

Performance-Centering Optimization for System-Level Analog Design Exploration

Xin Li¹, Jian Wang¹, Lawrence T. Pileggi¹, Tun-Shih Chen² and Wanju Chiang²

¹Dept. of ECE

Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A.

{xinli, jianw, pileggi}@ece.cmu.edu

²SoC Technology Center, Industrial Technology Research Institute

Rm. 382, Bldg. 11, 195 Sec. 4, Chung Hsing Road
Chutung, Hsinchu, Taiwan 310, R.O.C.

{tschen, wjchiang}@itri.org.tw

Abstract

In this paper we propose a novel analog design optimization methodology to address two key aspects of top-down system-level design: (1) how to optimally compare and select analog system architectures in the early phases of design; and (2) how to hierarchically propagate performance specifications from system level to circuit level to enable independent circuit block design. Importantly, due to the inaccuracy of early-stage system-level models, and the increasing magnitude of process and environmental variations, the system-level exploration must leave sufficient design margin to ensure a successful late-stage implementation. Therefore, instead of minimizing a design objective function, and thereby converging on a constraint boundary, we apply a novel *performance centering* optimization. Our proposed methodology *centers* the analog design in the performance space, and *maximizes* the distance to all constraint boundaries. We demonstrate that this early-stage design margin, which is measured by the volume of the inscribed ellipsoid lying inside the performance constraints, provides an excellent quality measure for comparing different system architectures. The efficacy of our performance centering approach is shown for analog design examples, including a complete clock data recovery system design and implementation.

1. Introduction

The challenges associated with large-scale analog system-level design exploration, which include topology selection and early-stage trade-off analysis, often create the bottleneck for mixed-signal system design. Various design automation approaches have been proposed for analog design space exploration [1]-[4] that are based on extracting the performance trade-off curve (called the Pareto optimal front) for each circuit block (e.g. Op Amp, LNA, etc.). Such trade-off curves represent the optimal (i.e. the best) performance values that each circuit topology can achieve with a given manufacturing process. Combining the performance trade-off curves of all circuit blocks and propagating them to system level, analog designers can quickly analyze the system-level design trade-offs and compare system architectures.

Most of the algorithms for calculating the Pareto optimal front can be classified into two categories: equation-based evaluations [1] and simulation-based evaluations [2]-[4]. The equation-based evaluations utilize analytic performance models, where each circuit-level performance (e.g. gain, bandwidth, etc.) is approximated by a close-form expression of the design variables (e.g. transistor sizes, bias current, etc.). The equation-based evaluations are extremely fast, but the accuracy is limited since it is often difficult to accurately capture all analog performance metrics by analytic equations. In contrast, the simulation-based approaches run numerical simulations to construct the circuit performance models; however, many modeling and/or simulation errors can still be introduced, such as those due to device modeling errors, layout parasitics, process variations, etc.

While simulation-based approaches are generally more

accurate at the expense of efficiency, it should be noted that there is a limit to the attainable accuracy at the system-level exploration stage due to the modeling, layout and process uncertainties. Moreover, simplifications and approximations are required at the system level to make the full system analysis and optimization feasible, yet they further increase the *lack of predictability*.

In this paper we propose a new analog system-level design methodology based on a novel *performance centering* optimization, which considers the lack of predictability for the existing performance trade-off models as a basic premise. Unlike most other analog circuit optimization formulations that minimize a cost function by pushing many performance constraints to their boundaries, we borrow the idea from traditional design centering for yield optimization [5]-[7] to derive a performance centering optimization approach for analog system-level design exploration. Our proposed methodology attempts to center the design within the performance space and maximize the inscribed ellipsoid lying inside the performance constraints. Importantly, our approach is *simultaneously* maximizing the design margin for all performance constraints, such that the resulting ellipsoidal volume represents a *quality* measure for the system-level architecture. For example, since the models are known to be imprecise, a small ellipsoidal volume indicates that the performance specifications will probably not be achievable at the circuit level or in the silicon implementation. We therefore propose how this ellipsoidal volume can be used as a quality metric for the assembly and exploration of the interconnected system-level components.

The essence of our performance centering approach is analogous to leaving sufficient design margin for all performance metrics. Such over-design strategies are *manually* applied by analog designers routinely. The novelty of our performance centering method, however, is to provide a *systematic* and *optimal* way to address the problem of preserving design margin for top-down design. The question we address in this paper is the how to mathematically formulate the system-level exploration problem so that we can optimally compare system architectures and hierarchically propagate the performance specifications from system level down to circuit/block level. After the circuit-level specifications are determined, each circuit block can be separately optimized by the circuit-level synthesis tools [11]-[14].

The remainder of the paper is organized as follows. In Section 2, we review the background on existent design space exploration techniques. Then we propose our performance centering approach in Section 3. The efficacy of the proposed performance centering method is demonstrated by several circuit examples in Section 4, followed by our conclusions in Section 5.

2. Background

2.1 Equation-Based Performance Space Models

The equation-based evaluation [1] starts from a set of analytic performance models:

$$p_m(X) \quad (m=1,2,\dots,M) \quad (1)$$

where $X = [x_1, x_2, \dots, x_N]^T$ represents the N design variables and $\{p_m, m = 1, 2, \dots, M\}$ corresponds to the M circuit-level performances.

The closed-form equations in (1) are derived to approximate the relations between all circuit-level performances and design variables. In [1], the author constrains each performance model to be in the form of a posynomial function g to facilitate convex optimization [8]:

$$g(x) = \sum_i c_i x_1^{\alpha_{i1}} x_2^{\alpha_{i2}} \dots x_N^{\alpha_{iN}} \quad (2)$$

where there are N real and positive variables $X = [x_1, x_2, \dots, x_N]^T$ with nonnegative coefficients $c_i \in R_+$ and real exponents $\alpha_{ij} \in R$.

