Journal of Circuits, Systems, and Computers
Vol. 28, No. 5 (2019) 1950073 (19 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0218126619500737

A Variability-Aware Robust Design Methodology for Integrated Circuits by Geometric Programming[*]

Yi Zhang[†], Junlong Zhou[‡], Li Chen[§] and Jin Sun[¶]

School of Computer Science and Engineering,
Nanjing University of Science and Technology,
200 Xiaolingwei Street, Nanjing,
Jiangsu 210094, P. R. China
[†]*yzhang@njust.edu.cn*
[‡]*jlzhou@njust.edu.cn*
[§]*915106840102@njust.edu.cn*
[¶]*sunj@njust.edu.cn*

Received 3 February 2018
Accepted 6 June 2018
Published 18 July 2018

Process variations have continuously posed significant challenges to the performance and yield of integrated circuits (ICs). The performance modeling and robust optimization method considering process variations has become an important research task in today's IC design. Aiming at solving the problems of strong nonlinearity and high-dimensional problems in circuit design, this paper proposes a general robust optimization method for ICs by geometric programming. This method first employs regularization sparse models to model a specific performance metric as a posynomial function in terms of design parameters, in order to reduce parameter space dimensionality and to accurately capture the nonlinear relationship between performance perturbations and process variations. Based on the posynomial performance models, this method further uses an uncertainty set to represent the uncertainties of process variations, and formulates the problem of robust optimization under process variations as a general geometric programming model that can be efficiently solved. Experimental results demonstrate that, the proposed method not only enhances the accuracy and efficiency of circuit performance modeling, but also improves the performance yield significantly compared with traditional circuit design methods.

Keywords: Integrated circuits; process variations; geometric programming; robust optimization; uncertainty set.

1. Introduction

The continuous technology scaling has led to the increasing complexity and uncontrollability in the manufacturing process of integrated circuits (ICs), which in

[*] This paper was recommended by Regional Editor Tongquan Wei.
[¶] Corresponding author.

turn causes an increase in the mismatch between prediction models and hardware fabrication.[1,2] Manufacturing variability resulting from random and/or systematic phenomena exerts a considerable influence on circuit performance, and makes it more challenging to characterize parameter variability and ensure robust designs.[3,4] For ICs in sub-65 nm technology, the variability in process parameters has become the dominate factor for parametric yield loss,[5,6] Therefore, it is of great importance to develop stochastic modeling and robust design techniques in the presence of process variations, for the purpose of improving circuit performance and robustness.

In most existing variability-aware design methods, the process variations are in general modeled as random variables. As a consequence, circuit performance that is dependent on associated design parameters may have substantial fluctuations around its nominal metric. With the continuously shrinking feature sizes of CMOS devices, the uncertainty in circuit performance propagating from process variations is more prone to cause violations of performance constraints.[7] As pointed out in the literature, the strong nonlinearity in circuit performance and the high dimensionality of parameter space are two emerging trends in advanced IC technologies.[8] On one hand, many response surface methodology (RSM)-based techniques employ linear RSM models that are not sufficiently accurate to capture the dependency of performance metric upon design parameters.[9–11] On the other hand, a variety of simulation-based approaches use polynomial fitting[12,13] or moment matching[14,15] algorithms to generate nonlinear performance models, which would inevitably lead to heavy computation burden under the circumstance of high-dimensional parameters. Moreover, these methods tend to cause the problem of over-fitting, resulting in less reliable prediction models. These two issues make it an extremely complicated and difficult task to explore design solutions that can achieve optimized circuit performance with a guaranteed robustness to process variations.

To overcome the above-mentioned difficulties in IC design, this paper proposes a general robust optimization method in the presence of process variations by geometric programming (GP).[16,17] To facilitate GP formulation, the proposed method proposes modeling a specific performance metric as a posynomial function in terms of design parameters, which can accurately capture the nonlinear relationship between performance perturbations and process variations. We further employ a sparse representation method[18,19] to reduce the dimensionality of performance model coefficients and to avoid the over-fitting problem. An iterative algorithm is developed to efficiently determine the sparse coefficients in posynomial performance models. Based on the posynomial performance model with sparse coefficients, this method further uses an uncertainty set to represent the uncertainties of process variations, and formulates the problem of robust optimization under process variations as a general GP framework that can be efficiently solved. Experimental results demonstrate that, the proposed method not only enhances the accuracy and efficiency of circuit performance modeling, but also improves the performance yield significantly compared with traditional circuit design methods.

The rest of this paper is organized as follows. Section 2 introduces the posynomial performance modeling approach based on a sparse regularization representation. Section 3 details the proposed GP framework for robust circuit design by using posynomial performance functions and an uncertainty set. Experimental results are demonstrated in Sec. 5. Finally, Sec. 6 concludes the whole paper.

2. Posynomial Performance Modeling

The transistor-level optimization of nanoscale circuits is an important research task at early design stage. This design task is generally formulated as a convex optimization problem that guarantees the global optimum of design objective.[20] For this reason, simulation data are popularly fitted into analytical forms of posynomials to satisfy the convexity requirement.[21,22] This section first introduces the fundamental concepts of posynomial function, and further presents the regularization-based method for generating accurate posynomial performance functions.

2.1. *Posynomial function*

In circuit design, it is necessary to take into account the process parameters of each CMOS device, including effective channel length (L_{eff}), threshold voltage (V_{th}) and oxide thickness (T_{ox}), etc. We use a parameter vector $X = [x_1, x_2, \ldots, x_n]$ to represent all associated design parameters. Following traditional RSM method, circuit performance can be modeled as a linear combination of selected basis functions dependent on process parameters:

$$y(X) = \sum_{m=1}^{M} \alpha_m \cdot g_m(X)\,, \tag{1}$$

where $y(X)$ denotes a specific performance metric, e.g., gain, bandwidth and phase margin, $\{g_m(X); m = 1, 2, \ldots, M\}$ are a set of basis functions in terms of design parameters, and α_m are model coefficients that indicate the corresponding basis functions' importances in performance model.

