Efficient Analog Circuit Optimization Using Sparse Regression and Error Margining

Mohamed Baker Alawieh[1], Fa Wang[2], Rouwaida Kanj[1], Xin Li[2] and Rajiv Joshi[3]
[1]ECE Department, American University of Beirut, Beirut, Lebanon
[2]ECE Department, Carnegie Mellon University, Pittsburgh, PA, USA
[3]IBM TJ Watson Lab, Yorktown Heights, NY, USA
{mma144, rk105}@mail.aub.edu, {fwang,xinli}@ece.cmu.edu, rvjoshi@us.ibm.com

ABSTRACT

In this paper, we propose a novel analog circuit optimization methodology for achieving high parametric yield. We solve the statistical worst-case optimization problem by a sequence of linear programings where performance metrics are fitted using sparse regression to take into account a large number of device-level parameters modeling process variations. In addition, we propose a margining mechanism to ensure accurate yield optimization with consideration of modeling errors. The efficacy of this method is demonstrated using two circuit examples where the cost function is minimized and high parametric yield (e.g., around 90%) is achieved compared to other conventional approaches.

1. INTRODUCTION

With the expeditious evolution of communication systems and consumer electronics, demand for low power and high performance Integrated Circuits (IC) has increased tremendously [1]-[2]. Meanwhile, continuous scaling of IC technologies imposes new challenges facing these integrated systems. Particularly, process variations manifest themselves as uncertainties in the performance and reliability of analog/RF design blocks. Robust analog circuit optimization in modern manufacturing is becoming increasingly important to ensure high yield [1]-[4].

Analog/RF circuit optimization has been extensively studied in literature [5]-[10]. Equation-based methods rely on analytical performance models to approximate circuit-level performance (e.g. bandwidth, gain, etc.) as a function of design variables [5]. It is not trivial, however, to build analytical equations for complex performance metrics particularly in the presence of process variations, and designers often rely on circuit-simulation based optimization [8]-[7]. Nominal optimization typically minimizes a cost function by pushing other performance constraints to their boundaries. However, due to process variations a nominally-optimized design can easily violate performance specifications. To overcome this problem, corner-based optimization was introduced. It optimizes the design at all process corners representing the extreme values of process parameters [5]. However, it is not guaranteed that the worst-case performance takes place at any one of these corners. Statistical optimization relies on circuit simulations to build response surface models [11]-[12] that represent performance as a function of the design variables and process variations [5]. Among these methods are the worst-case optimization and design-centering methods [8]-[9]. The worst-case method aims at optimizing the worst-case performance under process variations [8], whereas the design centering tries to find the design point furthest from all constraints boundaries [5]. Such methods are used in an iterative local optimization framework [8], [18]. Thus, at each iteration the performance models are built and a local optimization is performed over a small design sub-space, and the subspace is moved in the direction of the local optimum. This way, regression-based models, built over small subspaces, are able to accurately capture the performance behavior. In the past, these methods were able to efficiently optimize the analog design; however, today they suffer from the dimensionality problem. With hundreds to thousands of process variation parameters affecting a typical design, building the performance models using an over-determined system of equations requires a huge number of circuit simulations. Hence, facing the dimensionality challenge, conventional approaches seem ill-equipped to efficiently optimize analog designs.

In this paper, we propose a novel sparse regression based statistical optimization framework with an error margining mechanism to address the aforementioned challenges. We adopt sparse regression techniques to build the linear performance models using a small number of simulations [12]-[19]. These process-variation dependent models are then used to derive worst-case performance models which in turn are used in the local optimization framework [5], [8].

However, a new challenge arises due to modeling error. In practice, this error can lead to a discrepancy between the model estimation and the circuit behavior thereby affecting the true parametric yield. In [20], the authors map the error of posynomial performance models to an uncertainty set for robust optimization. In this paper, we adopt an error margining mechanism for the performance models based on error statistics to ensure high parametric yield.

The remainder of this paper is organized as follows. In Section 2, we provide a background review of sparse regression. We present the proposed statistical optimization framework in Section 3. The results and analysis are presented in Section 4, and the conclusions are presented in Section 5.

