Hindawi Publishing Corporation
EURASIP Journal on Embedded Systems
Volume 2010, Article ID 261583, 14 pages
doi:10.1155/2010/261583

Research Article

A Platform-Based Methodology for System-Level Mixed-Signal Design

Pierluigi Nuzzo,[1] Xuening Sun,[1] Chang-Ching Wu,[1] Fernando De Bernardinis,[2] and Alberto Sangiovanni-Vincentelli[1]

[1] *Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, 205 Cory Hall, Berkeley, CA 94720, USA*
[2] *Marvell Chip Design Center, Marvell Semiconductors, Viale Repubblica 38, 27100 Pavia, Italy*

Correspondence should be addressed to Pierluigi Nuzzo, nuzzo@eecs.berkeley.edu

Received 5 July 2009; Revised 8 December 2009; Accepted 1 February 2010

Academic Editor: Luca Fanucci

The complexity of today's embedded electronic systems as well as their demanding performance and reliability requirements are such that their design can no longer be tackled with ad hoc techniques while still meeting tight time to-market constraints. In this paper, we present a system level design approach for electronic circuits, utilizing the platform-based design (PBD) paradigm as the natural framework for mixed-domain design formalization. In PBD, a meet-in-the-middle approach allows systematic exploration of the design space through a series of top-down mapping of system constraints onto component feasibility models in a *platform library*, which is based on bottom-up characterizations. In this framework, new designs can be assembled from the precharacterized library components, giving the highest priority to design reuse, correct assembly, and efficient design flow from specifications to implementation. We apply concepts from design centering to enforce robustness to modeling errors as well as process, voltage, and temperature variations, which are currently plaguing embedded system design in deep-submicron technologies. The effectiveness of our methodology is finally shown on the design of a pipeline A/D converter and two receiver front-ends for UMTS and UWB communications.

1. Introduction

Modern electronic systems are becoming increasingly complex and heterogeneous. Telecommunication and multimedia applications require highly integrated, high-performance systems, where analog, RF, and digital components must be efficiently packaged into a single chip. Emerging sensor and actuator swarm applications, as well, demand customized mixed-domain systems to be embedded into a myriad of extreme physical environments to provide a variety of personal or broad-use services. On the other side, manufacturing technology is evolving deeper into the nanometer era, where leakage power, increasing process variations, reducing supply voltage, and worsening signal integrity conditions make it daunting even to assess the required performance specifications. To build future integrated systems, designers need to face several challenges, at all levels of abstraction, from system conception to physical implementation. Design complexity is indeed rising while, at the same time, time-to-market constraints are becoming tighter, and dependable systems need to be built out of increasingly unreliable components. Addressing the above challenges requires innovative solutions not only in manufacturing technologies and circuit architectures, but also in design methodologies and tools.

A disciplined design style that reduces iterations in the flow should be based on a rigorous formalism leveraging accurate and robust *performance modeling* techniques to guarantee that performance variables of each component are correctly propagated across the design hierarchy. Moreover, *fast, global optimization* techniques need to be deployed to provide the best design options, for a given application, within a well-constrained and characterized search space. Finally, a practical framework should promote design reuse, and the *separation of design concerns* to reduce system complexity and boost designers' productivity.

In this paper, we present a system-level design methodology for mixed-signal electronic circuits, which is inspired by the above principles, and leverages the platform-based design paradigm (PBD) [1, 2] as the natural formalization framework. In PBD, a platform is expressed as a collection of components and composition rules. A design is obtained by composing components of the platform in a platform instance. The refinement process consists of mapping a functional description into a set of interconnected components. The design space is systematically explored through a meet-in-the-middle approach in which top-down design constraints of the system are mapped onto bottom-up performance characterizations of the components in the platform library. Based on this paradigm, we provide a unified framework to assist designers at all levels of abstractions. At the system level, a global optimization technique provides the best design options by leveraging tradeoffs among all components, rather than composing systems using locally optimized components. At the component level, designs in different domains (e.g., RF, analog, or digital) can be concurrently characterized to provide an interface that offers smooth system integration while, at the same time, hiding implementation details. This *orthogonalization of concerns* allows making design decisions at the system level, where system tradeoffs can be evaluated across all RF, analog, and digital components. Moreover, the design process can be significantly shortened, because of the hierarchical approach enabled by our methodology, which progressively reduces the number of design variables. To ensure that reliable systems are produced, accurate and robust circuit performance models are crucial in our methodology, since high-level models should directly correspond to feasible physical implementations. Designs should therefore be robust to both modeling errors and process, voltage, and temperature variations (PVT), increasingly important as process parameters (minimum channel length, device threshold, supply voltage, etc.) decrease. As presented in [3], we include into our formulation techniques from design centering, traditionally adopted for digital design. With respect to [3], we add details on our performance models, in comparison with other modeling approaches, as well as on the mathematical derivation of the performance margin evaluation algorithm used in our robust optimization. Moreover, we apply our methodology to an additional example.

This paper is organized as follows. Section 2 gives an overview of the PBD methodology applied to the analog and mixed-signal domains. In Sections 3 and 4, we discuss the robust system-level design problem and provide its mathematical formulation within the PBD paradigm. In Section 5, we illustrate our methodology using three case studies, namely, a pipeline A/D converter and two RF front-ends, for UMTS and UWB receivers. Finally, we draw some conclusions in Section 6.

2. Analog Platform-Based Design

Performing system-level design space exploration and optimization in a systematic way can have a great impact on system performance and cost. In a wireless receiver, for example, it allows distributing design requirements (e.g., gain, NF, linearity) among the chain building blocks, and early evaluation of several tradeoffs, such as preselect filter selectivity and power consumption versus front-end linearity, or base-band filter selectivity versus ADC resolution.

In traditional analog and mixed-signal design flows, experienced architects conduct system-level design, and system specifications are empirically partitioned among the various functional blocks that circuit designers have to implement. In fact, since an effective system-level optimization is not achievable without accurate knowledge of the achievable performance of the several building blocks, final system performance may largely deviate from the expected one, which can result in silicon respins. To simultaneously achieve high-quality system integration starting from accurate circuit characterizations, analog-PBD (APBD) has been proposed and formulated in [4–7] as a meet-in-the-middle recursive process consisting in top-down optimization (platform mapping) and bottom-up characterizations, exporting the feasible design spaces of platform components to higher levels of abstraction.

Optimization is performed on behavioral models, that is, mathematical representations of electronic circuits, capturing their functionality as a function of a set of input, output, and configuration parameters. To allow information hiding and intellectual property (IP) protection, a feasible performance model is also provided for each circuit block, which exports the performance achievable by any available implementation of the block (in the platform library), without propagating implementation details. Performance models are built in a characterization process, as described in Section 2.1. Both models are accompanied by validity laws, that is, a set of constraints and inequalities delineating the validity regions of all component models and their compositions. An Analog Platform (AP) is therefore a library of components, each one decorated by the above set of models and laws. A design is a platform instance, that is, a correct composition of elements, implementing the desired function and, at the same time, optimizing a set of quality metrics.

2.1. Analog Performance Models. Performance models play a critical role in analog system-level design and particularly in platform-based design. Performance models are used to constrain the optimization process to achievable performances within the considered architecture space. Therefore, system-level design approaches have to consider the nature of performance models explicitly during system optimizations.

In the recent few years, a number of papers have appeared on the generation of performance models [8, 9] and even direct modeling of the feasibility region [10, 11]. The latter set of works aims at providing a classifier that separates feasible n-tuples of performances from unfeasible ones, without recurring to a regression-based approach. From the system-level perspective, feasibility models allow casting exploration problems in a more intuitive performance space rather than mapping down to implementation parameters. The number of variables in the optimization problems is

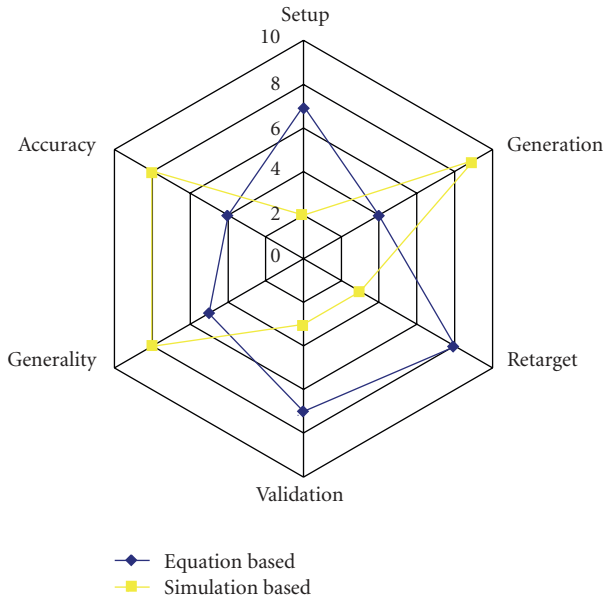

Figure 1: Equation-based and simulation-based approaches for generating performance models compared with multiple metrics. The 0–10 axes encode the difficulty of each figure of merit.

consequently reduced (at least in nondegenerate cases) and architecture selection becomes readily available as different implementation topologies may share common performance spaces.