As reported in plenty of research studies, many circuit performance specification can be cast into posynomial functions.[23,24] Posynomial is capable of accurately capturing the strong nonlinearity in circuit performance. More importantly, with posynomial performance models, circuit design can be formulated as a GP problem which is convex and therefore has the global optimum.[16] An optimization problem in which the objective and constraint functions are all posynomials is called a GP problem. A posynomial is mathematically defined as follows:

$$f_{\text{posy}}(X) = \sum_{d=1}^{D} f_{\text{mono,d}}(X) = \sum_{d=1}^{D} c_d \cdot x_1^{a_{1d}} x_2^{a_{2d}} \cdots x_n^{a_{nd}}\,, \tag{2}$$

where $\{f_{\text{mono,d}}; d = 1, 2, \ldots, D\}$ are a set of momonial functions, and $c_d \geq 0$ are nonnegative multiplicative coefficients. It is obvious that a posynomial is a nonnegative linear combination of monomials. Referring to Eq. (1), if we select a set of monomials as the basis functions in performance model (1) and impose nonnegative constrains upon model coefficients α_m's, a performance model in posynomial form can be generated by solving the undetermined coefficients.

2.2. *Regularization-based posynomial modeling*

This section details the regularization-based method for generating sparse posynomial performance models. In general, the model coefficients in Eq. (1) can be determined by solving the following set of equations at K sampling points:

$$y^{(k)} = \sum_{m=1}^{M} \alpha_m \cdot g_m(X^{(k)})\,, \tag{3}$$

where $X^{(k)}$ and $y^{(k)}$ stand for the design parameters and the corresponding performance metric at kth sampling point, respectively. Traditional RSM method requires a large number of samples in case of high-dimensional data, and is prone to cause over-fitting problem that degrades model quality. Considering that circuit performance usually is dominated by only a few basis functions, i.e., a majority of model coefficients can be compressed to zero, in this work we use a ℓ_1-norm regularization scheme to explore the sparsity of undetermined coefficients. It is worth emphasizing that, theoretically ℓ_0-norm is optimal for sparsity regularization, however, ℓ_0-regularization has been demonstrated to be NP-hard,[25] and ℓ_1-regularization is therefore mostly used as an alternative.

To be specific, assume that $Y = (y^{(1)}, y^{(2)}, \ldots, y^{(K)})$ represent the sample vector consisting of performance metrics at K sampling points $X^{(1)}, X^{(2)}, \ldots, X^{(K)}$, and H represent the sample matrix of all monomial basis functions:

$$H = \begin{bmatrix} h_1(X^{(1)}) & h_2(X^{(1)}) & \cdots & h_M(X^{(1)}) \\ h_1(X^{(2)}) & h_2(X^{(2)}) & \cdots & h_M(X^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(X^{(K)}) & h_2(X^{(K)}) & \cdots & h_M(X^{(K)}) \end{bmatrix}.$$

Based on ℓ_1-regularization scheme, the sparse representation of the fitting problem in Eq. (3) can be formulated into the following constrained optimization problem:

$$\begin{aligned} \text{minimize}: \quad & \|Y - H\alpha\|_2^2 + \lambda\|\alpha\|_1 \\ \text{subject to}: \quad & \alpha_m \geq 0 \; m = 1, 2, \ldots, M \\ \text{variables}: \quad & \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_M)\,, \end{aligned} \tag{4}$$

where $\|\cdot\|_2$ and $\|\cdot\|_1$ denote ℓ_2 and ℓ_1 operations, respectively, and α is a vector containing all model coefficients. The nonnegative constraints imposed on coefficients, $\alpha_m \geq 0$, guarantee that the generated performance function is in posynomial form. Referring to the objective function in Eq. (4), the first term is the sum of squared errors during the fitting procedure that indicates modeling accuracy, and the ℓ_1-regularization term measures the sparsity of coefficients since it forces those unimportant coefficients to be compressed to zero. In addition, the regularization parameter λ provides a trade-off between model sparsity and fitting accuracy. By solving this regularized and constrained optimization problem, a sparse posynomial model could be generated with a small set of samples without over-fitting.

In what follows, we present an iterative algorithm to efficiently solve the ℓ_1-regularization problem in Eq. (4) and obtain sparse posynomial coefficients. We perform the following transformation on the squared-error term of objective function:

$$\begin{aligned}\|Y - H\alpha\|_2^2 &= Y^TY - 2Y^T(H\alpha) + (H\alpha)^T(H\alpha) + (\lambda\mathbf{1})^T\alpha \\ &= \alpha^T(H^TH)\alpha + (\lambda\mathbf{1} - 2H^TY)^T\alpha\,,\end{aligned}$$

where $\mathbf{1}$ represents a M-dimensional vector of all ones. Letting $A = H^TH$ and $b = \lambda\mathbf{1} - 2H^TY$, the ℓ_1-regularization problem in Eq. (4) can be further converted into a quadratic programming program as follows:

$$\begin{aligned}\text{minimize}: &\quad \alpha^TA\alpha + b^T\alpha \\ \text{subject to}: &\quad \alpha \geq 0\,.\end{aligned} \tag{5}$$

To tackle the nonnegative constraints on α, an iterative algorithm is required. Following,[26,27] we introduce two nonnegative matrices, A^+ and A^-, to collect the positive terms and negative terms in A, respectively:

$$A^+ = \begin{cases} A_{ij}\,, & \text{if } A_{ij} > 0 \\ 0\,, & \text{if } A_{ij} \leq 0 \end{cases}, \quad A^- = \begin{cases} \|A_{ij}\|\,, & \text{if } A_{ij} < 0 \\ 0\,, & \text{if } A_{ij} \geq 0 \end{cases}.$$

The model coefficients can be eventually determined by the following iterative procedure starting with an initial guess of coefficient values:

$$\alpha_m \leftarrow \alpha_m \left[\frac{-b_m + \sqrt{b_m^2 + 4(2A^+\alpha)_m(2A^-\alpha)_m}}{(2A^+\alpha)_m}\right]. \tag{6}$$

By iteratively updating the the values of coefficients α_m's according to Eq. (6), a set of sparse and nonnegative coefficients are determined after iteration converges, and a sparse posynomial function is therefore obtained. The complete algorithmic flow of the proposed posynomial modeling method is summarized in Algorithm 1.