Considering a feasible set S for the design variables X , i.e. $X \in S$, the feasible performance space $P = [p_1, p_2, \dots, p_M]^T$ can be represented in the *implicit* form:

$$\{P | P(X) \quad X \in S\} \quad (3)$$

We refer to (3) as the implicit form because the feasible performance space is implicitly specified in the design variables X . The feasible performance space in (3) can be propagated to system level and optimized with all circuit-level design variables. Such an optimization problem is huge; however, it is computationally feasible if the performance models are in the posynomial form. An optimization with posynomial cost function and constraints can be formulated as a geometric programming problem [1], [8], [9]:

$$\begin{aligned} \text{minimize} \quad & g_0(X) \\ \text{subject to} \quad & g_k(X) \leq 1 \quad (k = 1, 2, \dots, K) \\ & x_n > 0 \quad (n = 1, 2, \dots, N) \end{aligned} \quad (4)$$

where all functions $\{g_i, k = 0, 1, \dots, K\}$ are posynomials. The geometric programming problem can be converted into a convex optimization and solved efficiently. For example, the state-of-the-art geometric programming solver can optimize thousands of variables in a few minutes [9]. The resulting values of all design variables can be then used as a starting point for a more fine-grained post-tuning as required in later design stages.

The feasible performance space can also be represented in the *explicit* form:

$$\{P | F(P) \leq 0\} \quad (5)$$

where $F(P) = [f_1(P), f_2(P), \dots, f_L(P)]^T$ is a nonlinear vector function containing L nonlinear scalar functions. The explicit form in (5) does not include any circuit-level design variables. After equation (5) is propagated to system level, the system-level optimization problem only has the circuit-level performances as the unknown variables. Such a problem is much smaller than that using the implicit representation in (3). However, after the optimization is solved, the circuit-level design variables are still unknown and further steps are required to determine their values.

The primary disadvantage of the equation-based approach is that various simplifications are generally applied and various second-order effects must be ignored when creating the closed-form performance models. It follows that the estimated performance trade-offs might have large errors in some cases.

2.2 Simulation-Based Performance Space Models

To address the accuracy concerns, simulation-based evaluation approaches [2]-[4] run detailed simulations to derive the circuit-level performances, extract the feasible performance space, and represent it in the form of (5) with significantly improved accuracy. However, regardless of the accuracy of the simulator, the lack of predictability at the system level will limit the resulting performance model precision. This lack of predictability can be traced to the system-level abstractions, as well as the inability to abstract modeling information from the

circuit and device levels due to the design uncertainty. Therefore, all of the following combine to increase the lack of predictability:

- *Device level*: e.g. device modeling errors and the uncertainties due to process variations.
- *Circuit level*: e.g. layout parasitics that cannot be accurately captured by running a schematic-level simulation.
- *System level*: e.g. the inaccuracy of the macromodels that are used to approximate the behaviors of circuit blocks. Without the macromodels, a numerical simulator might not be capable of analyzing the entire large-size system.

The key distinction between equation-based evaluation and simulation-based evaluation is the trade-off between accuracy and efficiency. It should be noted that due to the need to abstract the models (macromodels) to the system level to evaluate the interactions among components, there is a limit to the modeling accuracy which can be obtained in practice. For example, it is too expensive, if not impossible, to use a layout-in-the-loop approach to accurately capture the parasitic effect during the system-level optimization.

While many prior works have focused on generating the circuit-level performance trade-off curves, the problem of how to effectively use these trade-offs in system-level design has not been as thoroughly studied. More specifically, how to formulate the system-level optimization problem and make it insensitive to performance modeling errors is still an open question. For example, most traditional constrained optimizations typically minimize one cost function by pushing many performance constraints to their boundaries. As a result, the optimized design can easily fail the specifications even if a small modeling error exists. In this paper, we propose a novel performance centering methodology to improve the robustness of the system-level design exploration.

3. Performance Centering

Our performance centering methodology attempts to address two key problems in analog top-down design: (1) how to optimally compare different system architectures and select the best one for detailed implementation; and (2) how to hierarchically propagate the performance specifications from system level down to circuit level so that each circuit block can be designed separately. We develop two optimization formulations to address these two problems respectively.

3.1 Comparing System Architectures

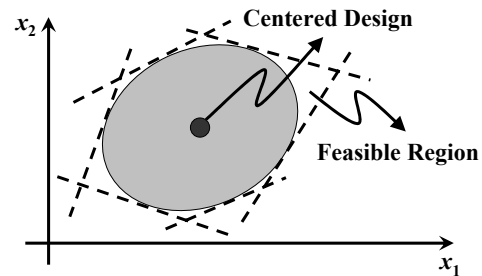


Fig. 1. Illustration of the traditional design centering with two design variables x_1 and x_2 in design space.

For explanation purposes, we define the system architecture as the circuit-level topologies and the interconnections among them.

Our performance centering approach is based on an adaptation of the traditional design centering method (for yield optimization) [5]-[7] to provide a criterion for comparing and contrasting system architectures. The basic idea of design centering is to make the design tolerant to uncertainties by maximizing the inscribed ellipsoid lying inside the feasible region in the design space. This is accomplished by simultaneously maximizing the distance to all constraint boundaries of the feasible design space, as shown in Fig. 1. Following optimization, the resulting ellipsoidal volume is an indicator of the design yield. For example, a larger ellipsoidal volume represents a higher probability that the design can meet the required specifications in silicon, and is less vulnerable to the various uncertainties outlined in Section 2.2.

Therefore, we propose a process of architecture selection whereby we compare different system designs in terms of the probability of achieving a successful implementation in later design stages. We propose to select the architecture based on the performance values and the indicated vulnerability to the various uncertainties.

