A New Framework for the Robust Design of Analog Blocks using Conic Uncertainty Budgeting

Claudio Talarico
Department of Electrical and Computer Engineering
Gonzaga University
Spokane, WA 99258, USA
Email: talarico@gonzaga.edu

Jin Sun
Orora Design Technologies
Issaquah, WA 98027, WA
Email: jinsun@orora.com

Priyank Gupta
Department of Electrical and Computer Engineering
The University of Arizona
Tucson, AZ 85721, USA
Email: pgupta@email.arizona.edu

Janet Wang
Department of Electrical and Computer Engineering
The University of Arizona
Tucson, AZ 85721, USA
Email: wml@ece.arizona.edu

Abstract **–** In nanoscale technologies process variability makes it extremely difficult to predict the behavior of manufactured integrated circuits (IC). The problem is especially exacerbated in analog IC where long design cycles, multiple manufacturing iterations, and low performance yields causes only few design to have the volume required to be economically viable. This paper presents a new framework that accounts for process variability by mapping the analog design problem into a robust optimization problem using a conic uncertainty model that dynamically adjust the level of conservativeness of the solutions through the introduction of the notion of budget of uncertainty. Given a yield requirement, the framework implements uncertainty budgeting by linking the yield with the size of the uncertainty set associated to the process variations depending on the design point of interest. Dynamically adjusting the size of the uncertainty set the framework is able to find a larger number of feasible solutions compared to other robust optimization frameworks based on the well known ellipsoidal uncertainty (EU) model. To validate the framework, we applied it to the design of a 90nm CMOS differential pair amplifier and compared the results with those obtained using the EU approach. Experimental results indicate that the proposed Conic Uncertainty with Dynamic Budgeting (CUDB) approach attain up to 18% more designs meeting target yield.

Keywords **–** Nanoscale Technology, Microelectronics, Process Variations, Robust Design Optimization, Geometric Programming, Uncertainty Set

I. Introduction

The difficulty of designing and manufacturing IC in nanoscale technologies is increasing to a point that only few products make the volume required to be economically viable [1]. This is particularly problematic in analog ICs where the large spread in circuit's performance metrics, caused by process variability, translates in a significant yield penalty [2], [3]. For this reason, it is essential for the design tools to incorporate uncertainty to obtain design circuits that are insensitive to random parameter variations as much as possible.

This paper describes the development of a framework for the design of analog IC blocks with a guaranteed performance yield bound. The method proposes to formulate the design with uncertainty problem as a robust optimization problem [4] to be mapped in the form of a special type of convex optimization problem called geometric programming (GP) [5], [6], [7].

In general, process variations can be divided in two categories: global variations and local variations. Global variations represents the variations occurring between different dies in the same wafer or different wafers. Local variations represent the variations between devices and interconnects within the same die. While global variations can be reasonably modeled as gaussian distributed random variables (RV), local variations are much more difficult to model. Local variations are random in nature, but they have strong spatial correlation and some variation sources are known to be non-gaussian with asymmetric distributions [8], [9]. Spatial correlation quantifies the fact that devices close to each other are more likely to have similar characteristics than devices far apart.

With process variations modeled as RVs, the most natural way to formulate the design with uncertainty problem is as a stochastic optimization problem [4], [10], [11]. Unfortunately, stochastic optimization has two major drawbacks. First, it assumes that uncertainty has an accurate probabilistic description, which often is not the case [4]. Second, it is plagued by high computational complexity [7]. For these reasons, in this work we use robust optimization rather than stochastic optimization. Robust optimization is an approach to optimization under uncertainty, in which the uncertainty model is not stochastic, but rather deterministic and set based [4], [12]. Robust optimization has recently become very popular [13], [14] thanks to the the development of fast interior point algorithms for convex optimization [7].

In robust optimization instead of looking for a solution that is robust to uncertainty in some probabilistic sense, we construct a solution that is feasible for any realization of the uncertainty in a given set. The resulting set-based design problems can be formulated as follows:

$$\begin{aligned} &\text{minimize: } f_0(x) \\ &\text{subject to: } f_i(x,u_i) \le f_{limit}^{(i)} \quad \forall u_i \in \mathcal{U}_i, i=1,...,m \end{aligned} \tag{1}$$

Where f_0 is the objective function to be optimized (i.e. the performance metric to be optimized), $x = (x_1, x_2, ..., x_n)$ is the vector of variables the objective function depends on (i.e. the design and process parameters), $f_i(x, u_i)$ are the performance constraints with uncertainty parameters u_i, and $f_{limit}^{(i)}$ is the maximum limit that can be imposed on the performance constraints function $f_i(x, u_i)$. The uncertainty parameters $u_i \in \mathbb{R}^k$ are assumed to take arbitrary values in the uncertainty sets $\mathcal{U}_i \subseteq \mathbb{R}^k$. The goal is to compute minimum cost solutions $x^* \in \mathbb{R}^n$ among all those solutions which are feasible for all realization of disturbances u_i within $\mathcal{U}_i$.

Requiring that the design solutions x^* must guarantee the performance constraints, lead to overly conservative design solutions and restrict significantly the applicability of robust optimization [4]. In practice is often sufficient to seek to "immunize" the design solutions in a more relaxed "probabilistic" sense. In this work, we address the restriction by using the concept of "budget of uncertainty". The notion of budget of uncertainty allows to structure the uncertainty set in a way that flexibly explore the trade-off between yield projection and design parameters variability and the solutions obtained are expected to be close to those obtained using stochastic methods. There are several possible ways the notion of budget of uncertainty can be implemented.

