observation is similar to what we observe for response surface modeling in Fig. 10.

Table IV compares the accuracy of three posynomial modeling approaches. The modeling error is measured using the same testing set that includes 500 random sampling points. In Table IV, direct fitting without template estimation [24] is the most accurate one since it takes into account all possible posynomial product terms in (6). However, although both the template estimation method and the proposed ROAD approach apply simplifications to reduce the computational cost, their fitting accuracy is still comparable to the direct fitting without template estimation. Table V shows the computational cost for all these three approaches. In this small example with only eight design variables, ROAD consumes more coefficient fitting time than the template estimation approach. However, the total computational cost (i.e., statistical analysis + coefficient fitting) of ROAD is still the smallest, since it requires fewer sampling points in the training set and, therefore, saves a significant amount of computational time when generating these training samples by statistical analysis.

4) *Robust Design*: With the preextracted posynomial performance models, the geometric programming problem is solved efficiently, taking 1–2 s for this LNA example. For testing and comparison, three different designs are generated. The initial design is created by hand analysis, the nominal design is optimized without considering process variations, and the robust design is synthesized for high parametric yield.

As shown in Table VI, the initial manual design contains a few circuit performance metrics that do not satisfy the design specifications. We apply ROAD to nominal optimization and robust optimization, respectively. Table VI shows the nominal and worst case performance values after both optimizations. The worst case performance is measured at the 1% or 99% point on the cumulative distribution function that is calculated from 1000 transistor-level Monte Carlo simulations in Spectre.

As shown in Table VI, the nominal ROAD optimization pushes several nominal performances (i.e., S21 and IIP3) to the boundaries of the design specifications, and therefore, the worst case performances violate the design requirements. The robust ROAD design, however, leaves sufficient margin for each performance metric. These margins are estimated from the probability distributions extracted by APEX [30] and are related to the parametric yield values of individual performance matrices. The additional performance margins enable the circuit to meet design specifications under process variations. For example, Fig. 12 plots the probability distribution of S21, where the robust design produced by ROAD satisfies the specification for a much larger fraction of the process variations than the nominal design.

TABLE VII
RESPONSE SURFACE MODELING ERROR FOR OP AMP ($\varepsilon\% = 5\%$)

Performance	Direct Fitting [20]-[21]	ROAD
Gain	1.46%	1.18%
UGF	0.75%	0.95%
Gain Margin	0.85%	1.09%
Phase Margin	0.14%	0.18%
Slew Rate	0.06%	0.07%
Swing	0.02%	0.02%
Power	0.19%	0.32%

TABLE VIII
RESPONSE SURFACE MODELING ERROR FOR OP AMP ($\varepsilon\% = 1\%$)

Performance	Direct Fitting [20]-[21]	ROAD
Gain	1.42%	1.07%
UGF	0.65%	0.79%
Gain Margin	0.76%	1.06%
Phase Margin	0.12%	0.08%
Slew Rate	0.03%	0.02%
Swing	0.01%	0.01%
Power	0.13%	0.20%

TABLE IX
RESPONSE SURFACE MODELING COST FOR OP AMP ($\varepsilon\% = 5\%$)

	Direct Fitting [20]-[21]	ROAD
# of Unknown Coefficients	903	83
# of Samples in Training Set	1922	328
Spectre Simulation (Sec.)	77282	13690
Coefficient Fitting (Sec.)	864.98	26.74

TABLE X
POSYNOMIAL MODELING ERROR FOR OP AMP ($\varepsilon\% = 5\%$)

Performance	Direct Fitting /w Template [24]	ROAD
Gain	0.112%	0.128%
UGF	0.091%	0.071%
Gain Margin	0.275%	0.159%
Phase Margin	0.049%	0.054%
Slew Rate	0.042%	0.050%
Swing	0.004%	0.003%
Power	0.146%	0.199%

TABLE XI
POSYNOMIAL MODELING ERROR FOR OP AMP ($\varepsilon\% = 1\%$)