Most importantly, it is not meaningful to simply compare two system architectures in terms of their maximized ellipsoidal volumes in the design space. The architecture selection problem is substantially different from design centering, since different system architectures consist of different circuit blocks and, therefore, have different design space structures. For example, the dimensions of the design spaces can be different for two architectures since they have different numbers of design variables, and it is not meaningful to compare their maximized ellipsoidal volumes. What is comparable is the performance space. All architecture candidates are toward implementing the same system function with a set of given system-level specifications. This observation motivates us to center the design in the performance space, maximize the distance to all performance boundaries, and use the maximized volume of the inscribed ellipsoid as a criterion to compare system architectures. Such a performance centering idea is illustrated by the example in Fig. 2.

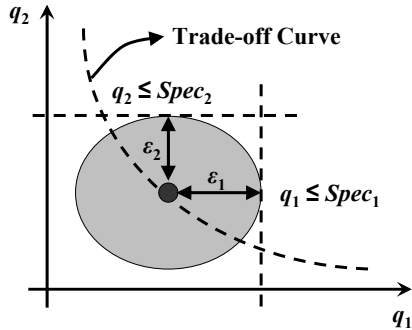


Fig. 2. Illustration of the performance centering with two performance specifications q_1 and q_2 in performance space.

We mathematically formulate and solve the performance centering problem as follows. We formulate two optimization problems that use the implicit performance space models in (3) and the explicit performance space models in (5), respectively.

A. Using Implicit Performance Space Models

For simplicity, we focus on the system design with three hierarchical levels, as shown in Fig. 3. We should note, however, that nothing precludes us from extending our methodology to more complex, multi-level system structures.

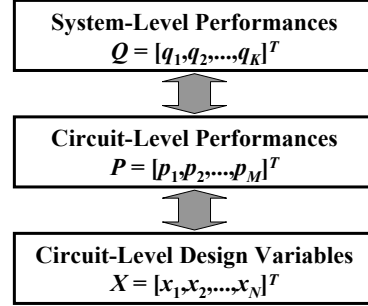


Fig. 3. System design with three hierarchical levels.

We assume that the relations between the system-level performances $\{q_k, k = 1, 2, \dots, K\}$ and the circuit-level performances $\{p_m, m = 1, 2, \dots, M\}$ are available. These relations $\{q_k(P), k = 1, 2, \dots, K\}$ can be derived by hand analysis [1] or approximated by regression modeling [4]. Combining $\{q_k(P), k = 1, 2, \dots, K\}$ and the circuit-level performance models $\{p_m(X), m = 1, 2, \dots, M\}$ yields the system-level performance equations $\{q_k(X), k = 1, 2, \dots, K\}$. In addition, we assume that these system-level performances can be approximated as posynomials. This posynomial assumption might not be true for all analog designs; however, it is valid for many analog circuits. It has been demonstrated that many analog circuit specifications can be cast into posynomial functions [1], [8], [9]. The posynomial property guarantees that the system-level optimization problem is convex and can be solved efficiently. Otherwise, the explicit performance space models in (5) must be used to eliminate all circuit-level design variables and render a small-size, affordable system-level optimization problem.

Without loss of generality, we normalize all system-level performance specifications to the standard form:

$$q_k(X) \leq 1 \quad (k = 1, 2, \dots, K) \quad (6)$$

where all $\{q_k(X), k = 1, 2, \dots, K\}$ are posynomials and K is the total number of the system-level performance metrics. The standard form in (6) has been utilized by the authors in [1], [8], [9] to formulate the analog sizing as a geometric programming problem. A performance specification $f(X) \geq 1$ can be written as $1/f(X) \leq 1$ in order to fit into the standard form in (6) [8].

Based on (6), the performance centering problem can be formulated as:

$$\begin{aligned} & \text{maximize} && \varepsilon_1 \cdot \varepsilon_2 \cdots \varepsilon_K \\ & \text{subject to} && \varepsilon_k = 1 - q_k(X) \quad (k = 1, 2, \dots, K) \\ & && \varepsilon_k > 0 \quad (k = 1, 2, \dots, K) \\ & && x_n > 0 \quad (n = 1, 2, \dots, N) \end{aligned} \quad (7)$$

where $\{\varepsilon_k, k = 1, 2, \dots, K\}$ is a set of variables that represent the lengths of the ellipsoidal axes (see Fig. 2). The optimization in (7) solves for the optimal values of $\{\varepsilon_k, k = 1, 2, \dots, K\}$ and $\{x_n, n = 1, 2, \dots, N\}$ such that the ellipsoidal volume $\varepsilon_1 \varepsilon_2 \cdots \varepsilon_K$ is maximized. Such an optimization can also be generalized to use other cost functions to measure the ellipsoidal size, e.g. $\varepsilon_1^2 + \varepsilon_2^2 + \dots + \varepsilon_K^2$.

The optimization problem in (7) does not match the geometric programming form in (4), since it maximizes a posynomial cost function and includes the posynomial equality constraints. However, equation (7) can be equivalently converted to:

$$\begin{aligned} & \text{minimize} && \varepsilon_1^{-1} \cdot \varepsilon_2^{-1} \cdots \varepsilon_K^{-1} \\ & \text{subject to} && q_k(X) + \varepsilon_k \leq 1 \quad (k = 1, 2, \dots, K) \\ & && \varepsilon_k > 0 \quad (k = 1, 2, \dots, K) \\ & && x_n > 0 \quad (n = 1, 2, \dots, N) \end{aligned} \quad (8)$$

Comparing (7) and (8), we note that maximizing the cost

function $\varepsilon_1 \cdot \varepsilon_2 \cdot \dots \cdot \varepsilon_K$ in (7) is equivalent to minimizing the cost function $\varepsilon_1^{-1} \cdot \varepsilon_2^{-1} \cdot \dots \cdot \varepsilon_K^{-1}$ in (8). In addition, maximizing $\varepsilon_1 \cdot \varepsilon_2 \cdot \dots \cdot \varepsilon_K$ will push all $\{\varepsilon_k, k = 1, 2, \dots, K\}$ to their maximal values. It follows that the inequality constraints $\{q_k(X) + \varepsilon_k \leq 1, k = 1, 2, \dots, K\}$ in (8) always become active, i.e. reach $\{q_k(X) + \varepsilon_k = 1, k = 1, 2, \dots, K\}$, after the optimization. According to these observations, we have the following theorem:

Theorem 1: The optimization problem in (8) is equivalent to the original problem in (7).

