Robust Analog/RF Circuit Design with Projection-Based Posynomial Modeling

Xin Li, Padmini Gopalakrishnan, Yang Xu and Lawrence T. Pileggi
Dept. of ECE, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
{xinli, pgopalak, yangx, pileggi}@ece.cmu.edu

Abstract

In this paper we propose a RObust Analog Design tool (ROAD) for post-tuning analog/RF circuits. Starting from an initial design derived from hand analysis or analog circuit synthesis based on simplified models, ROAD extracts accurate posynomial performance models via transistor-level simulation and optimizes the circuit by geometric programming. Importantly, ROAD sets up all design constraints to include large-scale process variations to facilitate the tradeoff between yield and performance. A novel convex formulation of the robust design problem is utilized to improve the optimization efficiency and to produce a solution that is superior to other local tuning methods. In addition, a novel projection-based approach for posynomial fitting is used to facilitate scaling to large problem sizes. A new implicit power iteration algorithm is proposed to find the optimal projection space and extract the posynomial coefficients with robust convergence. The efficacy of ROAD is demonstrated on several circuit examples.

1. 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. Firstly, device models have become increasingly complex in order to characterize the physical behavior of transistors at high frequencies. Secondly, at these frequencies, parasitic couplings become more important and more complex. Finally, but perhaps most importantly, with sub-wavelength photo-lithography, 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.

To address this increasing difficulty of manual design, various approaches have been proposed for analog synthesis [1]-[8]. These methods take a fixed circuit topology as input and optimize the component sizes to meet design specifications. Advanced stochastic algorithms such as simulated annealing and genetic programming have been applied to search the entire design space for a globally optimal solution [2]-[4]. However, stochastic search algorithms can be very slow, especially when process variations are simultaneously considered [5].

Recently, it has been demonstrated that many analog circuit specifications can be cast into posynomial functions. As such, analog circuit sizing can be formulated as a geometric programming problem which guarantees that a globally optimal solution can be determined [6]-[8]. However, the traditional geometric programming approach requires the creation of the posynomial design equations by hand. Manually derived equations apply various simplifications and ignore many second order effects. In addition, the authors in [6]-[8] use corner enumeration to achieve robust analog design, where design equations are listed at all process corners. As such, the number of total design equations increases exponentially in the number of independent process parameters. Furthermore, it is not guaranteed that the worst-case design will occur at one of these process corners.

In addition to modeling inaccuracy, process variations have an increasingly significant impact on circuit performance, thereby posing additional challenges for analog synthesis. For example, given a circuit that is synthesized for nominal process parameters, substantial device sizing may be required to accommodate large-scale process variations and improve product yield..

In this paper, we propose a novel RObust Analog Design tool (ROAD) to post-tune analog/RF circuits based on accurate transistor-level simulation models and with consideration of large-scale process variations. 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 synthesis by traditional analog optimization algorithms [1]-[8]. Simplified device/coupling models can be utilized in this step to ease the manual design or speedup the automatic synthesis. 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 optimize the tradeoff between yield and performance.

Compared with other robust design approaches [5], [9]-[11], the novelty of ROAD lies in our convex formulation of the robust design problem, which improves the optimization efficiency and, more importantly, helps to find a better solution. We find that, even for a small analog/RF design space, the circuit performance, e.g. the cost function for optimization, is not convex. This non-convex property makes it difficult to apply traditional optimization algorithms, such as a gradient method with linear search, since all such algorithms assume a convex cost function and can otherwise become stuck at a locally optimal solution. On the other hand, although the stochastic algorithms such as simulated annealing and genetic programming are more robust and can find the global optimum, they are computationally expensive when using detailed circuit simulation models.

ROAD combines the posynomial modeling and geometric programming to overcome this non-convex difficulty with low computation cost. We find that in a sufficiently small local design space, accurate posynomial models can be fitted for many circuit specifications from transistor-level simulation data. Given an initial design, ROAD first applies statistical analysis and approximates the best/worst-case circuit performances as posynomial functions in the local design space. It then optimizes the circuit by geometric programming, which can be transformed to a convex optimization that is easy to solve. This fitting and optimization procedure is repeatedly applied on successively narrowed local design space. Since ROAD sets up all design constraints with process variations and the posynomial model becomes increasingly accurate in the successively narrowed local design space, ROAD can converge to an accurate design with high yield.

0-7803-8702-3/04/$20.00 ©2004 IEEE.

Another novelty of ROAD is the utilization of a projection-based approach for posynomial fitting. Compared with traditional posynomial fitting techniques [12]-[15], the proposed projection-based method is more efficient and handles large-size problems. While the traditional projection theory generally trades accuracy for simplicity in terms of the dimension of projection space, we find that the rank-one projection is especially meaningful for posynomial modeling. The rank-one projection provides sufficient accuracy for approximating circuit performance and, most importantly, theoretical analysis shows that quadratic posynomials are invariant (i.e. remain posynomials) under the rank-one projection. In addition, a novel implicit power iteration algorithm is developed to find the optimal projection space and extract the posynomial coefficients. This iteration consists of a sequence of convex quadratic programming steps and exhibits robust convergence.

The remainder of this paper is organized as follows. In Section 2 we propose and study the challenging problems for analog/RF post-tuning. Then we propose our ROAD approach, including statistical analysis, posynomial fitting and geometric programming in Section 3. The efficacy of ROAD is demonstrated by several circuit examples designed in the IBM BiCMOS 0.25 μm process in Section 4. Finally, we draw conclusions in Section 5.