3. Robust Circuit Design by GP

3.1. *Robust design framework under process variations*

The traditional circuit design method typically aims at determining appropriate values of design parameters such that a performance metric of interest is maximized under a set of pre-specified constraints. This nonrobust design problem can be formulated as:

$$\begin{aligned} \text{minimize :} \quad & f_0(x) \\ \text{subject to :} \quad & g_j(x) \le g_{\text{limit}}^{(j)} \\ & X_{\min} \le X \le X_{\max} \\ \text{variables :} \quad & X = (x_1, x_2, \ldots, x_n)\,, \end{aligned} \tag{7}$$

where $f_0(X)$ is the objective function to be minimized, and $g_{\text{limit}}^{(j)}$'s represent impose a set of performance constraints, i.e., each performance metric cannot exceed a corresponding specification value. The decision variables include all associated design parameters. $X_{\min}$ and $X_{\max}$ are the lower and upper bound limits for design

Algorithm 1. Regularization-based Posynomial Performance Modeling

```
Input: simulation data at $K$ sampling points, i.e. $H$ and $Y$; a set of $M$ selected
    monomial basis function, i.e., $g_m(X)$
Output: a vector of sparse model coefficients $\alpha$
create the sample matrix of basis functions $H$;
compute $A = H^T H$ and $b = \lambda \mathbf{1} - 2H^T Y$;
create matrices $A^+$ and $A^-$ based on $A$;
set iterator $Iter = 0$;
set tolerance $TOL = 1\text{e-}3$;                        ▷ termination condition
initialize model coefficients $\alpha = \alpha^{\text{init}}$;
while $Iter < IterMax$ do                                 ▷ iteration limit not reached
    $\alpha_{\text{old}} = \alpha$;                       ▷ save the current $\alpha$
    for $m = 1, 2, \ldots, M$ do
        update $\alpha_m$ values according to (6);
    end for
    if $\|\alpha - \alpha_{\text{old}}\|_2 < TOL$ then    ▷ termination condition satisfied
        return coefficient vector $\alpha$;
    end if
    $Iter = Iter + 1$;
end while
return coefficient vector $\alpha$;
```

parameters, respectively. Note that $f_0(x)$ and $g_j(X)$'s in Eq. (7) are posynomial functions. It is obvious that the traditional design problem is in GP form.

We now discuss robust circuit design under the impact of process variations. The design parameters have been assigned random variations around their nominal values. In other words, the decision variables of the IC design problem are no longer deterministic quantities but become random variables. As a consequence, performance metric becomes variational with fluctuations deviated from its nominal metric. Let $\delta X = [\delta x_1, \delta x_2, \dots, \delta x_n]$ represent the variations in design parameters, the constraint functions $g_j(X_0)$ in Eq. (7) would be replaced by $g_j(X_0 + \delta X)$. Obviously, the performance metric under process variations is now a stochastic process, rather than a deterministic function. When conducting robust design, all performance metrics are required to not exceed their specification values $g_{\text{limit}}^{(j)}$'s for all possible perturbation values introduced by parameter variations, leading to the following robust performance constraints:

$$\max\{g_j(X_0 + \delta X)\} \le g_{\text{limit}}^{(j)} . \tag{8}$$

Under this circumstance, variability-affected circuit design in fact turns into a robust optimization problem with variational performance constraints.

3.2. *Second-order cone uncertainty set*

Random process variations are unrepeatable uncertainties due to insufficient process control and intrinsic fluctuations. The accuracy of the uncertainty characterization model significantly impact the quality of robust design solutions. According to the discussion in Sec. 3.1, instead of finding solutions robust to stochastic uncertainty in a probabilistic sense, robust design constructs a solution that is feasible for any realization of parameter variation. Robust optimization usually employs an uncertainty set to estimate parameter uncertainty. An uncertainty set is a deterministic set that is used to capture all possible perturbation values around a nominal point in the design space. By the aid of an uncertainty set, it is possible to reduce the robust design problem with the robust constraint in Eq. (8) to a completely deterministic problem.

In this paper, we employ a novel second-order cone (SOC) uncertainty set to characterize process variations and circuit uncertainties. Mathematically, a second order cone of dimension k is defined as[28]

$$\left\{ \begin{bmatrix} U \\ t \end{bmatrix} \middle| U \in R^{k-1},\ t \in R,\ \|U\| \le t \right\}, \tag{9}$$

where U is a vector of dimension $k - 1$. For the random vector X of design parameters with process variations, δX denotes the random perturbations around its nominal value X_0. By introducing an auxiliary variable s, the variation vector δX can be represented by an uncertainty second order cone (USOC), which is defined as a set:

$$\{(\delta X, s) | \|\delta X\|_2 = \|X - X_0\|_2 \leq s, s \geq 0\}\,, \tag{10}$$

where $\|\delta X\|_2 = \delta X^T \delta X$ is the two-norm of variation vector δX. To be more specific, we provide an example of three-dimensional SOC uncertainty set for a two-dimensional parameter vector X, which is represented by

$$\left\{ \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ s \end{bmatrix} \middle| \left\| \begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix} \right\|_2 \leq s,\ x_{1\,\min} \leq \delta x_1 \leq x_{1\,\max},\ x_{2\,\min} \leq \delta x_2 \leq x_{2\,\max} \right\}, \tag{11}$$

where $x_{1\,\max}$ and $x_{1\,\min}$ are the upper and lower bounds for parameter variation δx_1, the same as $x_{2\,\max}$ and $x_{2\,\min}$ for δx_2. As illustrated in Fig. 1, the SOC representation allows the elasticity of maximum parameter perturbation range. The auxiliary variable s in fact restricts how far the parameter variations can perturb from their nominal values. By varying this s variable, all possible parameter perturbations of X around X_0 can be bounded within the SOC region.

