Predictable Equation-Based Analog Optimization Based on Explicit Capture of Modeling Error Statistics

Ashish Kumar Singh, Kareem Ragab, Mario Lok, Constantine Caramanis, *Member, IEEE,* and Michael Orshansky

***Abstract*—Equation-based optimization using geometric programming (GP) for automated synthesis of analog circuits has recently gained broader adoption. A major outstanding challenge is the inaccuracy resulting from fitting the complex behavior of scaled transistors to posynomial functions. In this paper, we advance a novel optimization strategy that explicitly handles the error of the model in the course of optimization. The innovation is in enabling the successive refinement of transistor models within gradually reducing ranges of operating conditions and dimensions. Refining via a brute force requires exponential complexity. The key contribution is the development of a framework that optimizes efficient convex formulations, while using SPICE as a feasibility oracle to identify solutions that are feasible with respect to the accurate behavior rather than the fitted model. Due to the poor posynomial fit, standard GP can return grossly infeasible solutions. Our approach dramatically improves feasibility. We accomplish this by introducing robust modeling of the fitting error's sample distribution information explicitly within the optimization. To address cases of highly stringent constraints, we introduce an automated method for identifying a true feasible solution through minimal relaxation of design targets. We demonstrate the effectiveness of our algorithm on two benchmarks: a two-stage CMOS operational amplifier and a voltage-controlled oscillator designed in TSMC 0.18 μm CMOS technology. Our algorithm is able to identify superior solution points producing uniformly better power and area values under a gain constraint with improvements of up to 50% in power and 10% in area for the amplifier design. Moreover, whereas standard GP methods produced solutions with constraint violations as large as 45%, our method finds feasible solutions.**

***Index Terms*—Analog optimization, geometric programming (GP), robust optimization.**

I. Introduction

ONE OF THE challenging aspects of analog design is optimizing a given circuit topology to meet design specifications, such as gain, while minimizing cost metrics such as area and power. This process poses severe challenges due to the stringent requirements upon multiple mutually conflicting performance constraints. In any manual design strategy, success heavily depends on the designer's experience and design-specific intuition. Automated analog optimization promises to increase productivity by reducing design time. Efforts to automate analog design have taken two major routes. In one, the circuit topology is assumed to be fixed and only optimal device sizing is performed [26], [29]. In the other, the topology is also selected automatically [31], [33]. Our work focuses on the first class of approaches.

The existing work has fallen into two major categories based on how they evaluate a solution point to drive optimization: approaches relying on extensive use of SPICE simulations [27], [28], [30], [32]–[34], and those that construct an analytical model of circuit behavior and use the model to drive the optimization [14], [20], [29], [37]. At the first cut, the tradeoff is between the accuracy that SPICE-driven methods provide, versus the global structure that can be captured by equation-based methods and global fitting. We discuss these approaches in greater detail below, and discuss the novelty and contribution of this paper in the context of a taxonomy of past work that has been done.

The challenges of equation-based optimization are two-fold: 1) to ensure sufficient model accuracy, or at least, fidelity; and 2) do that in a functional form that lends itself to efficient optimization. The first-principles small-signal parameters derived based on long-channel transistor theory are not accurate for nanometer scale technologies. The small signal parameters g_m, g_d, g_{mb}, as well as overdrive voltage and the transistor capacitances, are complex functions of the biasing current and sizes. Device-level modeling of transistors is embedded into SPICE device models, such as BSIM4, and involves hundreds of variables. Directly working with such models appears infeasible since in order to be tractable equation-based optimization requires low-dimensional models, and, in particular, convexity.

Perhaps the most common technique along these lines models circuit performance constraints as *posynomial* functions [13], [20], [21]. Optimization problems using posynomial constraints and objective function can be solved efficiently using the convex optimization framework of the geometric programming (GP) paradigm relying on fast interior point

Manuscript received June 6, 2011; revised September 7, 2011 and January 4, 2012; accepted March 4, 2012. Date of current version September 19, 2012. This work was supported by the National Science Foundation, under Grant CCF-1116955. This paper was recommended by Associate Editor H. E. Graeb.

A. K. Singh is with Terra Technology, Chicago, IL 60173 USA (e-mail: ashish.singh@terratechnology.com).

K. Ragab, C. Caramanis, and M. Orshansky are with the Department of Electrical and Computer Engineering, University of Texas, Austin, TX 78705 USA (e-mail: ragab@mail.utexas.edu; caramanis@mail.utexas.edu; orshansky@mail.utexas.edu).

M. Lok is with Harvard University, Cambridge, MA 02138 USA (e-mail: mlok@seas.harvard.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2012.2199115

solution methods. Posynomials are generalized polynomials with positive coefficients and arbitrary real exponents [8]. Thus, a posynomial $f(x_1, \ldots, x_n)$ has the general form $\sum_{q=1}^{\beta} a_q x_1^{\alpha_{q1}} \cdots x_n^{\alpha_{qn}}$, where the αs are arbitrary and the as are positive. Each product term is called a *monomial*.

Relying on posynomial models is useful due to the fact that they yield globally solvable tractable optimization problems. The cost, however, is that the fitting error of posynomial models can be large. While in part this is due to the complex device physics, another important source of error is due to the limitations imposed by the requirement of modeling in a GP-compatible manner. In addition to limiting our fitting capability, GP compatibility forces us to eliminate referencing some key input variables, e.g., it has been proven difficult to capture the dependence on $V_{\rm ds}$ (drain to source voltage) in a posynomial form, in contrast to dependence on I (drain current), W (width), and L (gate length) [14]. As a result, the optimization model may differ greatly from the behavior of the physical device. As we quantitatively demonstrate later, the attributes of the solution generated by the optimization may be more than 30% off from the intended target constraints.

This paper proposes a novel optimization strategy that seeks to combine the accuracy of SPICE simulations, with the global optimality of GP-based methods. Our central idea is as follows. We numerically capture the error between the exact device behavior (via SPICE) and the GP models we fit, and we *use these measured errors* to design a robust optimization problem that takes into account specifically these errors. This allows us to find a feasible, but possibly suboptimal point. Then refining around this point, we obtain a higher accuracy GP model, which we again robustify according to the fitting errors between the GP model and SPICE. This iterative process is efficient, and as we show, produces high-quality solutions, greatly outperforming existing techniques.

An important clarification needs to be made with regard to our ability to check true feasibility of a solution and ("validation" phase) and the way the optimization is guided toward some solution points ("optimization" phase). The key point is that while a model used to drive optimization inevitably has errors due to its capturing of device-level or circuit-level behavior, we are able to exactly establish true feasibility of any given solution point. Thus, the basic validation strategy is direct SPICE validation in which a SPICE simulation is directly used to evaluate all key performances of interest without using any intermediate representation. In some cases, it is possible to use mixed SPICE and model-based validation in which device-level accuracy is captured through SPICE and the circuit-level behavior is evaluated via a circuit-level model when it is known to be accurate. This is not a fundamental feature of the algorithm, and its only advantage is a slight reduction in SPICE runtime.

A. Existing Approaches

It is useful to give a broad overview of existing methods in order to properly situate the contributions and novelty of this paper. We organize our discussion and literature survey primarily around four categories: the MOS model and the circuit model used, and then what we call the *optimization iteration* and the *evaluation feedback* steps. Most algorithms work by a combination of an optimization step, followed by some assessment of how good, in terms of feasibility and performance. The optimization iteration refers to the optimization step, while the evaluation feedback refers to this assessment, which in all iterative methods drives the next step.

Many successful local and global optimization methods are based on local evaluation of feasibility and optimality using direct SPICE simulation [27], [28], [30], [32]–[34]. These include algorithmic descent algorithms, such as gradient descent, as well as derivative-free optimization (DFO) methods developed for functions available only through black-box simulation. DFO methods are suitable for such purposes [11]. They typically utilize the concept of trust region in which the functional behavior is approximated *locally* by a quadratic function. Thus, such methods are essentially local [2], [7]. Direct SPICE-based methods also include widely used approaches, such as simulated annealing [15], [16], [34], nonconvex optimization [28], nonlinear equation solvers [19]. Like DFO or gradient descent, simulated annealing is accurate, since it is SPICE driven at all levels, optimization and evaluation, and there is no approximating model involved. This accuracy is a central advantage that these methods enjoy. In addition to being inherently accurate, simulated annealing is designed to be able to escape from local minima, according to the so-called cooling schedule [12], [18]. Essentially, the idea is that at the initial phases of the algorithm, steps in a "hill climbing" direction are permitted (in contrast to descent algorithms) in the name of exploration. While such methods are not typically grouped or discussed together, there is a primary distinction that conceptually unites them, and more importantly separates them from the type of method we present here. This is that the optimization step does not capture any substantive global information about the search space. That is, the methods make use of local properties, be they function evaluations or local first-order information, and thereby compute the next step (via strict descent, or via an annealing-type approach that allows local hill climbing and exploration) using local information.

A different approach that also focuses on global evaluation and information, yet uses ideas from convex optimization for the optimization step, is the work in [35]. Here, the constraints are locally linearized so that the approximating constraint set becomes a polytope. Then, borrowing computationally efficient techniques from interior point optimization the largest inscribed ellipsoid is fitted, and its center becomes the next update in the iteration.

There are various approaches that take a different view in order to capture some global structure of the problem. For instance, the work in [16] produces circuit-level equations by symbolic analysis. This approach does not, however, always yield convex optimization formulations, hence greatly jeopardizing our ability to efficiently find a global optimum. Furthermore, unlike our method where we use successive rounds of refined fitting and robust optimization, this approach uses a single-shot optimization.