There are two basic model *generation schemes*, equation based and simulation based (Figure 1). The first approach requires deriving analytical expressions to estimate performance from configuration (regression case) or to model the performance space (classification case). The second approach is based on statistical approximation techniques, where a set of performance samples is evaluated and exploited to build a performance model approximation. In order to compare the different schemes, it is useful to introduce some figures of merit for performance models. The *cost* of generating a performance model can be decomposed into different contributions: model setup, model generation, and model retargeting. In particular, the first two contributions are usually at odds and need to be traded off in real models. Another fundamental figure of merit is *accuracy*. Accuracy is usually assessed through some function (e.g., average or maximum) of the estimation error. We can further distinguish between two different kinds of error, the error on the training data and the generalization error. The last figure of merit we consider is *generality* of the approach, both in terms of classes of circuits and of the performance figures that can be captured. Analytical- and simulation-based models are at opposite ends of the spectrum of performance model schemes.

Analog platform performance models rely on Support Vector Machines (SVMs) as a way of approximating the classifier $\mathcal{P}$ discriminating the feasible performance space. Given a set $\{\mathbf{x}_n\}$ of simulated performance vectors (as detailed in [10]), SVM training selects a subset of vectors $\mathbf{x}_i$

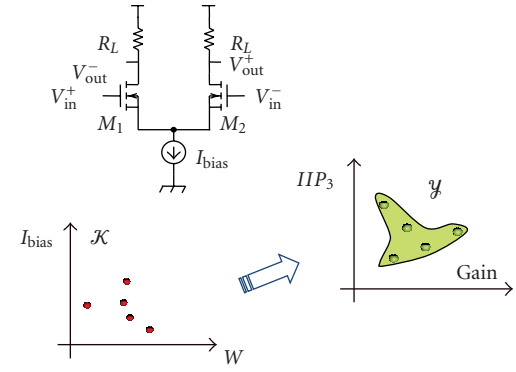

Figure 2: Schematic view of the performance model generation for a simple differential amplifier.

(support vectors) and corresponding weight coefficients α so that the classifier function is obtained as

$$f(x) = \operatorname{sgn}\left(\sum_i \alpha_i e^{-\gamma\|\mathbf{x}-\mathbf{x}_i\|^2} - \rho\right), \tag{1}$$

where ρ is a biasing term (also determined during training) and γ is an SVM parameter. As schematically shown in Figure 2, a set of design configurations (e.g., transistor size for the input differential pair and its bias current) is generated as points in the configuration parameter domain $\mathcal{K}$. Electrical simulation maps these points into vectors in the performance space $\mathcal{Y}$. SVMs are then used to classify the simulated points and generate a feasible performance model. Performance vectors $\mathbf{x}$ are obtained through simulation, so that maximum generality is available in terms of allowable circuits and performance figures. Moreover, SVMs can be generated so as to minimize the impact of false positives, that is, unfeasible performances classified as feasible. In fact, several case studies have shown that the approximation around support vectors is usually restricted in small regions, so that optimal predicted performances are very close to some actually simulated performance vectors. This is an amenable feature to enable effective hierarchical design with minimum risk of incurring in iterations and redesign.

3. Robust System-Level Design

Robust design and optimization have traditionally been closely related subjects. In fact, it is almost impossible to consider an aggressive optimization scheme without considering the robustness of the achieved solutions. System-level design should embrace robust approaches for two separate reasons. From the system level, mixed-signal design has to cope with model inaccuracies that are intrinsic to the behavioral models exploited in design explorations. The more complex the system, the larger the hierarchical structure of the design and the higher the risk when performing nominal design optimizations. In fact, composition of high-level models may provide results whose accuracy is not easily bounded, so either a costly iterative scheme between top-down system-level design and bottom-up verification or relaxed (robust)

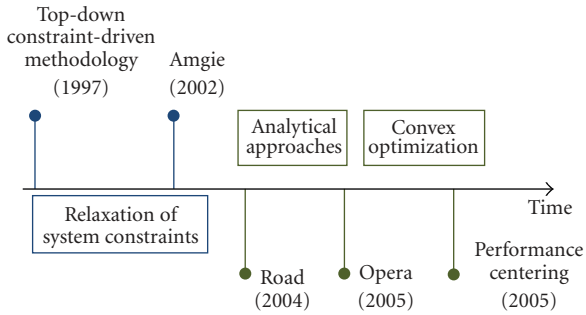

Figure 3: Main approaches to robust system-level analog and mixed-signal design in the last few years.

constraint propagation is adopted. From the implementation level, any performance model is subject to two kinds of inaccuracies: intrinsic modeling errors and process variability. While some control is available on the former source (even if potentially very expensive or restrictive), the latter cannot be solved with deterministic approaches.

Early approaches to computer-aided design centering in an analog context date back to the early 80s [12–14]. All the approaches have a common dependency on the model used to estimate performance degradation on design parameters and, if yield is actually considered, on joint probability functions used to compute yield expectations. However, robust optimization for analog design has not been developed at the same level as nominal optimization. The largest obstacle on the way is represented by the complexity of the resulting optimization problem, which is usually captured as a semiinfinite programming problem. In [15], a circuit optimizer based on simulation is enriched with robust design features, showing significant improvements albeit constrained with scaling issues for complex circuits. The lesson learned from early attempts of including process variations and mismatch in automated circuit design is the tremendous complexity of the resulting problem.

Models generated with classic approaches based on Response Surface Methodology (RSM) [16] can become too expensive to build because of the number of primal parameters and the complexity of the necessary simulations. Instead, we propose an alternative approach, based on approximate models to be developed at the system level.

3.1. Previous Approaches. Several robust approaches to analog design have been proposed during the past few years. Far from being exhaustive, we review those ones that we consider more relevant to the approach presented in this paper (Figure 3). Initially, relaxation of system constraints during top-down optimizations was exploited as an attempt to overcome poor architecture models. We can date back the first rigorous attempt in this direction with the top-down constraint-driven methodology presented in [17] and demonstrated in [18, 19]. Since in pure top-down approaches no detailed information is available on implementation as architectures have not been selected in the first design steps, the methodology formulates the optimization problem (constraint propagation problem) as the maximization of a set of *flexibility functions*. Flexibility functions are introduced to capture the complexity of implementing a specific set of performances. Therefore, in place of optimizing for power or area, the optimization problems maximize the "flexibility" of achieving the optimum set of performances (i.e., minimize the "effort" of implementation). Albeit rigorously formulated, the methodology was rather limited in performing aggressive optimizations because of the halo inherently inserted by the heuristic flexibility functions.

More recently AMGIE [20] proposed to carry out hierarchical design via a set of optimization problems where, at each abstraction level, component performances are bounded to predefined ranges. A robust approach is achieved inserting margins ΔP on all performances, so as to compensate for modeling inaccuracies. However, ΔP has to be determined a priori so that its final value is not the result of an optimization problem. In particular, the cost of meeting the margin on performances is not traded off with the potential improvements in system performances, that is, the sensitivity of the goal function on ΔP is not evaluated at all, leaving a wide discretionality in determining performance margins.

Recent advances in convex optimization [21] have revitalized analytical approaches to analog design and, consequently, robust design. ROAD [22] introduces a robust optimization approach based on posynomial performance models. To improve accuracy, a simulator-in-the-loop approach is selected and local posynomial models are generated around design points. It is then possible to deal with nonconvex design spaces exploiting the possibility of exactly solving large-scale convex programs. OPERA [23] introduces a robust geometric optimization problem to maximize yield over statistical variations. Design process variations are captured with confidence ellipsoids and approximated to yield a convex problem. The robust design formulation computes optimal design parameters to meet a predetermined yield target. Convex optimization approaches, however, tend to limit designers in selecting cost function and formulating their problems. The efficiency achieved in actually solving the problem may be then counterbalanced by the effort required to model the system and validate the analytical expressions used to set the problem. Moreover, classic approaches to system design with convex optimization are based on generating a flat optimization problem, where all circuit topologies have been selected, thus setting a challenging problem as system complexity grows and mixed-signal designs are approached.