An approach, that has recently gained popularity among several authors [1], [15] is to structure the uncertainty set using an ellipsoidal uncertainty (EU) model. The idea is to approximate the uncertainty set, using the maximal ellipsoidal that can be inscribed inside the design parameters variation region. In order to capture variability, the design and process parameters (i.e. transistors' length (L), width (W), threshold voltage (V_{th}), oxide thickness (T_{ox}), etc.) are modeled as random variables with some joint probability density function (pdf) $p(\mu, \Sigma)$, where μ is a vector of means and Σ is a covariance matrix. For any vector $X \in \mathbb{R}^n$ with random perturbations around its center point $X_0 \in \mathbb{R}^n$, and non singular covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$, the parameter variability can be estimated by an ellipsoid uncertainty set in $\mathbb{R}^n$:

$$\mathcal{U}_\epsilon = \left\{ X \text{ s.t. } (X - X_0)^T \Sigma^{-1} (X - X_0) \le \psi^2 \right\} \quad (2)$$

where ψ and Σ determine how far the uncertainty ellipsoid extends in every direction from X_0. Figure 1 illustrates an ellipsoid in $\mathbb{R}^2$. The length of the axis of the ellipsoid are proportional to ψ. The proportionality factors λ_1 and λ_2 are given by the eigenvalues of the matrix Σ, and the direction of the axis l_1 and l_2 are given by the eigenvectors.

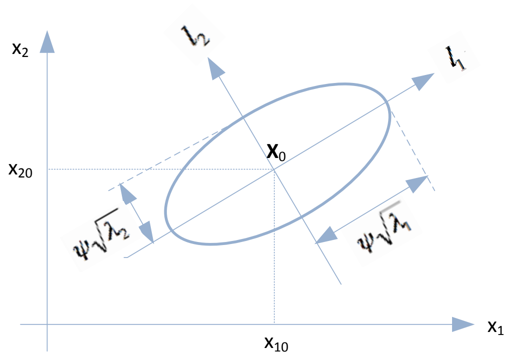

Fig. 1. Example of a two-dimensional ellipsoidal uncertainty set

If Σ is a symmetric and positive semidefinite matrix (that is, $\Sigma = \Sigma^T \ge 0$) an alternative representation of (2) can be obtained by making the substitution $\Sigma^{-1/2}(X - X_0) = u$:

$$\mathcal{U}_\epsilon = \left\{ X = X_0 + \Sigma^{1/2} u \text{ s.t. } \|u\|_2 \le \psi \right\} \quad (3)$$

where $\|u\|_2 = (u^T u)^{1/2}$ is the 2-nom of vector $u \in \mathbb{R}^n$. The ellipsoid represent an n-dimensional region, where the vector X varies around the center point X_0. The vector u represents the movement of X around X_0. The parameter variations are bounded within the ellipsoid region. Although, the approach can achieve satisfactory results [1], [15], often, it can be significantly improved. Figure 2 illustrates the major shortcoming of the EU method: the maximal inscribed ellipsoid in most cases leaves out significant regions of the feasible design space.

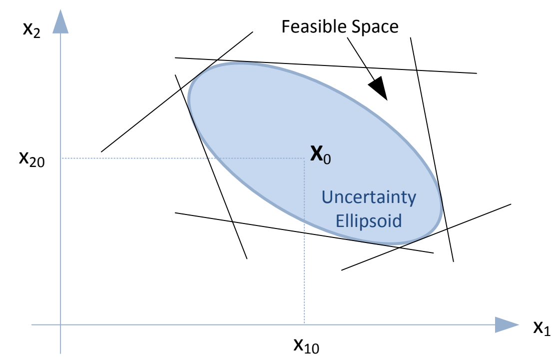

Fig. 2. Example of two-dimensional ellipsoidal uncertainty set giving modest coverage of the feasible design space

The reason for this shortcoming is that the size of the ellipsoid is solely determined by the uncertainty distribution of the design space (i.e. the variability of the design and process metrics). There is no information about how the design space maps into the performance space. For a desired performance yield η, neglecting the dependence of the performance metrics on the design and process metrics may lead to an over conservative estimation of the variations.

To address the above limitation we propose to use a second order Cone Uncertainty set with Dynamic adjustment of the uncertainty Budget (CUDB).

The rest of the paper is organized as follows. Section II describes: 1) how the framework models design and process parameter variations, 2) the implementation of the framework and 3), some of the underying algorithms. Section III discusses the results obtained with the CUDB method and compare them with the results obtained with the EU method. Finally, Section IV summarizes the results of our work and provides conclusions and future research directions.

II. Conic Uncertainty Dynamic Budgeting Framework

In the proposed framework, we flexibly adjust the level of conservativeness of the robust solutions by linking the probabilistic bounds on the constraint performance violations to the performance yield η:

$$\begin{aligned} &\text{minimize: } f_0(x) \\ &\text{subject to: } f_i(x, u_i) \leq f_{limit}^{(i)} \quad \forall u_i \in \mathcal{U}_i(\eta), i = 1...m \end{aligned} \tag{4}$$

A. Variability Model

To capture the variations that affect design and process parameters, we propose to use a second order conic uncertain set. Mathematically, a second order cone (SOC) in $\mathbb{R}^{n+1}$ is defined as [7]:

$$\{(X, s) \text{ s.t. } X \in \mathbb{R}^n, s \in \mathbb{R}, \|X\|_2 \leq s\} \subseteq \mathbb{R}^{n+1} \tag{5}$$

where X is the vector of random variables representing the design and process parameters, and s is the size of the uncertainty set. Figure 3 illustrates the boundary of a second-order cone in $\mathbb{R}^3$.