3.3. *Robust GP formulation*

Standard GP formulation requires that the objective and all constraint functions are posynomials. The objective function in Eq. (7) is already in posynomial form. We focus on formulating the set of constraint functions $g_j(X_0 + \delta X)$. For small parameter variations deviating from their nominal values, the variational constraint function $g_j(X_0 + \delta X)$ can be approximated by a first order Taylor series

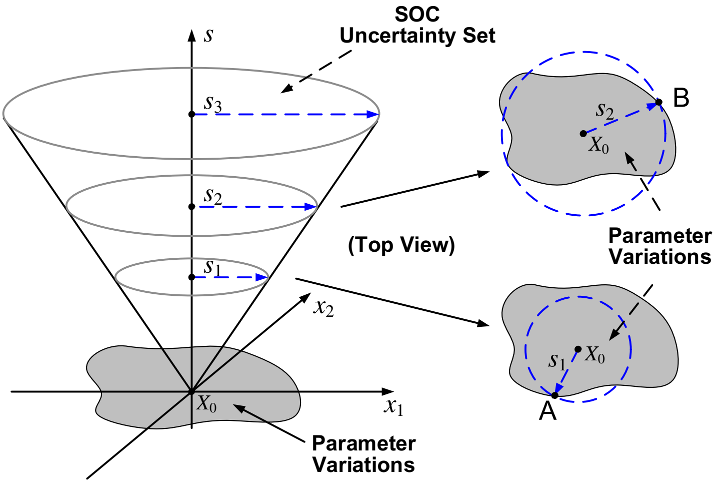


Fig. 1. An example of three-dimensional USOC representation for two-dimensional parameter variations.

expansion:

$$\begin{aligned} g_j(X_0 + \delta X) &= g_j(X_0) + \nabla g_j(X_0) \cdot (X + \delta X - X_0) \\ &= g_j(X_0) + \sum_{i=1}^{n} \left(\frac{\partial g_j}{\partial x_i} \right) \bigg|_{x_{i_0}} \delta x_i \,, \end{aligned} \tag{12}$$

where $\nabla g_j(X_0)$ represents the gradient of performance function g_j calculated at the nominal values of design parameters, and δX denotes the random variations around the nominal parameter values.

From Eq. (12) we observe that the variational function $g_j(X_0 + \delta X)$ consists of two components: (1) the deterministic part $g_j(X_0)$, which is in posynomial form; and (2) the variational part $\sum_{i=1}^{n} (\frac{\partial d_j}{\partial x_i})|_{x_{i_0}} \delta x_i$, which can be further decomposed into a gradient term and a parameter variation term. For the gradient term, since performance function g_j is modeled as a posynomial, it is not difficult to show that $(\frac{\partial g_j}{\partial x_i})|_{x_{i_0}}$, the derivative of posynomial function g_j at nominal point x_{i_0}, no longer keeps the posynomial property. On the other hand, all possible perturbation values in the variation term are required to satisfy the performance constraint. In other words, the complete variational function (12) has to be smaller than a pre-specified upper bound limit. This is equivalent to

$$g_j(X_0) + \max_{\forall \delta X} \left\{ \sum_{i=1}^{n} \left(\frac{\partial g_j}{\partial x_i} \right) \bigg|_{x_{i_0}} \delta x_i \right\} \leq g_{\text{limit}}^{(j)} \,, \tag{13}$$

which indicates that the maximum possible value of the variational function must be bounded by timing constraint $g_{\text{limit}}^{(j)}$. By above transformation, the variational performance constraints in Eq. (7) will be substituted by the deterministic constraint (13) in the following part.

After one step of transformation, the constraint function is still not in standard GP form, because of the gradient term and the parameter variation term in Eq. (13). Further transformations are required for GP formulation. We show that by employing the concept of a SOC uncertainty set and introducing several slack variables, the deterministic constraint function can be transformed into a set of standard posynomials.

We employ the SOC uncertainty set, which is described in Sec. 3.2, to address the parameter variation term δX. With parameter variations characterized in USOC representation, we focus on formulating the gradient term. The posynomial performance function can be rewritten as follows:

$$g_j(X) = \sum_{d=1}^{D} c_d \cdot \prod_{l=1}^{n} x_{l0}^{a_{ld}} \,. \tag{14}$$

The derivative of the performance function at x_{i0} is then given by

$$\left(\frac{\partial g_j}{\partial x_i}\right)\bigg|_{x_{i_0}} = \sum_{d=1}^{D} a_{id} c_d \cdot \underbrace{\prod_{l \neq i} x_l^{a_{ld}} \cdot x_i^{a_{id}-1}}_{\gamma_{id}(X_0)} . \tag{15}$$

By combining the results in Eqs. (14) and (15), we can express the constraint function (13) explicitly as

$$\sum_{d=1}^{D} c_d \prod_{l=1}^{n} x_{l_0}^{a_{ld}} + \max_{\forall \delta X} \left\{ \sum_{i=1}^{n} \left(\sum_{d=1}^{D} \gamma_{id}(X_0) \right) \cdot \delta x_i \right\} \leq g_{\text{limit}}^{(j)} . \tag{16}$$

In above equation, coefficient a_{id} stands for the exponent index in posynomial performance function (14). Therefore, $g_{ik}(X)$ could be either positive or negative, bringing the difficulty in GP formulation since a standard posynomial does not allow negative coefficients. To address this problem, we introduce two vectors $\Phi^+, \Phi^- \in R^n$ to collect the positive and negative coefficients in Eq. (16), respectively. To be more specific, the components of vectors Φ^+ and Φ^- are summarized as follows:

$$\begin{cases} \phi_i^+ = \sum_p g_{ip}(X) , & \text{for } \forall p \text{ s.t. } a_{ip} > 0 , \\ \phi_i^- = \sum_q g_{iq}(X) , & \text{for } \forall q \text{ s.t. } a_{iq} < 0 . \end{cases}$$

Accordingly, the constraint function in Eq. (16) can be converted into the following expression:

$$\sum_{d=1}^{D} c_d \prod_{l=1}^{n} x_{l_0}^{a_{ld}} + \max_{\forall \delta X} \{ \langle \Phi^+, \delta X \rangle + \langle \Phi^-, \delta X \rangle \} \leq g_{\text{limit}}^{(j)} , \tag{17}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors. Following the well-known Cauchy–Schwartz inequality, as well as the fact that $\|\delta X\|_2 \leq s$ in the SOC uncertainty set, an equivalent expression for Eq. (17) can be given by

$$\sum_{d=1}^{D} c_d \prod_{l=1}^{n} x_{l_0}^{a_{ld}} + \|\Phi^+\|_2 \cdot s + \|\Phi^-\|_2 \cdot s \leq g_{\text{limit}}^{(j)} , \tag{18}$$

where $0 \leq s \leq s_{\max}$ is an auxiliary variable introduced by the SOC uncertainty set.