978-1-5090-1213-8/16/$31.00 ©2016 IEEE

2. BACKGROUND REVIEW

In a traditional linear regression framework, a function $f(\mathbf{x})$ is modeled as a linear combination of the independent variables vector $\mathbf{x}$ [12] as follows:

$$f(\mathbf{x}) \approx \sum_{i=1}^{D} \alpha_i \cdot x_i + C \,, \tag{1}$$

where $\{\alpha_i;\ i = 1, 2, 3,\ldots, D\}$ are the model coefficients, $\mathbf{x}=\{x_i;\ i = 1, 2, 3,\ldots, D\}$ represents the D-dimensional variable vector, and C is a constant term. Given N sample points, the set of equations used to solve for the model coefficients in equation (1) is given by:

$$\mathbf{X} \cdot \boldsymbol{\alpha} + C = \mathbf{F} \,, \tag{2}$$

where

$$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_2^{(2)} & \cdots & x_D^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_D^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \cdots & x_D^{(N)} \end{bmatrix} \tag{3}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_D \end{bmatrix}^T \tag{4}$$

$$\mathbf{F} = \begin{bmatrix} f^{(1)} & f^{(2)} & \cdots & f^{(N)} \end{bmatrix}^T \,, \tag{5}$$

and $f^{(n)}$ and $x^{(n)}$ stand for the values of f and x at the n^{th} sample point respectively. Typically, N > D, and the system of equations (2) is overdetermined and the model parameters are determined via least squares method [21].

2.1 Sparse Regression

Hundreds of process variation variables affect state-of-the-art designs and pose a challenge to the modeling problem thereby requiring an unreasonably large number of sample points to fit the model using least squares method. The high-dimensional challenge, however, is accompanied by the special feature of sparsity [12]. Different methods have been proposed in literature to make use of this feature in model fitting [12]- [18]. Indeed, although the size of α is large due to the large number of variables, only few of these variables are required to estimate f, and the fitted coefficients' vector is expected to have only few non-zero elements. Therefore, it is possible to solve for the model coefficients from an under-determined system of equations using sparse regression. The problem can be formulated mathematically as an L_0-norm regularization [12]:

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} && \|\mathbf{X} \cdot \boldsymbol{\alpha} + C - \mathbf{F}\|_2^2 \\ &\text{s.t.} && \|\boldsymbol{\alpha}\|_0 \le \lambda \end{aligned} \,, \tag{6}$$

where $\|\bullet\|_0$ and $\|\bullet\|_2$ stand for the L_0-norm and L_2-norm respectively. The L_0-norm regularization problem in (6) is NP hard, and L_1-norm regularization is usually used instead to solve for a sparse solution as presented in (7).

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} && \|\mathbf{X} \cdot \boldsymbol{\alpha} + C - \mathbf{F}\|_2^2 \\ &\text{s.t.} && \|\boldsymbol{\alpha}\|_1 \le \lambda \end{aligned} \,, \tag{7}$$

$\|\bullet\|_1$ is the L_1-norm of a vector and represents the sum of the absolute values of the elements of the vector, and (7) can be restated by introducing slack variables as the convex optimization problem in (8) [21].

$$\begin{aligned} &\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} && \|\mathbf{X} \cdot \boldsymbol{\alpha} + C - \mathbf{F}\|_2^2 \\ &\text{s.t.} && \sum_{i=1}^{D} \beta_i \le \lambda \\ & && -\beta_i \le \alpha_i \le \beta_i \quad (i = 1, 2, \cdots, D) \end{aligned} \,. \tag{8}$$

However, Parameter λ is not known apriori, and its optimal value is determined by solving the optimization problem (8), for example using the interior point method, for different values of λ. This can be costly. Alternatively, orthogonal matching pursuit can be used [12] to identify the critical variables iteratively as described in Algorithm 1. Thus, the importance of a variable is determined by computing its correlation with $\mathbf{F}$:

$$c_i = \left|\mathbf{X}_i^T \cdot \mathbf{F}\right| \quad (i = 1, 2, \cdots, D) \,, \tag{9}$$

where $\mathbf{X}_i$ is the normalized i^{th} column of matrix $\mathbf{X}$. The vector $\mathbf{X}_s$ with the highest correlation is chosen and least squares fitting is used to estimate $\mathbf{F}$:

$$\mathbf{F} \approx \phi_1 \mathbf{X}_s + C \,. \tag{10}$$

After that, the residual R is computed using:

$$\mathbf{R} = \mathbf{F} - \phi_1 \mathbf{X}_s \,, \tag{11}$$

and the correlation between $\mathbf{R}$ and the remaining vectors of $\mathbf{X}$ is computed to get the next most correlated vector and include it in the model and so on until λ vectors are chosen. This method is further repeated for different values of λ.