Theorem 1 can be formally proved by using the Karush-Kuhn-Tucker optimality condition [10]. Due to the space limitation, the detailed proof is not included here.

The optimization in (8) is a geometric programming problem and, therefore, can be solved efficiently. In addition, the optimization formulation in (8) has two interesting properties.

- *Scaling-independent:* both the optimized design variables $\{x_n, n = 1, 2, \dots, N\}$ and the architecture selection result are independent on the performance scaling. For example, instead of normalizing the performance specifications as (6), we can scale the specifications by any pre-defined factors $\{\beta_k > 0, k = 1, 2, \dots, K\}$, e.g. $\{\beta_k q_k(X) \leq \beta_k, k = 1, 2, \dots, K\}$, and change the optimization problem (8) to:

$$\begin{aligned} & \text{minimize} && \varepsilon_1^{-1} \cdot \varepsilon_2^{-1} \cdot \dots \cdot \varepsilon_K^{-1} \\ & \text{subject to} && \beta_k \cdot q_k(X) + \varepsilon_k \leq \beta_k \quad (k = 1, 2, \dots, K) \\ & && \varepsilon_k > 0 \quad (k = 1, 2, \dots, K) \\ & && x_n > 0 \quad (n = 1, 2, \dots, N) \end{aligned} \quad (9)$$

The optimization of (9) yields the same optimal values of the design variables $\{x_n, n = 1, 2, \dots, N\}$ as that in (8). The maximal ellipsoidal volumes (i.e. $\varepsilon_1 \cdot \varepsilon_2 \cdot \dots \cdot \varepsilon_K$) computed from (8) and (9) differ by a scaling factor $\beta_1 \cdot \beta_2 \cdot \dots \cdot \beta_K$. However, these ellipsoidal volumes are only used for comparing system architectures. As long as the same normalization scheme is used for all architecture candidates, the resulting ellipsoidal volumes are scaled identically and their *relative* relation is unchanged. It follows that the architecture selection result is also independent on the performance scaling. The scaling independent property allows us to use the normalized specifications in (6) to formulate the performance centering problem.

- *Knowledge-compatible:* analog designers can easily add their design knowledge into the optimization formulation (8). For example, if a designer wants to leave the margin ε_i for the i -th performance as two times the margin ε_j for the j -th performance, he/she can explicitly add the constraint $\varepsilon_i = 2\varepsilon_j$ into the optimization. Such an equality constraint can be split into two posynomial inequality constraints $0.5\varepsilon_i \varepsilon_j^{-1} \leq 1$ and $2\varepsilon_i^{-1} \varepsilon_j \leq 1$ so that the optimization is still a geometric programming problem. Geometrically, adding such a constraint will adjust the ellipsoidal shape during the optimization. This is similar to defining the correlation among uncertainty parameters in traditional design centering [7]. However, the uncertainty correlation in performance centering is much more difficult to estimate than that in design centering. Since the knowledge of a given system architecture is limited at the early design stages, the correlation information can only be estimated based on the design experience.

B. Using Explicit Performance Space Models

If the posynomial assumption in the previous sub-section fails, the system-level optimization becomes a general nonlinear programming problem that must be solved by comprehensive algorithms, e.g. simulated annealing or genetic programming. In such cases, including all circuit-level design variables $\{x_n, n = 1, 2, \dots, N\}$ into the system-level optimization is infeasible, since it yields a huge problem and is too computationally expensive. The explicit performance space models in (5) should be utilized to eliminate all circuit-level design variables to make the system-level optimization tractable.

We focus on the three-level system structure in Fig. 3 and assume that the relations $\{q_k(P), k = 1, 2, \dots, K\}$ are known. After normalizing all system-level performance specifications to the standard form (6), the performance centering problem can be formulated as:

$$\begin{aligned} & \text{minimize} && \varepsilon_1^{-1} \cdot \varepsilon_2^{-1} \cdot \dots \cdot \varepsilon_K^{-1} \\ & \text{subject to} && q_k(P) + \varepsilon_k \leq 1 \quad (k = 1, 2, \dots, K) \\ & && f_l(P) \leq 0 \quad (l = 1, 2, \dots, L) \\ & && \varepsilon_k > 0 \quad (k = 1, 2, \dots, K) \end{aligned} \quad (10)$$

where $\{f_l(P), l = 1, 2, \dots, L\}$ stands for the L nonlinear scalar functions that specify the feasible performance space (see (5)). The optimization in (10) solves for the optimal values of $\{\varepsilon_k, k = 1, 2, \dots, K\}$ and $\{p_m, m = 1, 2, \dots, M\}$ such that the ellipsoidal volume $\varepsilon_1 \cdot \varepsilon_2 \cdot \dots \cdot \varepsilon_K$ is maximized. Similar to (8), the scaling-independent and knowledge-compatible properties are also valid for the optimization in (10).

In summary, the optimization formulations in (8) and (10) are developed to center the system-level design in the performance space by maximizing the inscribed ellipsoid lying inside the specification boundaries. The resulting ellipsoidal volume is used as a criterion to compare system architectures. A larger ellipsoidal volume implies a higher probability to achieve a successful implementation in later design stages.

The proposed architecture comparison can be directly applied to analog design based on component library, where basic analog blocks are pre-characterized with performance models. In this case, the performance centering approach can be used to run optimizations for all architecture candidates in the library and then automatically pick up the best one for detailed implementation.

It should be noted, however, that the quality of the architecture comparisons significantly depends on the quality (i.e. accuracy) of the performance models. If the performance models are extremely inaccurate and two system architectures are extremely close, the performance centering approach might not be able to distinguish these two architectures because the performance modeling errors are larger than the architecture difference. In such cases, more accurate performance models must be utilized to achieve an accurate architecture comparison.