The last formulation procedure is to introduce two more slack variables r_1 and r_2 to substitute the norm terms Φ^+ and Φ^-:

$$r_1 = \|\Phi^+\|_2, \quad r_2 = \|\Phi^-\|_2 .$$

By this substitution, the constraint function in (18) can be expressed as

$$\sum_{d=1}^{D} c_d \prod_{l=1}^{n} x_{l_0}^{a_{ld}} + r_1 \cdot s + r_2 \cdot s \le g_{\text{limit}}^{(j)} \,. \tag{19}$$

It is obvious that Eq. (19) is now a posynomial of the new decision variable set (X, r_1, r_2, s). Following the fact that $r_1^2 = (\Phi^+)^T\Phi^+$ and $r_2^2 = (\Phi^-)^T\Phi^-$, we need to further include the following two inequality constraints in the robust design framework:

$$(\Phi^+)^T\Phi^+ r_1^{-2} \le 1 \,, \tag{20}$$

$$(\Phi^-)^T\Phi^- r_2^{-2} \le 1 \,. \tag{21}$$

We show that the quadratic terms in Eqs. (20) and (21) are in standard posynomial form by expanding them:

$$(\Phi^+)^T\Phi^+ = \sum_{i,j} \phi_i^+ \phi_j^+ \,,$$
$$(\Phi^-)^T\Phi^- = \sum_{i,j} \phi_i^- \phi_j^- \,.$$

This expansion indicates that these two quadratic terms are summations of posynomials with all positive multiplicative coefficients, since all terms in ϕ_i^+ and ϕ_j^+ have positive coefficients, and all terms in ϕ_i^- and ϕ_j^- have negative coefficients. Therefore, the constraint functions in Eqs. (20) and (21) both are posynomials.

Putting all the formulation results together, we conclude that the variational constraint function (13) has been replaced by a set of posynomial constraints. In other words, we have formulated the robust design problem with variational performance constraints into a standard GP, which can be summarized as follows:

$$\begin{aligned} \text{minimize}: \quad & f_0(X_0) \\ \text{subject to}: \quad & g_j(X_0) + r_1 \cdot s + r_2 \cdot s \le g_{\text{limit}}^{(j)} \\ & (\Phi^+)^T\Phi^+ r_1^{-2} \le 1 \\ & (\Phi^-)^T\Phi^- r_2^{-2} \le 1 \\ & X_{\min} \le X_0 \le X_{\max} \\ & 0 \le s \le s_{\max} \\ \text{variables}: \quad & X_0, r_1, r_2, s \,, \end{aligned} \tag{22}$$

where X_0 represents nominal design parameters, r_1, r_2 are two slack variables, and variable s is introduced by the SOC uncertainty set. The objective and all constraint functions in Eq. (22) are in posynomial form. This formulated GP can be then efficiently solved by convex optimization tools. The final solution is the vector of nominal design parameters such that the objective performance metric is optimized while all performance constraints are satisfied in the presence of process variations.

4. Complexity Analysis

In this section, we perform a complexity analysis regarding the two main steps of the proposed robust optimization framework, i.e., the regularization-based posynomial performance modeling and the GP-based robust circuit design.

In the performance modeling step, the model coefficients are determined by the iterative procedure solving the ℓ_1-regularization problem formulated in Eq. (4). The main computational burden results from the iterative update of model coefficients (lines 7–16 in Algorithm 1). Since A^+ and A^- in the iterative equation (6) are two $M \times M$ matrices (where M is the number of monomials in posynomial performance model), the computational complexity during each iteration is roughly $\mathcal{O}(M^2)$. Assuming that the iteration count is C, the runtime complexity for generating a posynomial model is determined to be $\mathcal{O}(C \times M^2)$. It is observed in experiments that the iterative procedure typically converges within 5–10 iterations. In addition to fitting cost, the simulation cost for collecting sampling data by this regularization-based method also can be significantly reduced due to the sparse feature of model coefficients.

We now discuss the complexity of the robust optimization flow by employing the SOC uncertainty set, which can be formulated into a GP with posynomial performance functions. A GP can be efficiently solved by interior-point methods[29,30] that are widely used in existing GP solvers.[31,32] As pointed out in Ref. 20, the runtime complexity of solving a GP is in practice $O(p \times n^2)$, where p is the number of constraints, and n is the number of decision variables in GP. Considering that the number of constraints in the proposed design framework is roughly proportional to the number of design parameters, the complexity of this step can be estimated as $O(n^3)$.

According to the preceding analysis, the proposed framework for robust design is computationally efficient. On one hand, by establishing the regularized sparse representation of posynomial performance models, we can substantially reduce the number of required simulation samples but can obtain sparse model coefficients. On the other hand, based on the obtained posynomial models, we can formulate the robust design into a standard GP which can be solved in polynomial time.

5. Experimental Results

The proposed method was implemented and verified by MATLAB programming integrated with SPICE simulations. All simulations and experiments were performed on a quad-core 3.4 GHz machine with 16 GB memory. We use a high-gain, rail-to-rail output, fully CMOS operational amplifier (OpAmp) as the benchmark circuit. The schematic of this OpAmp is shown in Fig. 2. The design parameters include the effective channel length (L_{eff}), threshold voltage (V_{th}), and oxide thickness (T_{ox}) of each device. The process variations for all design parameters are assumed to follow