Recently, a hierarchical approach to robust system-level analog design has been presented [24]. Performance centering is sought through concurrent maximization of system-level flexibility based on behavioral models and implementation-level performance margins based on performance models. A possible limitation of the approach is the requirement of posynomial models to capture both system-level and implementation-level constraints. While this assumption is certainly acceptable for some classes of analog systems, it may be in practice a hard one to satisfy as it becomes increasingly difficult to guarantee (or even assess)

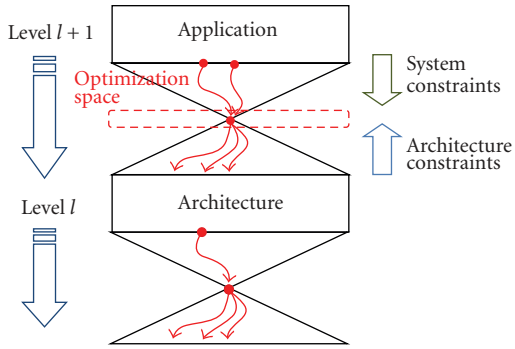

Figure 4: Platform mapping optimization process from level $l+1$ (denoted as *application* in this case) to level l (denoted as *architecture*).

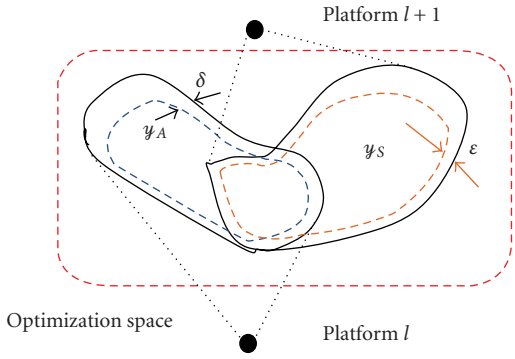

Figure 5: Enlarged view of the optimization space in Figure 4.

model convexity as design hierarchy becomes deeper and high-level behavioral models are exploited in mixed-signal design space explorations.

In our framework, we extend the hierarchical approach by removing the posynomial constraints on design formulation. As in [23, 24], robustness is achieved through maximization of margins with respect to system specifications. Extending the approach to analog platforms, we obtain a two-fold advantage. First, very accurate performance models (not constrained to be convex, posynomial, or even in explicit form) can be exploited to estimate implementation margins. It is then possible to accurately weigh implementation margins since model inaccuracies are kept to minimum levels. Second, arbitrary system behavioral models and constraints can be used to formulate the optimization problem since analog platform-based design relies on global stochastic optimization approaches to find optimal implementations. Designers can then specify their systems without recurring to posynomial approximations and capturing arbitrary nonconvex constraints.

4. Mathematical Formulation

The essence of APBD in its general formulation is pictorially represented in Figure 4 and consists of a bottom-up platform generation phase, where architectural constraints are characterized and exported to higher levels, and a top-down optimization phase, where system constraints are intersected with architectural constraints and the system cost is minimized. At the end of the optimization, system specifications are mapped on the available platform library and the process is repeated.

4.1. Nominal Optimization. In a nominal formulation, the optimization process mapping platform $l+1$ onto platform l is mathematically captured as

$$\min_{\kappa} \quad \text{cost}(\zeta) \qquad \text{s.t.} \quad \begin{cases} \zeta = \mathcal{F}(\kappa), \\ \mathcal{S}(\zeta) \leq 0, \\ \mathcal{P}(\kappa) \leq 0, \end{cases} \tag{2}$$

where ζ is a set of system performance indices, κ is a set of platform configuration parameters, $\mathcal{F}$ is the behavioral model used to map κ into ζ, $\mathcal{S}(\zeta)$ represents the set of constraints imposed on ζ by system specifications, and $\mathcal{P}(\kappa)$ captures the set of constraints on the configuration parameters κ imposed by the architecture space. The set of constraints in (2) can be visualized defining two sets in the optimization space. The system constraints $\mathcal{S}(\zeta) \leq 0$ define the set $\mathcal{Y}_S$ of feasible performances from the system perspective. The architectural constraints $\mathcal{P}(\kappa) \leq 0$ define, through the behavioral model $\mathcal{F}$, the set $\mathcal{Y}_A$ of achievable performances with the current architecture (platform). Figure 5 shows a pictorial representation of the two sets and how mapping is the minimization of the cost function on $\mathcal{Y}_S \cap \mathcal{Y}_A$. Nominal design optimization computes the vector κ that produces the minimum cost in (2). At optimum, the Karush-Kuhn-Tucker conditions require for active constraints that $\mathcal{S}(\zeta) = 0$ and $\mathcal{P}(\kappa) = 0$, which means that the optimized system is, in general, at the "edge" of implementability on several constraints from both a system and an architecture perspective. However, any modeling error in $\mathcal{F}$ may translate in actual performances ζ_{act} (computed with accurate models) to violate $\mathcal{S}$. Similarly, any modeling error in $\mathcal{P}$ may translate in platform l performances being unfeasible. When similar events occur, system design needs either to be iterated or degraded performances have to be accepted. Degradation may be rather severe and force costly redesigns when aggressive specifications are addressed. Even accurate models may fail if performance degradation is due to process parameter dispersion or temperature variation. In general, it is deemed unfeasible to export this information with performance models as for each circuit configuration $\overline{\kappa}$ a function has to be provided $\phi(\zeta; \overline{\kappa})$ which computes the probability density function of performance ζ given the circuit sizing $\overline{\kappa}$. As the approximation of ϕ usually relies on expensive Monte Carlo simulations around $\overline{\kappa}$, the generation of $\phi(\zeta, \kappa)$ over the entire configuration space $\mathcal{K}$ is hardly doable.

4.2. Robust Optimization. To address this problem, an alternate formulation of the optimization problem is required.

The sets of constraints $\mathcal{S}$ and $\mathcal{P}$ have to be satisfied with some margin so as to compensate for modeling inaccuracies. We can write the new set of constraints as $\mathcal{S}(\zeta) \leq \epsilon$ and $\mathcal{P}(\zeta) \leq \delta$. Margins have an intuitive interpretation, defining a sphere (as defined by the norm adopted) $S_\zeta(\zeta^*, \epsilon)$ for system constraints and $S_\kappa(\kappa^*, \delta)$ for performance constraints around the optimal pair $\{\zeta^*, \kappa^*\}$. The objective of the optimization problem is then changed so as to maximize margins δ and ϵ, which corresponds to the maximization of the volumes of the spheres around the optimum configuration and performance points. The original cost function is inserted as an added constraint with a dedicated ϵ_c. Given a minimum cost target $\tilde{c}$, at optimum ϵ_c is maximized constrained on the other margin variables, so that a tradeoff is evaluated between cost value and robustness during the optimization. Therefore, problem (2) becomes

$$\min_{\kappa} \left(\prod_i \delta_i\right)^{-1} \left(\prod_j \epsilon_j\right)^{-1} \quad \text{s.t.} \begin{cases} \zeta = \mathcal{F}(\kappa), \\ \text{cost}(\zeta) \leq \tilde{c} + \epsilon_c, \\ \mathcal{S}(\zeta) \leq \epsilon, \\ \mathcal{P}(\kappa) \leq \delta. \end{cases} \tag{3}$$

System-level constraints are usually available in explicit form; therefore $\mathcal{S}(\zeta) \leq \epsilon$ can be immediately written as

$$s_1(\zeta) - \epsilon_1 \leq 0, \ldots, s_p(\zeta) - \epsilon_p \leq 0, \quad \epsilon_1 > 0, \ldots, \epsilon_p > 0 \tag{4}$$

and included in the optimization problem. Additional constraints may be inserted to set specific relations on ϵ, for example, $\epsilon_1 = 2\epsilon_2$. The problem is more involved with performance models, as analog platforms provide $\mathcal{P}$ in implicit form with a nonlinear function $f(\kappa) \rightarrow \{-1, 1\}$. In this case, we interpret the margin δ in the following way. For a performance model $\mathcal{P}$, its frontier $\partial\mathcal{P}$ defines the boundary of the feasible region. Given a configuration point κ satisfying performance constraints $\mathcal{P}(\kappa) = 1$, its margin δ can be obtained finding the closest configuration $\hat{\kappa} \in \partial\mathcal{P}$ to κ and computing the norm of $\hat{\kappa} - \kappa$. If all components κ_i of κ have the same weight, then $\delta = \|\hat{\kappa} - \kappa\| \cdot \mathbf{1}$ (the performance constraint $\mathcal{P}(\kappa) = 1$ is consistent with the formulation in (2) as it is equivalent to the argument of sgn in (1) being ≤ 0 after a sign change). In this case, minimizing $(\prod_i \delta_i)^{-1}$ is equivalent to maximizing the volume of the sphere around κ that is enclosed in the feasible space (within its boundary $\partial\mathcal{P}$). The general case of different weights on different performance components can be immediately obtained adopting a different norm when computing $\|\hat{\kappa} - \kappa\|$. Since the different performances in the performance vectors used to generate $\mathcal{P}$ can differ in orders of magnitude, they are all preconditioned to be normalized in the interval $[-1, 1]$. In the following paragraph, we show how to compute δ based on the SVM representation of $\mathcal{P}$.