2. Post-Tuning for Analog/RF Circuits

2.1 Non-convex Analog/RF Design Space

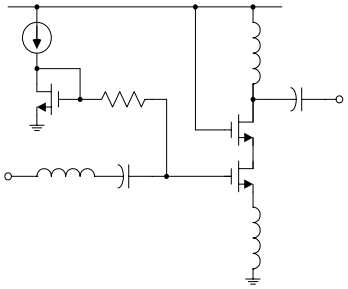

Fig. 1. Circuit schematic for LNA.

TABLE 1. Quadratic modeling result for LNA

Performance	Eigenvalue # of A		Eigenvalues of A	
	Negative	Positive	Minimal	Maximal
F0 (GHz)	4	8	-0.93	2.57
S11 (dB)	6	6	-150.29	660.77
S12 (dB)	6	6	-69.23	19.55
S21 (dB)	4	8	-11.57	66.16
S22 (dB)	7	5	-199.24	299.20
NF (dB)	4	8	-1.49	12.51
IIP3 (dBm)	8	4	-54.64	25.42
Power (mW)	6	6	-1.59	24.40

It is well known that the entire analog/RF design space is strongly nonlinear in general. In this paper, however, we find that the local analog/RF design space is not convex even for a small region. This non-convex property can be demonstrated by the following low noise amplifier example design in the IBM BiCMOS 0.25 μm process.

Shown in Fig. 1 is a low noise amplifier which includes 12 independent design variables and 8 design specifications. Given an initial design, we sample the local design space by a perturbation of ±5% on all design variables. The circuit performance in such a small local design space is fitted by the quadratic model:

$$f(x) = x^T \hat{A} x + \hat{B}^T x + \hat{C} \tag{1}$$

where $x = [x_1, \cdots, x_N]^T$ contains independent design variables, $\hat{A} \in R^{N \times N}$, $\hat{B} \in R^N$ and $\hat{C} \in R$ are model coefficients and N is the total number of design variables.

TABLE 1 shows the eigenvalue distribution of the quadratic coefficient matrix A for this LNA example. Note that, for most circuit performances, the A matrix includes both positive and negative eigenvalues with comparable magnitude. It, in turn, implies that the analog design space is *neither convex nor concave,* even for such a small/local perturbation region.

One important reason for the above observation is that many analog/RF circuit performances are expressed as the sum of the multiplications of design variables, e.g. $Gain = G_m \cdot Z_L$, $Area = W_1 \cdot L_1 + W_2 \cdot L_2 + \cdots$, etc. It is easy to verify that, for example, the function $f(x) = x_1 x_2$ is neither convex nor concave, since its corresponding quadratic coefficient matrix A has two eigenvalues $\lambda = \pm 0.5$. The multiplication of different design variables results in the cross product term $x_i x_j$ in the quadratic model (1), rendering a non-convex design space. Such a non-convex property of the analog/RF design space is called the "*xy cross product pattern*" or simply "*xy pattern*" in this paper.

The underlying xy pattern of analog/RF circuits makes it difficult to apply traditional optimization algorithms, such as a gradient method with linear search, to post-tuning, since all these algorithms assume a convex cost function and, therefore, is very likely to become stuck at a locally optimal solution. On the other hand, although the stochastic algorithms, such as simulated annealing and genetic programming, are much more robust in terms of finding the global optimum, they are computationally expensive. The above analysis, therefore, motivates us to utilize other more effective optimization algorithms, e.g. the geometric programming, for analog/RF post-tuning, which can handle the xy pattern with low computation cost.

2.2 Geometric Programming for Analog/RF Tuning

Let $x = [x_1, \cdots, x_N]^T$ be N real, positive variables. A function f is called *posynomial* if it has the form:

$$f(x) = \sum_{i=1} c_i x_1^{\alpha_{1i}} x_2^{\alpha_{2i}} \cdots x_N^{\alpha_{Ni}} \tag{2}$$

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.

It has been demonstrated that many analog circuit specifications can be cast into posynomial functions [6]-[8]. As such, analog circuit sizing can be formulated as a geometric programming problem:

$$\begin{aligned} &\textit{minimize} && f_0(x) \\ &\textit{subject to} && f_i(x) \le 1 \quad i = 1, \cdots, K \\ & && x_i > 0 \quad\;\; i = 1, \cdots, N \end{aligned} \tag{3}$$

where $f_0, f_1, \cdots, f_K$ are posynomial functions.

It is important to note that the xy pattern of analog/RF circuits can be easily handled by geometric programming, because the sum of the cross product terms is exactly a posynomial function. In fact, the posynomial functions $f_0(x), f_1(x), \cdots, f_K(x)$ in (3)

might not be convex. However, geometric programming defines a set of new variables $y_i = \log(x_i)$ $(i = 0,1,\cdots,N)$ and new functions $g_i = \log(f_i)$ $(i = 0,1,\cdots,K)$. The transformed cost/constraint functions $g_0(y), g_1(y), \cdots, g_K(y)$ are convex. In other words, instead of directly working on the original non-convex functions $f_i(x)$, geometric programming optimizes the transformed convex functions $g_i(y)$. That's the reason why geometric programming can be numerically robust and reach the global optimum while directly applying traditional algorithms, such as the gradient method with linear search, often gets stuck in the local minimum. For this reason, geometric programming is much better than the general nonlinear optimization algorithm for analog/RF post-tuning, since it can handle the xy pattern and reduce the probability of converging to local minimum.