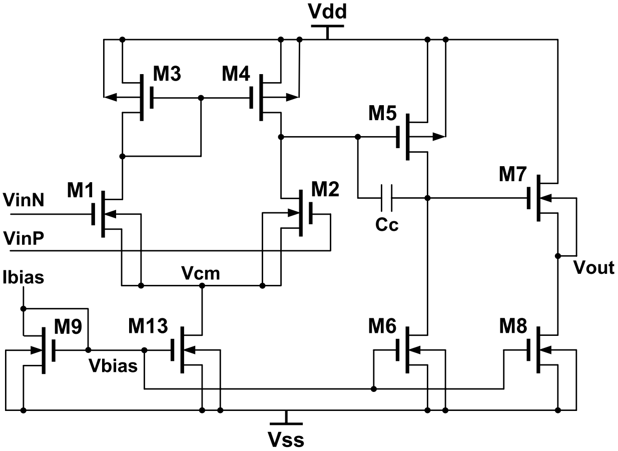


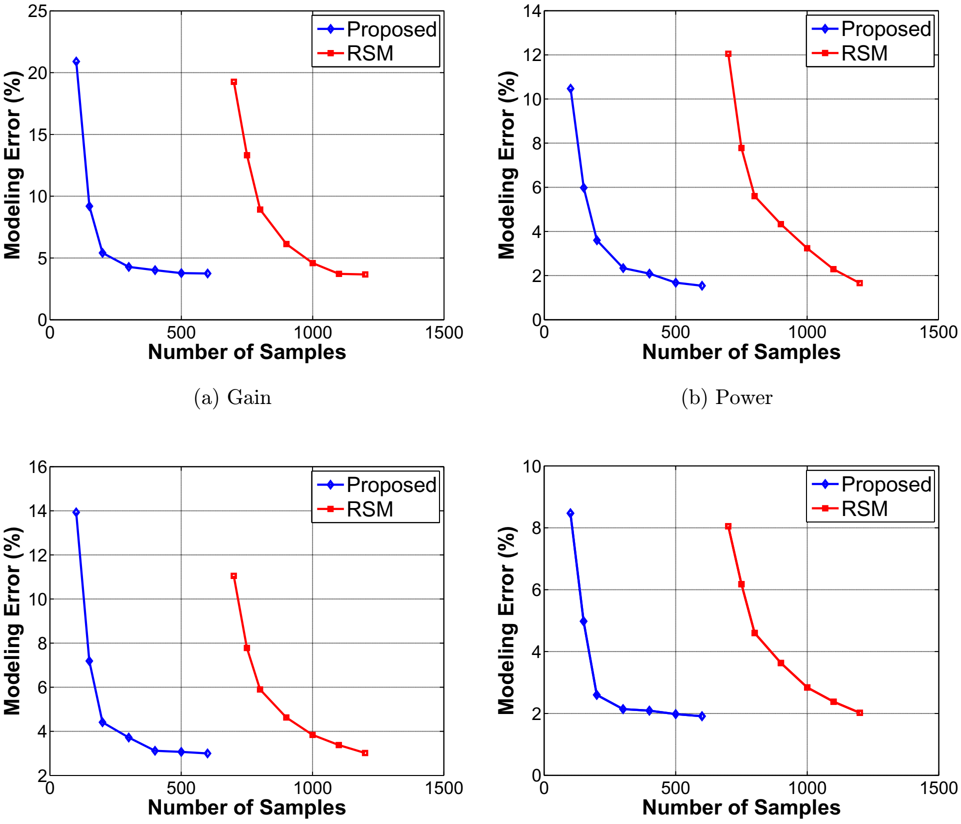
Fig. 2. Schematic of the OpAmp benchmark circuit.

Gaussian distributions. The 3σ values for L_{eff}, V_{th} and T_{ox} are specified as 30%, 30%, and 20% of their corresponding nominal values, respectively. The design range of each parameter is as follows: L_{eff} value ranges from 40 to 200 nm, V_{th} value ranges from 0.3 to 0.7 V, and T_{ox} value ranges from 0.5 to 5 nm. Before performance modeling, we first perform a sensitivity analysis to prune those parameters that have little impact on circuit performance, in order to reduce the dimension of design space. In addition, we used GGPLAB, an efficient GP solver in a convex optimization tool, to solve the formulated GP in Eq. (22) and obtain the optimal nominal values of design parameters that satisfy the robust performance constraints.

5.1. *Verification of modeling accuracy*

To verify the modeling accuracy of the proposed method, we take into account four key performance metrics of the OpAmp circuit, including gain, power consumption, bandwidth and offset voltage, to run SPICE simulations, collect sampling data and determine the coefficients in the performance model.

Figure 3 provides the modeling errors when fitting the four performance metrics by the proposed method and traditional RSM method with different sample volumes. It can be observed that as the number of samples increases, the modeling error improves for both methods. However, the proposed method requires a much smaller sample volume to achieve the same accuracy, and therefore significantly reduces the simulation cost. Using power specification for example, a modeling error of 1.66% is obtained by RSM method when the sample volume is as large as 1,200, while the proposed method requires merely 500 samples to achieve a roughly equal modeling error of 1.68%. The reason is that, circuit performance is usually dominated by the design parameters of several few devices. The regularization-based method in this



(a) Gain

(b) Power

(c) Bandwidth

(d) Offset

Fig. 3. Comparison of performance modeling errors by the proposed method and RSM method.

work has the ability to compress those unimportant coefficients in performance model to zero by exploring the sparsity of model coefficients. When the sample volume reaches 600, the proposed method provides modeling errors ranging from 1.54% to 3.75% for all four performance metrics of interest, which are superior over the accuracy achieved by RSM method with 1,200 samples.

Table 1. Comparison of computation time by the proposed method and RSM method.

	Simulation cost (s)		Fitting cost (s)		Total cost (s)		
Performance metric	RSM	Proposed	RSM	Proposed	RSM	Proposed	Speedup
Gain	2013	841	1.70	0.47	2014.70	841.47	2.39×
Power	1849	770	1.73	0.46	1850.73	770.46	2.40×
Bandwidth	2028	845	1.54	0.39	2029.54	845.39	2.40×
Offset	2020	844	1.46	0.44	2021.46	844.44	2.39×

Table 1 lists the computation costs required for generating performance model by the proposed method and RSM method for comparison. For the proposed method, the time costs are evaluated based on 500 samples, while the number of samples used for time cost evaluation is 1,200 for RSM method. From the second to seventh columns of Table 1, the simulation costs, fitting costs and total costs by both methods are presented. Due to the complexity of the benchmark circuit, simulation cost accounts for the majority of total computation cost. Relying on the regularized sparse representation, the proposed method improves computational efficiency by substantially reducing the number of required samples. The speedup of total computation time over RSM method is listed in the last column of Table 1.

