Bias-driven Robust Analog Circuit Sizing Scheme

Viswakiran Popuri
Electrical Engineering Department, University of California Los Angeles
{viswa}@ee.ucla.edu

***Abstract*—A robust automated analog circuit synthesis has been considered the holy grail in semiconductor CAD and design industry. We review the published literature on analog sizing problem. This work proposes a new bias-driven analog circuit sizing scheme. The importance of biasing in analog integrated circuits is emphasized. Our conjecture is that if the circuit is *well biased,* the variation of performance across process variations is lower and that its transient response is better than a circuit whose biasing is not optimized. We show that there is upto 100% reduction in the standard deviation values of performance specifications and there is a significant improvement (10-100%) in the total harmonic distortion just by biasing the circuit better.**

I. Introduction

Most real world signals are inherently analog. However, information storage, processing is done very efficiently in digital domain. This explains the booming market share of mixed signal integrated circuits (ICs) seen today with a reported average growth rates well above 20% [1]. Though the number of transistors is less, the engineering effort involved in designing analog circuits is not. Most steps in analog design are basically still handcrafted, ranging from extensive and repeated SPICE simulation runs through manual place and route with assistance of parameterized device generators. All this does not fit well with the short design cycles of time-to-market critical applications. Therefore, it is necessary to develop computer-aided design (CAD) tools that assist designers with the design of analog and mixed-signal integrated systems and eventually automate (large parts of) it.

Research in analog CAD is not as mature as the digital CAD. There are some fundamental differences between analog and digital circuits that prevent adaptation of digital CAD flow to analog design. Firstly, the digital design typically has fewer performance specifications (e.g., speed and power) than analog circuits (e.g., gain, accuracy, linearity, signal-to-noise ratio, impedance matching). In addition, all the specifications are tightly coupled in analog and the tradeoffs are often not as straight forward like in digital (speed vs power).

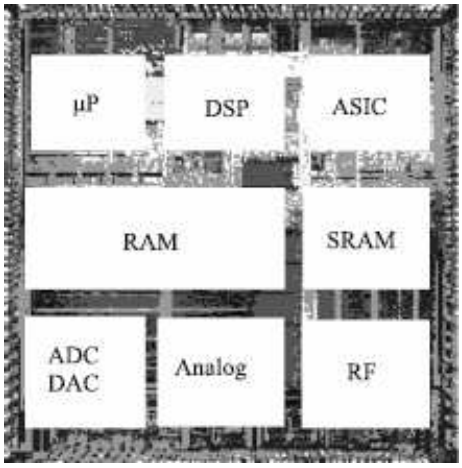


Fig. 1. A typical mixed signal IC

Secondly, the performance models in digital are typically static. This is not the case in analog as accuracy performance requires SPICE level simulations. Abstraction in digital domain can be leveraged and design can be done at a very high level e.g., at software level. In analog, the design needs to be custom crafted at all levels of hierarchy (architecture, schematic and layout) due to the sensitivity of the performance towards physical design.

Consequently, because of tighter coupling between specifications (leading to high dimensional performance models) and the performance models themselves requiring computationally expensive SPICE simulations, library-based design is impractical in analog circuits. Both the constraints are relatively relaxed in digital thus allowing standard cell library-based design.

Finally, integration of analog, RF and digital design onto the same die opens up a deluge of mixed signal integration issues like substrate noise etc. Hence, the analog CAD needs to be looked at from a new perspective. In this paper, we propose a bias-driven robust analog circuit sizing scheme where the bias information is incorporated in the optimization process and we see that this leads to a well-behaved design.

The organization of the paper is as follows. Section II describes the

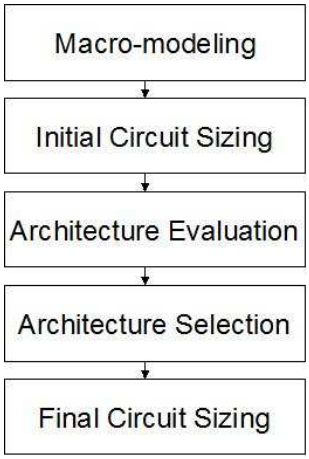


Fig. 2. Analog circuit design flow

analog circuit sizing problem and reviews the performance modeling and optimization procedures in the analog sizing approaches taken so far. Section III explains the proposed bias-driven robust analog circuit sizing methodology. Section IV shows the example circuits and results. Section V concludes the work and gives future research directions.

II. Analog Circuit Sizing: Review of Performance models and Optimization approaches

A. Analog Circuit Sizing Problem

Analog design procedure primarily consists of two stages: architecture selection and circuit sizing. Here we concentrate on the circuit sizing problem. In typical circuit sythesis tools, circuit sizing is formulated as a constrained optimization problem.

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g(x) \geq 0 \\ & h(x) = 0 \\ & X_L < x < X_H \end{aligned}$$

Where

Vector *x* corresponds to the design variables (e.g., Device dimensions).

f(x) is objective to be minimized (e.g., Area, Power etc).

g(x) corresponds to user-defined performance specifications (e.g., Minimum gain, SNR etc).

h(x) are equality constraints to be met (e.g., KCL, KVL equations).

Circuit synthesis is the inverse operation of circuit analysis, where sub-block parameters (such as device sizes and bias values) are given and resulting performance of the overall block is calculated (as is done in SPICE). During synthesis, the block performance is specified and values for the sub-block parameters needed to meet these performance specifications have to be determined. This inverse process is not a one-to-one mapping, but usually is an under-constrained problem with many degrees of freedom. Different analog circuit synthesis systems can essentially be classified in the way they eliminate these degrees of freedom. The two key aspects that differentiate the various analog sizing approaches are: performance modeling and optimization algorithm used in the design space exploration.

B. Performance Modeling

Here, we differentiate between behavioral and performance modeling [2]. Consider a system *S* transforming an input signal *E* into an output signal *Y* (Figure (4)). The mathematical modeling of this input-output relationship is called *behavioral modeling*. The figure depicts the same system seen from a designer's point of view: a number of design parameters (*X*) cause the system to exhibit a particular performance (*P*). The modeling of this relationship is called *performance modeling*. The following section concentrates on the latter.

Fig. 3. Electronic system seen (a) as a system that relates an input signal E to an output signal Y and (b) as a system for which a set of design parameters X have to be chosen to obtain a specified performance P

The performance models can be classified as shown in the Figure (4).