Geometric programming has been previously applied for analog synthesis in [6]-[8]. However, the existing geometric programming approach optimizes analog/RF circuits based on simplified models, similar to hand analysis equations. Our objective is to start with the solution derived from the optimization based on such equations and further optimize the design for robust performance using detailed device/variation modeling information.

2.3 Quadratic Posynomial Modeling

A key operation in ROAD is to extract posynomial performance models based on transistor-level simulation. Instead of modeling the circuit performance as full posynomial functions (2), ROAD utilities quadratic posynomial models [12]-[15]. The quadratic posynomial performance models are much easier to fit, and, more importantly, already provide sufficient accuracy to cover the local/small design space for post-tuning.

Several techniques have been proposed for quadratic posynomial model generation [12]-[15]. Considering the tradeoff between modeling accuracy and computation cost, direct fitting [14] is the most efficient of these previous approaches. Direct fitting approximates the posynomial function by a quadratic form:

$$f(x) = X^T AX + B^T X + C \tag{4}$$

where $X = \left[x_1^{-1}, \cdots, x_N^{-1}, x_1, \cdots, x_N\right]^T$ includes independent design variables and N is the total number of design variables. $A \in R_+^{2N \times 2N}$, $B \in R_+^{2N}$ and $C \in R_+$ are unknown posynomial coefficients which can be determined by the optimization:

$$\begin{aligned} &\textit{minimize} \quad \psi(A,B,C) = \sum_i \left(X_i^T AX_i + B^T X_i + C - \hat{f}_i\right)^2 \\ &\textit{subject to} \quad A \in R_+^{2N \times 2N}, B \in R_+^{2N \times 1}, C \in R_+ \end{aligned} \tag{5}$$

In (5), X_i and $\hat{f}_i$ are the value of X and the exact value of the function f for the i-th simulated sample, respectively. The cost function (5) is a positive semi-definite quadratic function restricted to a convex constraint set [14]. Therefore, the optimization problem is convex and is guaranteed to have a globally optimal solution.

In addition, the authors in [14] propose a heuristic template estimation to reduce the computation cost. The template estimation algorithm first fits the data samples to the quadratic function in (1). Then, in the second step, dominant posynomial terms are selected based on the value of $\hat{A}$, $\hat{B}$ and $\hat{C}$, and only these dominant terms are put into the cost function in (5) for final optimization. Note that, since the matrices $\hat{A}$ and $\hat{B}$ in (1) have much smaller sizes than the matrices A and B in (5), template estimation reduces the computation cost significantly.

Even with the template estimation, however, the number of problem unknowns in (1) is $O(N^2)$. The computational complexity for solving all these unknowns is of the order of $O(N^6)$. This high computation cost limits previous approaches [12]-[15] to small or medium size applications. In ROAD, we propose a novel projection-based posynomial modeling algorithm, which can reduce the computation cost significantly.

3. ROAD Methodology

Our proposed robust analog design tool (ROAD) is facilitated by a combination of statistical analysis, posynomial modeling and geometric programming. In this section, we describe the key algorithms used in ROAD and highlight the novelty.

3.1 Projection-Based Posynomial Modeling

The key disadvantage of the posynomial modeling algorithm in [14] is the need to compute all elements of matrix A in (5) or matrix $\hat{A}$ in (1). These matrices are 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, $\|\bullet\|_F$ 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 [17].

From matrix theory [18], for any symmetric matrix $A \in R^{n \times n}$, the optimal rank-p approximation with the least Frobenius-norm error is:

$$A_L = \sum_{i=1}^{p} \lambda_i P_i P_i^T \tag{6}$$

where λ_i is the i-th dominant eigenvalue, and $P_i \in R^{n \times 1}$ is the i-th dominant eigenvector. The eigenvectors in (6) define an orthogonal projector $P_1 P_1^T + \cdots + P_p P_p^T$, and every column in A_L is the *projection* of every column in A onto the subspace $span\{P_1, \cdots, P_p\}$. We use this orthogonal projector for posynomial fitting in this paper.

The above discussion is applicable to any quadratic form, without restricting A and A_L to be nonnegative. For posynomial fitting, we need to further prove that A_L is nonnegative, i.e. $A_L \in R_+^{n \times n}$, if $A \in R_+^{n \times n}$. For that, we need an additional theorem from matrix analysis.

> **Perron-Frobenius theorem** [17]: Let A be a real nonnegative matrix, i.e. $A \in R_+^{n \times n}$. Then $\lambda_1 = \rho(A)$, 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 1st dominant eigenvalue λ_1 and eigenvector P_1 for

nonnegative $A \in R_+^{n\times n}$ are both nonnegative, the rank-one projection $A_L = \lambda_1 P_1 P_1^T$ is nonnegative. In other words, *quadratic posynomials are invariant (i.e. remain posynomials) under the rank-one projection*.

On the other hand, the eigenvectors $P_1, P_2, \cdots$ are mutually orthogonal for a symmetric matrix A. Since P_1 is nonnegative, $P_2, P_3, \cdots$ must contain non-positive 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 above discussion to map a posynomial to posynomial. The main advantage of such a rank-one projection is that, for approximating the matrix $A \in R_+^{2N\times 2N}$ in (4), only λ_1 and $P_1 \in R_+^{2N\times 1}$ need to be solved, thus reducing the number of problem unknowns to $O(N)$. Compared with the problem size $O(N^2)$ in previous approaches [12]-[15], our proposed projection-based method is more efficient and can be applied to large-size problems. In addition, we find that the rank-one projection achieves high accuracy in many practical applications, which will be demonstrated by numerical examples in Section 4.