Algorithm 1: Orthogonal Matching Pursuit (OMP)

1. Start by setting $\mathbf{R} = \mathbf{F}$ and $\Omega = \{\}$. Initialize the index $k = 0$.
2. Compute the correlation between $\mathbf{R}$ and vectors of $\mathbf{X}$
$$c_i = \left|\mathbf{X}_i^T \cdot \mathbf{R}\right| \quad (i = 1, 2, \cdots, D) \,. \tag{12}$$
3. Identify index s of the largest value c_s. $\Omega = \Omega \cup \{s\}$.
4. Estimate $\mathbf{F}$ as a linear combination of all elements in Ω:
$$\min_{\phi} \left\| \sum_{i\in\Omega} \mathbf{X}_i \cdot \phi_i + C - \mathbf{F} \right\|_2^2 \tag{13}$$
5. Update $\mathbf{R} = \mathbf{F} - \sum_{i\in\Omega} \mathbf{X}_i \cdot \phi_i$
6. If $k < \lambda$, set $k = k + 1$ and go to Step 2. Otherwise, stop.

3. PROPOSED OPTIMIZATION FLOW

The objective of optimizing an analog circuit is to find the optimal design point that meets required performance specification while minimizing the cost represented usually by power and/or area. The optimization problem can be formulated as follows:

$$\begin{aligned} &\min_{\mathbf{x}} && f(\mathbf{x},\mathbf{p}) \\ &\text{s.t.} && G_m^l \le g_m(\mathbf{x},\mathbf{p}) \le G_m^u \quad (m = 1, 2, \cdots, M) \,, \\ & && l_d \le x_d \le u_d \qquad\qquad (d = 1, 2, \cdots, D) \end{aligned} \tag{14}$$

where, $\mathbf{x}$, and $\mathbf{p}=\{\ p_i;\ i = 1, 2, 3,\ldots, D_p\}$ are the design variables vector and vector of normalized process variations respectively. $f(\mathbf{x},\mathbf{p})$ is the performance metric to be minimized (e.g. power consumption), $\{g_m(\mathbf{x},\mathbf{p});\ m = 1, 2, \cdots, M\}$ contains the performance metrics (e.g. gain) constrained by the specifications $\{G^l_m$ and G^u_m; $m = 1, 2, \cdots, M\}$, and $\{l_d$; $d = 1, 2, \cdots, D\}$ and $\{u_d$; $d = 1, 2, \cdots, D\}$ define the lower and upper bounds for the d^{th} design variable respectively.

3.1 Optimization Methodology Overview

Figure 1 presents an overview of the proposed methodology. To enable statistical optimization, we propose to derive a worst case model for each of the performance metrics as a function of the design variables only. This requires first deriving models in the high-dimensional process and design space using true circuit simulations. Furthermore, the optimization may be impacted by the modeling error, and the true circuit yield may not adhere to that obtained from the model based optimization. To address this challenge, an error margining mechanism is adopted. Due to the nonlinearity of the underlying metrics, the approach aims at solving the given optimization problem over a local design sub-space in an iterative manner.