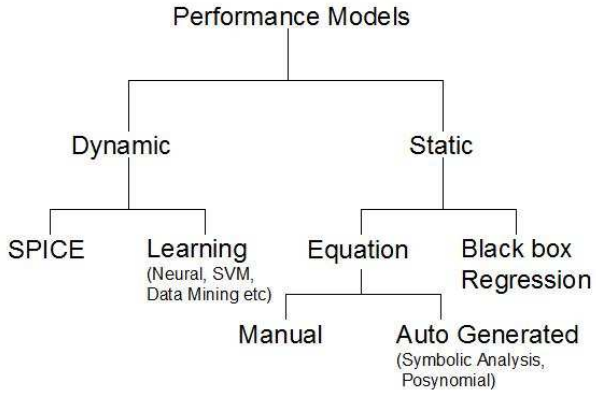


Fig. 4. Performance estimation approaches

1) Dynamic models: SPICE models offer state-of-the-art accuracy but have significant complexity and are hence slow to be simulated. They also run into convergence problems. However, the computation capacity increased tremendously over time. With the advent of parallel processing, simulation time is no longer a bottleneck. In the modern IC design flows, it has become mandatory to validate design with the best available models and hence latest analog sizing tools use the SPICE model in an iterative optimization loop [3] [4].

Another class of dynamic models is learning based models. They use neural networks [5], support vector machines [6] and data mining [7] techniques to build macro models. They require a training sequence, which is usually based upon SPICE simulation data and that covers significant portions of the design space. However, such models do not provide much insight. Also, the initial set up time is high considering that huge multi dimensional data needs to be analyzed.

2) Static models: Manually created performance models were used in the early approaches ([8], [9], [10]). Transistor was assumed to follow simple first order equations. The performance of the circuit is expressed in terms of the transistor parameters (small signal parameters like transconductance, output resistance etc) and bias currents. These are the textbook based equations which are still used in the initial hand calculations of analog design. Since the sizing problem is usually under-constrained, the assumptions and initial estimates are based on the experience of the designer.

As the circuits got complicated, writing analytical equations by hand became cumbersome. A symbolic simulator [11] was developed to read in circuit topology and automatically generate design equations needed to evaluate the circuit performance [12].

Almost all of the symbolic analysis research carried out in the past concerned the analysis of linear circuits in frequency domain. For lumped, linear, time invariant circuits, the symbolic network functions obtained are rational functions in the complex frequency variable *x* (*s* for continuous time and *z* for discrete time circuits) and the circuit elements p_j that are represented by a symbol (instead of a numerical value).

$$H(x) = \frac{N(x)}{D(x)} = \frac{\sum_i x^i . a_i(p_1, \ldots, p_m)}{\sum_i x^i . b_i(p_1, \ldots, p_m)} \quad (1)$$

In 1 the partially expanded form on the right the coefficients $a_i(\ldots)$ and $b_i(\ldots)$ of each power of *x* for both the numerator and denominator polynomial are symbolic polynomial functions in the circuit elements p_j. These polynomials can be in nested format or expanded into the sum-of-product form.

Consider the active RC filter of Figure (5). Starting from the circuit description of this filter, a symbolic analysis program will return the following symbolic expression for the transfer function, *H(s)*, of this filter. In Equation (**??**),G_i's denote the conductances of the resistors used.

$$\begin{aligned} H(s) = [&-G_4G_8(G_1G_2G_9 + G_1G_3G_9 + G_2G_6G_9 + \\ &G_2G_6G_1) + sG_7G_2(G_1G_3G_9 + G_1G_3G_1 - G_2G_5G_9 \\ &-G_2G_5G_1) - s^2G_2G_7C_1C_2(G_9 + G_1)]/[G_1(G_9 + \\ &G_1)(G_4G_6G_8 + sG_5G_7C_2 + s^2G_7C_1C_2)] \end{aligned} \quad (2)$$

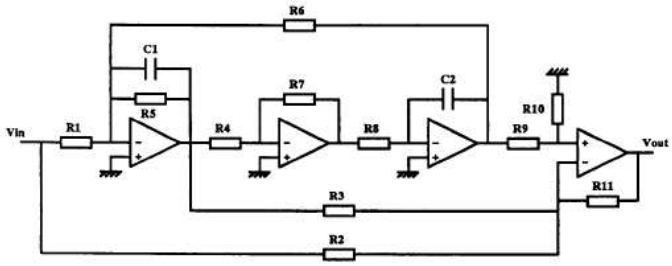


Fig. 5. Active RC filter to illustrate the principle of symbolic analysis.

Computer aided symbolic analysis is possible for the small signal analysis behavior (both linear and weakly nonlinear) of analog circuits up to the complexity of a few tens of transistors. The symbolic equations can also be used to provide insight into the behavior of an analog circuit.

Another efficient approach proposed to describe circuit performance is given in [13]. The performance is described with a set of analytical equations that have a special form (i.e., they are convex Figure (6)) and the design problem is cast as a convex optimization problem, which is then solved very efficiently and globally by numerical algorithms like geometric programming. The problem formulation for optimization is described in Section II-C2 .

It is shown in [13] that most of the basic transistor parameters like the gate overdrive, transconductance as well as the circuit performance specifications like the small signal gain, 3-dB bandwidth, common mode rejection ratio, slew rate, noise etc can be expressed as *posynomial* functions (defined in below) of device sizes.

Let *x* be a vector (x_1,...,x_n) of n real, positive variables. A function *f* is called a *posynomial* function of *x* if it has the form

$$f(x_1, \ldots, x_n) = \sum_{k=1}^{t} c_k x_1^{\alpha_{1k}} x_2^{\alpha_{2k}} \ldots x_n^{\alpha_{nk}} \quad (3)$$

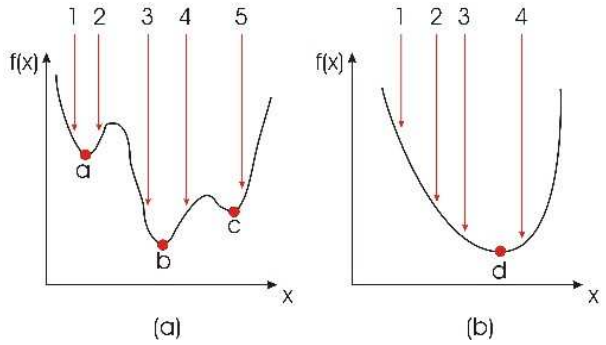


Fig. 6. Optimization problem with nonconvex objective function (a) and convex objective (b)

where $c_j \geq 0$ and $\alpha_{ij} \in \mathbb{R}$.

As seen in the Section II-C2, geometric programming is a very efficient way of reaching a global optimum for functions which can be expressed as *posynomials*. Geometric programming approach is also taken in [14]. In analog designs, however, it is not guaranteed that all the performance specifications can be cast into a convex function.

3) Black box regression models: As the transistor dimensions reduce, the transistor models grow in complexity and first order calculations result in significant errors. Then, the performance models were expressed as a function of the device sizes with the coefficients fitted using SPICE simulations. Circuit sizing methods used these symbolic models for optimizing the device sizes. A lot of research has been done on symbolic analysis and efforts were made to incorporate some non linear effects. The performance estimation is very fast with such symbolic models, however they suffer from inaccuracies. The accuracy of the modeling is improved by using SPICE simulation to fine tune the coefficients in the model.