3.2 Coefficient Fitting via Implicit Power Iteration

1. Start from an initial vector $Q_0 \in R_+^{2N\times 1}$ and set $k = 1$.
2. Compute $Q_{k-1} = Q_{k-1} / \|Q_{k-1}\|_F$.
3. Solve the convex quadratic programming

$$\textit{minimize} \quad \psi_k(Q_k, B_k, C_k) = \sum_i \left(X_i^T Q_k Q_{k-1}^T X_i + B_k^T X_i + C_k - \hat{f}_i\right)^2$$

$$\textit{subject to} \quad Q_k \in R_+^{2N\times 1}, B_k \in R_+^{2N\times 1}, C_k \in R_+$$

4. If the minimized cost function is unchanged, i.e.

$$\left|\psi_k^{\min}(Q_k, B_k, C_k) - \psi_{k-1}^{\min}(Q_{k-1}, B_{k-1}, C_{k-1})\right| < \varepsilon$$

where ε is the pre-defined error tolerance, then go to Step 5. Otherwise, $k = k+1$ and return Step 2.

5. The approximated posynomial function is

$$f(x) = X^T Q_k Q_{k-1}^T X + B_k^T X + C_k$$

Fig. 2. Implicit power iteration algorithm.

Since matrix $A \in R_+^{2N\times 2N}$ in (4) is not known in advance, we cannot use the standard matrix computation algorithm to compute its dominant eigenvalue λ_1 and eigenvector P_1. One approach is to solve the following optimization problem:

$$\textit{minimize} \quad \psi(Q, B, C) = \sum_i \left(X_i^T Q Q^T X_i + B^T X_i + C - \hat{f}_i\right)^2 \quad (7)$$

$$\textit{subject to} \quad Q \in R_+^{2N\times 1}, B \in R_+^{2N\times 1}, C \in R_+$$

Compared with (4), equation (7) approximates matrix A by QQ^T. Since the optimal rank-one projection is $A_L = \lambda_1 P_1 P_1^T$ with $\lambda_1 \geq 0$ and $P_1 \geq 0$, we expect the cost function $\psi(Q, B, C)$ in (7) to have the global optimum $Q = \sqrt{\lambda_1} P_1$. Unfortunately, $\psi(Q, B, C)$ is a 4th order function of Q and, therefore, might not be convex. There is no efficient optimization algorithm that can be guaranteed to find the globally optimal solution for $\psi(Q, B, C)$.

Instead of using the 4th order cost function in (7), we propose a novel implicit power iteration method to efficiently extract the unknown coefficients λ_1 and P_1. This implicit power iteration consists of a sequence of convex quadratic programming steps and exhibits robust convergence with an arbitrary initial point. An outline of this algorithm is shown in Fig. 2.

In Fig. 2, the cost function $\psi_k(Q_k, B_k, C_k)$ in Step 3 is a positive semi-definite quadratic function over a convex constraint set. Therefore, the quadratic programming in Step 3 is convex and is guaranteed to converge to its global optimum.

Next, we explain why the overall implicit power iteration yields the optimal rank-one projection $A_L = \lambda_1 P_1 P_1^T$. Note that, Step 3 in Fig. 2 approximates 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 over-determined linear equation:

$$Q_k Q_{k-1}^T = A \quad (8)$$

The least-square-error solution for equation (8) is given by [18]:

$$Q_k = A Q_{k-1} \cdot \left(Q_{k-1}^T Q_{k-1}\right)^{-1} = A Q_{k-1} \quad (9)$$

In (9), $Q_{k-1}^T Q_{k-1} = \|Q_{k-1}\|_F^2 = 1$, since Q_{k-1} is normalized in Step 2 in Fig. 2. Equation (9) tells us the interesting fact that the quadratic programming in Step 3 "implicitly" computes the matrix-vector product AQ_{k-1}, which is the basic operation required in the power iteration for dominant eigenvector computation [18].

Given the initial vector:

$$Q_0 = \alpha_1 P_1 + \alpha_2 P_2 + \cdots \quad (10)$$

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

$$Q_k = A^k Q_0 = \alpha_1 \lambda_1^k P_1 + \alpha_2 \lambda_2^k P_2 + \cdots \quad (11)$$

In (11), we ignore the normalization $Q_{k-1} = Q_{k-1} / \|Q_{k-1}\|_F$ which is nothing else but a scaling factor. This scaling factor will not change the direction of Q_k. Since $Q_0 > 0$ (Step 1 in Fig. 2) and $P_1 \geq 0$ (based on the Perron-Frobenius theorem), Q_0 is not orthogonal to P_1, i.e. $\alpha_1 \neq 0$ in (10). Therefore, $\alpha_1 \lambda_1^k P_1$ (with $|\lambda_1| > |\lambda_2| > \cdots$) will become more and more dominant over other terms. Q_k will asymptotically approach the direction of P_1, as shown in Fig. 3.

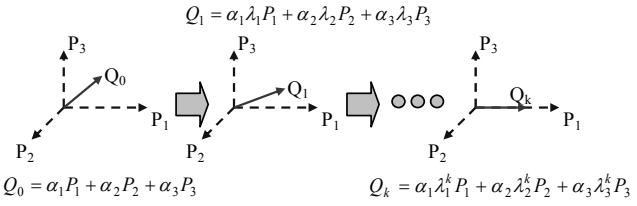

Fig. 3. Convergence of power iteration in 3-D space.