Performance	Direct Fitting /w Template [24]	ROAD
Gain	0.005%	0.005%
UGF	0.007%	0.008%
Gain Margin	0.006%	0.006%
Phase Margin	0.002%	0.002%
Slew Rate	0.005%	0.005%
Swing	0.001%	0.001%
Power	0.007%	0.009%

TABLE XII
POSYNOMIAL MODELING COST FOR OP AMP ($\varepsilon\% = 5\%$)

	Direct Fitting [20]-[21]	ROAD
# of Unknown Coefficients	630	137
# of Samples in Training Set	1458	343
Statistical Analysis [30] (Sec.)	1530.90	336.14
Coefficient Fitting (Sec.)	235.29	66.59

TABLE XIII
CIRCUIT PERFORMANCE AND OPTIMIZATION COST FOR OP AMP

	Specification	Initial Design	Nominal ROAD Optimization		Robust ROAD Optimization	
		Nominal	Nominal	Worst-Case	Nominal	Worst-Case
Gain (dB)	≥ 100	116.2	106.6	104.3	102.7	100.0
UGF (MHz)	≥ 10	11.76	9.96	9.05	10.94	9.93
Gain Margin	≥ 2	7.45	5.86	5.12	7.27	6.28
Phase Margin	≥ 60	65.82	59.64	55.10	63.48	59.50
Slew Rate (V/ μ s)	≥ 20	46.36	19.96	19.36	20.53	19.87
Swing (V)	≥ 0.5	1.00	1.00	1.00	1.00	1.00
Power (mW)	Minimize	2.54	0.73	0.79	0.79	0.86
Cost (# of Spectre Simulations)	—	—	5186		8038	

B. Two-Stage Folded-Cascode Operational Amplifier

Fig. 13 shows a two-stage folded-cascode operational amplifier, which includes 34 design variables and 8 design specifications. After principal component analysis [33] and variable screening [34], seven random factors are left to model the process variations for both active devices (i.e., MOSFETs) and passive components (i.e., capacitors). For simplicity, device mismatches are not considered in this example. However, it should be noted that nothing precludes us from including more detailed mismatch models in ROAD as well. In each posttuning iteration, we sample the local design space by a perturbation of $\pm\epsilon\%$ on all design variables. Similar to the LNA example, $\epsilon\%$ is initially set to 5%, and it is successively reduced to 1% during the iterations.

1) *Response Surface Modeling*: Tables VII and VIII compare the response surface modeling accuracy for both direct fitting [20], [21] and ROAD. ROAD achieves similar modeling accuracy as the direct fitting approach. In addition, comparing Tables VII and VIII, one would find that as the perturbation $\pm\epsilon\%$ is reduced from $\pm 5\%$ to $\pm 1\%$, the modeling accuracy is improved. The ROAD modeling error is controlled around 1% for all circuit performances in Table VIII, which guarantees high optimization accuracy at the end of the ROAD iterations. It should be noted that achieving small modeling error is extremely important for ROAD. The process and environmental variations in today's IC technologies typically introduce 20%–30% variations on circuit performance. If the modeling error is not sufficiently smaller than this value, the circuit yield cannot be accurately estimated and optimized via these models.

Table IX summarizes the computational cost for generating the response surface models. In this example, ROAD achieves $5.5\times$ speedup for Spectre simulation and $32\times$ speedup for coefficient fitting. Compared with the previous LNA example, the speedup of using ROAD becomes much more significant as the problem size increases.

2) *Posynomial Modeling*: Tables X and XI summarize the posynomial modeling accuracy for both direct fitting with template estimation [24] and ROAD. Direct fitting without template estimation [24] includes 2381 unknown coefficients in this example and is, therefore, too computationally expensive. From the data in Tables X and XI, both template estimation and ROAD yield similar modeling accuracy. However, as shown in Table XII, ROAD is four times faster for both statistical analysis and coefficient fitting.