4.3. Performance Margin Evaluation. The problem of finding $\hat{\kappa}$ given κ and $\mathcal{P}$ is analogous to the problem of finding the largest hyperellipsoid enclosed by $\partial\mathcal{P}$. Initially we start solving the case of hypersphere enclosure, extending to the general case at the end of this paragraph. By definition, $\hat{\kappa}$ is the point on the boundary $\partial\mathcal{P}$ which shows minimum distance from κ. To simplify notation, we set $\mathbf{x} = \hat{\kappa}$ and $\mathbf{a} = \kappa$. Therefore, we can obtain $\hat{\kappa}$ solving the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{x}} \quad \|\mathbf{x} - \mathbf{a}\|, \\ &\text{s.t.} \quad \mathbf{x} \in \partial\mathcal{P}, \end{aligned} \tag{5}$$

where $\partial\mathcal{P}$ is implicitly defined from (1) as

$$\sum_i \alpha_i e^{-\gamma\|\mathbf{x} - \mathbf{x}_i\|^2} - \rho = 0. \tag{6}$$

The optimization problem obtained substituting (6) into (5) is evidently nonlinear and can be interpreted as vector *projection* onto a nonconvex set. In fact, while the cost function in (5) is strictly convex, the equality constraint in (6) is nonlinear (and nonconvex). At optimum, the Karush-Kuhn-Tucker conditions require that

$$\begin{aligned} &\sum_{i=1}^{m} \alpha_i e^{-\gamma\|\mathbf{x} - \mathbf{x}_i\|^2} - \rho = 0, \\ &x_1 - a_1 = \lambda \cdot \gamma \cdot \left(\sum_{i=1}^{m} \alpha_i e^{-\gamma\|\mathbf{x} - \mathbf{x}_i\|^2} x_{i,1} - \rho x_1\right), \\ &x_2 - a_2 = \lambda \cdot \gamma \cdot \left(\sum_{i=1}^{m} \alpha_i e^{-\gamma\|\mathbf{x} - \mathbf{x}_i\|^2} x_{i,2} - \rho x_2\right), \\ &\vdots \\ &\vdots \\ &x_n - a_n = \lambda \cdot \gamma \cdot \left(\sum_{i=1}^{m} \alpha_i e^{-\gamma\|\mathbf{x} - \mathbf{x}_i\|^2} x_{i,n} - \rho x_n\right), \end{aligned} \tag{7}$$

where λ is the Lagrange multiplier, the first equation states the feasibility condition for **x,** and the other equations enforce that the gradient of the Lagrangian function L vanishes at any optimal point. System (7) originates from the equivalent problem obtained by (5) after squaring the cost function. For each $j \in \{1, \ldots, n\}$, the jth component of $\nabla L(\mathbf{x}, \lambda)$ can therefore be computed as follows:

$$(\nabla_{\mathbf{x}} L)_j = 2\left(x_j - a_j\right) - 2\gamma\lambda\left(\sum_{i=1}^{m} \alpha_i e^{-\gamma\|\mathbf{x} - \mathbf{x}_i\|^2}\left(x_{i,j} - x_j\right)\right). \tag{8}$$

By substituting (6) into (8), for each j, we finally obtain the equations in (7).

The nonlinear system (7) can be solved with Newton-Raphson (NR) providing quadratic convergence if $\mathbf{x}_0$ is "close" to $\hat{\mathbf{x}}$. $\|\hat{\mathbf{x}} - \mathbf{x}\|$ is therefore the radius of the largest hypersphere enclosed in $\partial\mathcal{P}$. However, the nonlinear nature

of (7) generates two problems. First, a multitude of solutions may exist, so we could achieve convergence on a point on $\partial\mathcal{P}$ which is not the closest to $\mathbf{x}$; second, NR may not converge at all if a sufficiently good initial guess is not provided. To cope with the above problems we first adapt to our problem a more sophisticated implementation of the NR method, similar to the *damped* Newton's method [21], which tries to improve on basic NR poor global convergence. Then we add some ad hoc heuristics to generate a good initial guess.

Solving for λ one of the equations in (7) and substituting the result into the other equations, we obtain an n-dimensional system in the unknown vector $\mathbf{x}$, which can be denoted as $\mathbf{F}(\mathbf{x}) = 0$. We then combine NR method with the minimization of the function $f = (1/2)\|\mathbf{F}\|^2$, in the sense that we accept the solution provided by each NR step only if the step considerably reduces f. If this does not happen, we backtrack along the NR direction $\mathbf{d}$ starting from the old point $\mathbf{x}_{\text{old}}$ until we have an acceptable new point $\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} + \nu\mathbf{d}$ $(0 < \nu \leq 1)$. Since the NR step is a descent direction for f, we are guaranteed to find an acceptable point by backtracking. The backtracking routine is based on the *line minimization* rule [25, 26] and consists in defining $g(\nu) \equiv f(\mathbf{x}_{\text{old}} + \nu\mathbf{d})$, as the restriction of f along $\mathbf{d}$, and finding ν so as to minimize g. To save on the number of function evaluations, a cubic approximation of g is actually computed based on available information on g and its derivative. Since the improved NR method can still occasionally fail converging on a local minimum of f, we can try a new starting point according to the following heuristics:

(i) we compute the distance along reference axes in $\mathbb{R}^n$ using bisection-based monodimensional methods. It is then possible to bound the distance of $\hat{\mathbf{x}}$. We observed that in practical cases whenever this bound is smaller than some $\delta_{\max}$ (whose actual value depends on normalization of $\mathbf{x}$) convergence is always achieved and the correct $\hat{\mathbf{x}}$ is returned by Newton-Raphson,

(ii) we set $\mathbf{x}_0 = \mathbf{a}$ to start iterations as we expect $\mathcal{P}$ to define a relatively "thin" feasible space. Whenever the previous heuristics is not satisfied, we run N NR iterations perturbing the initial point $\mathbf{x}_0$ in the direction of the axis where the minimum distance has been found in the previous point (iterations are aborted after a predetermined number) until the minimum distance solution is reached. We observed that $N = 5$ is generally sufficient to achieve convergence,

(iii) in case of nonconvergence, we return the bound computed in the first point. In practice, there is no consequence in doing this because it always happened for points deep in $\partial\mathcal{P}$ in our tests.

The above procedure can be extended to hyperellipsoids enclosure by scaling $\mathbf{x}$ with a unitary matrix $\mathbf{E}$ to obtain $\mathbf{x}' = \mathbf{E}\mathbf{x}$ and extending the previous approach on $\mathbf{x}'$. Margins found in this way need to be scaled back to the initial space through $\mathbf{E}^{-1}$. This allows selecting different margins on different performances.

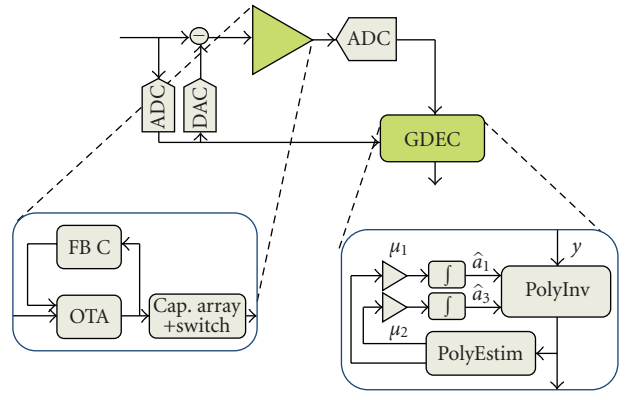

Figure 6: Pipelined converter simplified block diagram—feasible performance models have been generated for the blocks in green: the SHA and the gain error digital calibration block (GDEC).

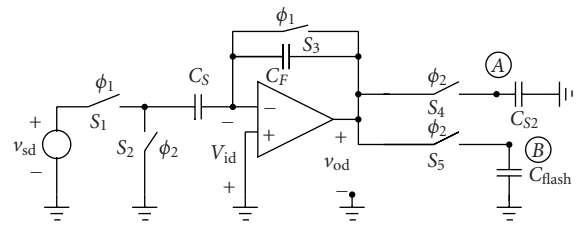

Figure 7: Single-ended equivalent (simplified) circuit of the switched capacitor SHA.

The overall algorithm complexity has been computed to be $O(n^2m + nmc_{\text{exp}})$ where n is the number of performance figures in $\mathcal{P}$, m is the number of performance vectors, and c_{exp} is the cost for evaluating the exponential function as in (1).