3.2 Propagating Performance Specifications

After the system architecture is selected, the equally important step is to propagate the performance specifications from system level down to circuit level and, hence, assign the performance specifications for all circuit blocks so that they can be designed separately. The specification propagation is a key step to enable the top-down design flow.

There are two challenging problems that one must consider when propagating performance specifications. Firstly, at the

circuit level, sufficient margin should be reserved for each circuit-level performance metric. Otherwise, if the circuit-level specifications are too tight, they will become infeasible at the detailed modeling and implementation stages.

Secondly, sufficiently large margins for system-level specifications are also required, since the relation between the system-level and circuit-level performances are not exactly known in the early design stages. Without these system-level performance margins, even if all circuit-level performance specifications are met, the system-level specifications will probably fail in the bottom-up verification.

Following these two observations, most traditional constrained optimizations seem ill-equipped to solve the specification propagation problem since they typically minimize one cost function by pushing many performance constraints to their boundaries. Instead, we are proposing to simultaneously maximize the design margins for all system-level and circuit-level performance metrics, as shown in Fig. 4.

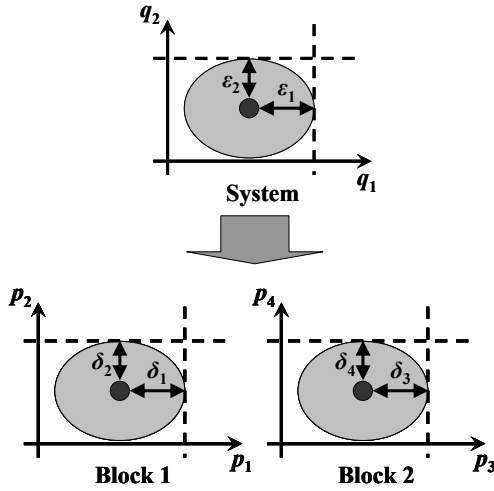


Fig. 4. Illustration of the specification propagation with two system-level specifications (q_1 and q_2) and four circuit-level specifications (p_1, p_2, p_3 and p_4). The performance centering is to maximize the overall design margin $\varepsilon_1 \cdot \varepsilon_2 \cdot \delta_1 \cdot \delta_2 \cdot \delta_3 \cdot \delta_4$.

The following is the mathematic formulation for optimally solving such a specification propagation problem. Two optimization formulations are developed to use the implicit performance space models in (3) and the explicit performance space models in (5), respectively.

A. Using Implicit Performance Space Models

Considering the three-level system structure in Fig. 3, where we represent the circuit-level performance specifications as $\{\tilde{p}_m, m = 1, 2, \dots, M\}$, i.e.:

$$p_m(X) \leq \tilde{p}_m \quad (m = 1, 2, \dots, M) \quad (11)$$

and approximate $\{p_m(X), m = 1, 2, \dots, M\}$ by posynomials. These performance specifications $\{\tilde{p}_m, m = 1, 2, \dots, M\}$ are the problem unknowns that will be determined during specification propagation. Similar to (6), equation (11) is the standard form to define the performance specifications. A specification $f(X) \geq \tilde{f}$ can be written as $1/f(X) \leq 1/\tilde{f}$ in order to fit into the standard form in (11) [8].

In addition, we assume that the system-level performances can be approximated as posynomial functions of the circuit-level specifications, i.e. $\{q_k(\tilde{P}), k = 1, 2, \dots, K\}$, where $\tilde{P} = [\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_M]^T$.

Similar to (6), we normalize the system-level performance specifications as:

$$q_k(\tilde{P}) \leq 1 \quad (k = 1, 2, \dots, K) \quad (12)$$

Combining (11) and (12), the specification propagation problem can be mathematically formulated as:

$$\begin{aligned} & \text{minimize} && \varepsilon_1^{-1} \varepsilon_2^{-1} \dots \varepsilon_K^{-1} \delta_1^{-1} \delta_2^{-1} \dots \delta_M^{-1} \\ & \text{subject to} && q_k(\tilde{P}) + \varepsilon_k \leq 1 \quad (k = 1, 2, \dots, K) \\ & && p_m(X) + \delta_m \leq \tilde{p}_m \quad (m = 1, 2, \dots, M) \\ & && \varepsilon_k > 0 \quad (k = 1, 2, \dots, K) \\ & && \delta_m > 0 \quad (m = 1, 2, \dots, M) \\ & && x_n > 0 \quad (n = 1, 2, \dots, N) \end{aligned} \quad (13)$$

where $\{\varepsilon_k, k = 1, 2, \dots, K\}$ denotes the design margins for the system-level performances and $\{\delta_m, m = 1, 2, \dots, M\}$ stands for the design margins for the circuit-level performances. It is straightforward to verify that the optimization in (13) is a geometric programming problem. It maximizes the ellipsoid defined by both system-level and circuit-level performance specifications. In other words, it simultaneously maximizes the design margins for all system-level and circuit-level performance metrics. After the optimization problem is solved, the circuit-level performance specifications $\{\tilde{p}_m, m = 1, 2, \dots, M\}$ are known and they can be used to design each individual circuit block separately in later design stages.

Compared with (13), the optimization in (8) does not leave any margin for circuit-level specifications. In the application of topology selection, different system architectures consist of different circuit blocks. Therefore, it is not meaningful to compare ellipsoidal volume at the circuit level, and only the system-level ellipsoid is maximized and compared to select the best system architecture. In contrast, for the problem of specification propagation, we need to distribute the design margins from system level down to circuit level. In such cases, the ellipsoid should be maximized at both levels, as shown in (13).

It is worth noting that the scaling-independent and knowledge-compatible properties that are described in Section 3.1.A are also valid for the optimization in (13). For example, based on one's design experience, it is possible to modify the cost function and constraints in (13) to make a different tradeoff between the system-level and circuit-level design margins.