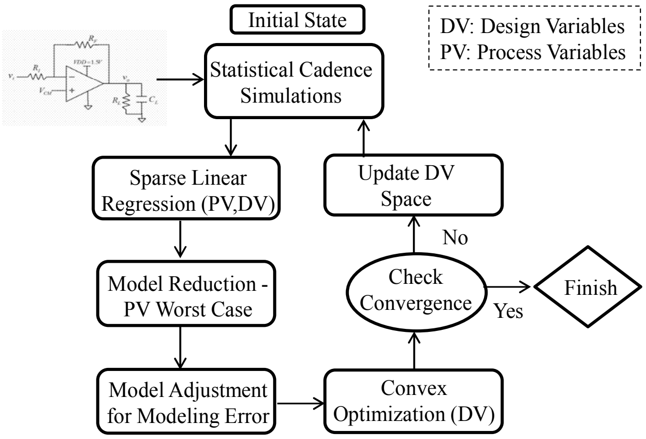

Initial State
DV: Design Variables
PV: Process Variables
Statistical Cadence Simulations
Sparse Linear Regression (PV,DV)
Update DV Space
No
Model Reduction - PV Worst Case
Check Convergence
Finish
Yes
Model Adjustment for Modeling Error
Convex Optimization (DV)

Figure 1. The flow chart summarizing the proposed optimization methodology is shown.

3.1.1 Local Design Space Approach

Most performance metrics are highly nonlinear over the entire design space and including the process parameters will introduce additional nonlinearity to the model [1]. In the proposed convex optimization framework, our goal is to represent the performance metrics with linear models. To address this challenge, we rely on a local design sub-space iterative optimization flow [5].The optimization starts from an initial design point and the design search space is defined by a design window around the starting point. Over this small sub-space, accurate transistor-level simulations obtained from Cadence environment are used to fit locally generated linear models to represent the performance metrics with good accuracy. This in turn enables solving a local optimization problem. The resulting local optimum would serve as the new center for the design sub-space of the next iteration. This method will continue spanning the design space until two consecutive local optima are very close; hence the method converges to the overall optimal design point. Without loss of generality, Figure 2 illustrates the iterative local design space approach for a two dimensional example.

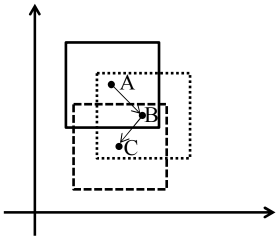


Figure 2. The iterative local optimization approach is shown for a two dimensional case. The optimization starts from A as the initial point, and the solid-line square as the local design search sub-space. B and C represent the new design centers for iterations 2 and 3 respectively.

3.1.2 Sparse Linear Regression

Due to the high dimensionality of the problem, we rely on the sparse regression based OMP method presented in Algorithm 1 for modeling the design performance metrics and cost functions. This enables accurate modeling with reduced number of transistor-level simulations, and is key to speeding up the optimization flow. The performance metrics, f_i, are modeled as linear functions of both the design variables and process parameters according to (15).

$$f(\mathbf{x},\mathbf{p}) \approx \sum_{i=1}^{D} \delta_i \cdot x_i + \sum_{i=1}^{P} \theta_i \cdot p_i + C\,, \tag{15}$$

$\{\delta_i$; $i = 1, 2, 3, \ldots, M\}$ and $\{\theta_i$; $i = 1, 2, 3, \ldots, P\}$ are the model coefficients associated with the design variables and process parameters respectively. Our goal is to find the smallest number of non-zero coefficients that guarantee the minimum modeling error; in practice, we sweep λ and select the value that results in the smallest modeling error.

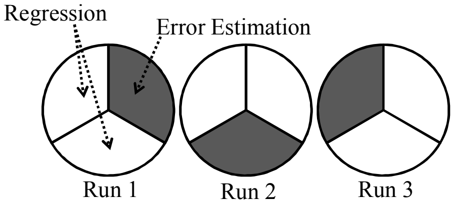


Figure 3. Example of 3-fold cross validation. Data is split into 3 different groups and error is estimated from 3 independent runs.

To avoid over-fitting, we rely on K-fold cross-validation such that the modeling error is computed from an independent set that is not used in the fitting stage [12]. Thus, the available data set is divided into K sets, as shown in Figure 3 and regression is done K times where in each time one of the groups is used to estimate the modeling error and all the remaining groups are used for regression. This will result in having K values for the modeling error, and the overall modeling error is computed as the average of the K error values.