5. Examples

In this section we apply the previous results to the case studies reported in [4, 27, 28]. The original designs are reformulated according to (3). The selection of good cost functions is a crucial issue in system-level optimization, with implications that may become subtle when maximizing robustness. In our experiments, we used the following cost prototype:

$$\frac{1}{\left(\prod_{i=1}^{k}\left(\alpha_i + \tanh\left(\beta_i\delta_i\right)\right)^{\mu_i}\right)^{1/k} \cdot \left(\sum_{j=1}^{r}\epsilon_j^{\theta_j}\right)^{1/r}}. \tag{9}$$

A few considerations may help explain the form of (9). First, the volumes of the δ ellipsoid and the ϵ hypercube increase with number of dimensions for constant margin; therefore an overall normalization is achieved with the powers $1/k$ and $1/r$ of σ and ϵ products. As far as architecture margins are concerned, we can partition $\delta = \{\delta_1\delta_2\ldots\}$, where δ_i refers to the single platform component. Elements $\delta_{i,j}$ of δ_i are strongly related describing an ellipsoid embedded in $\mathcal{P}_i$. Therefore a single element is sufficient to describe

Table 1: Performance of optimal ADC, OTA, and GDEC circuit for 3 different cost functions. M denotes the system and architecture margins.

Performance	(1)	M_1	(2)	M_2	(3)	M_3
DNL (LSB)	0.07	0.73	0.04	0.76	0.07	0.73
INL (LSB)	0.43	0.57	0.04	0.96	0.45	0.55
SNR (dB)	85.1	9.1	85.1	9.1	82.6	6.6
P_{ADC} (mW)	57.1	42.9	59.6	40.4	42.6	57.4
P_{OTA} (mW)	52.8	—	55.3	—	37.8	—
A_v	194	134	267	88.7	228	2.72
BW (KHz)	4269	1119	3768	739	2755	22.7
G	7.46	—	7.65	—	7.24	—
P_{GDEC} (mW)	4.2	—	4.2	—	4.8	—

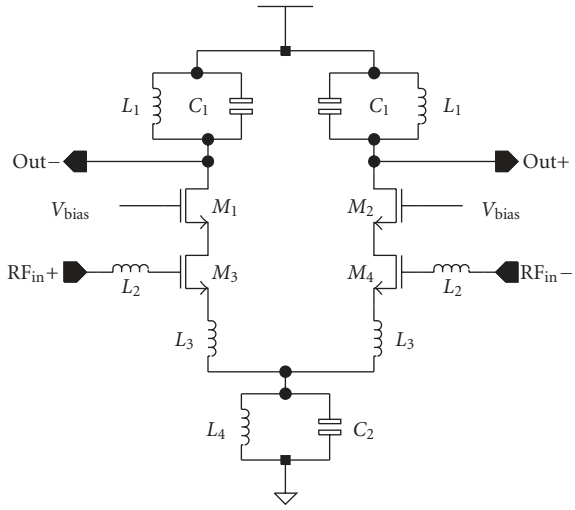

Figure 8: Schematic for the n-type input stage LNA used in the UMTS receiver front-end.

the margin of the ith component. If we consider that the composition of blocks is as robust as the weakest block, we can obtain a different cost function considering $\min_i(\alpha_i + \tanh(\beta_i\delta_{i,1}))$. The tanh function is used to saturate the sensitivity on δ as margins too wide may cause degenerate robustness/performance tradeoffs. Finally, if we analyze the Pareto optimal curves as a function of ϵ and δ, we can easily obtain that the relative importance of two ϵ parameters is controlled by

$$\theta_a \frac{\Delta\epsilon_a}{\epsilon_a} + \theta_b \frac{\Delta\epsilon_b}{\epsilon_b} = 0 \tag{10}$$

so that θ_a/θ_b sets the relative impact of variations of ϵ_a and ϵ_b. When δ and ϵ are considered, we obtain (for small $\beta\delta$)

$$\frac{\theta_a}{r}\frac{\Delta\epsilon_a}{\epsilon_a} + \frac{\mu_b}{k}\frac{\beta_b\delta_b}{\alpha_b + \beta_b\delta_b}\frac{\Delta\delta_b}{\delta_b} = 0, \tag{11}$$

which makes it clear how the parameter α can be used to control sensitivity on δ without recurring to exponent ranges that may generate numerical issues during optimization. Equations (10) and (11) can be used as guidelines to set parameters in (9), as exemplified in the following case studies.

As a final remark, we notice that architecture performance margins are taken on lower-dimensional models than the corresponding platform ones. In fact, some parameters are simply "ancillary" parameters required for correct composition of platform models, and as such not related with the robustness of the solution. One other parameter, which we did not include when computing margins at the component (architectural) level, is power. Power may be considered as an annotation on circuit performances. In fact, in our case studies, if a given circuit exhibits a larger (or smaller) power consumption with respect to the estimated one, it does not affect circuit performances (which is obviously not true if gain is not met, for example). We remark that this is an arbitrary design choice and is not related to the presented methodology. On the other hand, in our examples we introduce margins on power at the system-level to trade the *global* power consumption with the robustness of the solution. Also, area has not been exploited as a robustness criterion, but this can be seamlessly introduced in the robust optimization scheme to export at the system level area penalties involved in topology selection.

5.1. Pipeline ADC. In [4] we performed design space exploration of a 14-bit, 80 MS/s pipeline analog-to-digital converter (ADC) in 0.13 μm, 2.5 V analog supply CMOS technology. The simplified block diagram of the system is represented in Figure 6. The ADC is made up of 4 multibit stages and includes digital calibration circuits to enhance performance. In particular, the digital-to-analog subconverter (DAC) errors are canceled with the DAC Noise Cancelation (DNC) technique [29] and the first-stage Sample-and-Hold Amplifier (SHA) errors are corrected through a Gain and Distortion Error Correction (GDEC) algorithm as in [30]. The SHA gain and third-order distortion coefficients, a_1 and a_3, are first estimated from the digital back-end by a *PolyEstim* circuit. At the same time the distorted SHA characteristic is effectively inverted (rectified) by the *PolyInv* circuit. As shown in Figure 7, the interstage residue amplifiers are fully differential

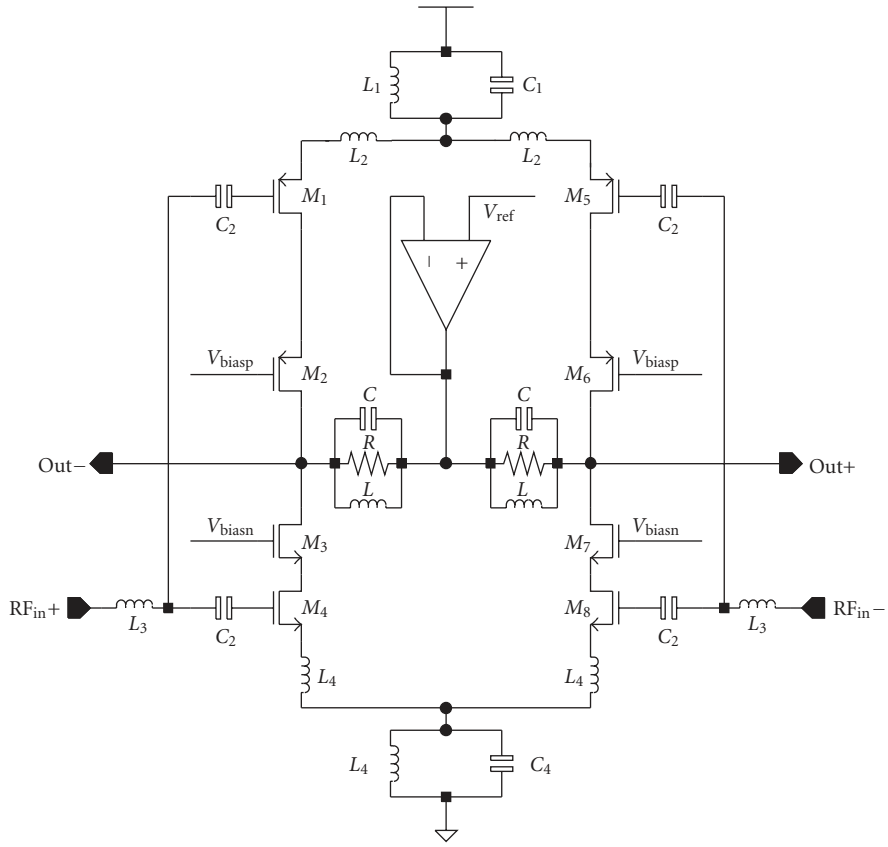

Figure 9: Schematic for the np-type input stage LNA used in the UMTS receiver front-end.