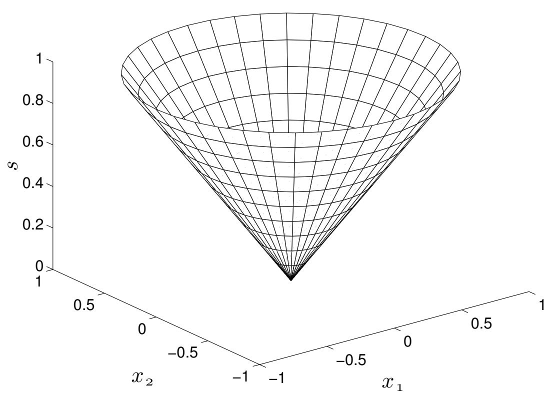

Fig. 3. Example of second-order cone in $\mathbb{R}^3$, $\{(x_1, x_2, s)$ s.t. $(x_1^2 + x_2^2)^{1/2} \leq s\}$

Choosing the appropriate s_{max} value, we can flexibly vary the uncertainty set to guarantee that all possible perturbation values in the feasible solution space are captured. If we denote with s the size of the uncertainty, and with δX the random perturbations around a nominal design X_0, the uncertainty set can be put in the form:

$$\begin{aligned} \mathcal{U}_\mathcal{C} = \{(\delta X, s) \text{ s.t. } \delta X \in \mathbb{R}^n, s \in \mathbb{R}, \\ \|\delta X\|_2 = \|X - X_0\|_2 \leq s, \\ s_{min} < s \leq s_{max}\} \subseteq \mathbb{R}^{n+1} \end{aligned} \tag{6}$$

where $s_{min} = 0$. If we consider a design point X_A, to estimate the variations around X_A (that is δX_A) we vary s from 0 to s_{maxA}, with s_{maxA} being the size of the uncertainty of the variations δX_A. At a different location X_B, there exist a different uncertainty set whose size is given by s_{maxB} that encompasses the variations δX_B. Note that to capture all possible perturbation values in the feasible region, not only we need to be able to vary the size of the set but also its shape. This is possible by extending the second-order cone from its standard form (6) to the general form:

$$\mathcal{U}_\mathcal{C} = \{\|A\delta X + b\|_2 \leq s_{max}\} \tag{7}$$

where $\delta X \in \mathbb{R}^n$, $A \in \mathbb{R}^{k \times n}$, and $b \in \mathbb{R}^n$. Figure 4 illustrate how the conic uncertainty model allows to change size (s_{max}) and shape (matrix A) of the uncertainty set to fit the feasibility space.

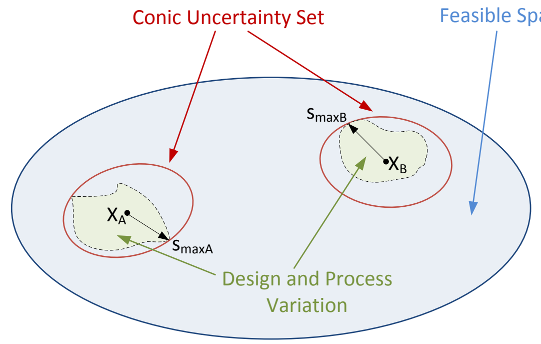

Fig. 4. The conic uncertainty model allows changing size and shape of the uncertainty set.

To determine the relationship between s_{max} and the nominal design parameters X_0 we use a fitting technique that rely on data obtained by sampling the design and process parameters variations. As discussed before, local variations are difficult to model, especially in presence of strong spatial correlation. Following [9], [16], [17], the entire die can be partitioned into a number of grid elements. Each grid element on the chip is denoted by its coordinate position: $l = (x, y)$. The variations at any two grids l_i and l_j on the same chip will be correlated. The correlation between two grid elements is assumed to depend on the Euclidean distance between them:

$$\begin{aligned} \rho(l_i, l_j) = \rho(\|l_i - l_j\|) = \\ = \rho\left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right) \end{aligned} \tag{8}$$

The covariance function of correlated variations can be then determined as:

$$\text{cov}(l_i, l_j) = \sigma_i \sigma_j \rho(\|l_i - l_j\|) \tag{9}$$

where σ_i and σ_j are the standard deviation of the variations at grid l_i and l_j respectively, and ρ_{ij} is the correlation function. Note that standard deviations σ_i and σ_j, and the correlation function ρ_{ij} across various dies will be different [9], therefore, it is not possible to use a unique uncertainty set for all design candidates.