In general, a mathematical model is assumed and the parameters of the model are fitted so that the model corresponds as closely as possible to the simulated or measured data of the real system. Standard mathematical fitting techniques like interpolation or least squares regression are well known. However, the use of these techniques in analog integrated circuit design has always been rather limited. Techniques from design of experiments provide a mathematical basis to select a limited but "optimal" set of samples points needed to fit a black box model. Well known and often used sampling schemes range from full and fractional factorial design, Placket-Bruman and Taguchi schemes, to Latin hypercube and even random design. The combination of design of experiments with a standard fitting technique often is called response surface modeling.

In symbolic analysis, models are derived via topology analysis. Its main weakness is that it is limited to linear and weakly nonlinear circuits. Leveraging SPICE simulation in modeling is promising because simulators readily handle nonlinear circuits, as well as environmental effects, manufacturing effects, and different technologies. However, the models are constrained to templates, which restricts the functional form.

In [15], symbolic models are generated with a more open ended functional forms (arbitrarily nonlinear) and at the same time ensure that the models are interpretable. Genetic programming is applied as a means of traversing the space of possible symbolic expressions. It provides a set of models that trade off normalized mean squared error and complexity, which is dependent on the number of basis functions etc. A grammar is specially designed to constrain the search to a canonical form for functions. In this approach, the posynomial performance models are handcrafted. In [16], SPICE simulation data is used to fit a second order polynomial. Then, this model is automatically be converted into a posynomial model, which can be solved using geometric programming.

Apart from the above models, there are various approaches that use cubic splines [17] with adaptive sampling etc. The regression models have better accuracy than the static models as they are generated using the simulation data. To summarize, in static models, the coverage of design space is not good enough to encompass a highly nonlinear design space of analog circuits. The sizing tools using these models could design a circuit to meet small signal specifications but might not fare well to meet the large signal specifications (which usually represent the nonlinear behavior of circuits).

C. Optimization procedures

The second aspect of the analog circuit sizing problem is the optimization procedure used in the design space exploration. The various methods employed are categorized in Figure (7).

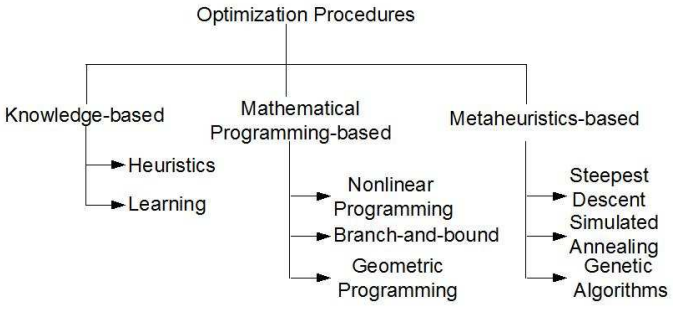


Fig. 7. Optimization procedures

1) Knowledge-based heuristics: Knowledge-based methods have been widely used in analog CAD. Examples include special heuristics based systems like IDAC [8], OASYS [9]; systems based on fuzzy logic like FASY [18].

In heuristics based approaches, the design knowledge is codified in a computer readable format. They are typically used in topology selection where the choice is based on decision trees (OPASYN [19]) derived from the designer's experience. For example, a high voltage gain requires a cascode structure, a high swing calls for a two stage opamp. To size a schematic, IDAC [8] makes use of three types of knowledge, i.e., knowledge specific to the schematic, general circuit knowledge (for instance, how to size cascode devices) and knowledge common for a family of circuits (for example, how to stabilize an amplifier, how to improve the slew rate, etc.). IDAC provides a set of topologies for several kinds of blocks, e.g., opamps, comparators, voltage and current references, etc., and can size these circuits to meet user specifications.

OASYS [9] approaches the problem with a hierarchically structured framework. The framework is knowledge intensive in that it relies heavily on the codification of mature analog design expertise. The analog circuit topologies are represented as a hierarchy of templates of abstract functional blocks (called design styles) each with associated detailed design knowledge. This hierarchical structure has two important features: it decomposes the design task into a sequence of smaller tasks with uniform structure, and it simplifies the reuse of design knowledge.

The idea of selecting from among mature design styles follows directly from standard manual design practices. Analog designers usually attempt first to find a known, mature topology to fit a given specification, and only embark on the much more difficult task of designing a new topology if this search fails to provide any reasonable candidates. Translation involves knowledge of how performance specifications for a high level block (a design style) should be transformed into specifications for each sub-block. For example, typical performance specifications that form the *input* to an opamp level translation task will include dc gain, slew rate, phase margin, etc. The *output* of this translation task is a set of designed sub-blocks, specifically, a set of *input* specifications for each of these sub-blocks. The process then repeats inside each sub-block.

BLADES [10] uses an expert system based approach. It uses both formal and intuitive knowledge in the design process and attacks the problem using divide-and-conquer strategy. The circuit is partitioned into sub-blocks and the specifications on the individual sub-blocks are derived from the global system specification. The design knowledge is classified as procedures to handle design problems systematically (e.g., the design of differential amplifiers from a differential pair, load and current source), or special rules to handle special situations (e.g., short circuit protection and compensation). Most of these rules are in the "if-then" format, where the "if" part represents the instance of the rule application, and the "then" part is the action to be taken. Text book design equations are used and the topology choice and sizing are simple rule based. This approach is similar to OPASYN.

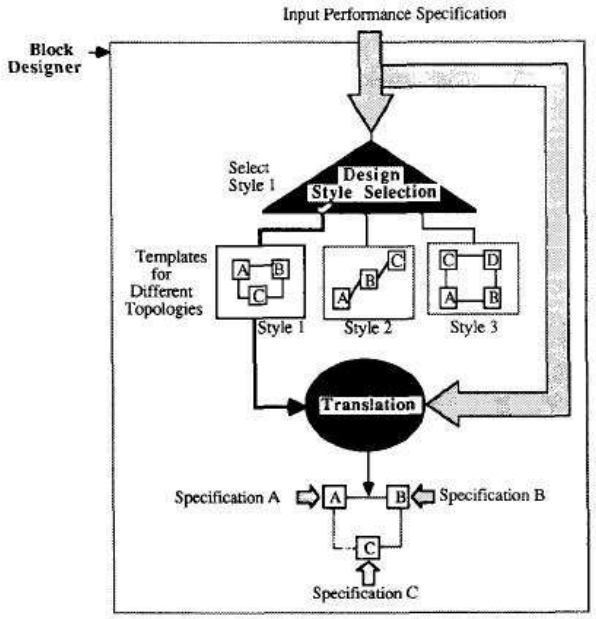


Fig. 8. OASYS: Topology selection and translation processes.

Fuzzy logic based systems have been specially proposed to deal with "uncertain" information and have proven to be very efficient in capturing human expertise. FASY [18] uses fuzzy logic in the topology selection process. Then, a two-phase optimizer sizes all the components of the selected topology to minimize a user-defined cost function. First phase uses simple analytical models with simulated annealing to avoid getting trapped in local minimum. Second phase uses SPICE in each movement and the results of the first optimization phase as a starting point and the wellknown Flectcher-Powell conjugate gradient method to obtain the final result.