B. Using Explicit Performance Space Models

The optimization formulation in (13) can be easily extended to use the explicit performance space models in (5):

$$\begin{aligned} & \text{minimize} && \varepsilon_1^{-1} \varepsilon_2^{-1} \dots \varepsilon_K^{-1} \delta_1^{-1} \delta_2^{-1} \dots \delta_M^{-1} \\ & \text{subject to} && q_k(\tilde{P}) + \varepsilon_k \leq 1 \quad (k = 1, 2, \dots, K) \\ & && p_m + \delta_m \leq \tilde{p}_m \quad (m = 1, 2, \dots, M) \\ & && f_l(P) \leq 0 \quad (l = 1, 2, \dots, L) \\ & && \varepsilon_k > 0 \quad (k = 1, 2, \dots, K) \\ & && \delta_m > 0 \quad (m = 1, 2, \dots, M) \end{aligned} \quad (14)$$

where $\{f_l(P), l = 1, 2, \dots, L\}$ stands for the L nonlinear scalar functions that specify the feasible performance space (see (5)). If the constraints in (14) are arbitrary functions, the optimization should be solved by comprehensive algorithms, e.g. simulated annealing or genetic programming.

In summary, the optimization formulations in (13) and (14) are developed to maximize the design margins for both system-level and circuit-level performance metrics. As such, the system-

level specifications are distributed to the circuit level so that each circuit block can be optimized individually by the circuit-level synthesis tools [11]-[14].

It must be noted, however, that our performance centering methodology does not guarantee a feasible system-level design – but rather it finds the design that is most capable of achieving the required specifications. Moreover, the quality of the performance centering results significantly depends on the quality (i.e. accuracy) of the performance models. If the performance models are extremely inaccurate and the design constraints are extremely tight, the performance centering approach might fail because the performance modeling errors are larger than the maximal design margins that are available. Therefore, as the design constraints become tighter, we must rely on more accurate performance models. Similar requirements have been noted for traditional design centering for yield enhancement [5]-[7]. When the process variations are too large to achieve a high product yield, better process control is required to reduce the manufacturing uncertainties and improve the yield.

4. Numerical Examples

In this section, we demonstrate the efficacy of the proposed performance centering approach using two design examples: an operational amplifier and a clock and data recovery circuit. All numerical experiments are run on a SUN — 1 GHz server.

4.1 Operational Amplifier

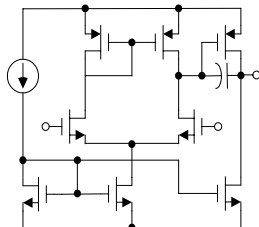


Fig. 5. Circuit schematic of a simple two-stage Op Amp.

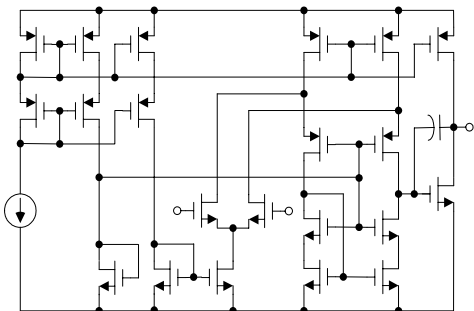


Fig. 6. Circuit schematic of a folded-cascode two-stage Op Amp.

Table 1. Design specifications and results for Op Amp

Performance	Spec	Result
V_{DD} (V)	= 2.5	—
Gain (dB)	≥ 100	102
UGF (MHz)	≥ 10	10.9
Offset (mV)	≤ 1.0	0.01
Phase Margin (degree)	≥ 60	63.2
Slew Rate (V/ μ s)	≥ 20	20.5
Swing (V)	≥ 0.5	1.00
Power (mW)	≤ 20	0.79

Fig. 5 and Fig. 6 show the circuit schematics of a simple two-stage Op Amp and a folded-cascode two-stage Op Amp, respectively. The purpose of this design example is to select a best Op Amp topology from Fig. 5 and Fig. 6 to meet the performance specifications in Table 1. The Op Amp circuits are based on these topologies as implemented in the IBM 0.25 μ m BiCMOS process.

A. Topology Selection

We construct the implicit performance space models using posynomial design equations, and formulate the performance centering problem as (8). The details of the Op Amp design equations can be found in [8]. In this example, since the performance models are approximated as posynomials, the performance centering problem in (8) can be efficiently solved using geometric programming algorithm, taking 1~2 seconds for this Op Amp example.

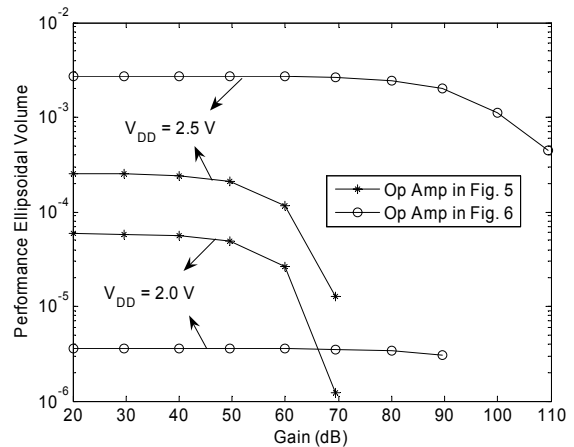


Fig. 7. Maximized ellipsoidal volume in performance space when applying different gain and V_{DD} specifications for Op Amp.