In the fitting procedure, to derive the correlation function $\rho(\|l_i - l_j\|)$ we employs the Matern model [18]. We start the fitting procedure by performing random sampling around a nominal design X_0 and capture the largest variation from the nominal value. The distance obtained is the s_{max} value associated to the particular nominal design X_0. After repeating the simulation a large number of times we collect all data pairs (X_0, s_{max}) and fit them to a quadratic relationship of the form:

$$s_{\max} = \sum_{i=1}^{n} \lambda_i x_i^2 + \sum_{i=1}^{n} \alpha_i x_i + \sum_{i,j}^{n} \beta_{ij} x_i p_j + k \tag{10}$$

Before describing the framework implementation is important to emphasize that the viability of the approach is based on recent theoretical work that formalizes how robust optimization problems can be solved via convex optimization [4], [19]. The framework maps the problem of designing an analog circuit in a special type of convex optimization problem called geometric programming (GP). A geometric program is an optimization problem of the form:

$$\begin{aligned} \text{minimize: } & f_0(x) \\ \text{subject to: } & f_i(x) \leq 1, \quad i = 1, ..., p \\ & g_i(x) = 1, \quad i = 1, ..., m \end{aligned} \tag{11}$$

where $f_0, f_1, ..., f_p$ are posynomial functions and $g_1, ..., g_m$ are monomial functions. f_0 is the objective function to be optimized (i.e. the performance metric to be optimized), $x = (x_1, x_2, ...x_n)$ is the vector of variables the objective function depends on (i.e. the design and process parameters) and $f_i(x)$ and $g_i(x)$ are the constraints. A function g is called a monomial if it has the form:

$$g(x_1, x_2, ...x_n) = c \prod_{i=0}^{n} x_i^{\alpha_i} \tag{12}$$

where $c \geq 0$ and α_i is a real number. A posynomial function f is a sum of monomials.

$$f(x_1, x_2, ...x_n) = \sum_{i=0}^{k} c_i \prod_{j=0}^{n} x_j^{\alpha_{ij}} \tag{13}$$

If the circuit design problem can be expressed in the form of a GP problem, then it is possible to use interior-point algorithms for solving the problem very efficiently in terms of computational complexity. Interior point methods for GP are: 1) extremely fast, 2) find globally optimal solution or provide proof of unfeasibility (i.e. specifications are too tight), and 3) independent of starting point [19] [20]. If for simplicity we consider the long channel model of a MOS transistor operating in saturation we easily realize that all relevant equations we need to describe an analog circuit can be put in the form of monomial functions of the process and design variables μC_{ox}, W, L and I_d:

$$\begin{aligned} V_{ov} &= \sqrt{2}\ \mu^{-0.5} C_{ox}^{-0.5} W^{-0.5} L^{0.5} I_d^{0.5} \\ g_m &= \sqrt{2}\ \mu^{0.5} C_{ox}^{0.5} W^{0.5} L^{-0.5} I_d^{-0.5} \\ C_{gs} &= \frac{2}{3} C_{ox}^{1} W^{1} L^{1} \end{aligned} \tag{14}$$

Analogously, it is possible to develop more complex GP models that take into account channel length modulation, body effect, short channel effect, and other second-order effects and non idealities necessary to adequately describe the large and small signal characteristics of nanoscale MOS transistors.

The key contribution of the approach proposed is to model the dependence between the variation in the performance constraints (e.g bandwidth, gain, power consumption, etc.) and the variation in the design parameters (e.g. transistor sizes, threshold voltage, oxide thickness, etc.). For a given yield requirement, the performance yield information is mapped back onto the design space and used to determine the associated size of the uncertainty set. With respect to the notion of budget of uncertainty associated to a given yield requirement, we can rewrite the expression of the conic uncertainty set as:

$$\{\|A\delta X(\eta) + b\|_2 \leq s_{max}\Omega(\eta)\} \tag{15}$$

where $\Omega(\eta)$ is a factor scaling the conic set size according to the specified yield requirement η. The scaling factor $\Omega(\eta)$ is restricted to lie in the interval $[0, 1]$. If $\Omega(\eta) = 0$, there is no protection against parameter variability. If $\Omega(\eta) = 1$, the performance constraint is completely protected against variability. If $\Omega(\eta)$ is between 0 and 1, there exists a trade off between yield protection and parameter variability. Assuming the physical parameter variations δX follow a multivariate gaussian distribution $\delta X \sim \mathcal{N}_n(\mu, \Sigma)$, then the yield associated variations $\delta X(\eta)$ will also be gaussian distributed and of the form $\delta X(\eta) \sim \mathcal{N}_n(\Omega\mu, \Omega^2\Sigma)$. The subscript n indicates that δX and $\delta X(\eta) \in \mathbb{R}^n$. Expressing the effect of the variations on the performance constraint functions in terms of the desired yield probability:

$$Prob\{f(X_0 + \delta X(\eta)) \leq f_{limit}\} \geq \eta \tag{16}$$

and approximating the constraint functions with a fist order Taylor expansion we can derive that:

$$\begin{aligned} & Prob\{f(X_0 + \delta X(\eta))) \leq f_{limit}\} \simeq \\ & \simeq \Phi\left(\frac{f_{limit} - \Omega\mu}{\Omega\sigma}\right) \geq \eta \end{aligned} \tag{17}$$

and therefore extract Ω from η by simply inverting the well know and tabulated [21] cumulative distribution function (CDF) $\Phi(\cdot)$.

B. Framework Implementation and underlaying algorithms

Figure 5 illustrates our proposed framework for optimizing the design of analog circuits and systems subject to process variations. The uncertainty budget introduced to account for variability is modeled using a second order cone set. The size and shape of the uncertainty set is dynamically adjusted based on yield requirement. The framework acts as a wrapper around the nominal design, the physical technology, the variability model, the performance requirements and the GP convex optimization software (in our case CVX [22]). The framework consists of six steps.

- Design exploration
- Performance extraction
- Objective and constrain functions construction
- SOC size fitting
- Yield mapping and constraints formatting
- GP convex optimization solver

The first step consists of simulating a large number of perturbations of the nominal circuit. The various circuit instances simulated are automatically generated based on the perturbation model provided. The nominal circuit and the technology model are provided in a SPICE-like format.