In [20], data mining techniques are used to extract a specs-to-topology decision tree, global nonlinear sensitivities on topology and sizing variables and determining analytical expressions of performance tradeoffs. A database containing thousands of pareto optimal designs across five objectives is used.

2) Mathematical programming: In mathematical programming based optimization, the analog sizing problem is formulated as a constrained nonlinear optimization problem ([19]) and it is solved using various nonlinear solvers like NPSOL. NPSOL employs a dense SQP algorithm and is especially effective for non linear problems whose functions and gradients are expensive to evaluate. The functions should be smooth but need not be convex.

Branch-and-bound is an approach used for solving such nonlinear optimization problems [21] with a guaranteed globally optimal solution. It is a systematic enumeration of all candidate solutions, where large subsets of fruitless candidates are discarded, by using upper and lower estimated bounds of the quantity being optimized. However, it tends to get very slow even for a sizeable number of parameters.

Convex functions are a special form of equations that lend themselves to very efficient global optimization algorithms. Extremely powerful interior-point methods are developed for general convex optimization problems. These methods can solve large problems, with thousands of variables and tens of thousands of constraints, very efficiently. The other main advantage is that the methods are truly global, i.e., the global solution is always found, regardless of the starting point. Infeasibility is unambiguously detected, i.e., if the methods do not produce a feasible point they produce a certificate that proves the problem is infeasible. Also, the stopping criteria are completely nonheuristic. At each iteration a lower bound on the achievable performance is given.

Geometric programming is a certain type of convex problem. A geometric programming is an optimization problem of the form

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 1, i = 1, \ldots, m. \\ & g_i(x) 1 i = 1, \ldots, p. \\ & x_i > 0 i = 1, \ldots, n. \end{aligned}$$

Where $f_0, \ldots, f_m$ are posynomial functions and $g_1, \ldots, g_p$ are monomial functions.

The *posynomial* functions are defined earlier. They are closed under sums, products, nonnegative scaling. Monomials are closed under products, division.

One of the main disadvantages is that the types of problems, performance specifications, and objectives that can be handled are far more restricted than any of the methods described above.

3) Metaheuristics: The simplest among the metaheuristics-based optimization is the steepest descent method (used in OPASYN). To find a local minimum of a function using steepest descent, one takes steps proportional to the negative of the gradient of the function at current point. Its main disadvantage is that it gets trapped in the local minima.

Simulated annealing (SA) is a popular method that can avoid becoming trapped in a locally optimal design. In SA, the algorithm replaces the current solution by a random "nearby" solution with a probability that depends on two factors: the difference between the corresponding function values and a global parameter T (called temperature), that is gradually decreased during the process. In principle it can compute the globally optimal solution, but in implementations there is not guarantee at all, since, for example, the cooling schedules called for in the theoretical treatments are not used in practice. Moreover, no realtime lower bound is available, so termination is heuristic. Like some knowledge-based and mathematical programming based methods, SA allows a very wide variety of performance measures and objectives to be handled. SA has been used in several tools such as ASTR/OBLX [22]. The main advantages of SA are that it handles discrete variables well, and greatly reduces the chances of finding a non globally optimal design. The main disadvantage is that it can be very slow, and cannot (in practice) guarantee a globally optimal solution.

In DARWIN [23], genetic algorithm (GA) is used to perform simultaneous topology selection and circuit sizing. GA belongs to a class of evolutionary algorithms that can be used to find near optimal solution for a wide variety of problems. GAs maintain a population of individuals P(t) for iteration t. During, each generation, all individuals are evaluated to give some measure of their fitness. On the basis of this fitness, part of the population is selected to maintain for the next generation. The vacant places in generation t+1 are filled up by new individuals, generated by means of applying crossover and mutation operators. SEAS [24] also uses GA for topology selection.

The GA is not well-suited for fine-tuning structures which are close to optimal solution [25]. Memetic algorithms [26] can be viewed as a special kind of GA with a local hill climbing. The role of local search in memetic algorithm is to locate the local optimum more efficiently than the GA.

The main disadvantage of SA and GA based approaches is that they tend to get very slow and huge number of iterations are required to reach the global optimal solution.

III. Bias-driven robust analog circuit sizing methodology

A. Importance of biasing in analog circuit sizing

Unlike digital circuits, analog circuits rely on good biasing for functionality and performance. For example, the transistor has to be biased in saturation region to act as a good amplifier. The sizing rules approach [27] talks about adding the bias constraint to the analog sizing optimization problem. The bias constraint is that all transistors should have a positive "margin" (V_{ds} - V_{dsat}). However, in an analog circuit, some transistors require more margin than others to stay in proper region of operation across signal swings and environmental variations (process, supply voltage and temperature). We proposed a bias-driven robust analog circuit sizing methodology where this aspect of bias requirement is captured while sizing the circuit. This is essential in deep sub-micron CMOS technologies for two reasons. First, the transition between linear and saturation regions is blurred and having just a positive margin does not ensure good transistor properties (e.g., having high gain) in all conditions. Second, as the supply voltages are scaling down, the voltage headroom is getting crunched. In a cascode structure, the transistors are left with little margins and again having the right combinations of margins proves advantageous in low power analog designs.

In the proposed method, margins of all the transistors are incorporated in the optimization routine as a constraint through a margin function (equation 4). Using three example circuits, we show that there are two apparent advantages in this methodology compared to a case where optimization is driven purely by performance specifications. They are:

- The variation of the performance specifications is reduced.
- The transient behavior of the circuit is improved.

The margin function is defined as below.

$$V_m(x) = \sum_{i=1}^{N} w_i \frac{(V_{ds} - V_{dsat})_{i,x}}{(V_{ds} - V_{dsat})_{i,0}} \tag{4}$$

Where

x represents the vector denoting the sizes of N transistors.

w_i is the weight allocated for the normalized margin of transistor i ($\sum$w=1).

$(V_{ds} - V_{dsat})_{i,x}$ denotes the margin of transistor i at size x.

$(V_{ds} - V_{dsat})_{i,0}$ denotes the margin of transistor i at initial size.

$V_m(x)$ denotes the overall margin of the design at vector x. For the initial design, its value is 1. A design with better margin would have a higher value of $V_m(x)$.

B. Calculation of the weights

In order to calculate the weights, first, the sensitivities of the margins of various transistors w.r.t the sizing of all transistors is calculated. The sensitivity towards the sizing serves two purposes: it tries to allocate higher margin to transistors which are more sensitive to the sizing during optimization, it also serves as a proxy for the sensitivity towards process variations. All the process variations (V_{th}, t_{ox}, μ and dimension variations) can be seen as variation of the strength of transistor and hence sizing can be seen as a proxy for the process variations.