After the iteration in Fig. 2 converges, we have $Q_{k-1} = Q_{k-1} / \|Q_{k-1}\|_F = P_1$ and $Q_k = AQ_{k-1} = \lambda_1 P_1$. $Q_k Q_{k-1}^T$ is the optimal rank-one projection $A_L = \lambda_1 P_1 P_1^T$. 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 [18].

The above discussion demonstrates that the implicit power iteration is provably convergent if A is symmetric. For an asymmetric A, we can show that Q_{k-1} and Q_k converge to the directions of the dominant left and right singular vectors of A. However, the global convergence is difficult to prove in that case.

3.3 Robust Design with Process Variations

Given a circuit topology, the circuit performance (e.g. gain, bandwidth) can be specified in terms of design variables (e.g. bias current, transistor sizes) and process parameters (e.g. V_{TH}, T_{OX}). Due to process variations, the process parameters must be modeled as random variables. Therefore, the circuit performance $f(x)$ with fixed design variables $x=[x_1, x_2, \cdots, x_N]$ is also a random 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: the mean value $f_M(x)$, the lower bound $f_{LOW}(x)$ and the upper bound $f_{UP}(x)$, as shown in Fig. 4. The values of these three metrics $f_M(x)$, $f_{LOW}(x)$ and $f_{UP}(x)$ can be computed by the APEX algorithm [19].

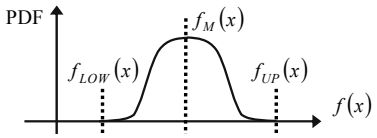

Fig. 4. Probability density function of performance $f(x)$.

ROAD classifies all design specifications into two categories: equality constraints (e.g. F0 = 2.14GHz) and inequality constraints (e.g. IIP3 ≥ 5dBm and NF ≤ 2dB). In order to achieve robust design, ROAD incorporates the process variations into the cost function or constraints during optimization. For example, the design specification $f(x)=Spec$ is translated to $f_M(x)/Spec \le 1$, $Spec/f_M(x) \le 1$ and $minimize[f_{UP}(x)-f_{LOW}(x)]$. In other words, we force the mean value of $f(x)$ to equal *Spec* and minimize the variance of $f(x)$ under process variations. TABLE 2 summarizes all the geometric programming constraints utilized in ROAD.

TABLE 2. ROAD formulation for geometric programming*

Category	Specification	Cost Function/Constraint
Equality	$f(x)=Spec$	$f_M(x)/Spec \le 1$ $Spec/f_M(x) \le 1$ $minimize[f_{UP}(x)-f_{LOW}(x)]$
Inequality	$f(x) \le Spec$	$f_{UP}(x)/Spec \le 1$
	$f(x) \ge Spec$	$Spec/f_{LOW}(x) \le 1$

ROAD first approximates $f_M(x)$, $1/f_M(x)$, $f_{UP}(x)-f_{LOW}(x)$, etc. as posynomial functions, and then optimizes design variables $x=[x_1, x_2, \cdots, x_N]$ via geometric programming. Note that such an optimization will converge to a robust design with high yield, since the optimization constraints in TABLE 2 are set up for both nominal conditions and process variations.

3.4 ROAD Iteration Scheme

1. Start from an initial design D_0 and perturbation size ±ε%. Set iteration index $k=1$.
2. Make a perturbation of ±ε% on all design variables of D_{k-1}. This perturbation defines a local design space S_k for k-th post-tuning. Generate a set of samples $\{x\}_k$ in S_k.
3. For every sampling point in $\{x\}_k$, run transistor-level simulation and apply the APEX algorithm [19] to compute $f_M(x)$, $f_{LOW}(x)$ and $f_{UP}(x)$.
4. Fit the cost function and constraints (e.g. $f_M(x)$, $1/f_M(x)$, $f_{UP}(x)-f_{LOW}(x)$, etc.) as posynomial functions using the proposed projection-based algorithm.
5. Run geometric programming and find the optimal design D_k in the design space S_k.
6. If the difference between D_k and D_{k-1} is smaller than a pre-defined error tolerance, then go Step 7. Otherwise, $k=k+1$ and return Step 2.
7. If ε% is smaller than the pre-defined minimal design space size, then stop iteration. Otherwise, $\varepsilon=\varepsilon/2$, $k=k+1$ and return Step 2.

Fig. 5. Outline of the overall ROAD iteration scheme.

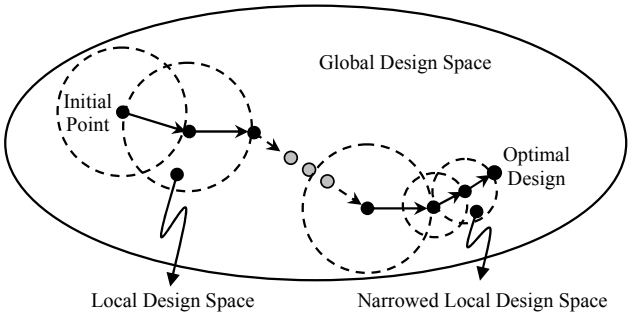

Fig. 6. Illustration of ROAD iterations.

As summarized in Fig. 5 and depicted in Fig. 6, starting from an initial design, ROAD iteratively improves the circuit performance and product yield by successive posynomial fitting and geometric programming. In each iteration, ROAD fits posynomial models in a local design space to approximate the worst/best-case circuit performance with consideration of process variations. Then, geometric programming is utilized to find the optimal design in that local design space.