In addition, it should be noted that the maximal ROAD posynomial modeling error is no more than 0.01% in Table XI

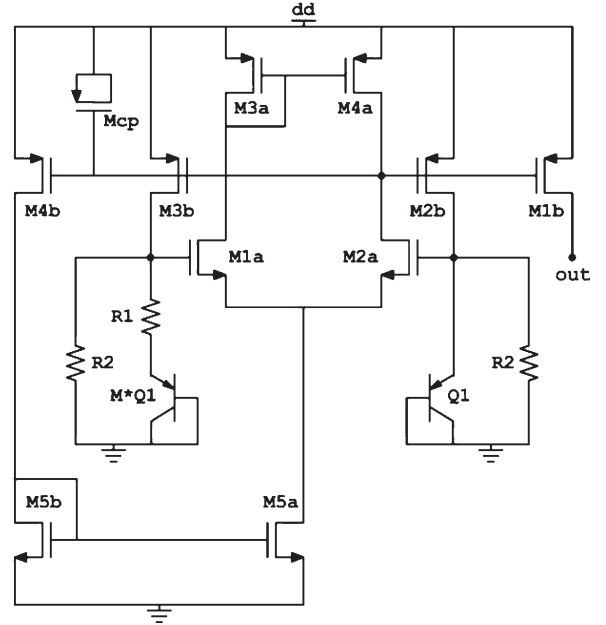


Fig. 14. Circuit schematic of a bandgap reference.

when the perturbation $\pm\epsilon\%$ is reduced to $\pm 1\%$. This small error implies a high optimization accuracy at the end of the ROAD iterations.

3) *Robust Design*: Table XIII shows the Spectre-simulated Op Amp performance before and after posttuning. The worst case performance in Table XIII is measured at the 1% or 99% point on the cumulative distribution function that is calculated from 1000 transistor-level Monte Carlo simulations in Spectre. As shown in Table XIII, the initial circuit is overdesigned, resulting in a large power consumption. The nominal ROAD optimization yields a much better design that meets all design specifications, but it overoptimizes the circuit with several nominal performances (e.g., unity gain frequency, phase margin, and slew rate) sitting on the boundaries of the design specifications, and therefore, the worst case performances violate the design requirements. Finally, the robust ROAD optimization takes into account the process variations and, therefore, leaves sufficient margin for each circuit performance metric. These margins are estimated from the probability distributions extracted by APEX [30] and are related to the parametric yield values of individual performance matrices. Compared with the initial design, the robust design generated by ROAD achieves more than three times power reduction in this example.

TABLE XIV
CIRCUIT PERFORMANCE AND OPTIMIZATION COST FOR BANDGAP

	Specification	Initial Design	Robust ROAD Optimization
Area (μm^2)	≤ 3600	3234.72	3414.29
Nominal Power (mW)	≤ 125	125.74	110.47
Nominal I_{OUT} (μA)	$= 10$	10.74	9.95
$\text{std}(I_{\text{OUT}})$	Minimize	0.2715	0.1950
Cost (# of Spectre Simulations)	—	—	6001

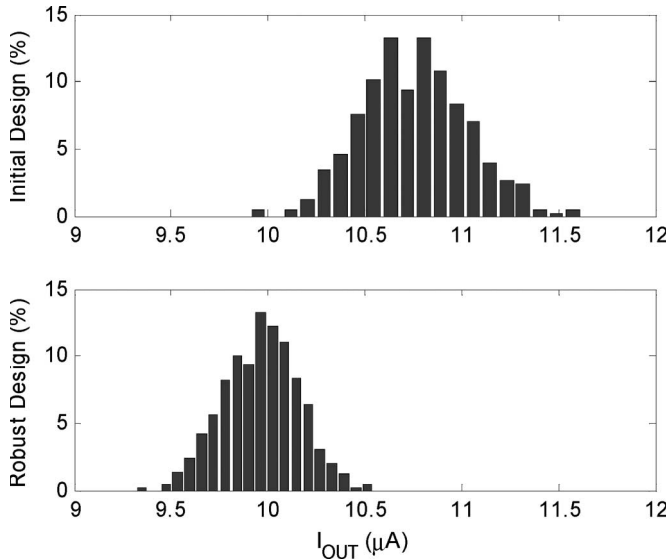


Fig. 15. Probability distribution of I_{OUT} achieved by the initial and robust designs.