The second step consists of analyzing all simulation results and extract from them a numerical function for each performance metric of interest. In our case the performance functions extracted were gain, bandwidth and power dissipation. However, thanks to its modularity the framework can be easily extended to any performance metrics.

The third steps is two fold. Initially, we approximate each performance function with a monomial, and then we further

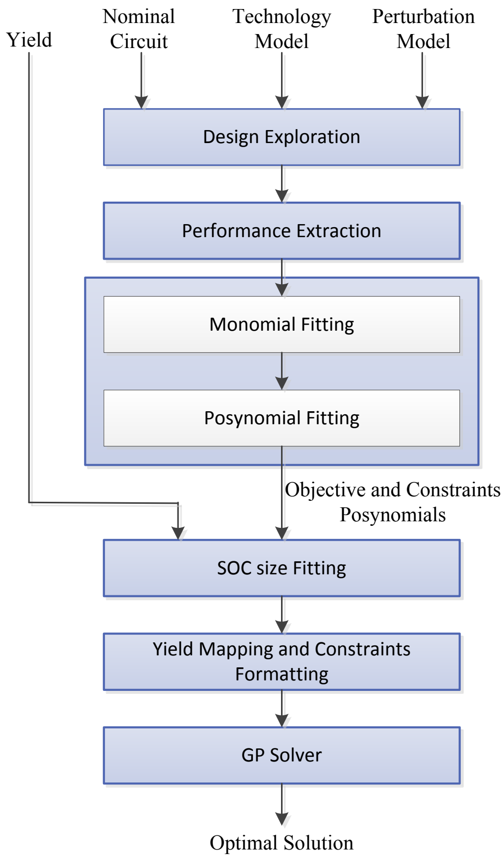

Fig. 5. Framework Structure

improve the approximation by taking the initial monomial and fitting it in a posynomial with K terms. In our framework we used K=5. The choice of K is a trade off between accuracy and computational time. Both the objective and inequality constraints functions must be put in posynomial form to be processable by the GP convex optimization solver. Algorithm 1 summarizes the steps for constructing a monomial function. Algorithm 2 summarizes the steps for transforming a monomial in a posynomial with K terms.

Algorithm 1 Monomial Fitting

```
procedure BuildMono(X^(i), f^(i))
▷ Given N data points (X^(i), f^(i)) with i=1,..,N
▷ X^(i) ∈ R^n_+ and f^(i) ∈ R_+
▷ fit the data with a monomial: f̂(X) = c x_1^α1 ... x_n^αn

    for i ← 1 to N do
        Y^(i) ← log X^(i)                    ▷ log transform
        z^(i) ← log f^(i)                    ▷ log transform
    end for
    ▷ find least-square approximation
    minimize Σ_{i=1}^N ( log c + α_1 y_1^(i) + ... + α_n y_n^(i) − z^(i) )

    return c, α_1, ...., α_n

end procedure
```

The fourth step computes the size of the SOC uncertainty of each constraints function according to (10).

The fifth step performs 1) the mapping between yield requirement and parameter variability summarized in (17), and 2) reformats the constraints functions back into a posynomial form. Since as part of the mapping process the constraints functions $f(X)$ are approximated by a first order Taylor expansion they take the form:

$$f(X) = f(X_0 + \delta X) \approx f(X_0) + \nabla f(X_0) \cdot \delta X = \\ = f(X_0) + \sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i} \right) \bigg|_{x_{i0}} \delta x_i \tag{18}$$

and can be expressed as:

$$f(X_0) + \max_{\forall \delta X} \left\{ \sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i} \right) \bigg|_{x_{i0}} \delta x_i \right\} \leq f_{limit} \tag{19}$$

Unfortunately, it can be easily worked out that the derivative term in (19) is not posynomial. This is because the coefficients $\alpha_{ik} c_k$ in (21) can be either positive or negative.

$$f(X) = \sum_{k=1}^{K} c_k \cdot \prod_{l=1}^{n} x_l^{\alpha_{lk}} \tag{20}$$

$$\frac{\partial f}{\partial x_i} = \sum_{k=1}^{K} \alpha_{ik} c_k \cdot \prod_{l \neq i}^{n} x_l^{\alpha_{lk}} \cdot x_i^{\alpha_{ik}-1} \tag{21}$$

By substituting (20) and (21) in (19) the constraints can be rewritten as in (23):

$$\sum_{k=1}^{K} c_k \cdot \prod_{l=1}^{n} x_l^{\alpha_{lk}} + \\ + \max_{\forall \delta X} \left\{ \sum_{i=1}^{n} \left(\underbrace{\sum_{k=1}^{K} \alpha_{ik} c_k \cdot \prod_{l \neq i}^{n} x_l^{\alpha_{lk}} x_i^{\alpha_{ik}-1}}_{= g_{ik}(X)} \right) \delta x_i \right\} \leq f_{limit} \tag{22}$$

$$\sum_{k=1}^{K} c_k \cdot \prod_{l=1}^{n} x_l^{\alpha_{lk}} + \max_{\forall \delta X} \left\{ \sum_{i=1}^{n} \left(\sum_{k=1}^{K} g_{ik}(X) \right) \delta x_i \right\} \leq f_{limit} \tag{23}$$