The overall iteration is performed in two levels: inner loop iteration and outer loop iteration. The inner loop iteration searches the optimal solution in the local design spaces that have the same size but are centered at various expansion points. During the k-th iteration, the k-th local design space is defined by a perturbation of ±ε% on all design variables of the previous iteration result D_{k-1}. Then, after the optimal design is found under the current perturbation size ±ε%, ROAD reduces ε by a factor of two and repeats the inner loop iteration for a finer-grained search. The iteration is stopped if no further improvement is identified

* TABLE 2 only considers positive *Spec* values. Negative *Spec* values can be normalized to positive ones through a proper scaling [14].

between two successive steps and the current ε% is smaller than the pre-defined minimal design space size. Since the posynomial 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 above ROAD iteration is mainly determined by the final value of ε%, i.e. the pre-defined minimal design space size in Step 7 of Fig. 5. This final ε% significantly impacts the posynomial modeling accuracy at the end of the iteration. Higher design accuracy can be achieved, if a smaller final ε% is utilized. On the other hand, the final ROAD design is not sensitive to the initial value of ε%. In many applications, the initial and final ε% can be typically selected as 5%~10% and 1%~5% respectively.

4. Numerical Examples

We demonstrate the ROAD flow as applied to circuit examples designed in the IBM BiCMOS 0.25 µm process. For each example 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 product yield. All the numerical experiments are performed on a SUN Sparc – 1GHz server.

4.1 Low Noise Amplifier

Shown in Fig. 1 is a low noise amplifier which includes 12 independent design variables and 8 design specifications. In each post-tuning iteration we sample the local design space by a perturbation of $\pm\varepsilon$% on all design variables. ε% is initially set to 5% and then successively reduced to 1% during iteration. Two independent sampling sets, called training set and testing set respectively, are generated. The training set is created by orthogonal arrays [20], which pick up the most important samples based on statistical analysis; this is used for posynomial coefficient fitting. For testing and comparison, we collect 500 random samples as the testing set and use them to verify the modeling accuracy.

A. Robust Convergence of Implicit Power Iteration

To test the convergence of the proposed implicit power iteration algorithm we pick 100 random initial vectors $Q_0 > 0$ and use them for running power iteration in posynomial fitting. We find that all 100 experiments reliably converge without a single failure.

B. Effect of Training Set Size

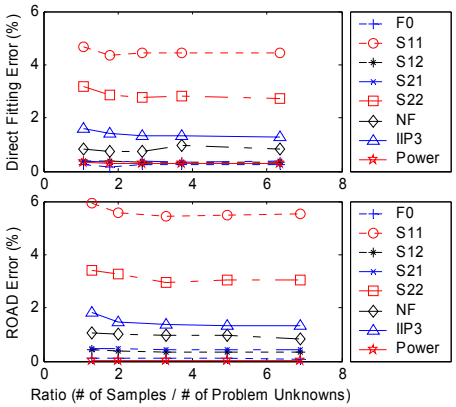

Fig. 7. Effect of training set size.

For both direct fitting [14] and ROAD, Fig. 7 shows the relation between posynomial modeling error and training set size, where the perturbation ε% is set to 5%. Studying Fig. 7 we find that the number of samples in the training set should be 2~3 times greater than the number of problem unknowns. Further increasing the number of samples doesn't have a significant impact on reducing the modeling error. This observation implies that the required number of samples depends on the number of problem unknowns. As the posynomial coefficient number is reduced in ROAD, we not only decrease the computation time for coefficient optimization, but also save a large portion of circuit simulation cost because of the smaller training set.

C. Modeling Accuracy and Cost

TABLE 3 and TABLE 4 compare the accuracy of three modeling approaches. Direct fitting without template estimation [14] is the most accurate since it takes into account all possible posynomial product terms in (4). However, although both the template estimation method and the proposed ROAD approach apply simplifications to reduce the computation cost, their fitting accuracy is still comparable to direct fitting without template estimation.

In addition, comparing TABLE 3 and TABLE 4, one would find that, as the perturbation $\pm\varepsilon$% is reduced from ±5% to ±1%, the posynomial modeling accuracy is greatly improved. The modeling error is less than 0.3% for all circuit performance in TABLE 4, which guarantees high design accuracy at the end of ROAD iteration. It should be noted that achieving small modeling error is extremely important for robust analog design. The process variations in today's IC technologies typically introduce 10%~20% variations on circuit performance. If the modeling error is not sufficiently less than this value, the circuit yield cannot be accurately estimated and optimized via these models.

TABLE 3. Posynomial modeling error for LNA (ε% = 5%)

Performance	Direct Fitting w/o Template	Direct Fitting with Template [14]	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 4. Posynomial modeling error for LNA (ε% = 1%)

Performance	Direct Fitting w/o Template	Direct Fitting with Template [14]	ROAD
F0	0.021%	0.021%	0.021%
S11	0.189%	0.208%	0.273%
S12	0.020%	0.020%	0.020%
S21	0.010%	0.012%	0.013%
S22	0.249%	0.255%	0.257%
NF	0.031%	0.033%	0.037%
IIP3	0.063%	0.064%	0.064%
Power	0.001%	0.001%	0.001%

D. Robust Design

With the pre-extracted posynomial performance models, the geometric programming is solved efficiently, taking 1~2 seconds for this LNA example. TABLE 5 shows the circuit performance

simulated by Spectre before and after the post-tuning. The initial manual design contains a couple of circuit performance metrics that don't satisfy the design specifications. We apply the transitional sequential quadratic programming (SQP) and our proposed ROAD algorithm for nominal optimization, i.e. without considering process variations, respectively. It is shown in TABLE 5 that the traditional SQP approach is stuck at the local minimum while ROAD yields better circuit performance, especially much smaller power consumption. However, we also observe that the nominal ROAD design includes several performances (S21 and IIP3 as shown by the grey entries in TABLE 5) sitting on the boundaries of the design specifications.