3.1.3 Worst Case Model

For statistical optimization, the worst case model for a given performance function is derived by adding the worst case variation of the process parameters to the constant term in the model. This results in the model being expressed as a function of the design variables only. From this perspective, no worst case model is needed for the cost function since the constant term does not play any role in the optimization. Thus, we focus on deriving the worst case model for the constraints. Given a general constraint

$$G^l \leq g(\mathbf{x},\mathbf{p}) \leq G^u \,, \tag{16}$$

the worst case model ensures that the specifications are met even for large process variation-induced deviations. In fact, for the worst case optimization we target the 1% and 99% points of the metrics' cumulative distribution functions Given that the process parameters are normalized and follow an independent standard normal distribution, the distribution of the process variation term in (15) is

$$\left(\sum_{i=1}^{P} \theta_i \cdot p_i\right) \sim N(0,\sigma^2) \,, \tag{17}$$

where

$$\sigma_p^2 = \sum_{i=1}^{P} \sigma_i^2 \cdot \theta_i^2 = \sum_{i=1}^{P} \theta_i^2 \,. \tag{18}$$

Note that the standard deviation of the i^{th} process parameter $\sigma_i=1 \;\; \forall \; i$. σ_p can be used to compute the maximum deviation of the performance away from the nominal. Since the performance metric is bounded from both sides, the constraint in (16) is replaced by two constraints corresponding to the upper and lower bound respectively.

$$\begin{aligned} g_{WC}^l(x) &\approx \sum_{i=1}^{D} \delta_i \cdot x_i + C - 3 \cdot \sigma \geq G^l \\ g_{WC}^u(x) &\approx \sum_{i=1}^{D} \delta_i \cdot x_i + C + 3 \cdot \sigma \leq G^u \end{aligned} \tag{19}$$

3.1.4 Error Margining

The worst case optimization problem ensures a robust design with high yield under the assumption that the linear models, originally fitted using sparse regression, are accurate. However, this is not the case due to the modeling error which can result in a mismatch between the model-based optimization results and the actual circuit behavior. In fact, the results of the model-based optimization may perfectly meet the required specifications, but, the actual circuit behavior may not match these results. To address this challenge, we propose to use a margining mechanism for the modeling error. Margining for the modeling error is achieved by assuming that the modeling error follows a normal distribution,

$$error \sim N(0,\sigma_e^2) \,, \tag{20}$$

where $\sigma_e{}^2$ is the variance of the residual between the simulation data and the data fitted according to (15).

To illustrate, we consider the scatter plot in Figure 4 (a). It presents the plot of the fitted data $\hat{g}(\mathbf{x})$ versus the simulation data for an arbitrary performance metric $g(\mathbf{x})$. Without loss of generality, we will consider the case of the lower bound constraint:

$$g(\mathbf{x}) \geq G^l \,. \tag{21}$$

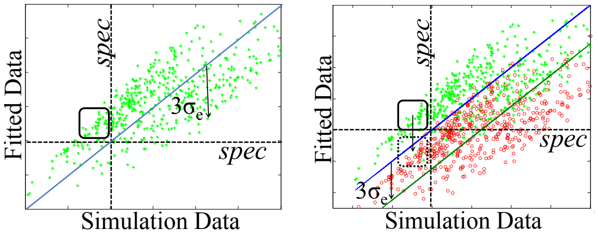
We notice that due to the modeling error, the points in the solid-line square which are considered 'passing' based on the model are 'failing' based on the actual simulation. We define

$$\hat{g}_E(\mathbf{x}) = g(\mathbf{x}) - 3\sigma_e \,. \tag{22}$$

Figure 4 (b) shows that, with 99% confidence, the systematic shift for error margining allows the following relation to hold

$$\hat{g}_E(\mathbf{x}) \geq G^l \Rightarrow g(\mathbf{x}) \geq G^l \,. \tag{23}$$

Hence, the simulation data will meet the specification if the fitted data with margining does. One can notice that the points in the solid-line square that were mispredicted as passing by the model in (a) are now pushed towards the failure region by the new model. With error margining, no points lie in the solid-line square in (b) which means that there is no mispredictions of fail by the model. The added margin in (22) ensures that*: if a point meets the specifications from the model perspective then it also meets the specifications in actual circuit simulations*.



(a) (b)

Figure 4. Scatter plots of an arbitrary metric (a) before and (b) after error margining. Points meeting the specification at the model level will pass the simulation test after margining.