C. Bandgap Reference

Fig. 14 shows a bandgap reference designed in the TSMC 0.13 μm process. The bandgap reference circuit includes 15 design variables and three design specifications. The primary goal of designing the bandgap reference is to reduce the output current variation caused by manufacturing and environmental fluctuations. After principal component analysis [33] and variable screening [34], 29 random factors are left to model the process and environmental variations, including both interdie variations and device mismatches. Similar to the previous two examples, $\varepsilon\%$ is initially set to 5%, and it is successively reduced to 1% during the ROAD iterations.

Table XIV shows the Spectre-simulated bandgap reference performance before and after posttuning. The standard deviation of the output current $\text{std}(I_{\text{OUT}})$ is calculated from 1000 transistor-level Monte Carlo simulations in Spectre. Note that the output current variation is reduced by approximately 30% after ROAD optimization. The probability distributions of the output current are plotted in Fig. 15.

V. CONCLUSION

In this paper, we propose a novel tool called ROAD to accurately posttune (i.e., locally optimize) analog/RF circuits for better performance and/or yield. Unlike nominal optimization, which might overoptimize the circuit, ROAD leaves sufficient margin for each circuit performance metric. These margins enable the circuit to meet design specifications even in the presence of large-scale process and environmental variations.

The novelty of ROAD lies in a projection-based approach for generating accurate performance (both polynomial and posynomial) models from transistor-level simulation. A crucial component of our approach is a novel implicit power iteration algorithm that is used to extract the unknown model coefficients with robust convergence. Compared with traditional modeling techniques, ROAD achieves significant runtime speedup and scales well with problem size.

Finally, it is important to mention that the projection-based modeling approach proposed in this paper is not limited to performance modeling for analog circuits. The same idea can be directly applied to solve the quadratic polynomial and posynomial modeling problems in many other engineering applications. In addition, the proposed sequential geometric programming (SGP) technique is also a general optimization scheme that can be applied to many other optimization problems. A detailed theoretical comparison between SGP and the traditionally used sequential quadratic programming would be an interesting direction for future research.

ACKNOWLEDGMENT

The authors would like to thank Dr. M. del Mar Hershenson and D. M. Colleran of Sabio Laboratories Inc. for providing the bandgap reference example. The authors would also like to thank all the anonymous reviewers for their excellent comments, resulting in an improvement in the quality of this paper.

REFERENCES

- [1] G. Gielen and R. Rutenbar, "Computer-aided design of analog and mixed-signal integrated circuits," *Proc. IEEE*, vol. 88, no. 12, pp. 1825–1852, Dec. 2000.
- [2] W. Nye, D. Riley, A. Sangiovanni-Vincentelli, and A. Tits, "DELIGHT: SPICE: An optimization-based system for the design of integrated circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 7, no. 4, pp. 501–519, Apr. 1988.
- [3] M. Krasnicki, R. Phelps, R. Rutenbar, and L. Carley, "MAELSTROM: Efficient simulation-based synthesis for custom analog cells," in *Proc. IEEE/ACM Des. Autom. Conf.*, 1999, pp. 945–950.
- [4] R. Phelps, M. Krasnicki, R. Rutenbar, L. Carley, and J. Hellums, "Anaconda: Simulation-based synthesis of analog circuits via stochastic pattern search," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 6, pp. 703–717, Jun. 2000.
- [5] M. Krasnicki, R. Phelps, J. Hellums, M. McClung, R. Rutenbar, and L. Carley, "ASP: A practical simulation-based methodology for the synthesis of custom analog circuits," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2001, pp. 350–357.
- [6] G. Plas, G. Debyser, F. Leyn, K. Lampaert, J. Vandebussche, G. Gielen, W. Sansen, P. Veselinovic, and D. Leenaerts, "AMGIE—A synthesis environment for CMOS analog integrated circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 9, pp. 1037–1058, Sep. 2001.
- [7] K. Francken and G. Gielen, "A high-level simulation and synthesis environment for $\Delta\Sigma$ modulators," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 8, pp. 1049–1061, Aug. 2003.