When the process variation information is included for robust design, however, the ROAD sizing is done so as to leave sufficient margin for each performance metric. These margins enable the circuit to meet design specifications under process variations. The efficacy of the robust design is further illustrated by Fig. 8, where Monte Carlo simulations are applied to plot the S21 distribution for both nominal and robust designs. Fig. 8 shows that the robust design produced by ROAD satisfies the specification for a much larger fraction of the process variations than the nominal design.

TABLE 5. Circuit performance and synthesis cost for LNA

	Spec	Initial	Nominal		Robust
			SQP	ROAD	ROAD
F0 (GHz)	= 2.1	2.07	1.87	2.10	2.10
S11 (dB)	≤ -10	-13.98	-8.64	-12.92	-11.92
S12 (dB)	≤ -25	-23.89	-26.15	-26.23	-26.26
S21 (dB)	≥ 15	14.60	14.63	15.00	15.46
S22 (dB)	≤ -10	-14.47	-10.07	-24.37	-17.27
NF (dB)	≤ 1.5	1.19	1.10	1.30	1.42
IIP3 (dBm)	≥ 5	4.97	5.80	4.99	5.51
Power (mW)	Min	13.23	11.46	7.47	9.34
Cost (Simulation #)	—	—	122	1980	2640

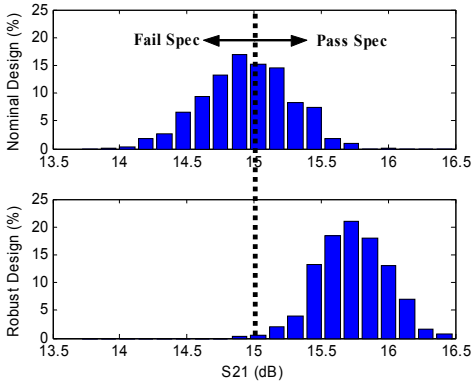

Fig. 8. Probability distribution for S21.

4.2 Scaling with Problem Size

Fig. 9 shows a two-stage folded-cascode Op Amp which includes 34 independent design variables and 8 design specifications. In each post-tuning iteration we sample the local design space by a perturbation of ±ε% on all design variables. Similar as the LNA example, ε% is initially set to 5% and it is successively reduced to 1% during iteration.

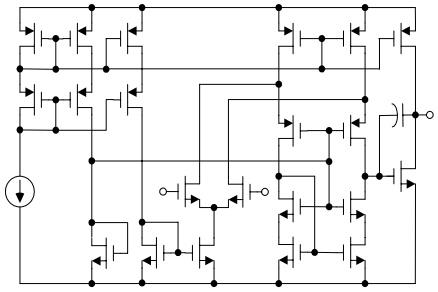
Fig. 9. Circuit schematic for Op Amp.

A. Modeling Accuracy and Cost

TABLE 6 and TABLE 7 summarize the modeling accuracy for both direct fitting with template estimation [14] and ROAD. Direct fitting without template estimation [14] includes 2381 unknown coefficients and, therefore, is computationally infeasible. From the data in TABLE 6 and TABLE 7, template estimation and ROAD have similar modeling accuracy. However, as shown in TABLE 8, ROAD is 4x faster for both Spectre simulation and coefficient fitting. We expect that as the problem size increases further, ROAD would achieve more speedup over the traditional methods.

In addition, it should be noted that the maximal ROAD modeling error is no more than 0.4% in TABLE 4, when the perturbation ±ε% is reduced to ±1%. This small error implies high design accuracy at the end of the ROAD iteration.

TABLE 6. Posynomial modeling error for Op Amp (ε% = 5%)

Performance	Template [14]	ROAD
Gain	0.11%	0.13%
UGF	0.09%	0.07%
Gain Margin	0.28%	0.16%
Phase Margin	0.05%	0.05%
Slew Rate	0.04%	0.05%
Swing	1.66%	1.92%
Power	0.15%	0.20%

TABLE 7. Posynomial modeling error for Op Amp (ε% = 1%)

Performance	Template [14]	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	0.402%	0.388%
Power	0.007%	0.009%

TABLE 8. Posynomial modeling cost for Op Amp (ε% = 5%)

	Template [14]	ROAD
Problem Unknown #	630	137
Sample # in Training Set	1458	343
Spectre Simulation (Sec.)	1.01×10^5	2.33×10^4
Coefficient Fitting (Sec.)	235.29	66.59

B. Robust Design

TABLE 9 shows the Spectre-simulated Op Amp performance before and after post-tuning. In TABLE 9, the initial circuit is over-designed, resulting in large power consumption. The nominal design optimized by the traditional sequential quadratic programming (SQP) is stuck at the local minimum, for which

several performances (Gain, Phase Margin and Slew Rate) still fail the design specifications. The nominal ROAD optimization yields a much better design that meets all design specifications, but it over-optimizes the circuit with several performances (UGF, Phase Margin and Slew Rate) sitting on the boundaries of the design specifications. Finally, the robust ROAD optimization takes into account the process variations and, therefore, leaves sufficient margin for each circuit performance metric during the optimization. Compared with the initial design, the robust design generated by ROAD achieves about 3x power reduction.