One class of methods is based on using limited sampling of the accurate model followed by the fitting of an approximate

model, such as a response surface model or a Kriging model [23]–[25]. In Kriging, a technique from mathematical geology, the underlying phenomenon is assumed to be a stochastic process and an optimal estimator identifies the area of the optimization space where more simulations should be done to achieve a more accurate approximate model [23]. These methods build approximate models entirely based on the samples from the accurate model, and do not incorporate additional knowledge of circuit behavior [25]. The work in [38] also takes an approach that aims to capture global structures by sampling across the optimization region, by generating a Pareto surface for the performance metrics, building upon the Kriging model. The evaluation feedback can be either model based, if deemed accurate enough, thereby reducing the number of SPICE simulations needed, or can be directly SPICE based. A fundamental issue is that the complexity of this method grows exponentially with the dimensionality of the problem, because this is typically how the complexity of global search grows in the dimension. That is, this method offers potentially significant improvements with respect to sampling requirements, but still ultimately has to perform a global exploration of the space.

Yet another family of approaches does attempt to use specific circuit knowledge, namely, the fact that a coarse simple model is available in our case in a closed form, based on the first-principles understanding of circuit behavior. That is, the perspective taken by the technique of input-space mapping that assumes that a simplified approximate model already exists [5]. Through a parameter extraction step, the space modeling technique establishes a mapping of data points in the spaces of fine and coarse model domains ensuring that they provide similar responses [3], [4]. In this way, a coarse model can be used for fast exploration, and then a mapping to the fine model space can be performed. Space mapping methods are primarily suited to unconstrained optimization, and it is unclear how they might be extended to the constrained setting, making them inappropriate for our problem. Indeed, our focus is on problems with highly nontrivial feasibility sets due to the multiple simultaneous constraints.

As we outline below, our method (and contribution) is based on the idea of leveraging convex optimization, using approximate models built from domain-specific knowledge of circuit models, and from samples taken from across the optimization region and evaluated via SPICE. The GP (and generally convex optimization) based approach is global, but without requiring exhaustive global exploration. Rather, it is the convex structure that captures global information about the feasible set. It is for this reason, for example, that linear programming can pick the optimal vertex of a polytope with exponentially many vertices, but while working only polynomially hard (and thus barely visiting even an exponentially small fraction of the vertices). The advantage of purely SPICE-based methods is the accuracy of the method. The promised advantage of convex approaches is the potential for better solutions closer to the global optimum, but without exhaustive search. The fundamental issue at hand is when the inaccuracies of convex approaches overwhelm this promise.

Thus, we now turn to some GP-based methods in the literature. The main pitfall is the inaccuracy of posynomial fitting over a large range; this has been observed in the literature [26], [29]. Accordingly, various attempts and approaches have been developed in order to rectify this significant shortcoming. In [26], the authors modeled transistor parameter behavior using piecewise linear models, hence improving accuracy. Unfortunately, this improved fit may lead to nonposynomial and, in particular, nonconvex constraints. Elsewhere in the literature, the modeling error is either fully ignored [14], or local refinement methods are used [29]. The work in [29] does local search in a small vicinity of the current solution. This requires directly fitting the circuit performance metrics. In comparison, single-transistor fitting that requires three or four variables can be done over a broader range with relatively better accuracy. In [26], a simple strategy for robustification utilizes the worst-case error. This can be highly overconservative because it may introduce robustness where none is needed. As a consequence, the feasible space of the optimization may be unduly and significantly reduced, resulting in degraded performance, or in the worst case, an inability to find a SPICE-simulation feasible solution.

The model-building step involved in the proposed algorithm is crucial. It is essential for the equation-based optimization strategy that a single-flat model is built to drive optimization. Another basic premise is that at some level such a model is constructed via regression. We also need to differentiate modeling needs at the MOS device (small-signal) level and the circuit level. There are two major possibilities for building a model and we distinguish: 1) a full regression model strategy; and 2) a mixed symbolic-regression model strategy. In the full regression model strategy, regression methods are used to fit the circuit-level function directly via regression, with the help of design of experiment methods. Fitting a highly accurate model for large circuits in this manner is a challenging task. Inevitably, the model that drives such a global exploration is not as accurate in a given local region compared to a locally fitted model (or a model-free algorithm performance locally). But it is exactly the advantage of our algorithm that we have a mechanism to explicitly account for errors and to drive optimization in such a setting to a true feasible (TF) point. Another issue is the number of SPICE simulations to run in this characterization/fitting stage since the complexity increase involved in flat versus hierarchical derivation of the final model is quite dramatic; it is exponentially more difficult in the number of transistors involved in an equation.

In the mixed symbolic-regression model strategy, device-level parameters are fitted to monomials via regression, and then combined according to a symbolic model. These symbolic models can be either based on first-principles analysis, e.g., a model for single-stage amplifier gain, or can be automatically derived using symbolic analyzers. If symbolic models are available and are reasonably accurate, using them can dramatically reduce the characterization effort in terms of SPICE simulations to run. In the paper, we used the mixed symbolic-regression model strategy for the first experiment (the op-amp), and the full regression model for the second experiment (the VCO). We discuss some further implications

Optimization Iteration	Evaluation Feedback	MOS Model	Circuit Model	References
Convex optimization with robustification and refinement	1) Direct SPICE validation 2) Mixed SPICE and model-based validation	Regression fitted to SPICE convex	1) Manual/symbolic convex 2) Regression fitted to SPICE convex	This paper
Single-shot optimization	None	Manual convex	Auto symbolic nonconvex	[15], [16]
Single-shot optimization	None	Manual convex	Manual convex global	[13], [14], [20], [21], [26]
Linearize constraints and move to center of max volume ellipsoid	SPICE	SPICE	SPICE	[35]
Local convex	SPICE	SPICE	SPICE	[29]
Move to Pareto surface generated by random sampling	Combination of Kriging model-based and SPICE-based depending on error	SPICE	SPICE	[38]
Local nonlinear optimization OR some variants of random search such as GA, SA	SPICE	SPICE	SPICE	[27], [30], [32]–[34]
None	SPICE	SPICE	SPICE	[36]

of using different modeling strategies once the details of the algorithm are presented.

We summarize this discussion, as well as the references, in the table above, organized around the main categories discussed: the driver of the optimization in the optimization phase, the nature of the evaluation and feedback phase, and the MOS model and circuit model choice.

As we detail below, the conceptual core of this paper is to develop an approach that harnesses the power of convex optimization, and the global information obtained from sampling, to avoid a global search, while nevertheless obtaining locally accurate results by introducing a robustness and refinement (RAR) phase, and using direct SPICE simulations for the evaluation phase of the iteration. Thus, this paper takes what we believe are important steps toward developing a systematic way of combining the accuracy of SPICE simulations, and the global optimization offered by GP-based approaches. Our work is based on the fundamental fact that fitting error creates a divergence between *model feasibility* and *true feasibility*: the solution is model feasible if it meets constraints under the approximate fitted model, and it is TF when it meets accurate, SPICE-verified constraints. The key idea is using data-driven robustification of the nominal model to optimize approximate functions.

B. Main Contributions and Outline

The central contribution of this paper is to harness the power of the convex optimization approach, while providing a principled way to find feasible solutions, thus moving us closer toward automating analog design and optimization. More specifically, the following is discussed.

1) We develop an efficient iterative algorithm that converges to a good solution that meets multiple performance constraint targets.
2) We address the (at times overwhelming) inaccuracies of fitting GP-compatible functions while still exploiting the benefits of an efficiently globally solvable formulation. This is in sharp contrast to existing algorithms that may result in solutions that may be off by upward of 30% from the desired targets.
3) The key enabler for the algorithm is the notion of global error-aware refinement via robustification. This uses the error statistics from the regression fit of the GP to the SPICE data, to build in custom-tailored and hence less conservative robustness. This, in turn, allows us to find a true feasible point, i.e., the point whose feasibility is verified by SPICE. We then refine the fitting range around that true feasible solution.
4) To allow robustification of multidimensional constraints, we introduce the important notion of the *coverage metric* for uncertainty set comparison. This allows us to map fitting error to robust optimization uncertainty sets. We show that this coverage metric serves as a successful proxy for true feasibility.
5) In case our methods return infeasibility for the user-given constraints due to insufficient coverage, we provide a scheme that finds a minimally relaxed set of constraints for which we are able to find a feasible solution.

Our method is independent of the fitting procedure used, and hence is flexible and modular. Thus, we believe the tools introduced here could be important for a broader class of problems.

The focus of this paper is on nominal problems and on effectively addressing modeling inaccuracy common to GP and other equation-based optimization methods. It is worth pointing out that if a feasible solution is overoptimized under nominal conditions, it may become infeasible under stochasticity of process and environmental variations. We view this as an important, though distinct problem, and a fruitful area for future work.

It is also important to identify the limitations of the proposed algorithm. We expect our algorithm to do well when symbolic models for circuit-level performances are available, thus eliminating the need to extract a convex circuit-level model via regression. In the absence of symbolic circuit-level models, our algorithm is premised on the ability to build a reasonably accurate convex circuit-level model via global regression. This may be difficult to do when the number of transistors is too high, i.e., for very large circuits, and the resulting convex model is grossly inaccurate. We believe that a

rapid initial sizing algorithm coupled with a more local SPICE-based approach is a promising way to bring out the benefits of this algorithm. Thus, the advantages of a convex optimization approach could be reaped, followed up with a local and very accurate SPICE-based approach.

The remainder of this paper is organized as follows. In Section II, we lay out the conceptual framework of our main approach. In Section III, we provide the details of the optimization. In Section IV, we address the case of stringent constraints and how to achieve constraint feasibility through minimally relaxing the constraints in an automated way. Finally, in Section V, we provide the numerical experiments that show the performance of our algorithm.