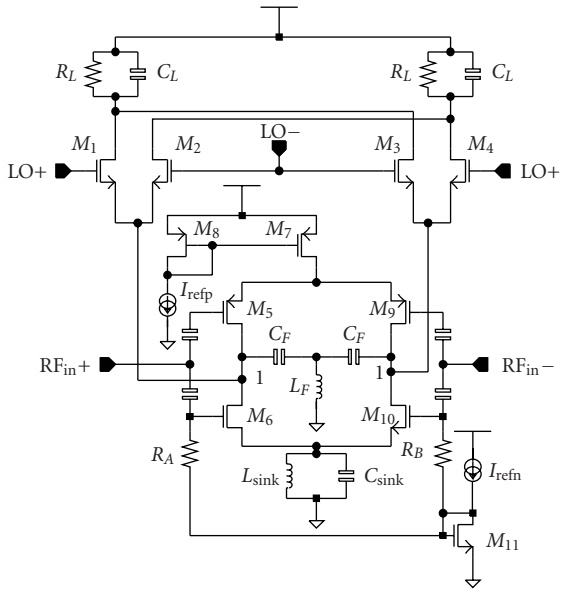

Figure 10: Schematic for the mixer used in the UMTS receiver front-end.

switched capacitor systems (FB C in Figure 6) implemented with a telescopic Operational Transconductance Amplifier (OTA). Loading effects and switches nonidealities are also included in the model. The OTA optimization needs to be performed under the hypothesis of operation of digital calibration circuits, as detailed in [5]. In order to perform efficient high-level exploration across the analog/digital boundary while reducing the complexity of the problem we provided characterizations and feasible performance models for the main blocks, that is, the digital calibration logic and the first-stage residue amplifier. The remaining part of the converter was considered ideal. Indeed, the first stage in a multibit pipeline ADC is the most critical block since the accuracy required in terms of gain and linearity is maximum; the remaining stages have been lumped into one block in our macromodel. Since the first stage provides the first 4 bits, a nominal gain G of 8 is required to the SHA. However, the presence of the digital correction circuit relaxes this constraint enabling power savings. In the nominal optimization, the cost function aims at minimizing power consumption P_{ADC} of the overall ADC subject to performance models and minimum system requirements on DNL, INL, and the signal-to-noise ratio (SNR) due to thermal noise. The architectural space includes four correction algorithms to invert the polynomial nonlinearity corresponding to different accuracy and power consumption levels, based on [5]. Performances are evaluated through the behavioral model $\mathcal{F}$ of the mixed-signal platform library, in which each component is embedded.

The extension to the robust approach of the optimization problem has been achieved through the following formulation, based on the cost template in (9):

$$\min_{\kappa} \quad \left(\prod_{j=1}^{4} \epsilon_j^{\theta_j}\right)^{-1/4} (\alpha+\delta)^{-\mu},$$

$$\text{s.t.} \begin{cases} \zeta = \mathcal{F}(\kappa), \\ P_{\text{ADC}} \le 100 \cdot (1+\epsilon_1)^{-1} \text{ mW}, \\ \text{DNL} \le 0.8 \cdot (1+\epsilon_2)^{-1} \text{ LSB}, \\ \text{INL} \le 1 \cdot (1+\epsilon_3)^{-1} \text{ LSB}, \\ \text{SNR} \ge 76 \cdot (1+\epsilon_4) \text{ dB}, \\ \mathcal{P}_{\text{SHA}}(\kappa_{\text{SHA}}) = 1, \quad \delta = \text{margin}(\mathcal{P}_{\text{SHA}}, \kappa_{\text{SHA}}), \\ \mathcal{P}_{\text{GDEC}}(\kappa_{\text{GDEC}}) = 1, \end{cases} \tag{12}$$

where ϵ_1, ϵ_2, ϵ_3, and ϵ_4 are system margins on power, DNL, INL, and SNR, respectively. δ, the architecture margin, is normalized in $[0, 1]$ and is computed by exploiting an ellipsoid in which weight for the OTA bandwidth (BW) and open-loop gain (A_v) is 2 times the other performance indices. The parameter α controls cost function sensitivity on δ, hence the architecture margin on the optimum.

Several optimizations with different cost parameter values were efficiently performed through simulated annealing, with an average time of 13 hours per run. Three meaningful results are reported in Table 1 to demonstrate how the tradeoffs between system margins (especially ϵ_1 on power) and architecture margins (especially on gain and bandwidth) can be thoroughly explored within our methodology. In (1) more emphasis has been given to the architectural constraint margins, setting $\theta_1 = 4$, $\theta_2 = \theta_3 = \theta_4 = 4/3$, $\alpha = 0$, and $\mu = 2$ thus obtaining higher δ values (e.g., up to 27% on bandwidth). On the other hand, in (2) and (3) focus is more on system margin maximization. For example, in (2) by setting $\theta_1 = 20$, $\theta_2 = \theta_3 = \theta_4 = 4$, $\mu = 1$, and $\alpha = 0$ we got a 17% margin on bandwidth. This lowers down to 0.8% in (3) where we set $\theta_1 = 8$, $\theta_2 = \theta_3 = \theta_4 = 2/3$, $\mu = 1/6$, and $\alpha = 0.8$ thus obtaining the overall minimum power solution.

We notice how in lower-power designs the system-level margin on SNR tends to decrease as well. Moreover, the unity gain frequency (and the bandwidth), which is the key parameter influencing the settling behavior of the SHA, tends to decrease thus impacting the accuracy of the system (i.e., INL, DNL, and G) and mandating more accurate and power expensive calibration circuits. We finally compare results in Table 1 with the optimal design reported in [4]. Using a nominal optimization technique, we obtained 52.5 mW ADC power consumption with approximately 9% margins. This implies that, in the nominal formulation, it was still possible to obtain reasonable architectural and system margins by acting both on optimization constraints and feasible performance model generation constraints as viable safety margin knobs. However, we had not chances to quantitatively explore and efficiently control the involved performance/margin tradeoffs as we have demonstrated here in the robust formulation.

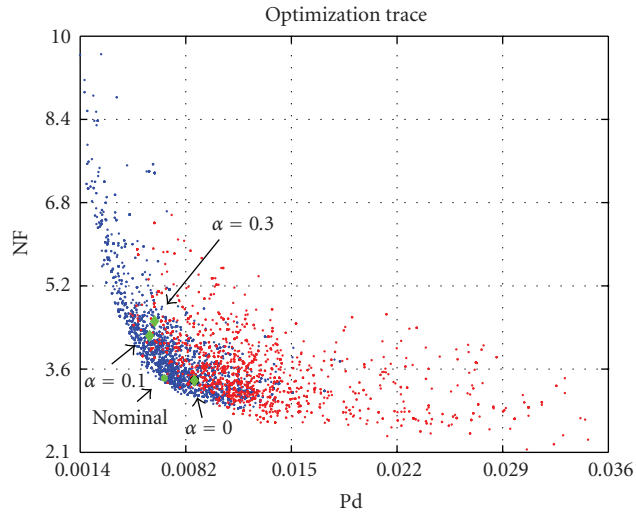

Figure 11: Optimization results compared with the nominal optimization trace (projections on the LNA NF-Power space). Red dots correspond to npMOS instances, blue dots to nMOS instances. Robust results do not lie on the Pareto optimal curve.

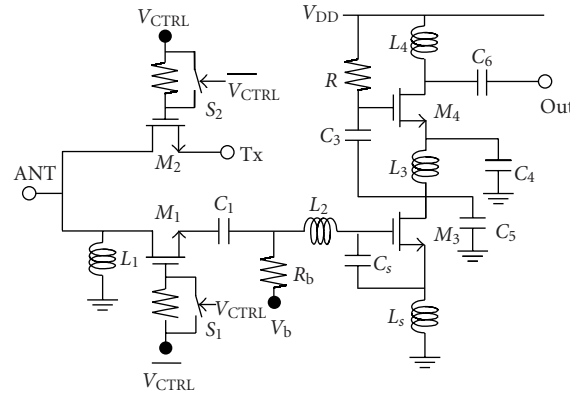

Figure 12: Schematic of the LNA used in the UWB receiver front-end together with its input matching network.

5.2. UMTS Front-End. In this and the following subsections, we demonstrate our methodology on RF systems. We start with robust optimization of the UMTS receiver front-end presented in [27]. The receiver consists of a Low-Noise Amplifier (LNA) and a mixer for a direct conversion UMTS receiver. All components were characterized and embedded in a platform library. In the nominal optimization, the cost function aims at minimizing power consumption of the overall receiver subject to compliance of standard UMTS tests and performance models. The architecture space is formed by two LNA topologies and one direct-conversion mixer, as reported in Figures 8, 9, and 10. The system-level

Table 2: UMTS receiver robust optimization results as a function of α. Larger values of α decrease the sensitivity on δ in (9). Note that the LNA topology is also affected by robust optimizations.