TABLE 9. Circuit performance and synthesis cost for Op Amp

	Spec	Initial	Nominal		Robust
			SQP	ROAD	ROAD
Gain (dB)	≥ 100	116.2	92.59	106.6	102.7
UGF (MHz)	≥ 10	11.76	10.33	9.96	10.94
Gain Margin	≥ 2	7.45	6.13	5.86	7.27
Phase Margin	≥ 60	65.82	58.46	59.64	63.48
Slew Rate (V/μs)	≥ 20	46.36	14.29	19.96	20.53
Swing (V)	≥ 0.5	1.00	1.00	1.00	1.00
Power (mW)	Min	2.54	0.53	0.73	0.79
Cost (Simulation #)	–	–	5034	5186	8038

5. Conclusions

In this paper, we propose a novel tool, ROAD, to post-tune analog/RF circuits quickly and accurately. ROAD utilizes a projection-based approach to generate accurate posynomial performance models from transistor-level simulation. In addition, it uses a novel implicit power iteration algorithm to extract the posynomial coefficients with robust convergence. Compared with previous posynomial modeling methods, ROAD achieves significant runtime speedup and scales well with problem size.

The novelty of ROAD lies in our convex formulation of the analog sizing problem and our inclusion of statistical analysis to incorporate process variations during optimization. Unlike the nominal design which might over-optimize 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 process variations.

Finally, it is important to mention that the projection-based approach proposed in this paper is not limited to posynomial modeling for analog synthesis. The same idea can be applied to quadratic polynomial or signomial modeling in many other engineering applications.

6. Acknowledgements

This work was funded in part by the MARCO Focus Center for Circuit & System Solutions (C2S2, www.c2s2.org) under contract 2003-CT-888.

7. References

[1] G. Gielen and R. Rutenbar, "Computer-aided design of analog and mixed-signal integrated circuits," *Proceedings of THE IEEE*, vol. 88, no. 12, pp. 1825-1852, Dec. 2000.

[2] M. Krasnicki, R. Phelps, R. Rutenbar and L. Carley, "MAELSTROM: efficient simulation-based synthesis for custom analog cells," *IEEE DAC*, pp. 945-950, 1999.

[3] R. Phelps, M. Krasnicki, R. Rutenbar, L. Carley and J. Hellums, "Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search," *IEEE Trans. CAD*, vol. 19, no. 6, pp. 703-717, Jun. 2000.

[4] 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," *IEEE ICCAD*, pp. 350-357, 2001.

[5] T. Mukherjee, L. Carley and R. Rutenbar, "Efficient handling of operating range and manufacturing line variations in analog cell synthesis," *IEEE Trans. CAD*, vol. 19, no. 8, pp. 825-839, Aug. 2000.

[6] M. Hershenson, S. Mohan, S. Boyd and T. Lee, "Optimization of inductor circuits via geometric programming," *IEEE DAC*, pp. 994-998, 1999.

[7] M. Hershenson, A. Hajimiri, S. Mohan, S. Boyd and T. Lee, "Design and optimization of LC oscillators," *IEEE ICCAD*, pp. 65-69, 1999.

[8] M. Hershenson, S. Boyd and T. Lee, "Optimal design of a CMOS Op-Amp via geometric programming," *IEEE Trans. CAD*, vol. 20, no. 1, pp. 1-21, Jan. 2001.

[9] S. Director, P. Feldmann and K. Krishna, "Statistical integrated circuit design," *IEEE JSSC*, vol. 28, no. 3, pp. 193-202, Mar. 1993.

[10] A. Dharchoudhury and S. Kang, "Worst-case analysis and optimization of VLSI circuit performances," *IEEE Trans. CAD*, vol. 14, no. 4, pp. 481-492, Apr. 1995.

[11] F. Schenkel, M. Pronath, S. Zizala, R. Schwencker, H. Graeb and K. Antreich, "Mismatch analysis and direct yield optimization by spec-wise linearation and feasibiliity-guided search," *IEEE DAC*, pp. 858-863, 2001.

[12] W. Daems, G. Gielen and W. Sansen, "Simulation-based automatic generation of signomial and posynomial performance models for analog integrated circuit sizing," *IEEE ICCAD*, pp. 70-74, 2001.

[13] W. Daems, G. Gielen and W. Sansen, "A fitting approach to generate synbolic expressions for linear and nonlinear analog circuit performance characteristics," *IEEE DATE*, pp. 268-273, 2002.

[14] W. Daems, G. Gielen and W. Sansen, "An efficient optimization-based technique to generate posynomial performance models for analog integrated circuits," *IEEE DAC*, pp. 431-436, 2002.

[15] T. Eeckelaert, W. Daems, G. Gielen and W. Sansen, "Generalized posynomial performance modeling," *IEEE DATE*, pp. 250-255, 2003.

[16] D. Bertseksa, *Nonlinear Programming*, Athena Scientific, 1999.

[17] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, 1996.

[18] G. Golub and C. Loan, *Matrix Computations*, The Johns Hopkins Univ. Press, 1996.

[19] X. Li, J. Le, P. Gopalakrishnan and L. Pileggi, "Asymptotic probability extraction for non-Normal distributions of circuit performance," *IEEE ICCAD*, 2004.

[20] A. Hedayat, N. Sloane and J. Stufken, *Orthogonal Arrays: Theory and Applications*, New York: Springer-Verlag, 1999.