II. Exploiting Modeling Error Statistics to Drive Optimization

A. High-Level Algorithm and Approach

Our strategy is based on two key observations. First, while any given posynomial model may have significant errors, we can precisely assess the true feasibility of the solution using SPICE. Thus, despite the difficulty in obtaining a globally accurate fit, we have a local oracle of true feasibility. However, while we can determine true feasibility of any given design point $(\mathbf{W}, \mathbf{L}, \mathbf{I})$, i.e., membership in the set of true feasible points, we cannot optimize over this set, i.e., find the best set, since this problem is known to be nondeterministic polynomial (NP)-hard and hence intractable [9]. The second is the observation that the fitting error is a consequence of seeking a global fit. Obtaining a better fit over a smaller range is, not surprisingly, much more readily achievable. Yet a brute-force search for a "good" limited range of variables (width, length, and others) over which to fit the transistor parameters and, subsequently, to optimize, is a hopeless avenue, as its computational complexity grows exponentially. Even seeking to reduce the range of each variable by just a factor of $1/d$ would lead to d^N possible variable ranges to consider, where N is the number of modeled transistor parameters.

Most equation-based optimization flows that have been proposed rely on the following sequence of basic steps in setting up the optimization [14], [26], [29]. (We refer to them as the "standard approach").

1) First, the simulation data generated by SPICE are used to fit the parameters of each transistor, using linear least-squares regression, as posynomial functions of width, length, and biasing currents, computed over the full initial range of the variables.
2) Second, the fitted models are used within equations that capture the design specifications, such as gain, bandwidth, and others; the result is a posynomial formulation of the circuit optimization.
3) Finally, the resulting geometric program is solved using standard or specialized convex optimization solvers [1], [17]. We denote this solution by $\mathbf{V} \equiv (\mathbf{W}, \mathbf{L}, \mathbf{I})$, where the bold face indicates vector notation, i.e., $\mathbf{W}, \mathbf{L}$, and $\mathbf{I}$ are the vectors of all the width, length, and current variables, respectively, in the circuit.

As our computational results demonstrate, this standard approach (and variants) is essentially unable to produce solutions that are reliably, i.e., in a predictable manner, true feasible. Two fundamental new ideas are required: robustness to fitting error, and refinement. Adding robustness ensures that we find solutions that SPICE simulations show to be feasible to the design constraints, and subsequently refinement allows for more accurate fitting. We note that feasibility at intermediate stages is critical, as it is difficult to justify refining the range of the variables around a point that is not feasible.

We add robustness to fitting errors using the paradigm of robust optimization (see [6] for the basic details, algorithms, tractability). The essence of robust optimization is to build in deterministic protection to parameter uncertainty in a chosen uncertainty set $\mathcal{U}$. That is, given $\mathcal{U}$, the solution to the resulting robust optimization problem is *guaranteed to be feasible* under any variation of the optimization parameters in the given uncertainty set. As an example, if we have a design constraint that has the form

$$\frac{g_m(V)}{g_d(V)} \geq c$$

the robust version of this would then become

$$\frac{g_m(V) \cdot e_1}{g_d(V) \cdot e_2} \geq c \qquad \forall (e_1, e_2) \in \mathcal{U}$$

meaning that the constraint must be satisfied for all values of the error parameters e_1 and e_2 in the uncertainty set $\mathcal{U}$. Using duality techniques from convex optimization, we reformulate these constraints in a tractably solvable fashion [6], [22]; see the Appendix for details).

We can now give the high-level description of our algorithm.

1) *Initialize*: apply the standard procedures to obtain $\mathbf{V_0} \equiv (\mathbf{W}_0, \mathbf{L}_0, \mathbf{I}_0)$.[1]
2) *Evaluate feasibility of solution*: if the solution $\mathbf{V}$ is *true feasible*, then go to Step 4). If it is not true feasible, then proceed to Step 3).
3) *Increase uncertainty set size*: increase the robustness of the algorithm by increasing the size of the uncertainty set $\mathcal{U}$. Solve the robust GP with the uncertainty set $\mathcal{U}$, to obtain $\mathbf{V}$, and return to Step 2).
4) *Refine variable range*: given the feasible point $\mathbf{V}$, we refine the allowed range of the variables, and return to Step 1), performing a new (and hence better) posynomial fit to the transistor parameters. At each step, we refine the range by shrinking it by a constant multiplicative factor.

The difficult challenge that stems from the above formulation is finding the "right" uncertainty set $\mathcal{U}$. The size and the form of the set control the amount of robustness built into the problem: if too little robustness is used where more is needed, we may not find a true feasible solution. If, however, too much robustness is added where less is required, we may not be able to find a model-feasible solution.

[1]That is, fitting parameters over a broad range, and solving the GP.

The existing theory of robust optimization does not tell us how the uncertainty sets should be chosen. The specific problem is that the uncertainty sets we create are multidimensional. Not only is a brute-force search in this space hopeless (exponential complexity), but there is no *a priori* well-defined way to even compare two candidate uncertainty sets without running the full optimization. Developing precisely such a means of comparison, and using it to find good uncertainty sets, is one of the main contributions of this paper. We now turn to this.

B. Robustness and True Feasibility

Step 3) in our high-level algorithm description above requires us to enlarge the uncertainty set each time the optimization outputs a solution that is not true feasible, to "encourage" the subsequent solution to be true feasible, but without overly penalizing the objective value. This is an important open problem in robust optimization, with the primary issue being computational complexity. Searching by brute force for such an uncertainty set requires exponential efforts. On the other hand, one can show that optimizing over the set of uncertainty sets is nonconvex and hence intractable. This is still true even over the more limited set of rectangular-type uncertainty sets needed for Robust-GP to be tractable. A key contribution of our work is in developing a fitting-error-driven approach for selecting an uncertainty set, thus circumventing such complexity problems.

What does not work. It is helpful to discuss two potential naive approaches. One approach is to simply let $\mathcal{U}$ be a uniform box of dimension equal to the number of transistor parameters, and then slowly increase its size, hoping that a true-feasible solution is found before the problem becomes infeasible. This reduces the search to a single dimension (since the dimensions of the box are scaled in the lock step) and hence is tractable. While this strategy does sometimes succeed in obtaining a true-feasible solution, we have found that it also often leads to model infeasibility, and in either case, the objective value suffers more than required. The intuition as to why such an approach fails is simple: we may be adding too much robustness where none is needed, and too little where more is needed. Another possible approach is to solve the nominal problem, look at constraints that are violated according to SPICE, and increase the robustness requirement of the transistor parameters participating in those true-infeasible constraints. The conceptual, and as we have found, practical problems with such an approach are that we are basing our uncertainty set selection (and ultimately our iterative optimization strategy) on the behavior of an infeasible point.

What does work. We need robustness because the fit is inaccurate. Our key idea is to design the uncertainty sets based on the error in the fit for each parameter. To do this, we define the concept of *coverage* of an uncertainty set. The coverage of an uncertainty set captures how much of the true function behavior, as represented by SPICE-generated tables, is covered by the fitted function with an added uncertainty set, and we can measure it directly. We define coverage for each parameter of a given transistor. Let f denote the posynomial function of a given transistor parameter. Then given an uncertainty set $\mathcal{U}$, we define the *coverage metric of function* f to be the fraction of entries in the SPICE-generated tables of transistor behavior for which the error between the fitted function, f, and the exact function, belongs to $\mathcal{U}$

$$\text{coverage}(f, U) = \mathcal{P}_n(\text{error}(f_{\text{exact}}, f_{\text{approx}}) \in \mathcal{U})$$

where $\mathcal{P}_n$ denotes the *empirical* distribution, and $\text{error}(f_{\text{exact}}, f_{\text{approx}})$ denotes the error between f_{exact}, and f_{approx}, can be defined either additively, or multiplicatively, as is more suitable for the GP constraint of interest here. Thus, coverage is the fraction of points for which the error falls inside the uncertainty set $\mathcal{U}$. When $\mathcal{U}$ is empty, the coverage is the fraction of points in the table for which the fitted function is exact. As the uncertainty set $\mathcal{U}$ grows, the coverage increases to 100% (Fig. 2). An attractive property of the coverage metric that enhances its effectiveness as a guide to optimization is that it is not sensitive to outliers, since it simply computes the fraction of points whose fit is within the tolerance of the uncertainty set.

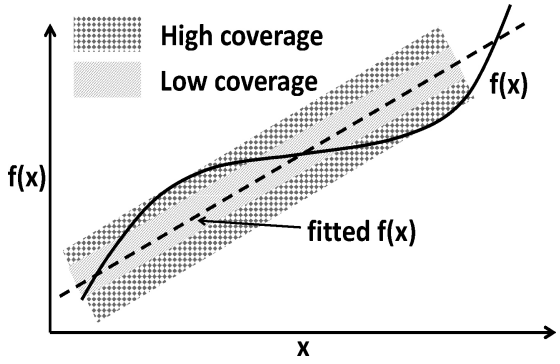

Fig. 1. Growth of uncertainty set based on coverage increase is demonstrated in the above 1-D optimization problem.

Given multiple transistor parameters, we define their joint coverage to be the minimum of their coverages. This gives us a meaningful measure for the quality of the uncertainty set, without the need for solving the robust optimization problem; coverage becomes a proxy for how strongly a set pushes the optimization problem to produce a true-feasible point. We use this notion of coverage, to develop an algorithm for increasing the uncertainty sets, to be used in Step 3) of the algorithm above. Essentially, this is done by solving a bicriterion problem that has the form of a geometric program. The key observation is that maximizing coverage subject to the constraint that the increase in the objective value of the problem corresponding to the uncertainty set is no more than some fixed value α, can be rewritten as a convex optimization problem. Thus, by replacing the intractable condition that an uncertainty set guarantee true feasibility, with the condition that the uncertainty set guarantee a desired level of coverage, we now have a tractable algorithm for selecting a coverage-optimal uncertainty set. We describe the details in the next section, and here give the high-level algorithm for Step 3).

Step 3):

1) If the problem from Step 3) is true-infeasible, increase the coverage requirement by the chosen step-size, Δ: $p \leftarrow p + \Delta$.