In the first step, the circuit is provided an initial sizing. This could be done in many ways. First way is to provide a constant overdrive to all the transistors. Second way is heuristic (used here) and is an output of an "initial" design from a designer. Third way is to run the optimization on the circuit using a subset of performances as constraints. The key is to provide a decent starting point so that the sensitivities (of the margin w.r.t sizing) do not change drastically with sizing.

The second step in weight calculation is to change the size of each transistor one at a time and calculate the sensitivity of the margin w.r.t sizing perturbation. For each transistor, the root mean square value of all the sensitivities is computed and is denoted by $Sens_i$ (See equation **??**). The sensitivity matrix is shown below. First row represents the transistor index. The last row shows the computed $Sens_i$'s. The remaining rows show the margins of all the transistors for size perturbations.

$$\begin{pmatrix} 1 & \dots & i & \dots & N \\ Vm_{1,1} & \dots & Vm_{1,i} & \dots & Vm_{1,N} \\ Vm_{2,1} & \dots & Vm_{2,i} & \dots & Vm_{2,N} \\ \dots & \dots & Vm_{j,i} & \dots & \dots \\ Vm_{M,1} & \dots & Vm_{M,i} & \dots & Vm_{M,N} \\ Sens_1 & \dots & Sens_i & \dots & Sens_N \end{pmatrix}$$

$$Sens_i = \sqrt{\sum_{j=0}^{M} \left(\frac{(Vm_{j,i} - Vm_{0,i})}{\Delta W/L}\right)^2} \tag{5}$$

Third and the crucial step is to shape the sensitivities using "coefficient-shaping factor". Coefficient-shaping factor for each transistor is chosen based on the sensitivity of the final performance

on this transistor's margin. Currently, this is chosen manually. However, this could be chosen more intelligently such that the sensitivity is shaped accurately. The sensitivities are multiplied with the coefficient-shaping factor and then normalized (such that $\sum w_i = 1$) to result in the weights w_i's.

The margin function is provided as a constraint to the optimization engine. This performance + bias-driven optimization is henceforth denoted as PBO. The pure performance-driven optimization is denoted as PO.

As a comparison, PO is performed with a simpler bias margin constraint of having a flat margin specification on all the transistors. This optimization is denoted as POFB.

IV. Examples and Results

Analog circuit optimizer (shown in Figure (9)) available in the Cadence *Analog Design Environment* is used as the sizing engine for performing PO, PBO and POFB on three circuits. Monte Carlo simulation (500 runs) is used to extract the statistics of the performance specifications. The histograms of the first example are shown in the Figure (10).

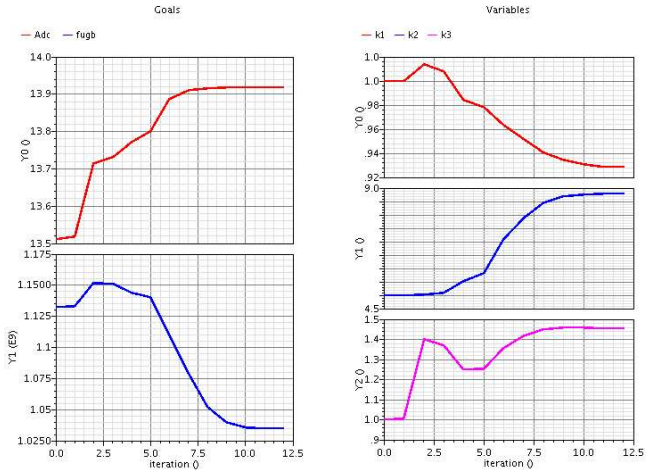
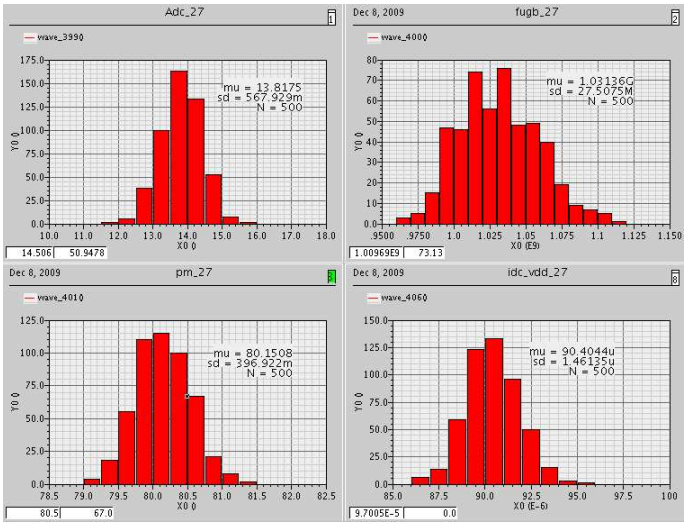


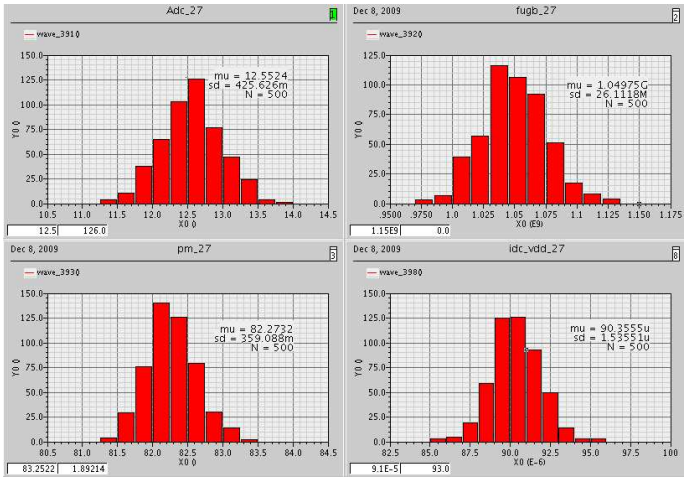
Fig. 9. Circuit size optimization tool in Cadence Environment. The performance (goals) variation across iterations are shown on the left. The parameter (variables) perturbations across iterations are shown on the right.

A. Reduction in the variation of performance specifications

1) Single-stage Differential amplifier: The first example circuit is a simple differential amplifier (schematic in Figure (11)). The $Sens_i$'s, coefficient-shaping factors and the weights calculated are shown in Table I. Also, the relative sizes and margins of transistors after PO, PBO and POFB are shown in the same table. Note the coefficient



(a)



(b)

Fig. 10. Histograms of ac performance specifications of opamp1 optimized using PO (a) & PBO (b)

shaping factor is chosen as 0 for M1 as it is estimated heuristically that its margin would not affect the performance specification significantly. The targets for the performance specifications are given based on the topology. Currently, phase margin and supply current are not constrained in all three designs.

The Table II has the statistics of the performance specifications resulting from Monte Carlo runs (N=500). The σ values of the specifications are seen to be lower for the design optimized using PBO than the one optimized using PO.