In our methodology, the worst-case constraint in (19) is modified to account for error margining as follows:

$$g_{WCE}^l(\mathbf{x}) \approx g_{WC}^l(\mathbf{x}) - 3 \cdot \sigma_e \geq G^l \,. \tag{24}$$

Similar to the previous section, we are only interested in margining the constraint functions. The final version of the optimization problem with error margining is given by:

$$\begin{aligned} &\min_{\mathbf{x}} && f(\mathbf{x}) \\ &\text{s.t.} && g_{WCE_m}^l(\mathbf{x}) \geq G_m^l \quad (m=1,2,\cdots,M) \\ & && g_{WCE_m}^u(\mathbf{x}) \leq G_m^u \quad (m=1,2,\cdots,M) , \\ & && l_d \leq x_d \leq u_d \qquad (d=1,2,\cdots,D) \end{aligned} \tag{25}$$

3.2 Algorithm Flow

For a given iteration, the convex optimization problem is solved inside the local design sub-space using the worst case models with error margins. The new local optimum solution is compared with the previous solution and the optimization flow is terminated if both are within a certain tolerance. Else, the new design sub-space is centered at the new solution and another optimization iteration is performed again including building new models and margining…etc. The overall optimization framework is summarized in Algorithm 2.

Algorithm 2: Overall Optimization Framework

1. Start from an initial design point $\mathbf{x}^{(0)}$ derived from hand-calculation, and set $\mathbf{x}_s = \mathbf{x}^{(0)}$.
2. Run Cadence simulations to get sample points.
3. Build the linear models using sparse regression.
4. Derive the worst case models using equations (19).
5. Add the error margins according to equation (22).
6. Solve the convex optimization for $\mathbf{x}^{new}$ over the local design sub-space.
7. If $|\mathbf{x}^{new}$ □ $\mathbf{x}_s| \geq \zeta$, where ζ is a user-specified tolerance, set $\mathbf{x}_s = \mathbf{x}^{new}$, and update the design sub-space, and go to Step 2. Else, stop; the final solution is $\mathbf{x}_s$.

4. RESULTS AND ANALYSIS

In this section, we demonstrate the efficacy of the proposed iterative optimization framework using two circuit examples: (i) a two-stage operational amplifier (OpAmp) and (ii) a low noise amplifier (LNA) presented in Figure 5 and Figure 6 respectively. Simulations are performed using Cadence 45nm gpdk. All analysis is performed on a 3.4GHz Linux server with 8GB memory.

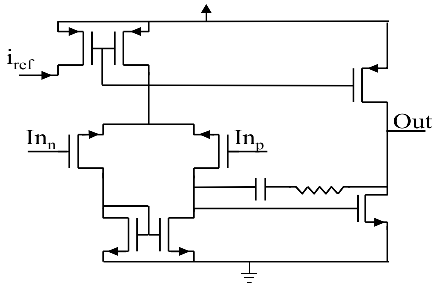


Figure 5. The schematic of the two-stage OpAmp.

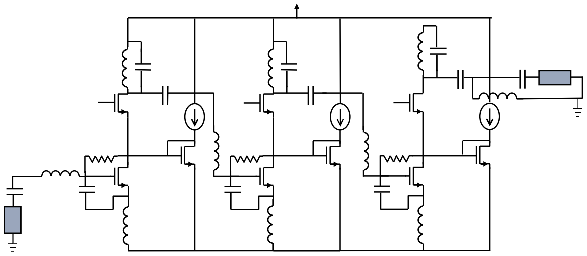
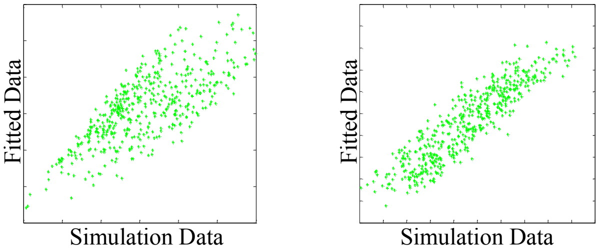


Figure 6. The schematic of the LNA used.