2) Find the uncertainty set $\mathcal{U}$, which provides coverage at least p and increases the objective value by the minimum amount. As described in detail below, we show that this can be done by solving a quasiconvex optimization problem, which can thus be solved by a combination of bisection and robust GP.
3) With this uncertainty set, solve the robust geometric program. If the returned solution is true feasible, then move on to Step 4) of the main algorithm. If it is not true feasible, then increase the coverage requirement by Δ, and return to Step b).

We end this section with two remarks. First, when using the mixed symbolic-regression modeling strategy, the robustification strategy is explicitly targeted largely at the device-level inaccuracy. In other words, the premise is that it is the device-level inaccuracy that is significant and that the errors of the circuit-level model can be tolerated. If the device-level models are accurate, but the circuit-level model is not, our algorithm will terminate at a point that is not true feasible after reaching coverage of 100%. We note that SPICE evaluations easily reveal such a phenomenon; we believe that our robustness methods could be adapted to handle just such a case, although this is not part of this paper. We believe that this point is still a good initial point for local optimization methods, although we do acknowledge that the main motivation for our work is the observation that device-level inaccuracies are typically quite significant.

Second, as introduced above, the coverage metric is based on the uniform weighing of all points in the table, since coverage essentially counts points that fall in or out of the uncertainty set. This approach is conceptually motivated. By allowing a certain range for the optimization variables, implicitly, it is a statement that the variables can take on any values in that range, and hence there is no reason to treat poor fitting quality in one part of the range differently from another part of the range. Nevertheless, the proposed algorithm is general and flexible, and could be tailored to other side information by using weighted coverage metrics.

III. Algorithm Details

We have described at a high level the key pieces of the algorithm. We provide the details in this section, and put the pieces together. First, we require the fitted model for each transistor parameter needed to set up the constraints and the objective, which are in terms of various performance metrics such as bandwidth, loop gain, power. Let there be q transistor parameters $M = (m_1, \ldots, m_q)$, where each m_i is a monomial function of (W, L, I). The nominal constraints of the circuit optimization are built up from the $\{m_i\}$ in a manner consistent with GP, namely, in a multiplicative manner. Thus, constraints will take the form (The coefficients γs are restricted to be positive and the exponents θs are real numbers to conform to a posynomial format)

$$\frac{\gamma \prod_{i\in I} m_i^{\theta_i}}{\sum_{l\in L} \gamma_l \prod_{j\in J_l} m_j^{\theta_{l,j}}} \geq 1.$$

Parameter fitting: To fit the parameters of these constraints, a set of SPICE simulations is carried out at the characterization phase to create a table typically capturing the values of the channel conductance (g_d) and transconductance (g_m), overdrive voltage (V_{gt}), and transistor capacitances (C_{gs}, C_{gd}, and C_{gb}). These measurements are made over a range of transistor width, length and bias current values, separately for NMOS and PMOS transistors. Next, a model is fitted to the collected data. We perform a single-monomial (unknown exponents) fit by applying a log transformation and subsequently taking the exponential to recover the transistor parameters as a single monomial. Via regression, we obtain coefficients and exponents for the best-possible, in the least-squares sense, monomial fit. Thus, for example, our model for g_m becomes $g_m \approx aW^bL^cI^d$, where the values a, b, c, and d are determined through regression to the data in the table. In this way, the coefficients of the constraints of the above form are determined.

Robust optimization: Robust optimization is not tractable for all possible uncertainty sets (see [6], the references therein for more details). In our setting, robust GP is tractable for rectangular and ellipsoidal uncertainty sets. In this paper, we develop our framework using rectangular uncertainty sets, although our framework is expandable to ellipsoidal uncertainty as well. Moreover, for the robustified geometric program to again be expressible as a geometric program, we formulate the uncertainty as affecting the constraints in a multiplicative manner, with an error parameter e_i corresponding to each transistor parameter m_i. In the interval rectangular uncertainty model, each e_i is constrained to lie in an interval $[-k_i, k_i]$. For example, for the transconductance parameter g_m, we have $g_m = aW^bL^cI^d \exp(e)$, where the error term e belongs to an interval $[-k, k]$.

For multiple constraints, we express the uncertainty set as

$$\mathcal{U} \stackrel{\triangle}{=} \mathcal{U}(\mathbf{k}) = \prod_{i=1}^{q} [-k_i, k_i].$$

Thus, an uncertainty set is characterized by the q-dimensional vector $\mathbf{k} = (k_1, \ldots, k_q)$. The *robustified constraints* now have the form

$$\frac{\gamma \prod_{i\in I} m_i^{\theta_i} \cdot (\exp e_i)^{\theta_i}}{\sum_{l\in L} \gamma_l \prod_{j\in J_l} m_j^{\theta_{l,j}} \cdot (\exp e_j)^{\theta_{l,j}}} \geq 1 \quad \forall (e_1, \ldots, e_q) \in \mathcal{U}(\mathbf{k}).$$

For a given uncertainty set $\mathcal{U}(\mathbf{k})$, letting M denote the set of transistor parameters as above, and V denote the complete set of variables (W, L, I) for each transistor, the robust GP that we need to solve is (We use the notation Constraint$(V, M, \mathbf{k}) \leq 1$ to refer to a collection of posynomial constraints defined in terms of variables V and M, and a vector of constants $\mathbf{k}$)

$$\begin{aligned} \text{Min}_V \quad & \text{Objective}(V) \\ \text{s.t.} \quad & \text{Constraints}(V, M, \mathbf{k}) \leq 1. \end{aligned} \tag{1}$$

The objective function, like the constraints, is a posynomial function of transistor parameters and variables in V. The constraints are now additionally a function of the uncertainty set $\mathcal{U}(\mathbf{k})$, which we also denote by $\mathbf{k}$ to shorten notation. The nominal GP corresponds to setting $\mathbf{k} = 0$.

A. Selecting Coverage and Robustness

Each time the robust GP returns a solution that is not feasible, we must increase the uncertainty set $\mathcal{U}$, by increasing the vector $\mathbf{k}$. Note that this cannot be done by sweeping or brute force, since any such attempt is exponential in q, the number of transistor parameters.

Let Coverage(k_i, i) denote the coverage of parameter m_i, and Coverage$(\mathbf{k})$ the overall coverage. Then, given $\mathcal{U}(\mathbf{k})$, Coverage(k_i, i) is equal to the number of entries in the table where the error is within a multiplicative factor of $\exp(\pm k_i)$, and Coverage$(\mathbf{k}) = \text{Min}_{i=1}^{q}\text{Coverage}(k_i, i)$. At each step where a true-infeasible solution is returned, our algorithm calls for us to increase the overall coverage, while minimally increasing the objective value of the resulting robust optimization problem. Thus, we want to solve the following coverage maximization problem (We represent by Constraint$(V, M, \mathbf{k}) \leq 1$ a collection of posynomial constraints defined in terms of variables V, M, and $\mathbf{k}$s):

$$\begin{aligned} \text{Maximize}_{\mathbf{k},V} \quad & \text{Coverage}(\mathbf{k}) \\ \text{s.t.} \quad & \text{Objective}(V) \leq \text{obj}_i * (1 + \frac{\alpha}{100}) \\ & \text{Constraints}(V, M, \mathbf{k}) \leq 1. \end{aligned} \tag{2}$$

In the above problem, the constraint set is a collection of posynomials; however, the objective is not. Moreover, we found that it cannot be well approximated by a monomial expression needed for GP compatibility. We also do not have the option to use a posynomial fit with multiple terms since this will violate GP compatibility. This is because we are maximizing the objective, and hence have to invert the objective in order to put it into a standard GP form (that requires minimizing). We are able to show, however, that this problem is quasiconvex. In particular, we show that finding the uncertainty set that increases coverage by at least a factor of β while not increasing the objective value by more than a factor of α can be cast as a geometric program.

Notice that Coverage(k_i, i) is essentially the empirical distribution of the absolute value of the fitting error for the log of parameter m_i. We treat $\exp(k_i)$ as a problem variable. Thus, we let $k_i' = \exp(k_i)$, and analogously for the vector: $\mathbf{k}' = \exp(\mathbf{k})$. By taking the inverse of the distribution function, we get $k_i' = \exp(k_i) \geq \exp(\text{Coverage}(k_i, i)^{-1}(\beta))$. With these new variables, we can check whether an uncertainty set exists, which increases coverage by at least β, while increasing the objective value by at most α, by solving the following geometric program:

$$\begin{aligned} \text{Maximize}_{\mathbf{k}',V} \quad & 1 \\ \text{s.t.} \quad & \text{Objective}(V) \leq \text{obj}_i * (1 + \frac{\alpha}{100}) \\ & \text{Constraints}(V, M, \mathbf{k}') \leq 1 \\ & \mathbf{k_i'} \geq \exp(\text{Coverage}(k_i, i)^{-1}(\beta)) \quad \forall i \in [1, q]. \end{aligned} \tag{3}$$

Thus, via a line search over β, we can now solve the problem of finding an uncertainty set that increases coverage while minimally deteriorating the objective value.

We can now describe all the steps of the algorithm as follows.

Input: SPICE table, initial range, user-selected parameter Δ. Initialize $\alpha = 0$ and $\mathbf{k} = \mathbf{0}$.

1) Solve problem (1) to obtain solution $\mathbf{V} \equiv (\mathbf{W}, \mathbf{L}, \mathbf{I})$.
2) Use SPICE simulation to assess whether the solution is true feasible.
3) If $\mathbf{V}$ is not true feasible, set $\alpha \leftarrow \alpha + \Delta$ and obtain a new uncertainty set and hence new value $\mathbf{k}$. Solve problem (1) to obtain new solution $\mathbf{V}$. Return to Step 2).
4) While the range is larger than the minimum size, shrink the range around the true-feasible solution $\mathbf{V}$, set $\alpha = 0$ and $\mathbf{k} = \mathbf{0}$, and return to Step 1).
5) Report true-feasible solution $\mathbf{V}$.