2) Single-stage Differential amplifier with telescopic cascode: The second example circuit is a differential amplifier with telecscopic

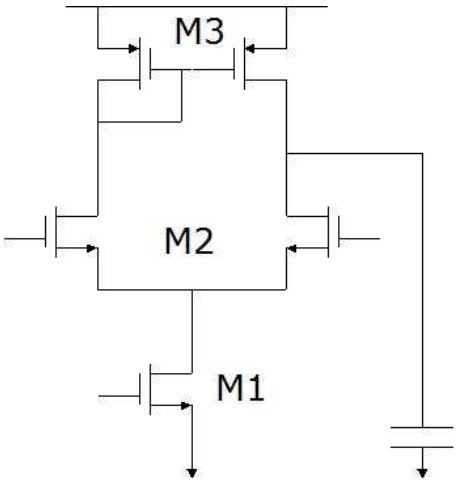


Fig. 11. Opamp1: Single-stage Differential amplifier.

TABLE I

OPTIMIZATION PARAMETERS AND RESULTS OF OPAMP1.

Transistor	M1	M2	M3
$Sens_i$	0.3262	0.5080	0.1658
Coeff. Shaping Factor	0	1	1
w_i	0.0	0.7539	0.2451
Relative Size(PO)	0.93	8.79	1.45
Relative Size(POF)	7.211	8.931	1.496
Relative Size(PBO)	1.1	7.51	4.36
Margin(PO) in mV	200	242	409
Margin(POFB) in mV	266	244	408
Margin(PBO) in mV	207	345	374

cascode (schematic in Figure (12)). The $Sens_i$'s, coefficient-shaping factors and the weights calculated are shown in Table III. Also, the relative sizes and margins of transistors after performance-driven optimization (PO) and performance + bias-driven optimization (PBO) are shown in the same table. Note the coefficient shaping factor is chosen as 0 for M1 as it is estimated heuristically that its margin would not affect the performance specification significantly. Similarly it is chosen as 2 for M4 and M5 transistors. The targets for the performance specifications are given based on the topology. The target for margin function is 1.5 as cascode has lesser leeway in margin improvement due to stacking and its value is 0.932 for PO and 1.321 for PBO.

The Table IV has the statistics of the performance specifications resulting from Monte Carlo runs (N=500). The σ values of the specifications are seen to be lower for the design optimized using PBO than the one optimized using PO.

3) Two-stage Miller compensated OTA: The third example circuit is a two-stage miller compensated OTA (schematic in Figure (13)). The $Sens_i$'s, coefficient-shaping factors and the weights calculated

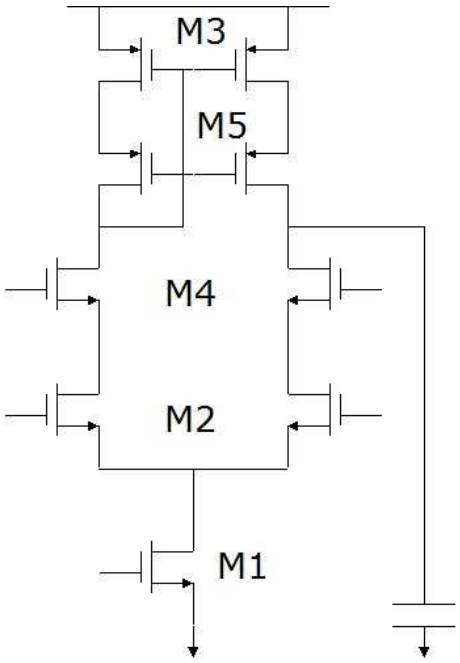


Fig. 12. Opamp2: Single-stage Differential amplifier with telescopic cascode

TABLE III

OPTIMIZATION PARAMETERS AND RESULTS OF OPAMP2.

Transistor	M1	M2	M3	M4	M5
$Sens_i$	0.074	0.3532	0.1111	0.4362	0.0255
Coeff. Shaping Factor	0	1	1	2	2
w_i	0.0	0.2546	0.08	0.6287	0.0367
Relative Size(PO)	1.44	0.79	0.23	0.25	0.76
Relative Size(POF)	8.209	0.77	0.14	0.246	0.97
Relative Size(PBO)	1.266	0.51	0.46	0.23	0.77
Margin(PO) in mV	61	90	64	181	237
Margin(POFB) in mV	98	92	92.3	112	238
Margin(PBO) in mV	41	165	53	181	228

are shown in Table V. Also, the relative sizes and margins of transistors after performance-driven optimization (PO) and performance + bias-driven optimization (PBO) are shown in the same table. Note the coefficient shaping factor is chosen as 0 for M1 as it is estimated heuristically that its margin would not affect the performance specification significantly. The targets for the performance specifications are given based on the topology.

The Table VI has the statistics of the performance specifications resulting from Monte Carlo runs (N=500). The σ values of the specifications are seen to be lower for the design optimized using PBO than the one optimized using PO.

B. Improvement in Transient behavior

For evaluating the transient response of the designs optimized using PO and PBO, total-harmonic-distortion (THD) is chosen as

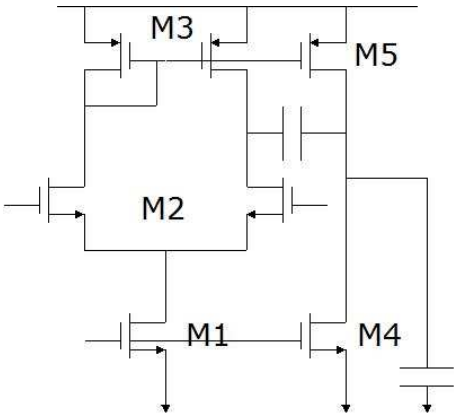


Fig. 13. Opamp3: Two-stage Miller Compensated OTA

TABLE V

OPTIMIZATION PARAMETERS AND RESULTS OF OPAMP3.

Transistor	M1	M2	M3	M4	M5
$Sens_i$	0.226	0.2311	0.0759	0.2967	0.1687
Coeff. Shaping Factor	0	1	1	1	1
w_i	0.0	0.2992	0.0982	0.3841	0.2184
Relative Size(PO)	0.68	4.952	1.704	4.952	1.704
Relative Size(POFB)	9.9	11.6	3.5	11.6	3.5
Relative Size(PBO)	2.49	8.13	7.62	8.13	7.62
Margin(PO) in mV	164	374	370	445	507
Margin(POFB) in mV	279	391	353	518	547
Margin(PBO) in mV	240	439	332	572	481

the performance metric as it captures the nonlinearities of the circuit. The opamps are configured in a unity feedback mode and a sinusoid of frequency 100 MHz and amplitude varying from 10mV to 200mV is given as the input and THD at the output is measured. The THD vs Input amplitude plots are shown in Figures (14) (15) (16).