5.2. *Verification of circuit robustness*

To verify the effectiveness of the proposed robust design methodology, we solve the formulated GP problem with robust performance constraints under process variations, and obtain the optimal nominal values of design parameters such that the objective performance metric is minimized. We rely on the obtained nominal design values to evaluate the performance perturbations propagated from parameter variations. We assume the design objective is to minimize the power consumption with a gain constraint, i.e., the variability-induced fluctuations in gain metric cannot exceed a given gain specification. For a gain specification of 50 dB, we randomly generate 20,000 samples around the nominal values of design parameters obtained by the proposed GP framework. Figure 4 shows the distribution of gain fluctuations

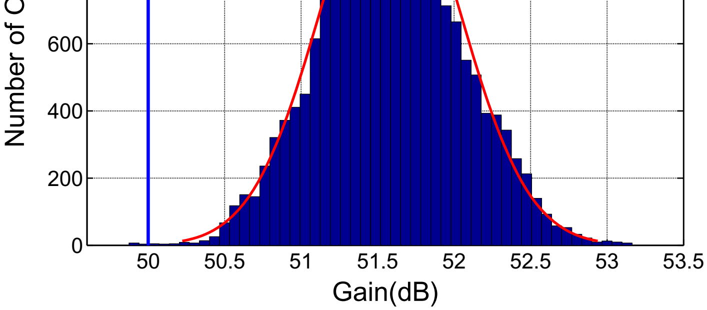


Fig. 4. Schematic of the OpAmp benchmark circuit.

evaluated at all samples. As can be observed from the figure, the vast majority of the fluctuation values satisfy the gain constraint. To be specific, only 10 out 20,000 samples produce gain metrics that are below the gain specification, indicating a gain yield of 99.95%. On the other hand, without consideration of process variations, the traditional design method may lead to a nonrobust solution that is close to the specification value and therefore a low gain yield. For comparison, the nonrobust solution with the same gain constraint produces about 9,200 out of 20,000 samples that violate the gain specification. The gain yield of the nonrobust solution is merely about 54%.

To further demonstrate the advantages of the proposed method over nonrobust method, we select a set of gain specification values ranging from 49.5 to 52.0 dB, and observe the power consumptions as well as corresponding power yields by both methods. The results are listed in Table 2. It can be observed that, for different specification values, the gain metrics obtained the proposed robust method can satisfy the gain constraint, with at least a gain yield of 99.87%. In contrast, not taking process variations into account, the design values obtained by the nonrobust method cause significant losses in power yield. The comparison results justify that, to ensure the robustness against process variations, on average the robust method leads to about 18% power overhead compared with the nonrobust method, but the robust solutions completely satisfy the performance constraints, and therefore are much more reliable than the nonrobust solutions.

Moreover, we perform a set of experiments to evaluate the robust solution obtained by the proposed method under multiple performance constraints. We use power consumption as the design objective, and require that the performance fluctuations in gain, bandwidth, and offset voltage propagated from process variations cannot exceed their pre-specified values. The gain specifications are selected from {50 dB, 51 dB, 52 dB}, the bandwidth specifications are selected from {400 MHz, 500 MHz}, and the offset specifications are selected from {25 uV, 30 uV}, respectively. Table 3 presents the power consumptions by using the proposed robust method and nonrobust method under the constraint of a combinations of various

Table 2. The power consumptions and power yields for different gain specifications.

	Traditional method		Proposed method	
Gain specification (dB)	Power (mW)	Yield (%)	Power (mW)	Yield (%)
49.5	11.80	54.19	13.76	99.94
50.0	12.03	53.94	13.99	99.95
50.5	12.17	52.26	14.37	99.90
51.0	12.64	51.54	14.82	99.90
51.5	12.95	51.45	14.92	99.87
52.0	13.16	50.79	15.39	99.89

Table 3. The power consumptions and power yields for different combinations of multiple performance specifications (gain, bandwidth, and offset voltage).

Performance specification			Traditional method			Proposed method		
Gain (dB)	Bandwidth (MHz)	Offset (uV)	Power (mW)	Yield (%)	Runtime (s)	Power (mW)	Yield (%)	Runtime (s)
50	400	30	12.44	61.76	0.86	14.93	99.93	4.52
50	400	25	12.75	61.70	0.84	15.30	99.93	4.50
50	500	30	13.40	59.89	0.93	16.08	99.85	5.13
50	500	25	13.86	60.99	0.92	16.63	99.90	4.78
51	400	30	13.80	58.26	0.95	16.63	99.85	5.02
51	400	25	14.29	58.54	0.99	17.15	99.84	4.39
51	500	30	14.75	58.17	0.95	17.10	99.86	4.55
51	500	25	15.07	56.87	0.97	18.08	99.84	4.83
52	400	30	15.65	57.20	0.86	18.78	99.91	4.99
52	400	25	16.39	55.56	0.95	19.67	99.84	4.97
52	500	30	18.07	55.06	1.00	21.68	99.80	5.19
52	500	25	18.50	54.94	0.92	22.20	99.82	4.81

performance specifications. By assigning random variations to the obtained design solutions, the performance yields under multiple performance constraints are also provided. In addition, the comparison of computation cost between the robust and nonrobust methods is also provided in Table 3. Similar to the robust design solutions under one single constraint, the solutions obtained by the proposed method under multiple constraints are robust to process variations, leading to performance yields over 99.80% with various performance specifications.

6. Conclusions

This paper presents a novel robust design framework for strongly nonlinear circuits by geometric programming. This framework uses a regularization-based method to model circuit performance as a posynomial function of design parameters with sparse model coefficients. Relying on the established posynomial models, robust circuit design under process variations can be formulated into an easy-to-solve optimization problem in standard GP form, by employing a SOC uncertainty set to characterize the random variations of design parameters. Experimental results on an OpAmp circuit show that the posynomial performance models can accurately capture the strongly nonlinear dependence of circuit performance metric upon design parameters. More importantly, the formulated GP facilitates the convex optimization of circuit performance under process variations. The robust design solutions generated by the proposed GP framework have achieved significant yield improvements compared with traditional nonrobust methods.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 71501096 and 61502234, the Natural Science Foundation of Jiangsu Province of China under Grant No. BK20150785, the Project funded by China Postdoctoral Science Foundation under Grant No. 2015M581801, the Fundamental Research Funds for the Central Universities under Grant No. 30916011325, and the National Key Research and Development Program of China under Grant No. 2016YFE0108000.