We find that this algorithm is computationally efficient, and results in greatly improved performance over competing methods. Interestingly, we find that the difference between our algorithm and standard GP algorithms is significant not only in terms of the value of the solution. Indeed, as we report in Section V, we find that often times our algorithm results in a solution that is *in a different part of the feasible region* than what standard GP returns. In particular, GP augmented by local search (e.g., by first solving a GP and then using DFO to find a local optimum) still performs worse, indicating that our global search procedure is directly tied to the success of our method. We report this and other computational experiments in Section V.

IV. Handling Infeasibility by Relaxation

Given that optimizing circuit performance exactly is NP-hard, an immediate although unfortunate corollary is that in the worst case, even finding a true-feasible point is NP-hard. Thus, any tractable method must be prepared for the contingency that a true-feasible solution is not found. Specifically, in our case, the increase of uncertainty set may lead to the solution becoming model-infeasible, i.e., the algorithm not being able to find any solution, before we find a true-feasible solution. To deal with this, we demonstrate that using our framework, and in particular, leveraging the concept of coverage, it is possible to find the "least relaxed" set of specifications, for which our method can find a true-feasible solution. The motivation and goal in this effort is again that of automated design.

The central idea is the observation that the set of constraint target values that have a given lower bound on coverage, in fact, are posynomial representable. Indeed, this is the advantage of coverage; it is a good proxy for true feasibility, yet it is captured via posynomial constraints. Thus, the problem of relaxation of constraint targets can be formulated as a GP problem. Our method provably provides the least amount that the constraints can be relaxed to achieve high (any pre-specified value) coverage. In our experiments, we find that a coverage range of 50%–80% is usually sufficient for most cases. This is inherently a multicriterion optimization problem, since it involves the relaxation of potentially multiple constraints. Our method is flexible, allowing the designer to specify which constraints are more important, and subsequently performing the relaxation according to a weighted ratio objective. Thus, the constraints that are deemed more important are relaxed by a smaller percentage while others can be relaxed more.

The constraints we seek to modify are the circuit-performance user-specified constraints. Other constraints present in the problem capture the structural circuit constraints and the internal current and voltage relations, and we are not seeking to modify those. Let us explicitly denote the user-specified constraints, including their right-hand sides (i.e., the specified values) by

$$f_i(V, M, \mathbf{k}') \leq P_i \quad \forall i \in C.$$

The vector $\mathbf{P}$ is the set of performance targets provided by the user and it is this that we seek to minimally relax in order to increase coverage. Thus, we treat the elements in $\mathbf{P}$ as variables so that we can then describe the set of constraint targets that allow high coverage β. We use Constraints_Str$(V, M, \mathbf{k}') \leq 1$ to denote the structural constraints that are not being modified. The augmented problem can be described by a set of posynomial constraints

$$\begin{aligned} \text{Maximize}_{\mathbf{k}', V, \mathbf{P}} \quad & 1 \\ \text{s.t.} \quad & \text{Objective}(V) \leq \text{obj}_i * (1 + \frac{p}{100}) \\ & \text{Constraints\_Str}(V, M, \mathbf{k}') \leq 1 \\ & f_i(V, M, \mathbf{k}') \leq P_i \quad \forall i \in C \\ & \mathbf{k_i}' \geq \exp(\text{Coverage}(k_i, i)^{-1}(\beta)). \end{aligned}$$

Since we may need to prioritize relaxation of individual constraints, rather than do that in a uniform manner, we introduce a weighting factor is $\mathbf{w}$. Finally, relax is the relaxation factor. Then, the minimal constraint relaxation problem is as follows:

$$\begin{aligned} \text{Minimize}_{\mathbf{k}', V, \text{relax}} \ & \text{relax} \\ \text{s.t.} \ & \text{Objective}(V) \leq \text{obj}_i * (1 + \frac{p}{100}) \\ & \text{Constraints\_Str}(V, M, \mathbf{k}') \leq 1 \\ & f_i(V, M, \mathbf{k}') \leq P_i * w_i * \text{relax} \ \forall i \in C \\ & \mathbf{k_i}' \geq \exp(\text{Coverage}(k_i, i)^{-1}(\beta)). \end{aligned}$$

Table III explores the effectiveness of this approach.

V. Experimental Results

In this section, we report our numerical experiments to validate the performance of presented algorithms. Devices were characterized using 180 nm TSMC high-performance technology models. First, we illustrate that the monomial fitted to SPICE data using least-squares regression may exhibit very high errors over certain portions of its range. In Fig. 2, we show the histogram of the fitting errors for the output conductance parameter g_d of a PMOS transistor. The transistor is simulated in HSPICE to predict the g_d and drain current for a set of width, length, V_{gs}, and V_{ds} values. The samples are generated by varying the gate length from 180 nm to 1.8 μm and gate width varying from 180 nm to 18 μm, both in increments of 20%. The V_{gs} ranges from 0.65 to 1.8 V and V_{ds} ranges from 0.35 to 1.8 V. Then, g_d was fitted as a monomial function of width, length, and drain current.

The fitted equation was $g_d = 0.079 W^{0.22} L^{-0.84} I^{0.73}$, where the unit of g_ds is $\frac{\mu A}{V}$, W and L are in units of μm, and I is

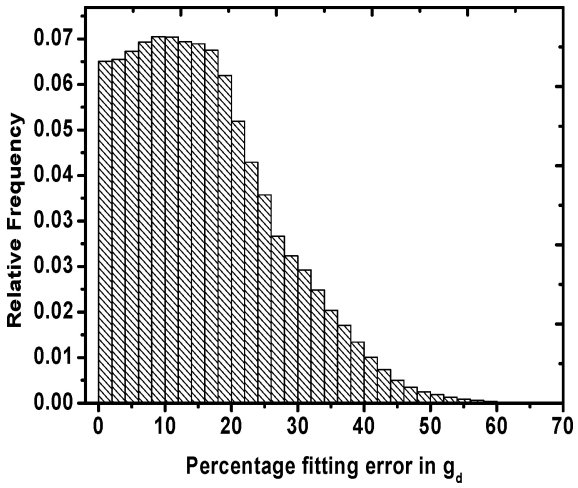

Fig. 2. Histogram shows the distribution of fitting errors for g_d for a PMOS transistor.

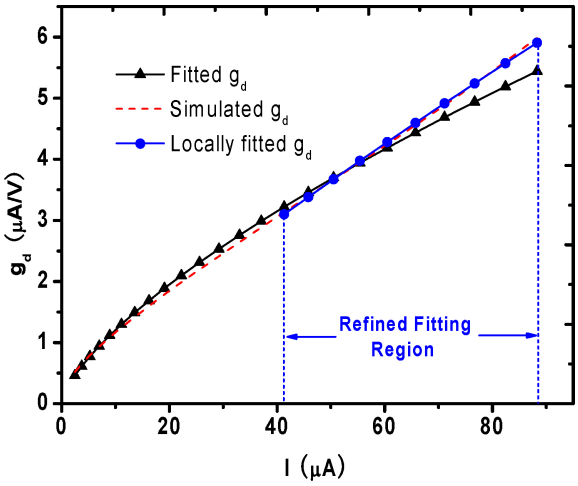

Fig. 3. Fitting improvement with refinement of the fitting region. The errors of the global fit are as large as 10%. The worst-case error is 1% for the narrower region.

measured in units of μA. The fitting errors are significant. Fig. 2 shows that a high number of samples yielded a fitting error higher than 20%. The rms error of the fit is 19%. Importantly, the maximum error is 69%. This indicates that there is a danger of optimizing around a region in which the model fit is very poor. Similar trends are observed for other fitted functions.

We next demonstrate the improvement in fitting accuracy through refinement by an example shown in Fig. 3. To simplify presentation, we restrict the fit to a 1-D single-variable fit. We do refinement by generating samples with W, L, and V_{ds} fixed, and varying only V_{gs} from 0.65 to 1.8 V in increments of 0.05. We show the fitting of g_d for a PMOS transistor as a monomial function of current. The worst-case error is 10% when fitting over the V_{gs} range of 0.65–1.8 V. However, when we restrict the range of V_{gs} to be 1.05–1.8 V, the worst-case error is reduced to 1.3%. The point is that even modest refinement of the range (factor of 2 here) can dramatically improve the fitting error (about a factor of 8).

We next report the outcomes of numerical experiments that validate the performance of our algorithm, and, in particular,

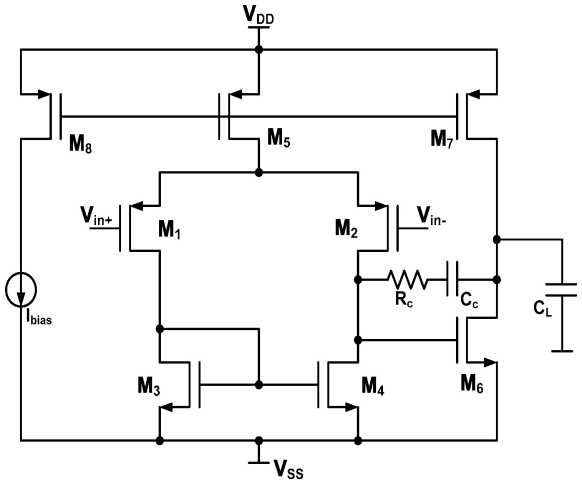

Fig. 4. Two-stage operational amplifier used for numerical experiments.