α	nom.	0	0.1	0.3
Gain (dB)	30.8	29.9	31.2	27.7
Power (mW)	10.9	14.4	12.9	11.7
D_2 (dBm)	−99.1	−99.7	−100.2	−99.5
D_3 (dBm)	−98.7	−99.5	−100.3	−99.7
ϵ_2/ϵ_3 (dB)	–	0.7/3.5	1.2/4.3	0.5/3.7
NF (dB)	6.8	6.9	6.4	6.9
Gain_{LNA} (dB)	18.2	15.4	17.3	15.7
NF_{LNA} (dB)	3.5	3.8	4.5	5.8
$\text{Power}_{\text{LNA}}$ (mW)	5.7	7.3	6.1	4.2
Topology	n	n	n	np
Margin	1.3%	12.5%	6.9%	3.6%
CG (dB)	12.6	14.5	13.9	10
$\text{Power}_{\text{MIX}}$ (mW)	5.1	7.1	6.8	5.9
Margin	0.9 %	15.8%	6.1%	4.8%

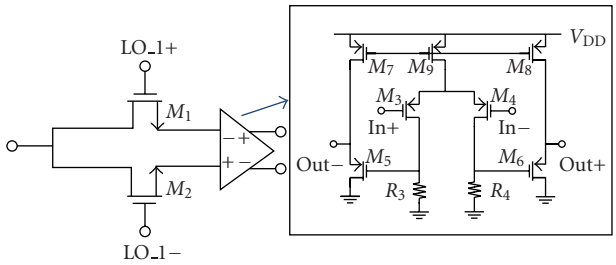

Figure 13: Schematic of the passive mixer and buffer used in the UWB receiver.

constraints (directly derived from UMTS specifications) are compactly formulated with

$$\begin{aligned} D_2 &\triangleq \frac{1}{G_R^2}(P_2 + N + P_{\text{rm}}) \le -99\,\text{dBm}, \\ D_3 &\triangleq \frac{1}{G_R^2}(P_3 + N + P_{\text{rm}}) \le -96\,\text{dBm}, \end{aligned} \tag{13}$$

where P_2 and P_3 are the output-referred second- and third-order distortion powers, respectively, N the output-referred noise power, P_{rm} the output-referred power due to reciprocal mixing, and G_R the front-end gain. The standard specifies the conditions in which system performance has to be assessed. All quantities are evaluated through the receiver behavioral model $\mathcal{F}$, described in [27]. Exploiting the robust formulation (3) and the cost function template (9), the following robust optimization problem has been obtained:

$$\begin{aligned} \min_{\kappa} \quad & \Big((\alpha + \tanh(40 \cdot \min(\delta_L, \delta_M))) \\ & \cdot \left(\epsilon_1^{\theta} \cdot \epsilon_2^2 \cdot \epsilon_3 \cdot \epsilon_4^{-2}\right)^{1/4}\Big)^{-1}, \\ \text{s.t.} \quad & \begin{cases} \zeta = \mathcal{F}(\kappa), \\ P \le 25\,\text{mW} \cdot (1+\epsilon_1)^{-1}, \\ D_2 \le -99 - \epsilon_2\,\text{dBm}, \\ D_3 \le -96 - \epsilon_3\,\text{dBm}, \\ \dfrac{C_L}{C_M} = 1 + \epsilon_4, \\ \mathcal{P}_{\text{L}}(\kappa_{\text{L}}) \ge 1, \quad \delta_{\text{L}} = \text{margin}(\mathcal{P}_{\text{L}}, \kappa_{\text{L}}), \\ \mathcal{P}_{\text{M}}(\kappa_{\text{M}}) \ge 1, \quad \delta_{\text{M}} = \text{margin}(\mathcal{P}_{\text{M}}, \kappa_{\text{M}}), \\ \epsilon_i > 0. \end{cases} \end{aligned} \tag{14}$$

The parameter α has been used to control the amount of margin on δ and thus the architecture margin at optimum. The tanh term has been set as to saturate at margins larger than 15% (δ is normalized in $[0,1]$). ϵ_1 determines power consumption margin and its weight is controlled by the parameter θ. ϵ_2 and ϵ_3 set the margin on minimum interference requirements. Since in a direct conversion receiver second-order terms are crucial, we increased its weight squaring ϵ_2. Finally, ϵ_4 measures the mismatch on the interface capacitance between LNA and mixer, and has to be minimized, as detailed in [27] in order to guarantee correct platform composition.

An optimization trace projected onto the Power-NF plane for the LNA is reported in Figure 11. The robust

Table 3: UWB RF front-end receiver robust optimization results as a function of α. Larger values of α decrease the sensitivity on δ in (15).

α	nom.	0	0.1	0.3
Gain (dB)	18.32	19.84	17.72	19.64
Power (mW)	7.95	8.59	8.33	10.28
NF (dB)	4.16	4.16	3.99	3.75
IIP_3 (dBm)	−24.07	−24.68	−21.00	−18.65
ϵ_1		0.75	0.80	0.46
ϵ_2		0.84	1.01	1.25
ϵ_3		5.32	9.00	11.35
Gain_{LNA} (dB)	19.42	21.57	20.30	21.56
NF_{LNA} (dB)	3.81	4.00	3.72	3.56
$IIP_{3\text{LNA}}$ (dBm)	−9.32	−11.13	−10.46	−11.49
$\text{Power}_{\text{LNA}}$ (mW)	5.8	6.38	6.39	8.44
Margin	0.22%	32.0%	24.6%	14.2%
CG (dB)	−1.11	−1.73	−2.59	−1.9
NF_{MIX} (dB)	12.79	11.81	12.32	12.07
$IIP_{3\text{MIX}}$ (dBm)	−4.47	−2.81	−0.29	3.83
$\text{Power}_{\text{MIX}}$ (mW)	2.15	2.21	1.93	1.84
Margin	0.04 %	17.13%	0.1%	0.1%

approach is able to perform architecture selection between the LNA topologies, as shown in Table 2. Larger values for the α parameter allow more aggressive optimizations, as shown by lower-power consumption levels. Moreover, it is evident that the optimal point does not lie on the Pareto optimal curve of the LNA performances, as was the case in the nominal design in [27]. In this example, area occupation is not directly traded with system robustness against variations. Table 2 shows the performances at optimum together with the main performance indices and corresponding margins. In this case, since direct conversion architectures are extremely sensitive to second-order distortion, we exploited an ellipsoid to compute δ_M so that the second-order distortion coefficient weight is 3 times the other performance indices. Overall, compared to the optimal nominal design, a significant increase in power is observed (+32% for the case $\alpha = 0$), but the final system allows for wide margins to compensate modeling inaccuracies and layout effects.

5.3. UWB Front-End. In this subsection, we proceed with optimization, under robustness constraints, of a UWB receiver based on the architecture in [28]. The RF front-end includes two main building blocks, which were both characterized and embedded into a platform library. The first block, shown in Figure 12, consists of the Tx/Rx switch (M_1 and M_2), the wideband (3.1–4.8 GHz) input matching network (L_1, L_2, L_S, and C_S), and the LNA, which features a stagger tuning technique to achieve gain flatness over the wideband of interest. The second block, represented in Figure 13, includes a passive mixer (M_1 and M_2) and a low-noise buffer amplifier (M_3–M_8) to boost the mixer gain.

In the nominal optimization, we aimed at minimizing power consumption (P) of the RF front-end while meeting system constraints on IIP_3, total gain G, and noise figure (NF). Similar to (14), the robust optimization problem is formulated as follows:

$$\min_{\kappa} \quad ((\alpha + \tanh(40 \cdot \min(\delta_L, \delta_M))) \cdot (\epsilon_1 \cdot \epsilon_2 \cdot \epsilon_3))^{-1},$$

$$\text{s.t.} \begin{cases} \zeta = \mathcal{F}(\kappa), \\ P \le 15 \cdot (1+\epsilon_1)^{-1}\,\text{mW}, \\ \text{NF} \le 5 - \epsilon_2\,\text{dB}, \\ IIP_3 \ge -30 + \epsilon_3\,\text{dBm}, \\ \text{G} \ge 15\,\text{dB}, \\ \mathcal{P}_L(\kappa_{\text{LNA}}) \ge 1, \quad \delta_L = \text{margin}(\mathcal{P}_{\text{L}}, \kappa_{\text{L}}), \\ \mathcal{P}_M(\kappa_{\text{M}}) \ge 1, \quad \delta_{\text{M}} = \text{margin}(\mathcal{P}_{\text{M}}, \kappa_{\text{M}}), \\ \epsilon_i > 0, \end{cases} \tag{15}$$

where the system performance figures (power, NF, gain, and IIP_3) are calculated from the κ_{L} and κ_{M} using RF cascade equations, as follows:

$$\begin{aligned} P &= P_L + P_M, \\ \text{NF} &= 10\log\left(10^{\text{NF}_L/10} + 10^{(\text{NF}_L - G_L)/10}\right), \\ IIP_3 &= -10\log\left(10^{-IIP_{3L}/10} + 10^{(G_L - IIP_{3M})/10}\right). \end{aligned} \tag{16}$$

The behavioral model $\mathcal{F}$ is then built out of (16), with some additional *validity laws* enforcing correct block composition. As in the UMTS optimization problem, α has been used to control the amount of margin on δ, hence the architecture margin at optimum. For this particular application, power, NF, and IIP_3, the most critical performance parameters, have

been given the same relative weight. ϵ_1 dictates the power consumption margin, while robustness with respect to gain and second-order distortion variations is less of a concern for our UWB communication system. ϵ_2 and ϵ_3 set the system-level margins on noise figure and IIP_3 of the optimum system design. As in (14), the tanh function saturates when margins become too large (>15%).