References

1. M. Orshansky, S. Nassif and D. Boning, *Design for Manufacturability and Statistical Design: A Constructive Approach* (Springer, Berlin, Germany, 2008).
2. M. Radfar and J. Singh, A yield improvement technique in severe process, voltage, and temperature variations and extreme voltage scaling, *Microelectron. Reliab.* **54** (2014) 2813–2823.
3. X. Dong and L. Zhang, Process-variation-aware rule-based optical proximity correction for analog layout migration, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **36** (2017) 1395–1405.
4. L. Cheng, W. Xu, F. Ren, F. Gong, P. Gupta and L. He, Statistical timing and power analysis of VLSI considering non-linear dependence, *Integr. VLSI J.* **47** (2014) 487–498.
5. W. Zhang, K. Balakrishnan, X. Li, S. Saxena, S. Saxena, A. Strojwas and R. A. Rutenbar, Efficient spatial pattern analysis for variation decomposition via robust sparse regression, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **32** (2013) 1072–1085.
6. H. Yu, X. Liu, H. Wang and X. D. Tan, A fast analog mismatch analysis by an incremental and stochastic trajectory piecewise linear macromodel, *Design Automation Conf.*, Taipei, Taiwan, 2010, pp. 211–216.
7. J. Sun, L. Xiao, J. Tian, H. Zhou and J. Roveda, Surrogating circuit design solutions with robustness metrics, *Integr. VLSI J.* **52** (2016) 1–9.
8. X. Li, Finding deterministic solution from underdetermined equation: Large-scale performance variability modeling of analog/RF circuits, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **29** (2010) 1661–1668.
9. J. Singh, Z.-Q. Luo and S. S. Sapatnekar, A geometric programming-based worst case gate sizing method incorporating spatial correlation, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **27** (2008) 295–308.
10. S. Yaldiz, U. Arslan, X. Li and L. Pileggi, Efficient statistical analysis of read timing failures in SRAM circuits, *International Symp. Quality Electronic Design*, San Jose, CA, USA, 2009, pp. 617–621.
11. H. Chang and S. S. Sapatnekar, Statistical timing analysis under spatial correlations, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **24** (2005) 1467–1482.
12. H. Tennakoon and C. Sechen, Efficient and accurate gate sizing with piecewise convex delay models, *Proc. of the Design Automation Conf.*, San Diego, CA, USA, 2005, pp. 807–812.
13. S. Vrudhula, J. M. Wang and P. Ghanta, Hermite polynomial based interconnect analysis in the presence of process variations, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **25** (2006) 2001–2011.

14. X. Li, J. Le, P. Gopalakrishnan and L. T. Pileggi, Asymptotic probability extraction for nonnormal performance distributions, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **26** (2006) 16–37.
15. F. Gong, S. Basir-Kazeruni, L. He and H. Yu, stochastic behavioral modeling and analysis for analog/mixed-signal circuits, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **32** (2012) 24–33.
16. S. Boyd, S. J. Kim, L. Vandenberghe and A. Hassibi, A tutorial on geometric programming, *Optim. Eng.* **8** (2007) 67.
17. R. J. Duffin, E. L. Peterson and C. Zener, *Geometric Programming: Theory and Application* (Wiley).
18. L. Clemmensen, T. Hastie, D. Witten and B. Ersbøll, Sparse discriminant analysis, *Technometrics* **53** (2011) 406–413.
19. C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, Berlin, Germany, 2006).
20. S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, New York, 2004).
21. V. Aggarwal and U.-M. O'Reilly, Simulation-based reusable posynomial models for MOS transistor parameters, *Design, Automation & Test in Europe Conf. & Exhibition*, Nice, France, 2007, pp. 1–6.
22. S. Maji and P. Mandal, Effcient approaches to overcome non-convexity issues in analog design automation, *International Symp. Quality Electronic Design*, Santa Clara, CA, USA, 2012, pp. 566–571.
23. Y. Zhang, B. Liu, B. Yang, J. Li and S. Nakatake, CMOS Op-amp circuit synthesis with geometric programming models for layout-dependent effects, *International Symp. Quality Electronic Design*, Santa Clara, CA, USA, 2012, pp. 464–469.
24. X. Li, P. Gopalakrishnan, Y. Xu and L. T. Pileggi, Robust analog/RF circuit design with projection-based performance modeling, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **26** (2006) 2–15.
25. B. K. Natarajan, Sparse approximate solutions to linear systems, *SIAM J. Comput.* **24** (1995) 227–234.
26. L. Wu, Y. Yang and H. Liu, Nonnegative-lasso and application in index tracking, *Comput. Stat. Data Anal.* **70** (2014) 116–126.
27. F. Sha, Y. Lin, L. K. Saul and D. D. Lee, Multiplicative updates for nonnegative quadratic programming, *Neural Comput.* **19** (2007) 2004–2031.
28. M. S. Lobo, L. Vandenberghe, S. Boyd and H. Lebret, Applications of second-order cone programming, *Linear Algebra Appl.* **284** (1998) 193–228.
29. Y. E. Nesterov and A. S. Nemirovskii, *Interior Point Polynomial Methods in Convex Programming: Theory and Algorithms* (SIAM, Philadelphia, PA, 1994).
30. K. O. Kortanek, X. Xu and Y. Ye, An infeasible interior-point algorithm for solving primal and dual geometric programs, *Math. Program.* **76** (1997) 155–181.
31. S. P. Boyd, GGPLAB: A simple matlab toolbox for geometric programming (2006), available at: https://stanford.edu/∼boyd/ggplab/.
32. M. Edvall and F. Hellman, User's guide for TOMLAP/GP (2005), available at: https://tomopt.com/docs/TOMLAB_GP.pdf.