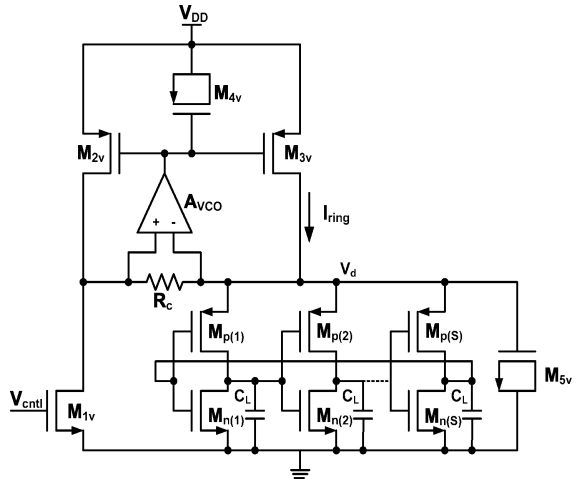

Fig. 5. Voltage-controlled oscillator used for numerical experiments.

compare our algorithm to the existing global GP-based solution. We compare the proposed RAR-based optimization with the prior equation-based global search method employing GP. We refer to the prior method as the standard optimization. We demonstrate the effectiveness of our algorithm by using it to optimize the area of a two-stage CMOS operational amplifier and power for a voltage-controlled oscillator. These two examples have been used as validation vehicles in several prior related publications [10], [29] (Figs. 4, 5). The two-stage amplifier circuit is made up of eight transistors. The typical design metrics include gain, unity gain bandwidth, slew rates, common-mode rejection ratios, phase margin, and areas. For this circuit, we rely on the well-known, "first-principles" models of circuit-level performances as functions of small-signal device-level models, which we fit directly to SPICE data. In Table I, we show several of the models used. (In Table I, C_1, C_2, and C_{out} are the capacitances at the gates of transistors M_6, M_3 and at the output node, respectively.) We verified that these models have good accuracy as long as the small-signal device-level models obtained from SPICE by regression are accurate. For that reason, we used the mixed SPICE and model-based validation strategy in this experiment.

TABLE I
MODELS OF CIRCUIT-LEVEL PERFORMANCES USED IN OPTIMIZATION

Gain	$\frac{gm_2 \times gm_6}{(gd_2+gd_4)\times(gd_6+gd_7)}$
Pole (p_1)	$\frac{gm_1}{2\pi \times gain \times C_c}$
Pole (p_2)	$\frac{gm_6 \times C_c}{2\pi C_1 C_c + 2\pi C_1 C_{\text{out}} + 2\pi C_c C_{\text{out}}}$
Pole (p_3)	$\frac{gm_3}{2\pi C_2}$
Pole (p_4)	$\frac{gm_6}{2\pi C_1}$
UGB	$\frac{g_m}{2\pi C_c}$
Phase margin	$\frac{\pi}{2} - 0.75(\frac{ugb}{p_2})^{0.7} - \frac{ugb}{p_3} - \frac{ugb}{p_4}$

As discussed previously, when circuit-level models are not available, the full regression model strategy can be used. In this case, we build models directly through DOE and regression, relating circuit-level performance metrics directly to device parameters (W, L, I). This is the modeling strategy we use in the second experiment based on the voltage-controlled oscillator (VCO) to fit the model of VCO frequency. The voltage-controlled oscillator has minimum and maximum frequency constraints, as well as saturation constraints for all the transistors. We also have a constraint on transistor sizes, which sets the transistor lengths and widths within the range of 180 nm to 1.8 μm. In the initial iteration, we use a global fit across the full range. In subsequent iterations, we refine the fitting range by 20% in each iteration, after finding a true feasible solution. In the validation phase, we use direct SPICE simulation to establish true feasibility.

The paramount benefit of the proposed algorithm is that it offers a guaranteed way of meeting multiple design specifications. Thus, the second set of experiments on solving multiple-constraint problems aims at demonstrating the degree to which the standard method can be infeasible, while our method meets all of the constraints. We find that because of the large fitting errors the standard optimization method often produces solutions that grossly violate the constraints, especially when multiple constraints are used.

In the experiment for the two-stage amplifier, we use area as the objective to minimize, and have constraints on gain, unity gain bandwidth (UGB), slew rate, common-mode rejection ratio (CMRR), phase margin (PM), and negative power supply rejection ratio (PSRR). In Table II, we present the comparison results in terms of percentage of constraint violations for minimum area optimization. As the results demonstrate, while our algorithm meets the target constraints, the standard optimization is unable to find a feasible solution, and the solution produced in some cases grossly violates the target constraints. We also used our method to perform a tradeoff analysis between individual circuit performances. An example of such a Pareto curve showing the tradeoff between the gain and unity-gain bandwidth for the two-stage amplifier circuit is given in Fig. 6. We note that the standard method violates the target gain constraint on average by 26%. In this experiment, the total number of design variables was 25. The runtime of our method was, on average, 42 min on a 2.93 GHz processor, when using 10 as the number of refinement steps and 21 as the number of coverage increment steps with

TABLE II
AREA MINIMIZATION FOR THE TWO-STAGE AMPLIFIER BENCHMARK CIRCUIT

Performance	Spec	RAR	Standard	Spec	RAR	Standard	Spec	RAR	Standard
		% Violation	% Violation		% Violation	% Violation		% Violation	% Violation
Gain (dB)	≥66	0	16	≥67.3	0	18.5	≥65.6	0	15.1
UGB (MHz)	≥5	0	0	≥5.5	0	0	≥6	0	0
Slew rate (V/μs)	≥9	0	1.7	≥10	0	0	≥5	0	0
CMRR (dB)	≥66	0	47.8	≥65.9	0	31.7	≥64.6	0	44.2
Phase margin (°)	≥60	0	0	≥45	0	0	≥60	0	0
Negative PSRR (dB)	≥74.8	0	17.8	≥74.8	0	12.8	≥74	0	11.4
Area (μm^2)	MIN	329	237	MIN	296.4	219	MIN	270.4	228.8

The standard method leads to significant constraint violations while the proposed method is able to meet all the constraints.

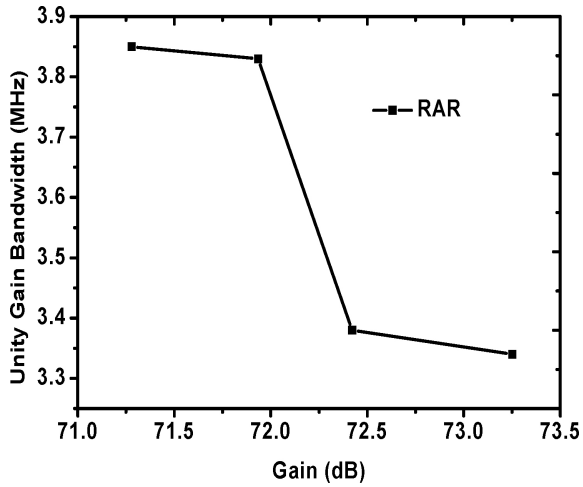

Fig. 6. Tradeoff curve between gain and bandwidth generated by our method. The standard method violates the gain constraint on average by 26%.

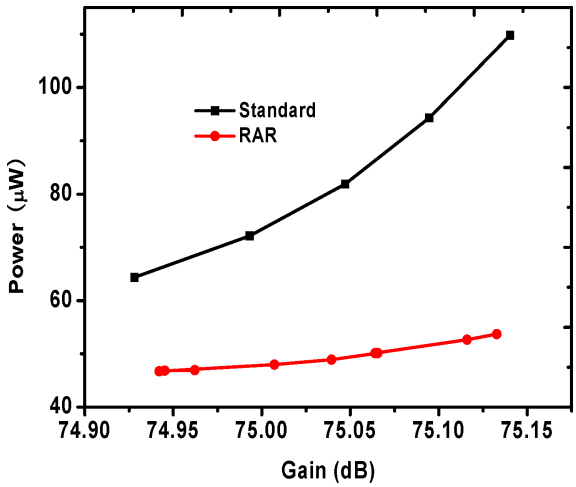

Fig. 7. Power versus gain Pareto curve. The proposed algorithm is uniformly better and maximum power savings are 50% at fixed gain.

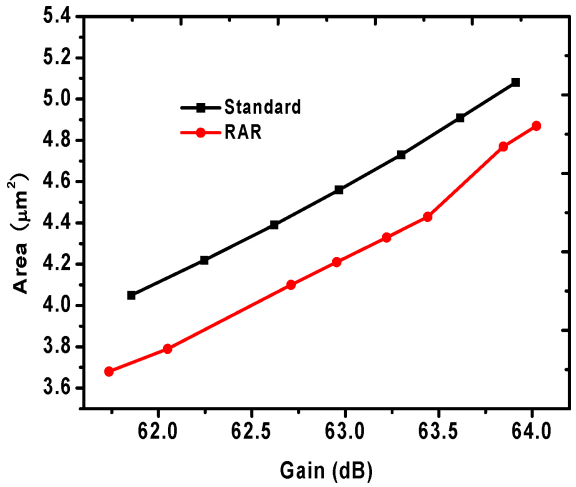

Fig. 8. Area versus gain Pareto curve. The proposed algorithm is uniformly better and maximum area savings are 10% at fixed gain.

a coverage increment of 5% per step. The number of actual SPICE simulations to check true feasibility was 124. We show a similar set of results for a voltage-controlled oscillator in Table IV. In this experiment, the number of design variables was 17. The average runtime was 47 min using 3 as the number of refinement iterations, and using 11 as the number of coverage iterations with a coverage step increment of 10% per iteration. The number of SPICE simulations required was 27 for the verification of true feasibility.

We note that the objective function of the standard approach appears better than what our approach finds—however, given that the standard optimization produces solutions that violate constraints by up to 47%, it is not clear that the objective function value is meaningful, or even how to devise a fair numerical comparison. A useful comparison would be based on comparing Pareto surfaces, generated by sweeping the values of all the constraints and plotting against the optimal values obtained, which is difficult to do for problems with multiple constraints.

Now, we consider a few cases wherein the constraints are so stringent that we are not able to get a true feasible solution using our robustification step. The primary reason is the low coverage as discussed in Section III. We use the strategy described in Section III to come up with a minimal relaxation of the user-given constraints so that we can get a true feasible solution. The results are depicted in Table III. We are able to find a true feasible solution for the relaxed versions while the standard solution is not able to find a true feasible solution and violations are up to 30%.