- [8] M. Hershenson, S. Mohan, S. Boyd, and T. Lee, "Optimization of inductor circuits via geometric programming," in *Proc. IEEE/ACM Des. Autom. Conf.*, 1999, pp. 994–998.
- [9] M. Hershenson, A. Hajimiri, S. Mohan, S. Boyd, and T. Lee, "Design and optimization of LC oscillators," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 1999, pp. 65–69.
- [10] M. Hershenson, S. Boyd, and T. Lee, "Optimal design of a CMOS Op-Amp via geometric programming," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 1, pp. 1–21, Jan. 2001.
- [11] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [12] S. Director, P. Feldmann, and K. Krishna, "Statistical integrated circuit design," *IEEE J. Solid-State Circuits*, vol. 28, no. 3, pp. 193–202, Mar. 1993.
- [13] Z. Wang and S. Director, "An efficient yield optimization method using a two step linear approximation of circuit performance," in *Proc. IEEE Eur. Des. Test Conf.*, 1994, pp. 567–571.
- [14] F. Schenkel, M. Pronath, S. Zizala, R. Schwencker, H. Graeb, and K. Antreich, "Mismatch analysis and direct yield optimization by piecewise linearization and feasibility-guided search," in *Proc. IEEE/ACM Des. Autom. Conf.*, 2001, pp. 858–863.
- [15] K. Antreich, H. Graeb, and C. Wieser, "Circuit analysis and optimization driven by worst-case distances," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 13, no. 1, pp. 57–71, Jan. 1994.
- [16] A. Seifi, K. Ponnambalam, and J. Vlach, "A unified approach to statistical design centering of integrated circuits with correlated parameters," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 46, no. 1, pp. 190–196, Jan. 1999.
- [17] A. Dharchoudhury and S. Kang, "Worst-case analysis and optimization of VLSI circuit performances," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 14, no. 4, pp. 481–492, Apr. 1995.
- [18] G. Debyser and G. Gielen, "Efficient analog circuit synthesis with simultaneous yield and robustness optimization," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 1998, pp. 308–311.
- [19] T. Mukherjee, L. Carley, and R. Rutenbar, "Efficient handling of operating range and manufacturing line variations in analog cell synthesis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 8, pp. 825–839, Aug. 2000.
- [20] G. Box and N. Draper, *Empirical Model-Building and Response Surfaces*. Hoboken, NJ: Wiley, 1987.
- [21] R. Myers and D. Montgomery, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. New York: Wiley Interscience, 2002.
- [22] W. Daems, G. Gielen, and W. Sansen, "Simulation-based automatic generation of signomial and posynomial performance models for analog integrated circuit sizing," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2001, pp. 70–74.
- [23] —, "A fitting approach to generate symbolic expressions for linear and nonlinear analog circuit performance characteristics," in *Proc. IEEE/ACM Des., Autom. Test Eur.*, 2002, pp. 268–273.
- [24] —, "An efficient optimization-based technique to generate posynomial performance models for analog integrated circuits," in *Proc. IEEE/ACM Des. Autom. Conf.*, 2002, pp. 431–436.
- [25] T. Eeckelaert, W. Daems, G. Gielen, and W. Sansen, "Generalized posynomial performance modeling," in *Proc. IEEE/ACM Des., Autom. Test Eur.*, 2003, pp. 250–255.
- [26] G. Golub and C. Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1996.
- [27] Y. Saad, *Iterative Methods for Sparse Linear Systems*. New York: PWS, 1996.
- [28] H. Graeb, S. Zizala, J. Eckmueller, and K. Antreich, "The sizing rules method for analog integrated circuit design," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2001, pp. 343–349.
- [29] G. Stehr, M. Pronath, F. Schenkel, H. Graeb, and K. Antreich, "Initial sizing of analog integrated circuits by centering within topology-given implicit specifications," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2003, pp. 241–246.
- [30] X. Li, J. Le, P. Gopalakrishnan, and L. Pileggi, "Asymptotic probability extraction for non-Normal distributions of circuit performance," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2004, pp. 2–9.
- [31] X. Li, J. Le, L. Pileggi, and A. Strojwas, "Projection-based performance modeling for inter/intra-die variations," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2005, pp. 721–727.
- [32] A. Hedayat, N. Sloane, and J. Stufken, *Orthogonal Arrays: Theory and Applications*. New York: Springer-Verlag, 1999.
- [33] G. Seber, *Multivariate Observations*. Hoboken, NJ: Wiley, 1984.
- [34] K. Low and S. Director, "An efficient methodology for building macro-models of IC fabrication processes," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 8, no. 12, pp. 1299–1313, Dec. 1989.