The obstacle caused by the presence of negative coefficients can be solved by properly reformatting the constraints expressions. To this end we introduce two vectors $\phi_+, \phi_- \in \mathbb{R}^n$ to collect the positive and negative coefficients. Thanks to the introduction of the vectors ϕ_+, ϕ_- we can further manipulate (23):

$$\sum_{k=1}^{K} c_k \cdot \prod_{l=1}^{n} x_l^{\alpha_{lk}} + \max_{\forall \delta X} \left\{ \sum_{i=1}^{n} \left(\sum_{k=1}^{K} g_{ik}(X) \right) \delta x_i \right\} = \\ = \sum_{k=1}^{K} c_k \cdot \prod_{l=1}^{n} x_l^{\alpha_{lk}} + \max_{\forall \delta X} \{ \langle \phi_+, \delta X \rangle + \langle \phi_-, \delta X \rangle \} \leq f_{limit} \tag{24}$$

Algorithm 2 Posynomial Fitting

procedure BUILDPOSY$(c, \alpha_1, ..., \alpha_n, X^{(i)}, f^{(i)})$
▷ Given N data points
▷ $(X^{(i)}, f^{(i)})$ with i=1..N , $X^{(i)} \in \mathbb{R}^n_+$, $f^{(i)} \in \mathbb{R}_+$
▷ and its monomial approximation (c, $\alpha_1, ..., \alpha_n$)
▷ fit the data with a K terms posynomial:
▷ $\hat{f}(X) = \sum_{j=1}^{K}(c/K)x_1^{\lambda_{j,1}}...x_n^{\lambda_{j,n}}$

  ▷ initialize posynomial to monomial
  $\hat{f}(X) \leftarrow \sum_{j=1}^{K}(c/K)x_1^{\alpha_1}...x_n^{\alpha_n}$
  ▷ compute root mean square error of monomial
  rmseMono $\leftarrow \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{f}^{(i)} - f^{(i)}\right)^2}$
  ▷ set target rmse
  rmseTarget $\leftarrow 0.9 \times$ rmseMono ▷ improve 10%
  $p \leftarrow 1$ ▷ perturbation trials
  $index \leftarrow 0$ ▷ track best posynomial
  rmsErr $\leftarrow$ rmseMono ▷ smallest acceptable rmse
  repeat
    ▷ pick $K \times n$ normally distributed random values
    ▷ $\delta_{j,l}$ (elements of matrix Δ)
    $\Delta \leftarrow randn(K, n)$ ▷ mean=0, stddev=1
    ▷ make the standard deviation 0.2 $|\alpha_l|$
    for $j \leftarrow 1$ to K **do**
      for $l \leftarrow 1$ to n **do**
        $\delta_{j,l} \leftarrow 0.2 * \alpha_l * \delta_{j,l}$
        $\lambda_{j,l} \leftarrow \alpha_l + \delta_{j,l}$
      end for
    end for
    ▷ perturb the exponents in the monomial terms
    $\hat{f}_p(X) \leftarrow \sum_{j=1}^{K}(c/K)x_1^{\lambda_{j,1}}...x_n^{\lambda_{j,n}}$
    ▷ compute root mean square error of posynomial
    rmsePoly $\leftarrow \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{f}_p{}^{(i)} - f^{(i)}\right)^2}$
    if rmsePoly $\leq$ rmsErr **then**
      rmsErr $\leftarrow$ rmsePoly
      index $\leftarrow p$ ▷ pick the best among the trials
    end if
    $p \leftarrow p + 1$
  until rmsePoly $\leq$ rmseTarget or $p \geq p_{max}$
  if $index < 1$ **then** ▷ fitting did not converge
    ▷ must return the original monomial
    for $j \leftarrow 1$ to K **do**
      for $l \leftarrow 1$ to n **do**
        $\lambda_{j,l} \leftarrow \alpha_l$
      end for
    end for
  end if

  return Λ ▷ size$(\Lambda) = K \times (n+1)$

  ▷ $\Lambda = \begin{bmatrix} c/K & \lambda_{1,1} & ... & \lambda_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ c/K & \lambda_{K,1} & ... & \lambda_{K,n} \end{bmatrix}$

end procedure

where $\langle a, b\rangle$ denotes the inner product of two vectors a and b. If we use Cauchy-Schwartz inequality:

$$\langle a, b\rangle \leq \|a\|_2 \cdot \|b\|_2 \tag{25}$$

and recall the SOC uncertainty is modeled as $\|\delta X\|_2 \leq s$ we can rewrite (24) as:

$$\begin{aligned} &\sum_{k=1}^{K} c_k \cdot \prod_{l=1}^{n} x_l^{\alpha_{lk}} + \|\phi_+\|_2 \cdot s + \|\phi_-\|_2 \cdot s \leq f_{limit} \\ &s_{min} \leq s \leq s_{max} \end{aligned} \tag{26}$$

At this point, applying the properties of the 2-norm of a vector we can introduce two new positive variables r_1 and r_2:

$$\begin{aligned} r_1 = \|\phi_+\|_2 \Leftrightarrow r_1^2 = \phi_+^T\phi_+ \\ r_2 = \|\phi_-\|_2 \Leftrightarrow r_2^2 = \phi_-^T\phi_- \end{aligned} \tag{27}$$

and conclude that the non-posynomial constraints in (19) can be replaced by a set of posynomial constraints in the variable set (X, r_1, r_2, s). The set of posynomial constraints can be put in the form:

$$\begin{aligned} &\sum_{k=1}^{K} c_k \cdot \prod_{l=1}^{n} x_l^{\alpha_{lk}} + r_1 \cdot s + r_2 \cdot s \leq f_{limit} \\ &\phi_+^T\phi_+ r_1^{-2} \leq 1 \\ &\phi_-^T\phi_- r_2^{-2} \leq 1 \\ &X_{min} \leq X \leq X_{max} \\ &s_{min} \leq s \leq s_{max} \end{aligned} \tag{28}$$

Finally, the sixth step solves the GP convex optimization problem. An optimal assignment of the design parameters that minimizes the performance objective subject to the given constraints in presence of process variations is obtained.