We carried out the experiments to measure the goodness of the global search versus the local refinement procedure. For this we took for comparison a local refinement method similar to the method proposed in [29]. The central idea of the local refinement scheme is to do refinement and refitting

TABLE III
AREA MINIMIZATION RESULTS WITH STRINGENT CONSTRAINTS

Performance	Spec	Relaxed Spec	RAR	Standard	Spec	Relaxed Spec	RAR	Standard
			% Violation	% Violation			% Violation	% Violation
Gain (dB)	$\geq$75.6	72	0	23	$\geq$72.5	70.4	0	23.2
UGB (MHz)	$\geq$1	1	0	0	$\geq$4	4	0	0
Slew rate (V/μs)	$\geq$1	1	0	0	$\geq$5	3.9	0	0
CMRR (dB)	$\geq$75.6	74	0	36.5	$\geq$72.5	70.4	0	27.2
Phase margin (°)	$\geq$70	62	0	0	$\geq$45	33.24	0	0
Negative PSRR (dB)	$\geq$83.8	83	0	1.2	$\geq$74	72	0	0
Area (μm^2)	MIN	MIN	303	251.6	MIN	MIN	498.3	365

We obtain a true feasible solution after minimal constraint relaxation. The standard method leads to significant constraint violations even for the relaxed constraints.

TABLE IV
POWER MINIMIZATION IN A VOLTAGE-CONTROLLED OSCILLATOR BENCHMARK CIRCUIT

Performance	Spec	RAR	Standard	Spec	RAR	Standard	Spec	RAR	Standard
		% Violation	% Violation		% Violation	% Violation		% Violation	% Violation
Min VCO freq. (GHz)	$\leq$1	0	30	$\leq$1.2	0	28.3	$\leq$1.2	0	30.8
Max VCO freq. (GHz)	$\geq$1.25	0	0	$\geq$1.4	0	0	$\geq$1.5	0	0
Power (μW)	MIN	69.9	36.7	MIN	69.4	44.6	$\geq$5	76.2	50.13

The standard method leads to significant constraint violations while the proposed method is able to meet all the constraints.

TABLE V
PROPOSED GLOBAL SOLUTION SEARCH VIA REFINEMENT AND ROBUSTIFICATION IS ABLE TO MAINTAIN A GLOBAL SEARCH FOR SEEKING THE OPTIMAL SOLUTION WHICH IS SUPERIOR TO THE PURELY LOCAL SEARCH BASED METHOD

Variable	RAR Solution	Local Refinement	Variable	RAR Solution	Local Refinement
W_1 (nm)	978	1404	L_1 (nm)	1781	1780
W_2 (nm)	978	1404	L_2 (nm)	1781	1780
W_3 (nm)	180	189	L_3 (nm)	1782	1594
W_4 (nm)	180	189	L_4 (nm)	1782	1594
W_5 (nm)	180	180	L_5 (nm)	451	424
W_6 (nm)	1782	1653	L_6 (nm)	933	741
W_7 (nm)	1783	1783	L_7 (nm)	451	424
W_8 (nm)	656	553	L_8 (nm)	451	424
Objective (μm^2)	243	267	Capacitance (fF)	250	274

in a very small region around the current solution, thus finding a locally optimal solution. After solving the problem restricted to this region, we get a new solution. The local neighborhood of this solution serves as the local fitting region for the next iteration. The fitting is done by using SPICE accurate values. Clearly, this method to begin with requires an initial true feasible point [29]. In order to provide this starting point, we set the initial starting point as the solution produced by our method in the first phase of robustification without any refinement step. We demonstrate through a specific example that the difference in performance between our algorithm and the standard GP algorithm may not be due to merely local improvement. That is, our combination of fitting-error-driven robustification and gradual refinement is better suited to the global exploration of the space, than what GP alone, or even GP and robustness, can accomplish. We demonstrate this by showing that local refinement can produce a solution that is far from global optimal and in a different region from the solution our algorithm produces. This is a fundamental affirmation of the conceptual underpinnings of this paper, which seeks to combine global methods (GP, and more generally convex optimization) with local accuracy as given by SPICE. The results are reported in Table V.

In some restricted cases—when only a single constraint is used—it is possible to compare the Pareto optimality of our method to that of the standard method across a range of design values. The design process is intrinsically a multiobjective optimization process and the optimal solutions lie on the multi-dimensional Pareto surfaces. It is hard to present Pareto curves in more than two dimensions, so we can largely demonstrate the effectiveness of the algorithm by showing the value of the objective function that can be obtained for a single constraint. We show the value of amplifier gain (used as the constraint) against the objective area, in one case, and power, in the other case. In the first experiment, the Pareto curve was generated by sweeping the value of target gain over the range of [74.3 db 75.3 db] and optimizing the power using the standard optimization method that uses the nominal fitting models for transistor parameters as well as the proposed RAR method. The results are shown in Fig. 7 and indicate that we can obtain uniformly better solutions with up to 50% savings in power. We generated a similar tradeoff curve for minimum area in a

different range of gain values. This experiment also demonstrates that our algorithm produces uniformly better solutions with up to 10% area savings. The results are shown in Fig. 8.

VI. Conclusion

In this paper, we presented a set of algorithmic solutions that aimed to explicitly utilize the knowledge of modeling error in the fitted equations to drive optimization. The algorithm was based on two key concepts of refinement and robustness. A novel concept of coverage was used to optimally construct the uncertainty sets. The results were promising and showed that significant improvements were possible in terms of the value of the achievable cost functions, as well as in terms of reliably meeting performance constraints in the presence of large modeling errors.

Acknowledgment

The authors would like to thank K. He for his help with some experiments and illustrations.

Appendix
Tractable Formulation for Geometric Programming

In the Appendix, we provide some details on how we obtain tractable robust GP formulations. In this paper, we have focused on rectangular uncertainty regions modeling posynomial fitting errors. In cases where the corner point of the rectangle gives the worst-case uncertainty, the robust GP reduces to a nominal GP, and hence is formulated and solved with the usual GP methods. The interesting setting is when this is not the case. Here, we need to apply some transformations to make the problem tractable. The first step is the standard log-exponential transformation that converts the GP into a convex form. It turns out that this log-exponential transformation can be uniformly approximated to arbitrary accuracy by a piecewise linear function. Thus, we replace the constraints by piecewise linear functions, obtaining an LP. Robust linear programming is well studied, and has tractable reformulations [6]. For rectangular uncertainty, the robust LP can be rewritten as equivalent LP; for ellipsoidal uncertainty, the problem can be rewritten as an equivalent (convex) second-order cone problem. Both of these can be solved efficiently.

Below, we illustrate this transformation process through some examples. Suppose that we have a constraint of the form $\frac{g_d}{g_m} + g_m \leq 1$, where we model g_m and g_d as (uncertain) monomials: $g_m = aW^bL^cI^d\text{error}(g_m)$ and $g_d = eW^fL^gI^h\text{error}(g_d)$. In the rectangular uncertainty model, the error parameters $\text{error}(g_m)$ and $\text{error}(g_d)$ belong to a box uncertainty set $\mathcal{U}$.

In the posynomial form, the constraint becomes

$$\frac{e \times \text{error}(g_d)}{a \times \text{error}(g_m)} W^{f-b}L^{g-c}I^{h-d} + a \times error(g_m) \times W^bL^cI^d \quad \leq \quad 1.$$

We introduce variables $W = \exp(\overline{W})$, $L = \exp(\overline{L})$, $I = \exp(\overline{I})$, and $e = \exp(\overline{e}), a = \exp(\overline{a})$. Similarly, we introduce variables to model error parameters in the log domain by $\text{error}(g_d) = \exp(\overline{\text{error}(g_d)})$, $\text{error}(g_m) = \exp(\overline{\text{error}(g_m)})$.

On plugging in these variables and taking the log, we get the following convex constraint:

$$\log(\exp(\overline{e} - \overline{a} + \overline{\text{error}(g_d)} - \overline{\text{error}(g_m)} + \overline{W} \times (f-b) + \overline{L} \times (g-c) + \overline{I} \times (h-d)) + \exp(\overline{a} + \overline{\text{error}(g_m)} + \overline{W}b + \overline{L}c + \overline{I}d))) \quad \leq \quad 0.$$

Using techniques from [22], we can approximate these constraints to arbitrary accuracy using a piecewise linearization function. This can, in turn, be modeled as a linear program, and in particular, the coefficients of this linear program will be linear functions of the error parameters $\overline{\text{error}(g_m)}$ and $\overline{\text{error}(g_d)}$. Consequently, the resulting optimization problem is a robust linear program with polyhedral uncertainty. This can then be converted to an equivalent LP, thus being efficiently solved using out-of-the-box LP solvers [6].

References

[1] *MOSEK: Mosek Optimization Software* [Online]. Available: http://www.mosek.com

[2] N. M. Alexandrov, J. E. Dennis, R. M. Lewis, and V. Torczon, "A trust region framework for managing the use of approximation models in optimization," *Structural Multidisciplinary Optimiz.*, vol. 15, no. 1, pp. 16–23, 1998.

[3] J. Bandler, R. Biernacki, S. H. Chen, P. Grobelny, and R. Hemmers, "Space mapping technique for electromagnetic optimization," *IEEE Trans. Microwave Theory Tech.*, vol. 42, no. 12, pp. 2536–2544, Dec. 1994.

[4] J. Bandler, Q. Cheng, S. Dakroury, A. Mohamed, M. Bakr, K. Madsen, and J. Sondergaard, "Space mapping: The state of the art," *IEEE Trans. Microwave Theory Tech.*, vol. 52, no. 1, pp. 337–361, Jan. 2004.

