Robust System Level Design with Analog Platforms

F. De Bernardinis[§]
f.debernardinis@ing.unipi.it

P. Nuzzo[§]
pierluigi.nuzzo@iet.unipi.it

A. Sangiovanni Vincentelli[†]
alberto@eecs.berkeley.edu

[†]Department of Electrical Engineering and Computer Science
University of California, Berkeley

[§]Dipartimento di Ingegneria dell'Informazione
Universita di Pisa, Italy

ABSTRACT

An approach to robust system level mixed signal design is presented based on analog platforms. The bottom-up characterization phase of platform components provides accurate performance models that export architectural constraints to the system level. From the one side, performance models can be affected by residual errors and usually do not consider process variations and modeling uncertainties. Conversely, behavioral models cannot match accurate circuit level simulations, so that during the mapping (exploration) process circuit configurations difficult to be realized may be obtained. We propose a methodology that extends techniques from optimization and design centering to system level analog design exploiting general, implicit architectural constraints to control the robustness of the solution. The approach allows quantitative extension of robust techniques to hierarchical designs. Its effectiveness is illustrated with the design of a pipeline A/D converter and a UMTS receiver front-end.

Categories and Subject Descriptors: B.7.2 [Integrated Circuits]: Design Aids

General Terms: Algorithms

Keywords: robust hierarchical design, system-level design, analog platforms.

1. INTRODUCTION

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. Early approaches to computer aided design centering in an analog context date back to the early 80s [1, 2, 3]. 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, that is usually captured as a semi-infinite programming problem. In [4], 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. A direct extension of these techniques to system level analog and mixed-signal design is therefore deemed very unlikely to happen. Alternative approaches based on approximate models must be developed at the system level, where the models generated with classic approaches based on Surface Response Methodology (SRM) [5] become too expensive to build because of the number of primal parameters and the complexity of the necessary simulations.

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) 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.

In this paper, we extend the Analog Platform Based Design (APBD) paradigm [6] with robust design techniques. As in other approaches, e.g. [7] and [8], robustness is achieved through maximization of margins with respect of system specifications. However, we argue that the performance models which annotate platform components allow accurate robust approaches since there are no constraints (e.g. convexity or explicit form) on the original models. Furthermore, a general optimization approach based on simulated annealing allows exploiting arbitrary cost functions when formulating the optimization problem, thus increasing designers' flexibility in performing system level design. The examples that end this paper show the effectiveness of the approach.

ICCAD'06, November 5 9, 2006, San Jose, CA

2. BACKGROUND

2.1 Previous Approaches

Several robust approaches to analog design have been proposed during the past few years. Initially, relaxation of system constraints during top-down optimizations where 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 [9] and demonstrated in [10, 11]. 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 [12] 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, i.e. 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 [13] have revitalized analytical approaches to analog design and, consequently, robust design. ROAD [14] 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 generated around design points. It is then possible to deal with non-convex design spaces exploiting the possibility of exactly solving large scale convex programs. OPERA [7] 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 [8]. 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) model convexity as design hierarchy becomes deeper and high-level behavioral models are exploited in mixed-signal design space explorations.

In this paper, we extend the hierarchical approach removing the posynomial constraints on design formulation. Extending the approach to analog platforms, we obtain a two-fold advantage. First, very accurate performance models (not constrained to be posynomial) 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. Designer can then specify their systems without recurring to posynomial approximations and capturing arbitrary non-convex constraints.

2.2 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, general approaches to system-level robust design 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 [15, 16] and even direct modeling of the feasibility region [17, 18]. 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 consequently reduced (at least in non-degenerate cases) and architecture selection becomes readily available as different implementation topologies may share common performance spaces.

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 $\{\boldsymbol{x_n}\}$ of simulated performance vectors (as detailed in [17]), SVM training selects a subset of vectors $\boldsymbol{x_i}$ (support vectors) and corresponding weight coefficients α so that the classifier function is obtained as

$$f(x) = \mathrm{sgn}(\sum_i \alpha_i e^{-\gamma \|\boldsymbol{x}-\boldsymbol{x_i}\|^2} - \rho) \qquad (1)$$

where ρ is a biasing term (also determined during training) and γ is an SVM parameter. Performance vectors $\boldsymbol{x}$ 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, i.e. unfeasible performances classified as feasible. In fact, several case studies

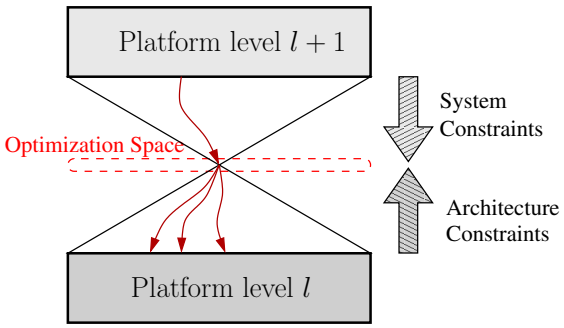

Figure 1: Platform mapping optimization process from level $l+1$ to level l.