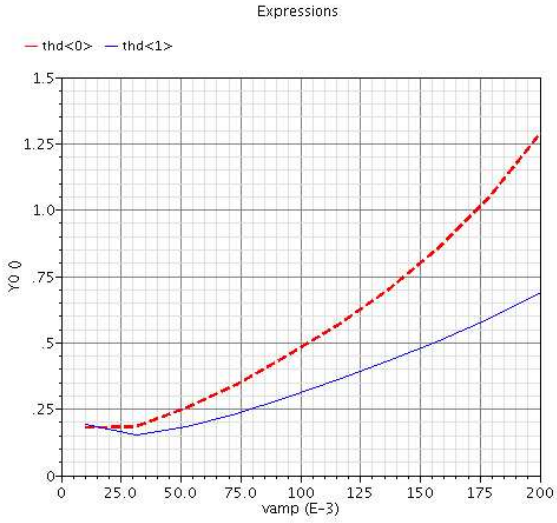


Fig. 14. Total Harmonic Distortion in opamp1 optimized using PO (Dotted line) & PBO (Solid line)

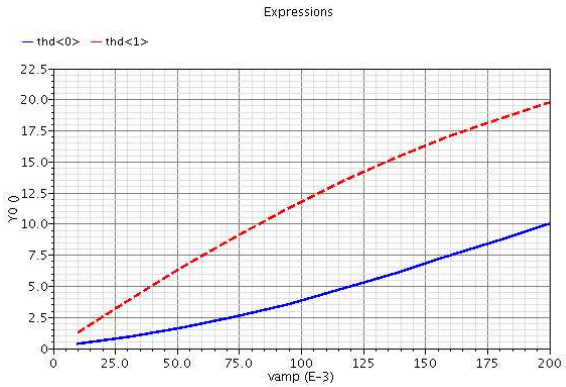


Fig. 15. Total Harmonic Distortion in opamp2 optimized using PO (Dotted line) & PBO (Solid line)

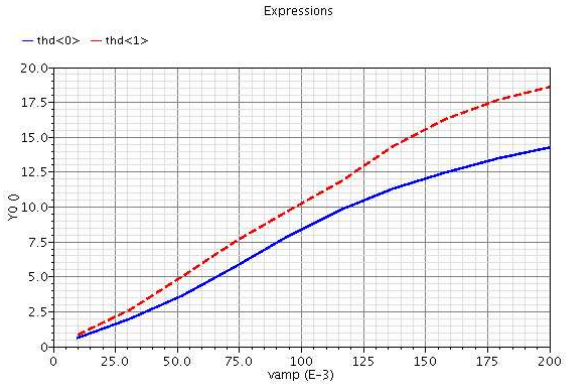


Fig. 16. Total Harmonic Distortion in opamp3 optimized using PO (Dotted line) & PBO (Solid line)

It is seen that the transient response is significantly better in PBO when compared to PO. In the case of opamp2, an improvement of 100% is seen. The nonlinearity in a transistor comes from two sources: transconductance, junction capacitance. For a given voltage swing, the nonlinearity because of the junction capacitance is fixed. The transconductance of the amplifier changes with the bias voltages, although the change is not significant when enough margin is provided to the transistor. This explains why the THD of the opamps sized using PBO is considerably lower than those sized using PO. Note that THD was not provided as a performance specification in the optimization. The improvement in THD comes as a by-product of biasing the circuit better. This could potentially circumvent the need to run transient simulation (which is computationally expensive and could have convergence issues for some designs) in an optimization loop in order to improve the THD.

TABLE VII

THD FOR THE THREE OPAMPS DESIGNED USING PO AND PBO

THD (%) at max Input	Opamp1	Opamp2	Opamp3
PO	1.25	19	18
PBO	0.7	10	14
% Improvement	50	100	10

V. CONCLUSIONS AND FUTURE WORK

A new bias-driven robust analog circuit sizing methodology is proposed in this work. The results bring out a key concept: biasing the circuit well results in a well behaved performance. The benefits are many folded. Reduction in the variation of performance potentially could eliminate the need to optimize the circuit implicitly for yield. Also, improvement in transient response comes as an extra benefit. Both the metrics i.e., variance of small signal specifications and nonlinear transient specifications are seemingly unrelated. But improvement in both because of margin optimization points strongly to the conjecture that biasing is the key for a robust as well as high performance design. As explained in Section III-A, the advantages of this scheme only increase in deep sub-micron technologies.

This scheme could be incorporated in an Analog Design Framework where in:

- The circuit is partitioned into groups (could be designer's input or done automatically by analyzing the topology).
- The performance specifications are split into two groups: (i) Bias dependent specifications (e.g., THD) and (ii) Bias independent specifications (e.g., Slew rate)
- The relative sizing within the group is optimized using PBO with the bias dependent specifications as the goals and constraints.
- The groups are then scaled up or down using constant current density scaling (scale both width and current such that the biasing is left undisturbed) to meet the bias independent specifications.

This methodology would still have some extent of nonlinearity between the groups, but this is still manageable than trying to optimize the entire circuit at the same time.

REFERENCES

[1] G. V. der Plas, G. Debyser, F. Leyn, K. Lampaert, J. Vandenbussche, G. G. E. Gielen, W. M. C. Sansen, P. Veselinovic, and D. Leenaerts, "Amgie-a synthesis environment for cmos analog integrated circuits," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 20, no. 9, pp. 1037–1058, 2001.

[2] G. G. E. Gielen, T. McConaghy, and T. Eeckelaert, "Performance space modeling for hierarchical synthesis of analog integrated circuits," in *DAC*, pp. 881–886, 2005.

[3] W. Nye, D. Riley, A. Sangiovanni-Vincentelli, and A. Tits, "Delight.spice: an optimization-based system for the design of integrated circuits," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 7, pp. 501–519, Apr 1988.

[4] R. Phelps, M. Krasnicki, R. Rutenbar, L. Carley, and J. Hellums, "Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 19, pp. 703–717, Jun 2000.

[5] G. Wolfe and R. Vemuri, "Extraction and use of neural network models in automated synthesis of operational amplifiers," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 22, pp. 198–212, Feb. 2003.

[6] F. De Bernardinis, M. I. Jordan, and A. S. Vincentelli, "Support vector machines for analog circuit performance representation," in *DAC '03: Proceedings of the 40th annual Design Automation Conference*, pp. 964–969, 2003.

[7] H. Liu, A. Singhee, R. Rutenbar, and L. Carley, "Remembrance of circuits past: macromodeling by data mining in large analog design spaces," in *Design Automation Conference, 2002. Proceedings. 39th*, pp. 437–442, 2002.

[8] M. Degrauwe, O. Nys, E. Dijkstra, J. Rijmenants, S. Bitz, B. Goffart, E. Vittoz, S. Cserveny, C. Meixenberger, G. van der Stappen, and H. Oguey, "Idac: an interactive design tool for analog cmos circuits," *Solid-State Circuits, IEEE Journal of*, vol. 22, pp. 1106–1116, Dec 1987.

[9] R. Harjani, R. A. Rutenbar, and L. R. Carley, "Oasys: a framework for analog circuit synthesis," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 8, no. 12, pp. 1247–1266, 1989.

[10] F. El-Turky and E. Perry, "Blades: an artificial intelligence approach to analog circuit design," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 8, pp. 680–692, Jun 1989.