In Table 3, we report the optimal performance as a function of α, together with the main performance figures and their margins. As in the UMTS case, larger values of α imply more aggressive optimizations, better performance, and lower margins. However, this does not necessary translate into lower-power consumption, in this case, since the system-level margin with respect to power has the same weight as the other margins. Since the system performance is more sensitive to NF_L and IIP_{3M}, the ellipsoids used to compute δ_L and δ_M were selected so that the LNA noise figure and the mixer IIP_3 have a weight which is twice the one of the other performance indices. Overall, the final system allows for much wider margins with respect to the nominal solution, albeit at the cost of increased power consumption (+30% for the case $\alpha = 0.3$).

As a final comment on the results, we could not perform a Monte Carlo analysis on the actual circuits for any design since the complexity of our systems rules out the possibility of performing any reasonable number of simulations to get meaningful results. In fact, this was an important motivation to introduce robustness early in the design cycle starting from the system level.

6. Conclusions

Platform-Based Design (PBD) is a promising methodology for embedded system design, aiming to improve design productivity by encouraging design reuse, orthogonalization of concerns, and system-level optimization. In this paper, we have illustrated the extension of PBD to mixed-signal systems. Furthermore, to ensure robustness with respect to both model and design uncertainties, we have proposed the application, within the PBD framework, of design-centering techniques. The proposed approach allows robust hierarchical design without any assumption on the mathematical properties of the system models, leading to a general formulation that can be used for robust automatic design-space exploration.

To demonstrate the effectiveness of the proposed design methods in different domains, we presented three case studies: a mixed-signal pipeline ADC and two RF front-ends, respectively, for UMTS and UWB receivers. In all cases, designs were efficiently composed from precharacterized components, as well as optimized at the system level, demonstrating the flexibility of the approach and significant improvements in terms of robustness.

References

[1] A. Sangiovanni-Vincentelli, "Quo vadis, SLD? Reasoning about the trends and challenges of system level design," *Proceedings of the IEEE*, vol. 95, no. 3, pp. 467–506, 2007.

[2] A. Sangiovanni-Vincentelli, "Defining platform-based design," *EE-Design*, 2002.

[3] F. De Bernardinis, P. Nuzzo, and A. Sangiovanni-Vincentelli, "Robust system level design with analog platforms," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers (ICCAD '06)*, pp. 334–341, San Jose, Calif, USA, November 2006.

[4] F. De Bernardinis, P. Nuzzo, and A. Sangiovanni-Vincentelli, "Mixed signal design space exploration through analog platforms," in *Proceedings of the 42nd Design Automation Conference (DAC '05)*, pp. 875–880, Anaheim, Calif, USA, June 2005.

[5] P. Nuzzo, F. De Bernardinis, and A. Sangiovanni-Vincentelli, "Platform-based mixed signal design: optimizing a high-performance pipelined ADC," *Analog Integrated Circuits and Signal Processing*, vol. 49, no. 3, pp. 343–358, 2006.

[6] F. De Bernardinis, *System level mixed signal design with analog platforms*, Ph.D. dissertation, University of California, Berkeley, Calif, USA, 2005.

[7] J. M. Rabaey, F. De Bernardinis, A. M. Niknejad, B. Nikolić, and A. Sangiovanni-Vincentelli, "Embedding mixed-signal design in systems-on-chip," *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1070–1087, 2006.

[8] H. Liu, A. Singhee, R. A. Rutenbar, and L. R. Carley, "Remembrance of circuits past: macromodeling by data mining in large analog design spaces," in *Proceedings of the 39th Design Automation Conference (DAC '02)*, pp. 437–442, New Orleans, La, USA, June 2002.

[9] T. Kiely and G. Gielen, "Performance modeling of analog integrated circuits using least-squares support vector machines," in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition (DATE '04)*, vol. 1, pp. 448–453, Paris, France, February 2004.

[10] F. De Bernardinis, M. I. Jordan, and A. Sangiovanni-Vincentelli, "Support vector machines for analog circuit performance representation," in *Proceedings of the 40th Design Automation Conference (DAC '03)*, pp. 964–969, Anaheim, Calif, USA, June 2003.

[11] G. Stehr, H. Graeb, and K. Antreich, "Analog performance space exploration by Fourier-Motzkin elimination with application to hierarchical sizing," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers (ICCAD '04)*, pp. 847–854, San Jose, Calif, USA, November 2004.

[12] R. Brayton, G. Hachtel, and A. Sangiovanni-Vincentelli, "A survey of optimization techniques for integrated-circuit design," *Proceedings of the IEEE*, vol. 69, no. 10, pp. 1334–1362, 1981.

[13] S W. Director and G. D. Hachtel, "The simplicial approximation approach to design centering," *IEEE Transactions on Circuits and Systems*, vol. 24, no. 7, pp. 363–372, 1977.

[14] K. K. Low and S. W. Director, "A new methodology for the design centering of IC fabrication processes," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, no. 7, pp. 895–903, 1991.

[15] E. Ochotta, R. Rutenbar, and R. Carley, *Synthesis of High Performance Circuits*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[16] G. Box and N. Draper, *Empirical Model-Building and Response Surfaces*, John Wiley & Sons, New York, NY, USA, 1987.

[17] H. Chang, E. Charbon, U. Choudhury, et al., *A Top-Down Constraint-Driven Design Methodology for Analog Integrated Circuits*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[18] H. Chang, E. Liu, R. Neff, et al., "Top-down, constraint-driven design methodology based generation of n-bit interpolative current source D/A converters," in *Proceedings of the Custom Integrated Circuits Conference (CICC '94)*, pp. 369–372, San Diego, Calif, USA, May 1994.

[19] I. Vassiliou, H. Chang, A. Demir, E. Charbon, P. Miliozzi, and A. Sangiovanni-Vincentelli, "Video driver system designed using a top-down, constraint-driven methodology," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers (ICCAD '96)*, pp. 463–468, San Jose, Calif, USA, November 1996.

[20] G. Van Der Plas, G. Gielen, and W. Sansen, *A Computer-Aided Design and Synthesis Environment*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.

[21] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[22] P. Gopalakrishnan, Y. Xu, and L. T. Pileggi, "Robust analog/RF circuit design with projection-based posynomial modeling," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers (ICCAD '04)*, pp. 855–862, San Jose, Calif, USA, November 2004.

[23] Y. Xu, I. Nausieda, K.-L. Hsiung, S. Boyd, X. Li, and L. Pileggi, "OPERA: optimization with ellipsoidal uncertainty for robust analog IC design," in *Proceedings of the 42nd Design Automation Conference (DAC '05)*, pp. 632–637, Anaheim, Calif, USA, June 2005.

[24] J. Wang, L. T. Pileggi, T.-S. Chen, and W. Chiang, "Performance-centering optimization for system-level analog design exploration," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers (ICCAD '05)*, pp. 421–428, San Jose, Calif, USA, November 2005.

[25] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, UK, 2nd edition, 1992.

[26] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Nashua, NH, USA, 1995.

[27] F. De Bernardinis, S. Gambini, F. Vincis, F. Svelto, R. Castello, and A. Sangiovanni-Vincentelli, "Design space exploration for a UMTS front-end exploiting analog platforms," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers (ICCAD '04)*, pp. 923–930, San Jose, Calif, USA, November 2004.

[28] Y. Li, C.-C. Wu, A. Sangiovanni-Vincentelli, and J. M. Rabaey, "Design and optimization of an MB-OFDM ultra-wideband receiver front-end," in *Proceedings of the 4th IEEE International Conference on Circuits and Systems for Communications (ICCSC '08)*, pp. 502–506, Shanghai, China, May 2008.

[29] I. Galton, "Digital cancellation of D/A converter noise in pipelined A/D converters," *IEEE Transactions on Circuits and Systems*, vol. 47, no. 3, pp. 185–196, 2000.

[30] B. Murmann and B. E. Boser, "A 12-bit 75-MS/s pipelined ADC using open-loop residue amplification," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 12, pp. 2040–2050, 2003.