III. EXPERIMENTAL RESULTS

To test our approach we applied it to the design of a 90 nm CMOS differential amplifier and compared the results with those obtained using the ellipsoidal uncertainty method. The amplifier schematic is shown in Figure 6. The framework is implemented using MATLAB integrated with HSPICE for circuit simulation, and CVX for GP optimization. The goal of the design was minimizing power consumption while satisfying a minimum gain requirement.

$$\begin{aligned} \text{minimize: } & \text{Power}\,(W, L, V_{\text{th}}, T_{ox}) \\ \text{subject to: } & G_v\,(W + \delta W, L + \delta L, V_{\text{th}} + \delta V_{\text{th}}, T_{ox} + \delta T_{ox}) \\ & \geq G_{\text{limit}} \end{aligned} \tag{29}$$

where G_v is the DC voltage gain expressed in dB, and G_{limit}=51 dB is the lower bound limit we want to satisfy. The design variables used in the optimization process include the design parameters W (channel width) and L (channel length) and the process parameters V_{th} (threshold voltage) and T_{ox} (oxide thickness).

In this example, we set a 100% yield requirement, and perform robust design optimization using both the EU model and the CUDB model. The results obtained running Montecarlo simulations show that the optimal design achieved using the EU method causes about 16% performance violation (Figure

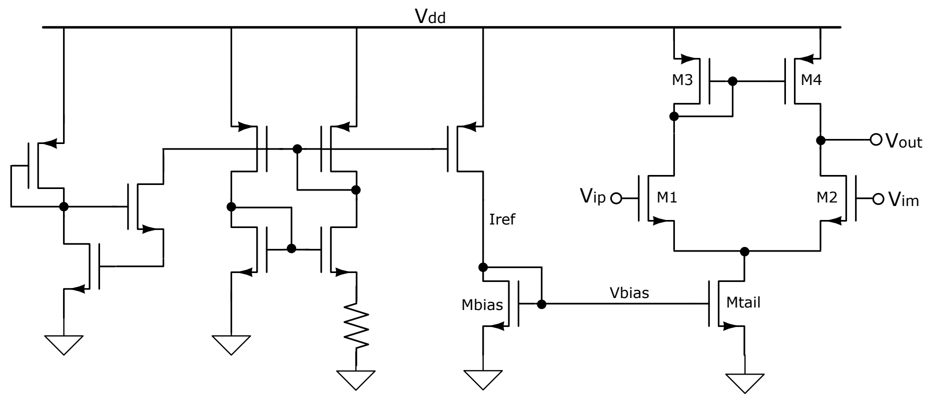

Fig. 6. Schematic of CMOS differential amplifier

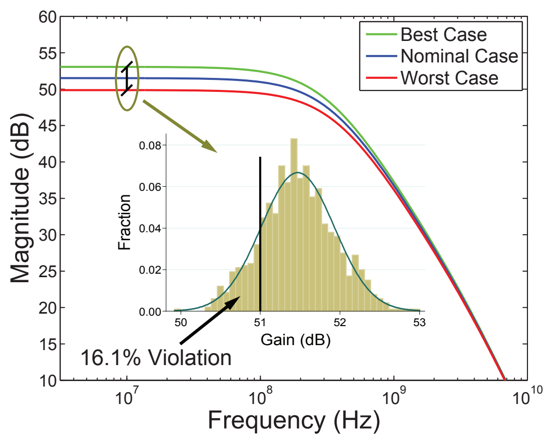

Fig. 7. Histogram of gain distribution for differential pair design using EU

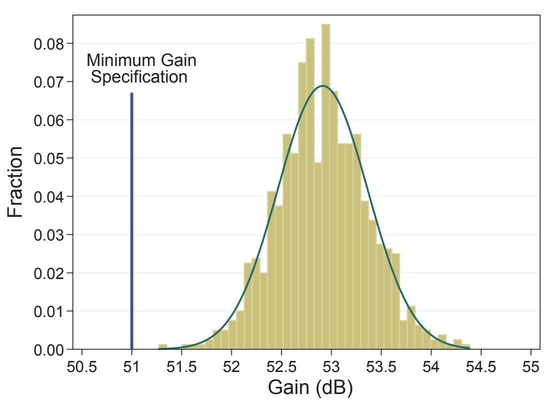

Fig. 8. Histogram of gain distribution for differential pair design using CUDB

7), while the solution generated by the proposed method perfectly meets the the required gain specification (Figure 8).

The proposed method (CUDB) maps the performance space back onto the design space to form a yield-associated uncertainty set. On the contrary, the EU method relies only on the distribution variations in design space. Due to the highly nonlinear behavior of analog circuits, neglecting the mapping relationship between design parameters and performance may lead to a very conservative estimation even if the yield specification can be guaranteed.

Figure 9 compares the power consumption achieved by EU and CUDB for different yield specifications. At 100% yield specification, a minor reduction of power consumption (about 8%) by the proposed method can be observed. The improvement increases significantly when lower yield is required, and becomes stable with yield specification below 80%. On average the proposed method has achieved a 18% reduction of design cost.