[11] G. Gielen, H. Walscharts, and W. Sansen, "Isaac: a symbolic simulator for analog integrated circuits," *Solid-State Circuits, IEEE Journal of*, vol. 24, pp. 1587–1597, Dec 1989.

[12] G. Gielen, P. Wambacq, and W. Sansen, "Symbolic analysis methods and applications for analog circuits: a tutorial overview," *Proceedings of the IEEE*, vol. 82, pp. 287–304, Feb 1994.

[13] M. Hershenson, S. Boyd, and T. Lee, "Optimal design of a cmos op-amp via geometric programming," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 20, pp. 1–21, Jan 2001.

[14] P. Mandal and V. Visvanathan, "Cmos op-amp sizing using a geometric programming formulation," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 20, pp. 22–38, Jan 2001.

[15] T. McConaghy, T. Eeckelaert, and G. Gielen, "Caffeine: Template-free symbolic model generation of analog circuits via canonical form functions and genetic programming," *Design, Automation and Test in Europe Conference and Exhibition*, vol. 2, pp. 1082–1087, 2005.

[16] W. Daems, G. G. E. Gielen, and W. M. C. Sansen, "Simulation-based generation of posynomial performance models for the sizing of analog

integrated circuits," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 22, no. 5, pp. 517–534, 2003.

[17] R. Vemuri and G. Wolfe, "Adaptive sampling and modeling of analog circuit performance parameters with pseudo-cubic splines," in *Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on*, pp. 931–938, Nov. 2004.

[18] A. Torralba, J. Chavez, and L. Franquelo, "Fasy: a fuzzy-logic based tool for analog synthesis," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 15, pp. 705–715, Jul 1996.

[19] H. Koh, C. Sequin, and P. Gray, "Opasyn: a compiler for cmos operational amplifiers," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 9, pp. 113–125, Feb 1990.

[20] T. McConaghy, P. Palmers, G. G. E. Gielen, and M. Steyaert, "Automated extraction of expert knowledge in analog topology selection and sizing," in *ICCAD*, pp. 392–395, 2008.

[21] P. C. Maulik, L. R. Carley, and R. A. Rutenbar, "A mixed-integer nonlinear programming approach to analog circuit synthesis," in *DAC '92: Proceedings of the 29th ACM/IEEE Design Automation Conference*, pp. 698–703, 1992.

[22] E. Ochotta, R. Rutenbar, and L. Carley, "Astrx/oblx: Tools for rapid synthesis of high-performance analog circuits," in *Design Automation, 1994. 31st Conference on*, pp. 24–30, June 1994.

[23] W. Kruiskamp and D. Leenaerts, "Darwin: Cmos opamp synthesis by means of a genetic algorithm," in *DAC '95: Proceedings of the 32nd annual ACM/IEEE Design Automation Conference*, pp. 433–438, 1995.

[24] Z.-Q. Ning, T. Mouthaan, and H. Wallinga, "Seas: a simulated evolution approach for analog circuit synthesis," in *Custom Integrated Circuits Conference, 1991., Proceedings of the IEEE 1991*, pp. 5.2–1–4, May 1991.

[25] D. Goldberg, *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.

[26] B. Liu, F. V. Fernández, G. Gielen, R. Castro-López, and E. Roca, "A memetic approach to the automatic design of high-performance analog integrated circuits," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 14, no. 3, pp. 1–24, 2009.

[27] T. Massier, H. Graeb, and U. Schlichtmann, "The sizing rules method for cmos and bipolar analog integrated circuit synthesis," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 27, pp. 2209–2222, Dec. 2008.

TABLE II

STATISTICS OF THE PERFORMANCE SPECIFICATIONS OF OPAMP1.

Perf.	Adc (V/V)			fugb (G Hz)			pm (deg)			ivdd (μA)			margin func.			Area (sq.um)		
Target	20			1			Unconstrained			Unconstrained			2			Unconstrained		
Stat.	PO	POFB	PBO	PO	POFB	PBO	PO	POFB	PBO	PO	POFB	PBO	PO	POFB	PBO	PO	POFB	PBO
μ	13.8	13.8	12.55	1.03	1.047	1.05	80.15	80.13	82.27	90.4	91.61	90.4	0.973	0.978	1.264	58.3	78.3	75.7
σ	0.57	0.567	0.425	0.028	0.035	0.026	0.39	0.395	0.36	1.46	2.52	1.53	NA	NA	NA	NA	NA	NA
min	11.5	11.6	11.25	0.96	0.94	0.97	79	79	81.25	86	84	85	NA	NA	NA	NA	NA	NA
max	15.5	15.2	13.75	1.12	1.14	1.12	81.25	81	83.25	95	100.2	95	NA	NA	NA	NA	NA	NA

TABLE IV

STATISTICS OF THE PERFORMANCE SPECIFICATIONS OF OPAMP2.

Perf.	Adc (V/V)			fugb (G Hz)			pm (deg)			ivdd (μA)			margin func.			Area (sq.um)		
Target	20			1			Unconstrained			Unconstrained			1.5			Unconstrained		
Stat.	PO	POFB	PBO	PO	POFB	PBO	PO	POFB	PBO	PO	POFB	PBO	PO	POFB	PBO	PO	POFB	PBO
μ	113.8	112	104.74	0.956	0.923	0.873	78.94	77.8	80.28	76.1	75.3	73.8	0.932	0.883	1.321	17.4	38.6	17.3
σ	12.48	16.78	6.04	0.027	0.016	0.026	0.41	0.29	0.45	2.8	3.19	3.2	NA	NA	NA	NA	NA	NA
min	60	40	85	0.89	0.84	0.77	78	76.5	79.6	67	65	62.5	NA	NA	NA	NA	NA	NA
max	135	140	120	0.98	0.95	0.92	81	80	81.1	83.5	85	82.5	NA	NA	NA	NA	NA	NA

TABLE VI

STATISTICS OF THE PERFORMANCE SPECIFICATIONS OF OPAMP3.

Perf.	Adc (V/V)			fugb (G Hz)			pm (deg)			ivdd (μA)			margin func.			Area (sq.um)		
Target	20			1			Unconstrained			Unconstrained			1.5			Unconstrained		
Stat.	PO	POFB	PBO	PO	POFB	PBO	PO	POFB	PBO	PO	POFB	PBO	PO	POFB	PBO	PO	POFB	PBO
μ	163.5	127	141	193	170	161	35.9	40	48	189	204	199	1.047	1.394	1.213	59.1	152	151
σ	9.58	8.41	9.07	6.63	5.96	4.82	1.46	1.28	0.9	2.38	5.08	3.44	NA	NA	NA	NA	NA	NA
min	130	100	115	175	145	140	31	36	45	181	190	185	NA	NA	NA	NA	NA	NA
max	190	150	165	215	185	175	39.25	43.5	50	196	220	210	NA	NA	NA	NA	NA	NA