Fig. 7 shows a comparison between the Op Amp topologies in terms of the maximized ellipsoidal volume in the performance space. As we would expect, the two-stage folded-cascode topology is better than the simple two-stage one when the power supply voltage is sufficiently high to provide the necessary voltage headroom. In this example, we find that a sufficient voltage is 2.5 V, whereas, the folded-cascode topology appears to be inferior to the simple two-stage one once the supply voltage is dropped to 2.0 V. Perhaps less obvious, however, we find that for extremely high gain specification, the quality measure (i.e. the ellipsoidal volume) for the simple Op Amp once again falls below that for the folded-cascode Op Amp, even at a 2.0 V supply. This indicates that the folded-cascode configuration would provide a better topology for detailed implementation even at $V_{DD} = 2.0$ V if the gain requirement is high enough. Given the performance specifications in Table 1, the folded-cascode Op Amp topology in Fig. 6 provides larger design margin (i.e. larger ellipsoidal volume) and, therefore, represents our preferred topology for these design specifications in the IBM 0.25 μ m technology.

It is important to consider that as IC technologies continue to scale, many traditional analog circuit topologies will begin to break down owing to reduced power supply voltages and/or device nonidealities such as excessive leakage current. It is essential to understand the limitation of each topology during the system-level design. The proposed performance centering approach offers a systematic way to quickly identify these ineffective topologies within the early design stages.

B. Detailed Sizing

Once the circuit topology is determined (Fig. 6), various circuit-level optimization tools (e.g. [11]-[14]) can be utilized to carefully optimize the device sizes using accurate transistor-level simulation. Table 1 shows the Spectre simulation results following detailed sizing. The final circuit performance in Table 1 meets all of the design requirements.

For testing and comparison, we apply the same detailed sizing to the topology in Fig. 5. As we would expect, the circuit-level optimization cannot produce a feasible design that meets the specifications in Table 1. The main difficulty is that the simple two-stage Op Amp in Fig. 5 cannot achieve an extremely high gain of 100 dB.

4.2 Clock and Data Recovery Circuit

Next, we consider the application of our methodology to the analog system design shown in Fig. 8, which is the block diagram of a clock and data recovery (CDR) circuit for a 2.5 Gbps synchronous optical network communication channel (OC-48) [15]. The CDR system in Fig. 8 is implemented in the TSMC 0.25 μm process and consists of three major circuit blocks: the phase and frequency detector (PFD), the charge pump (CP as shown in Fig. 9) and the voltage controlled oscillator (VCO as shown in Fig. 10). In this example, the PFD is pre-determined, since it is a pure digital circuit and is of less interest to our analog optimization. Our objective is to apply the proposed performance centering approach to systematically design the CP and VCO so that the overall CDR meets the specifications of the OC-48 standard. Such a large-scale design problem is accomplished via three individual steps: system-level design, circuit-level sizing and system-level verification.

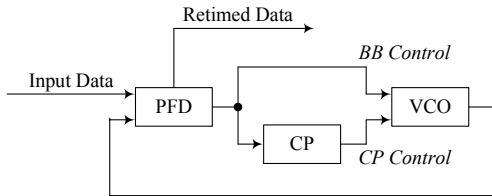


Fig. 8. Block diagram of a clock and data recovery circuit.

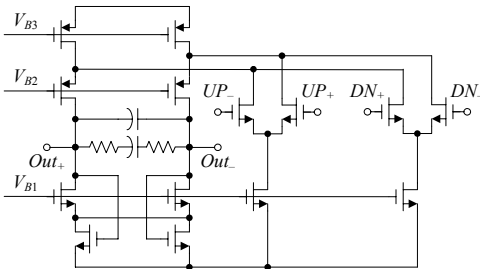
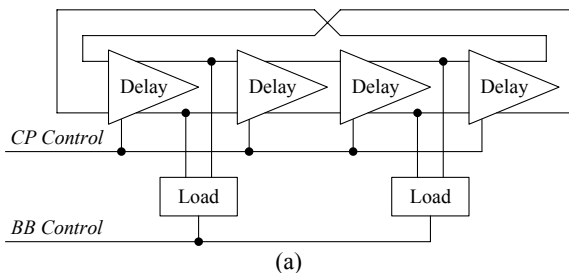


Fig. 9. Circuit schematic of the charge pump.



(a)

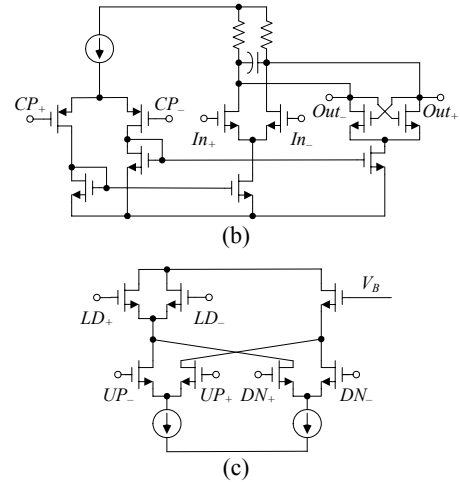


Fig. 10. Circuit schematic of the voltage controlled oscillator. (a) Block diagram of the VCO. (b) Circuit schematic of the delay block. (c) Circuit schematic of the load block.

Table 2. System-level specifications and results for CDR

Performance		Spec	Result
Data Rate (Gbps)		= 2.5	—
Power (mW)		≤ 60	43.98
Jitter Generation	Peak-to-Peak (UI)	≤ 0.1	0.026
	RMS (UI)	≤ 0.01	0.007
Jitter Transfer	Bandwidth (MHz)	≤ 2.0	0.79
	Peaking (dB)	≤ 0.1	0.00

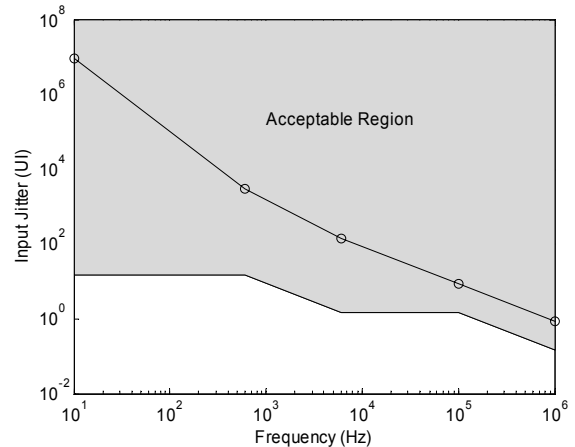


Fig. 11. Jitter tolerance specifications and results for CDR.