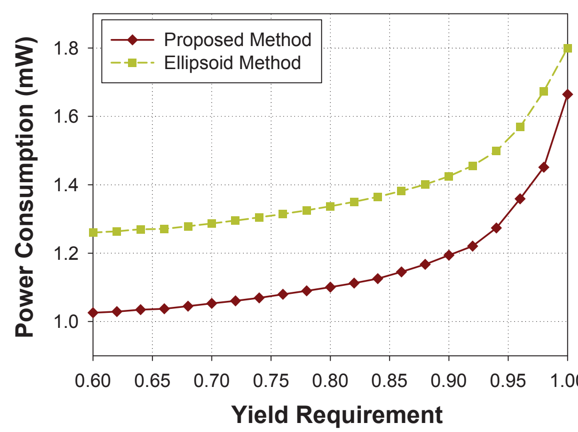

Fig. 9. Comparison of power consumption between ellipsoidal method and proposed method for differential pair design

IV. Conclusion

This paper presents a new robust optimization framework for the design of analog circuits and systems in presence of process variations. The proposed framework is based on a conic uncertainty model that incorporates the concept of budget of uncertainty by associating the performance yield requirement with the size and shape of the conic uncertainty set. With process variations characterized through the proposed model, the design of analog circuits and systems can be posed in the form of a robust optimization problem that can be reformulated and efficiently solved as a general GP convex problem. Experimental results based on the application of the framework on the design of a differential amplifier show that, when compared to other existing methods, our method produces a larger number of feasible solutions and a design performance improvement up to 18%. Future work will include validation on larger scale designs and iterative refinements of the models based on earlier results. We also plan to extend the framework to as many performance metrics as possible including noise, stability and settling time.

REFERENCES

[1] Y. Xu, K.-L. Hsiung, X. Li, L. T. Pileggi, and S. P. Boyd, "Regular analog/RF integrated circuits design using optimization with recourse including ellipsoidal uncertainty," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 5, pp. 623–637, 2009.

[2] D. Boning and S. Nassif, "Models of process variations in device and interconnect," in *Design of High-Performance Microprocessor Circuits, chapter 6*, A. Chandrakasan, W. J. Bowhill, and F. Cox, Eds. Piscataway, NJ: Wiley-IEEE Press, 2001, pp. 98–115.

[3] M. Orshansky, L. Milor, and C. Hu, "Characterization of spatial intrafield gate cd variability, its impact on circuit performance, and spatial mask-level correction," *IEEE Trans. Semicond. Manuf.*, vol. 17, no. 1, pp. 2–11, 2004.

[4] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM Review*, vol. 53, no. 3, pp. 464–501, 2011.

[5] M. M. Hershenson, S. P. Boyd, and T. H. Lee, "Optimal design of a cmos op-amp via geometric programming," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 1, pp. 1–21, 2001.

[6] R. J. Duffin, E. L. Peterson, and C. Zener, *Geometric Programming: Theory and Application*. Hoboken, NJ: John Wiley and Sons, 1967.

[7] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY: Cambridge University Press, 2004.

[8] L. Cheng, J. Xiong, and L. He, "Non-linear statistical static timing analysis for non-gaussian variation sources," in *Proc. DAC*, 2007, pp. 250–255.

[9] B. Hargreaves, H. Hult, and S. Reda, "Within-die process variations: How accurately can they be statistically modeled?" in *Proc. ASPDAC*, 2008, pp. 524–530.

[10] G. Dantzig, "Linear programming under uncertainty," *Management Sci.*, pp. 197–206, 1955.

[11] G. Infanger, *Planning under Uncertainty: Solving Large-Scale Stochastic Linear Programs*. San Francisco, CA: Boyd and Fraser, 1994.

[12] D. Bertsimas and M. Sim, "Robust convex optimization," *The Price of Robustness*, vol. 52, no. 1, pp. 35–53, 2004.

[13] D. Bertsimas and A. Thiele, "Robust and data-driven optimization: Modern decision-making under uncertainty," in *Tutorials in Operations Research: Models, Methods, And Applications for Innovative Decision making*. INFORMS, 2006.

[14] A. Ben-Tal and A. Nemirovski, "Robust solutions of linear programming problems contaminated with uncertain data," *Mathematical Programming*, vol. 88, no. 3, pp. 411–424, 2000.

[15] J. Singh, V. Nookala, Z. Luo, and S. Sapatnekar, "A geometric programming-based worst case gate sizing method incorporating spatial correlation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 53, no. 3, pp. 464–501, 2011.

[16] J. Xiong, V. Zolotov, and L. He, "Robust extraction of spatial correlation," in *Proc. ISPD*, 2006, pp. 2–9.

[17] H. Chang and S. Sapatnekar, "Statistical timing analysis considering spatial correlation using a single pert-like traversal," in *Proc. ICCAD*, 2003, pp. 621–625.

[18] M. L. Stein, *Interpolation of Spatial Data*. New York, NY: Springer, 1999.

[19] D. M. Colleran, C. Portmann, A. Hassibi, C. Crusius, S. Mohan, S. Boyd, T. H. Lee, and M. Hershenson, "Optimization of phase-locked loop circuits via geometric programming," in *Proc. CICC*, 2003, pp. 377–380.

[20] S. Boyd, S. J. Kim, L.Vandenberghe, and A. Hassibi, *Tutorial on Geometric Programming*. Stanford, CA: Stanford University, 2004.

[21] M. Abramovitz and I. Stegun, *Handbook of Mathematical Functions*. New York, NY: Dover Pub., 1965.

[22] S. P. Boyd. (2013) CVX research. [Online]. Available: http://cvxr.com/cvx/