Xin Li (S'01–M'06) received the B.S. and M.S. degrees in electronics engineering from Fudan University, Shanghai, China, in 1998 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2005.

He was a summer Intern with Extreme DA, Palo Alto, CA, in 2004. Since the summer of 2005, he has been a Systems Scientist with the Department of Electrical and Computer Engineering, Carnegie Mellon University. His current research interests include modeling, simulation, and synthesis for analog/RF and digital systems.

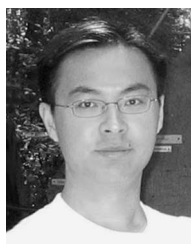
Dr. Li received the IEEE/ACM William J. McCalla International Conference on Computer Aided Design Best Paper Award in 2004.



Padmini Gopalakrishnan (S'98) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Madras, India, in 1997, and the M.S. degree in electrical and computer engineering from the University of Texas, Austin, in 1999. She is currently working toward the Ph.D. degree in electrical and computer engineering at Carnegie Mellon University, Pittsburgh, PA.

She spent three years with the R&D Team of Monterey Design Systems, Sunnyvale, CA, from 1999 to 2002. Her research interests span various aspects of VLSI design methodologies and CAD algorithms, including applications of graph theory and optimization theory.

Ms. Gopalakrishnan received the IEEE/ACM William J. McCalla International Conference on Computer Aided Design Best Paper Award in 2004.



Yang Xu (S'97–M'05) received the B.S. and M.S. degrees in electrical engineering from Fudan University, Shanghai, China, in 1997 and 2000, respectively, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2004.

He is currently with Qualcomm, Inc., working on RF receiver design. His research interests include the design and optimization of RF/analog and mixed-signal circuits and systems, circuit noise analysis and macromodeling, and digital media over wireless systems.



Lawrence T. Pileggi (S'85–M'89–SM'94–F'01) received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 1989.

He is the Tanoto Professor of electrical and computer engineering and the Director of the Center for Silicon System Implementation at Carnegie Mellon University. From 1984 through 1986, he was with Westinghouse Research and Development, where he was recognized with the corporation's highest engineering achievement award in 1986. From 1989 through 1995, he was a Faculty Member with the University of Texas, Austin. In January of 1996, he joined the faculty at Carnegie Mellon University. He has consulted for several electronic design automation (EDA) and semiconductor companies. He is a coauthor of *Electronic Circuit and System Simulation Methods* (McGraw-Hill, 1995) and *IC Interconnect Analysis* (Kluwer, 2002). He has published over 200 refereed conference and journal papers and holds 14 U.S. patents. His research interests include various aspects of digital and analog design and EDA.

Dr. Pileggi served as the Technical Program Chairman of the 2001 International Conference on Computer Aided Design (ICCAD) and as the Conference Chairman of the 2002 ICCAD. He received the Best CAD Transactions Paper Awards in 1991 and 1999. He received a Best Paper Award from the Design Automation Conference in 2003, and a Best Paper Award from the 2004 ICCAD. He received a Presidential Young Investigator Award from the National Science Foundation in 1991. In 1991 also, and again in 1999, he received the Semiconductor Research Corporation (SRC) Technical Excellence Award. In 1993, he received an Invention Award from the SRC and, subsequently, a U.S. patent for the RICE simulation tool. In 1994, he received the University of Texas Parent's Association Centennial Teaching Fellowship for excellence in undergraduate instruction. In 1995 and 2005, he received the Faculty Partnership Awards from IBM.