The transistor sizes and resistors, capacitors, and/or inductors values represent the design variables to optimize for. The goal is to find the optimal design point that minimizes the total power (P) consumption while meeting certain performance metric specification criteria. For the OpAmp the performance metrics include the Unity Gain Frequency (UGF), Gain (G), Phase Margin (PM), Slew Rate (SR), Input Offset (IO), and Output Swing(OS). For the LNA the performance metrics include the Noise Figure (NF), Forward Reflection (S11), Forward Transmission (S21), and Reverse Reflection (S22).

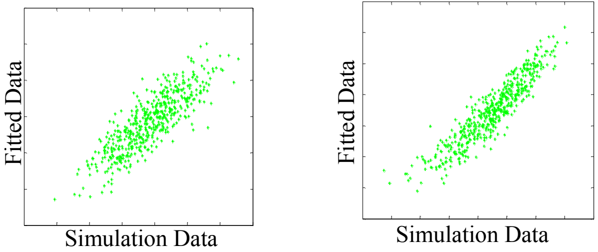
4.1 Performance Modeling

The variables included in the performance models are the design variables and the corresponding process and mismatch variables. To build these models, 400 transistor-level simulations are used to fit 2049 coefficients for the OpAmp design at each iteration. Likewise, 300 simulations are used to fit 569 coefficients for the LNA. Using sparse regression, the number of critical non-zero coefficients is found to range between 20 and 50. Figure 7 and Figure 8 present scatter plots reflecting the modeling accuracy for different performance metrics within a local design space. Figure 9 presents the average relative error throughout all the optimization iterations for the different performance metric models. The average error ranges between 2%-8% and the models corroborate well with the simulations.



(a) Gain (b) Power

Figure 7. Scatter plots for OpAmp performance metrics.



(a) S21 (b) S22

Figure 8. Scatter plots for LNA performance metrics.

4.2 Optimization Results

We implement Algorithm 2, and we compare the following optimization flows.

(i) Nominal: reflecting design variable-based modeling only with no account for process variations.

(ii) Statistical: traditional statistical worst-case optimization without error margining.

(iii) Proposed: our proposed methodology.

Table 1 and Table 2 present the optimization results in terms of the maximum power and the individual performance metrics yield for the OpAmp and LNA circuits respectively. It is clear that the proposed methodology is able to converge to a robust design point with high yield. Thus, we note 90%-94% yield improvement for the proposed methodology compared to the other two approaches. This comes at an extra cost of 5% to 15% power consumption for the OpAmp and LNA respectively. The runtime is of the same order as that of the statistical approach.

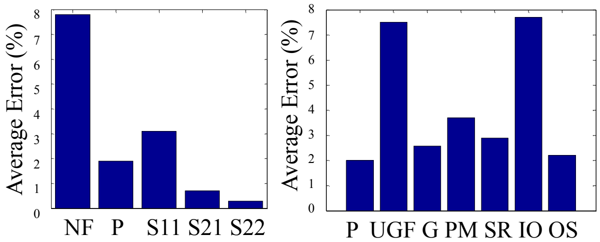


Figure 9. Average modeling error values for the different performance metrics.

Table 1. Final Optimization Results for OpAmp.

	Nominal	Statistical	Proposed
Maximum power (uW)	329.1	331.3	382
UGF yield (%)	100	100	94
G yield (%)	100	100	100
PM yield (%)	0	0	100
IO yield (%)	44	100	100
OS yield (%)	100	100	100
SR yield (%)	100	100	100

Table 2. Final optimization results for LNA.

	Nominal	Statistical	Proposed
Maximum power (mW)	14.65	14.68	15.44
NF yield (%)	12	100	100
S11 yield (%)	28	4	100
S12 yield (%)	100	100	100
S22 yield (%)	100	100	100

5. CONCLUSIONS

In this paper, we develop a novel optimization algorithm for analog circuits. Our objective is to find the optimal design point that maximizes the parametric yield and reduce a given cost function (e.g., power consumption). The aforementioned optimization problem is solved by a sequence of linear programings where the cost and constraint functions are approximated using sparse regression. Moreover, we propose a margining mechanism that takes into account modeling error during circuit optimization. Our experimental results of an OpAmp and an LNA demonstrate that the proposed technique can efficiently optimize the analog circuit while achieving high parametric yield compared to other conventional approaches.