A. System-Level Design

The purpose of the system-level design is to systematically propagate the OC-48 specifications down to the circuit level, i.e. determine the circuit-level specifications for the CP and VCO. The system-level CDR specifications consist of several requirements on jitter and power, as shown in Table 2 and Fig. 11.

We construct the CDR performance models using posynomial design equations, resulting in a geometric programming problem for the performance centering formulation in (13). The details of the CDR performance equations can be found in [15]. In this example, the optimization includes 58 variables and 83 constraints. It takes a few seconds to solve such a geometric programming problem. Table 3 shows the optimized circuit-level

specifications for both the CP and the VCO.

Table 3. Circuit-level specifications and results for CDR

Block	Performance	Spec	Result
PFD	Power (mW)	—	36.46
CP	I_{CP}/C_{CP} (A/F)	≤ 16.62	14.45
	Power (mW)	≤ 2.08	0.34
VCO	F0 (GHz)	$= 1.25$	1.22
	CP K_{VCO} (GHz/V)	≤ 3.08	1.29
	BB Tune Range (MHz)	≥ 1.92	2.73
		≤ 3.10	
	PNoise @ 1MHz (dBc)	≤ -76	-84
	Power (mW)	≤ 8.11	7.19

B. Circuit-Level Sizing

After the performance specifications for the CP and VCO are determined, circuit-level synthesis tools [11]-[14] are applied to optimize each circuit block separately. Table 3 shows the circuit-level performances after detailed sizing. Note that the circuit-level sizing is successful for both the CP and the VCO, i.e. all performance specifications in Table 3 are satisfied. This demonstrates that the circuit-level specifications optimized by our performance centering approach are feasible for both circuits.

C. System-Level Verification

As the final step in the design flow, we connect the PFD, CP and VCO together and run the Spectre simulation for the entire CDR. Table 2 and Fig. 11 show the transistor-level Spectre simulation results for the jitter and power performances. In this example, all system-level CDR performances satisfy the OC-48 requirements, as expected from our system-level optimization.

5. Conclusions

A new performance centering methodology has been proposed for robust (i.e. error-tolerant) analog system-level design exploration. By centering the design in the performance space, our proposed performance centering approach provides a quality measure of comparing system-level architectures and a systematic way of propagating the performance specifications from system level down to circuit level. As is demonstrated by numerical examples, the proposed performance centering approach provides successful system-level designs even using simple performance models.

6. Acknowledgements

This work has been supported by the MARCO Focus Center for Circuit & System Solutions (C2S2, www.c2s2.org) under contract 2003-CT-888 and the Industry Technology Research Institute, Taiwan (ITRI, www.itri.org.tw).

7. References

- [1] M. Hershenson, "Efficient description of the design space of analog circuits," *IEEE/ACM DAC*, pp. 970-973, 2003.
- [2] B. Smedt and G. Gielen, "WATSON: design space boundary exploration and model generation for analog and RF IC design," *IEEE Trans. CAD*, vol. 22, no. 2, pp. 213-224, Feb. 2003.
- [3] G. Stehr, H. Graeb and K. Antreich, "Performance trade-off analysis of analog circuits by normal-boundary intersection," *IEEE/ACM DAC*, pp. 958-963, 2003.
- [4] G. Stehr, H. Graeb and K. Antreich, "Analog performance space exploration by Fourier-Motzkin elimination with application to hierarchical sizing," *IEEE/ACM ICCAD*, pp. 847-854, 2004.
- [5] H. Abdel-Malek and A. Hassan, "The ellipsoidal technique for design centering and region approximation," *IEEE Trans. CAD*, vol. 10, no. 8, pp. 1006-1014, Aug. 1991.
- [6] K. Antreich, H. Graeb and C. Wieser, "Circuit analysis and optimization driven by worst-case distances," *IEEE Trans. CAD*, vol. 13, no. 1, pp. 57-71, Jan. 1994.
- [7] A. Seifi, K. Ponnambalam and J. Vlach, "A unified approach to statistical design centering of integrated circuits with correlated parameters," *IEEE Trans. CAS-I*, vol. 46, no. 1, pp. 190-196, Jan. 1999.
- [8] M. Hershenson, S. Boyd and T. Lee, "Optimal design of a CMOS Op-Amp via geometric programming," *IEEE Trans. CAD*, vol. 20, no. 1, pp. 1-21, Jan. 2001.
- [9] M. Hershenson, "Design of pipeline analog-to-digital converters via geometric programming," *IEEE/ACM ICCAD*, pp. 317-324, 2002.
- [10] D. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1999.
- [11] R. Phelps, M. Krasnicki, R. Rutenbar, L. Carley and J. Hellums, "Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search," *IEEE Trans. CAD*, vol. 19, no. 6, pp. 703-717, Jun. 2000.
- [12] G. Plas, G. Debyser, F. Leyn, K. Lampaert, J. Vandebussche, G. Gielen, W. Sansen, P. Veselinovic and D. Leenaerts, "AMGIE — a synthesis environment for CMOS analog integrated circuits," *IEEE Trans. CAD*, vol. 20, no. 9, pp. 1037-1058, Sep. 2001.
- [13] F. Schenkel, M. Pronath, S. Zizala, R. Schwencker, H. Graeb and K. Antreich, "Mismatch analysis and direct yield optimization by spec-wise linearization and feasibility-guided search," *IEEE/ACM DAC*, pp. 858-863, 2001.
- [14] X. Li, P. Gopalakrishnan, Y. Xu and L. Pileggi, "Robust analog/RF circuit design with projection-based posynomial modeling," *IEEE/ACM ICCAD*, pp. 855-862, 2004.
- [15] B. Razavi, *Phase-Locking in High-Performance Systems: From Devices to Architectures*, IEEE Press, 2003.