[5] J. Bandler, M. Ismail, J. Rayas-Sanchez, and Q.-J. Zhang, "Neuromodeling of microwave circuits exploiting space-mapping technology," *IEEE Trans. Microwave Theory Tech.*, vol. 47, no. 12, pp. 2417–2427, Dec. 1999.

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

[7] A. J. Booker, J. E. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset, "A rigorous framework for optimization of expensive functions by surrogates," *Structural Multidisciplinary Optimiz.*, vol. 17, no. 1, pp. 1–13, 1999.

[8] S. Boyd, S.-J. Kim, L. Vandenberghe, and A. Hassibi, "A tutorial on geometric programming," *Optimiz. Eng.*, vol. 8, no. 1, pp. 67–127, 2007.

[9] M. Chiang, "Geometric programming for communication systems," *Foundat. Trends Commun. Inform. Theory*, vol. 2, nos. 1–2, pp. 1–154, 2005.

[10] D. Colleran, C. Portmann, A. Hassibi, C. Crusius, S. Mohan, S. Boyd, T. Lee, and M. del Mar Hershenson, "Optimization of phase-locked loop circuits via geometric programming," in *Proc. Custom Integr. Circuits Conf.*, 2003, pp. 377–380.

[11] A. R. Conn, K. Scheinberg, and L. N. Vicente, "Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points," *SIAM J. Optimiz.*, vol. 20, pp. 387–415, Apr. 2009.

[12] A. Corana, M. Marchesi, C. Martini, and S. Ridella, "Minimizing multimodal functions of continuous variables with the simulated annealing algorithm," *ACM Trans. Math. Softw.*, vol. 13, pp. 262–280, Sep. 1987.

[13] M. del Mar Hershenson, "CMOS analog circuit design via geometric programming," in *Proc. Am. Contr. Conf.*, 2004, pp. 3266–3271.

[14] M. del Mar Hershenson, S. Boyd, and T. Lee, "GPCAD: A tool for CMOS op-amp synthesis," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 1998, pp. 296–303.

[15] G. Gielen, H. Walscharts, and W. Sansen, "Analog circuit design optimization based on symbolic simulation and simulated annealing," *IEEE J. Solid-State Circuits*, vol. 25, no. 3, pp. 707–713, Jun. 1990.

[16] G. G. Gielen and W. M. Sansen, *Symbolic Analysis for Automated Design of Analog Integrated Circuits*. Norwell, MA: Kluwer, 1991.

[17] M. Grant and S. Boyd. (2010, Oct.). *CVX: Matlab Software for Disciplined Convex Programming, Version 1.21* [Online]. Available: http://cvxr.com/cvx

[18] B. Hajek, "Cooling schedules for optimal annealing," *Math. Oper. Res.*, vol. 13, pp. 311–329, May 1988.
[19] J. Harvey, M. Elmasry, and B. Leung, "STAIC: An interactive framework for synthesizing CMOS and biCMOS analog circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 11, no. 11, pp. 1402–1417, Nov. 1992.
[20] M. Hershenson, "Design of pipeline analog-to-digital converters via geometric programming," in *Proc. Int. Conf. Comput.-Aided Des.*, 2002, pp. 317–324.
[21] M. del Mar Hershenson, S. Boyd, and T. Lee, "Optimal design of a CMOS op-amp via geometric programming," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 20, no. 1, pp. 1–21, Jan. 2001.
[22] K.-L. Hsiung, S.-J. Kim, and S. Boyd, "Tractable approximate robust geometric programming," *Optimiz. Eng.*, vol. 9, no. 2, pp. 95–118, 2008.
[23] D. Huang, T. Allen, W. Notz, and R. Miller, "Sequential kriging optimization using multiple-fidelity evaluations," *Structural Multidisciplinary Optimiz.*, vol. 32, no. 5, pp. 369–382, 2006.
[24] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *J. Global Optimiz.*, vol. 21, no. 4, pp. 345–383, 2001.
[25] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Global Optimiz.*, vol. 13, no. 4, pp. 455–492, 1998.
[26] J. Kim, J. Lee, and L. Vandenberghe, "Techniques for improving the accuracy of geometric-programming based analog circuit design optimization," in *Proc. Int. Conf. Comput.-Aided Des.*, 2004, pp. 863–870.
[27] M. Krasnicki, R. Phelps, R. A. Rutenbar, and L. R. Carley, "MAELSTROM: Efficient simulation-based synthesis for custom analog cells," in *Proc. ACM/IEEE Des. Automat. Conf.*, Jun. 1999, pp. 945–950.
[28] F. Leyn, W. Daems, G. Gielen, and W. Sansen, "Analog circuit sizing with constraint programming modeling and minimax optimization," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 3. Jun. 1997, pp. 1500–1503.
[29] X. Li, P. Gopalakrishnan, Y. Xu, and L. Pileggi, "Robust analog/RF circuit design with projection-based posynomial modeling," in *Proc. Int. Conf. Comput.-Aided Des.*, 2004, pp. 855–862.
[30] P. Maulik and L. Carley, "High-performance analog module generation using nonlinear optimization," in *Proc. IEEE Int. ASIC Conf.*, Sep. 1991, pp. 13–15.
[31] P. Maulik, L. Carley, and R. Rutenbar, "Integer programming based topology selection of cell-level analog circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 14, no. 4, pp. 401–412, Apr. 1995.
[32] P. Maulik, M. Flynn, D. Allstot, and L. Carley, "Rapid redesign of analog standard cells using constrained optimization techniques," in *Proc. IEEE Custom Integr. Circuits Conf.*, May 1992, pp. 65–69.
[33] T. McConaghy, P. Palmers, G. Gielen, and M. Steyaert, "Simultaneous multi-topology multi-objective sizing across thousands of analog circuit topologies," in *Proc. Des. Automat. Conf.*, 2007, pp. 944–947.
[34] E. Ochotta, R. Rutenbar, and L. Carley, "Synthesis of high-performance analog circuits in ASTRX/OBLX," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 15, no. 3, pp. 273–294, Mar. 1996.
[35] G. Stehr, M. Pronath, F. Schenkel, H. Graeb, and K. Antreich, "Initial sizing of analog integrated circuits by centering within topology-given implicit specification," in *Proc. IEEE/ACM ICCAD*, Nov. 2003, pp. 241–246.
[36] C. Toumazou, G. Moschytz, and B. Gilbert, *Trade-offs in Analog Circuit Design: The Designer's Companion, Part 1*. Norwell, MA: Kluwer, 2002.
[37] Y. Xu, K.-L. Hsiung, X. Li, I. Nausieda, S. Boyd, and L. Pileggi, "OPERA: Optimization with ellipsoidal uncertainty for robust analog IC design," in *Proc. Des. Automat. Conf.*, 2005, pp. 632–637.
[38] G. Yu and P. Li, "Yield-aware analog integrated circuit optimization using geostatistics motivated performance modeling," in *Proc. IEEE/ACM ICCAD*, Nov. 2007, pp. 464–469.

Ashish Kumar Singh received the B.Tech. degree in computer science from the Indian Institute of Technology Kanpur, Kanpur, India, in 2001, and the M.S. degree from the Royal Institute of Technology, Stockholm, Sweden, and the Ph.D. degree in electrical engineering from the University of Texas, Austin, in 2003 and 2007, respectively.

He is currently a Senior Researcher with Terra Technology, Chicago, IL. His current research interests include inventory optimization in the supply chain networks under demand uncertainty.

Dr. Singh received the IEEE/ACM William J. McCalla Best Paper Award in the International Conference on Computer-Aided Design in 2006.

Kareem Ragab received the B.Sc. degree in electrical engineering from Ain Shams University, Cairo, Egypt, in 2003, where he ranked top in the Department of Electronics and Communication Engineering and graduated with the highest honor, and the M.Sc. degree in electronics engineering from the same university in 2008. During his Masters thesis, he focused on the design of high-frequency gm-C filters for radio receivers. He is currently pursuing the Ph.D. degree in integrated circuits and systems with the Department of Electrical and Computer Engineering, University of Texas, Austin.

Since 2008, he has been with the University of Texas. His research interests include analog, mixed-signal, radio-frequency circuit, system design, and optimization. His current research interests include the design of low-power digitally assisted data converters.

Mario Lok received the B.S. degree (with the highest honor) in engineering physics from the University of British Columbia, Vancouver, BC, Canada, in 2008, and the M.S. degree in electrical engineering from the University of Texas at Austin, Austin, in 2010, researching on statistical adaptive circuit design. He is currently pursuing the Ph.D. degree in power electronics design for milligram-scaled robots with Harvard University, Cambridge, MA.

Constantine Caramanis (M'06) received the Ph.D. degree in electrical engineering and computer sciences from the Massachusetts Institute of Technology, Cambridge, in 2006.

Since 2006, he has been with the faculty of the Department of Electrical and Computer Engineering, University of Texas, Austin. His current research interests include robust and adaptable optimization, machine learning and high-dimensional statistics, with applications to large-scale networks, and computer-aided design.

Dr. Caramanis received the NSF CAREER Award in 2011.

Michael Orshansky received the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley (UC Berkeley), in 2001.

He is currently an Associate Professor of electrical and computer engineering with the Department of Electrical and Computer Engineering, University of Texas, Austin. Prior to joining the University of Texas, he was a Research Scientist and Lecturer with the Department of Electrical Engineering and Computer Sciences, UC Berkeley. He is the author, with S. Nassif and D. Boning, of the book, *Design for Manufacturability and Statistical Design: A Constructive Approach*. His current research interests include design optimization for robustness and manufacturability, statistical timing analysis, and design in fabrics with extreme defect densities.

Dr. Orshansky received the National Science Foundation CAREER Award in 2004 and the ACM SIGDA Outstanding New Faculty Award in 2007. He received the 2004 IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING Best Paper Award, and the Best Paper Award in the Design Automation Conference in 2005, the International Symposium on Quality Electronic Design in 2006, and the International Conference on Computer-Aided Design in 2006.