6. ACKNOWLEDGMENT

The authors would like to acknowledge the University Research Board (URB) at the American University of Beirut for funding student Mohamad Alawieh during this research study.

7. REFERENCES

[1] G. Yu and P. Li, “Yield-aware hierarchical optimization of large analog integrated circuits,” *ICCAD*, pp. 79-84, 2008.

[2] G. Gielen and R. Rutenbar, “Computer-aided design of analog and mixed signal integrated circuits,” *Proc. of the IEEE*, vol. 88, no. 12,pp. 1825-1852, 2000.

[3] S. Tiwary, P. Tiwary and R. Rutenbar, “Generation of yield-aware pareto surfaces for hierarchical circuit design space exploration,” *DAC*, pp. 31-36, 2006.

[4] Y.-S. Park and W-Y Choi, “On-chip compensation of ring vco oscillation frequency changes due to supply noise and process variation,” *IEEE TCAS-II*, vol. 59, no. 2, pp. 73-77, 2012.

[5] X. Li, J. Le and L. Pileggi, “Statistical performance modeling and optimization,” *Foundations and Trends in Electronic Design Automation*, vol. 1, no. 4, pp. 331-480, 2006.

[6] P. Mandal and V. Visvanathan, “CMOS Op-Amp sizing using a geometric programming formulation,” *IEEE TCAD*, vol. 20, no. 1, pp. 22-38, 2001.

[7] K. Kasamsetty, M. Ketkar and S. Sapatnekar, “A new class of convex functions for delay modeling and its application to the transistor sizing problem,” *IEEE TCAD*, vol. 19, no. 7, pp. 779-788, 2000.

[8] X. Li and L. Pileggi, “Efficient parametric yield extraction for multiple correlated non-Normal performance distributions of analog/RF circuits,” *DAC*, pp. 928-933, 2007.

[9] G. Debyser and G. Gielen, “Efficient analog circuit synthesis with simultaneous yield and robustness optimization,” *ICCAD*, pp. 308-311, 1998.

[10] A. Seifi, K. Ponnambalam and J. Vlach, “A unified approach to statistical design centering of integrated circuits with correlated parameters,” *IEEE TCAS-I*, vol. 46, no. 1, pp. 190-196, 1999.

[11] T. McConaghy and G. Gielen, “Template-free symbolic performance modeling of analog circuits via canonical-form functions and genetic programming,” *IEEE TCAD*, vol. 28, no. 8, pp. 1162-1175, 2009.

[12] X. Li and H. Liu, “Statistical regression for efficient high-dimensional modeling of analog and mixed-signal performance variations,” *DAC*, pp. 38-43, 2008.

[13] X. Li, “Finding deterministic solution from underdetermined equation: large-scale performance modeling by least angle regression,” *DAC*, pp. 364-369, 2009.

[14] W. Zhang, T. Chen, M. Ting and X. Li, “Toward efficient large-scale performance modeling of integrated circuits via multi-mode/multi-corner sparse regression,” *DAC*, pp. 897-902, 2010.

[15] X. Li, W. Zhang and F. Wang, “Large-scale statistical performance modeling of analog and mixed-signal circuits,” *CICC*, 2012.

[16] X. Li, “Finding deterministic solution from underdetermined equation: large-scale performance modeling of analog/RF circuits,” *IEEE TCAD*, vol. 29, no. 11, pp. 1661-1668, 2011.

[17] Y. Wang, M. Orshansky and C. Caramanis, “Enabling efficient analog synthesis by coupling sparse regression and polynomial optimization,” *DAC*, 2014.

[18] X. Li, P. Gopalakrishnan, Y. Xu and L. Pileggi, “Robust analog/RF circuit design with projection-based posynomial modeling,” *ICCAD*, pp. 855-862, 2004.

[19] Y. Zhang, S. Sankaranarayanan and F. Somenzi,"Sparse Statistical Model Inference for Analog Circuits under Process Variations", *ASP-DAC*, pp. 449-454, 2014.

[20] A. K. Singh, K. Ragab, M. Lok, C. Caramanis and M. Orshansky, “Predictable equation-based analog optimization based on explicit capture of modeling error statistics,” *IEEE TCAD*, vol. 31, no. 10, pp. 1485-1498, 2012.

[21] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.