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 PLATFORM BASED DESIGN

3.1 Formulation

Analog platform-based design [6] is a meet-in-the-middle design paradigm that allows effective system level design based on libraries of components (platforms). The essence of the approach is pictorially represented in Fig. 1 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. The optimization process mapping platform $l+1$ onto platform l is mathematically captured as

$$\text{s.t.} \begin{array}{l} \min_{\kappa} \ \text{cost}(\zeta) \\ \left\{ \begin{array}{l} \zeta = \mathcal{F}(\kappa) \\ \mathcal{S}(\zeta) \leq 0 \\ \mathcal{P}(\kappa) \leq 0 \end{array} \right. \end{array} \tag{2}$$

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). Fig. 2 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 con-

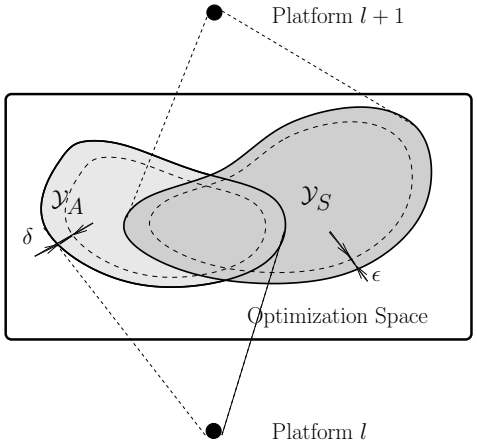

Figure 2: Enlarged view of the optimization space in Fig. 1.

straints 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.

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, the problem (2) becomes

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

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

$$\begin{cases} s_1(\zeta) - \epsilon_1 \leq 0 \\ \vdots \\ s_p(\zeta) - \epsilon_p \leq 0 \\ \epsilon_1 > 0, \ldots, \epsilon_p > 0 \end{cases} \tag{4}$$

and included in the optimization problem. Additional constraints may be inserted to set specific relations on ϵ, e.g. $\epsilon_1 = 2\epsilon_2$. The problem is more involved with performance models, as analog platforms provide $\mathcal{P}$ in implicit form with a non-linear 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 $\mathcal{P}(\kappa) = 1$[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 $\delta = ||\hat{\kappa} - \kappa|| \cdot \mathbf{1}$. In this case, minimizing $\left(\prod_i \delta_i\right)^{-1}$ is equivalent to maximizing the volume of the sphere around κ that is inclosed 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 pre-conditioned to be normalized in the interval $[-1, 1]$. In the following paragraph, we show how to compute δ based on the SVM representation of $\mathcal{P}$.

3.2 Performance Margin Evaluation

The problem of finding $\hat{\kappa}$ given κ and $\mathcal{P}$ is analogous to the problem of finding the largest hyper-ellipsoid enclosed by $\partial\mathcal{P}$. Initially we start solving the case of hyper-sphere 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 $\boldsymbol{x} = \hat{\kappa}$ and $\boldsymbol{a} = \kappa$. Therefore, we can obtain $\hat{\kappa}$ solving the following optimization problem

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

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

$$\sum_i \alpha_i e^{-\gamma\|\boldsymbol{x} - \boldsymbol{x_i}\|^2} - \rho = 0 \tag{6}$$

The optimization problem obtained substituting (6) into (5) is evidently nonlinear. At optimum, the Karush-Kuhn-Tucker conditions require that

$$\begin{cases} \sum_{i=1}^m \alpha_i \mathrm{e}^{-\gamma\|\boldsymbol{x} - \boldsymbol{x_i}\|^2} - \rho = 0 \\ x_j - a_j = \lambda \cdot \gamma \cdot \left(\sum_{i=1}^m \alpha_i \mathrm{e}^{-\gamma\|\boldsymbol{x} - \boldsymbol{x_i}\|^2} x_{i,j} - \rho x_j\right) \\ j = 1, \ldots, n; \\ \lambda \geq 0; \end{cases} \tag{7}$$

where λ is the Lagrange multiplier. The nonlinear system (7) can be solved with Newton-Raphson (NR) providing quadratic convergence if $\boldsymbol{x}_0$ is "close" to $\hat{\boldsymbol{x}}$. $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|$ is therefore the radius of the largest hyper-sphere 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 $\boldsymbol{x}$; 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, the *damped* Newton's method [13], 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 a n-dimensional system in the unknown vector $\boldsymbol{x}$, which can denoted as $\boldsymbol{F}(\boldsymbol{x}) = 0$. We then combine NR method with the minimization of the function $f = \frac{1}{2}\|\boldsymbol{F}\|^2$, 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 $\boldsymbol{d}$ starting from the old point $\boldsymbol{x_{old}}$ until we have an acceptable new point $\boldsymbol{x_{new}} = \boldsymbol{x_{old}} + \nu\boldsymbol{d}$ $(0 < \nu \leq 1)$. 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 [19] and consists in defining $g(\nu) \equiv f(\boldsymbol{x_{old}} + \nu\boldsymbol{d})$, as the restriction of f along $\boldsymbol{d}$, 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:

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

- we set $\boldsymbol{x}_0 = \boldsymbol{a}$ 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 $\boldsymbol{x}_0$ in the direction of the axis where the minimum distance has been found in the previous point (iteration are aborted after a predetermined number) until the minimum distance solution is reached. We observed that $N = 5$ is generally sufficient to achieve convergence;

- in case of non-convergence, 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 hyper-ellipsoids enclosure by scaling $\boldsymbol{x}$ with a unitary matrix $\mathbf{E}$ to obtain $\boldsymbol{x}' = \mathbf{E}\boldsymbol{x}$ and extending the previous approach on $\boldsymbol{x}'$. 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.

The overall algorithm complexity has been computed to be $O(n^2m + nmc_{exp})$ 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).

[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.

4. EXAMPLES

In this section we apply the previous results to the case studies already reported in [6] and [20]. 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

$$\frac{1}{\left(\prod_{i=1}^{k}\left(\alpha_i+\tanh(\beta_i\delta_i)\right)^{\mu_i}\right)^{\frac{1}{k}}\cdot\left(\prod_{j=1}^{r}\epsilon_j^{\theta_j}\right)^{\frac{1}{r}}} \tag{8}$$

A few considerations may help explaining the form of (8). 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 $\frac{1}{k}$ and $\frac{1}{r}$ of δ 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 the margin of the i^{th} component. If we consider that the composition of blocks is as robust as the weakest block, we can obtain a different cost function considering $\min_i\left(\alpha_i+\tanh(\beta_i\delta_{i,1})\right)$. 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

$$\theta_a\frac{\Delta\epsilon_a}{\epsilon_a}+\theta_b\frac{\Delta\epsilon_b}{\epsilon_b}=0 \tag{9}$$

so that $\frac{\theta_a}{\theta_b}$ sets the relative impact of variations of ϵ_a and ϵ_b. When δ and ϵ are considered, we obtain (for small $\beta\delta$)

$$\frac{\theta_a}{r}\frac{\Delta\epsilon_a}{\epsilon_a}+\frac{\mu_b}{k}\frac{\beta_b\delta_b}{\alpha_b+\beta_b\delta_b}\frac{\Delta\delta_b}{\delta_b}=0 \tag{10}$$

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 (9) and (10) can be used as guidelines to set parameters in (8), 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

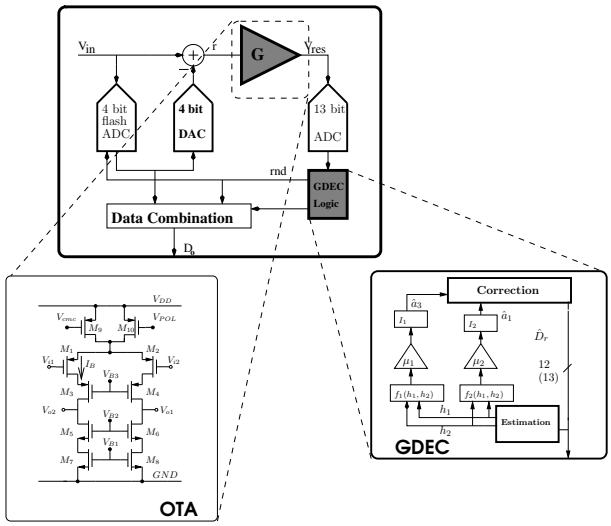

Figure 3: Pipelined converter simplified block diagram—Feasible performance models have been generated for the blocks in grey: the SHA and the gain error digital calibration block (GDEC).

criterion, but this can be seamlessly introduced in the robust optimization scheme to export at the system level area penalties involved in topology selection.

4.1 Pipeline ADC

In [6] we performed design space exploration of a 14-bit, 80 MS/s pipeline analog-to-digital converter (ADC) in 0.13μm, 2.5V analog supply CMOS technology. The simplified block diagram of the system is represented in Figure 3. The ADC is made up of 4 multi-bit stages and includes digital calibration circuits to enhance performance. In particular, the Digital-to-Analog sub-Converter (DAC) errors are canceled with the DAC Noise Cancelation (DNC) technique [21] and the first stage Sample-and-Hold Amplifier (SHA) errors are corrected through a Gain and Distortion Error Correction (GDEC) algorithm as in [22]. The inter-stage residue amplifiers are fully differential switched capacitor systems (Figure 4) implemented with a telescopic Operational Transconductance Amplifier (OTA). The OTA optimization need to be performed under the hypothesis of operation of digital calibration circuits, as detailed in [23]. 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, i.e. the digital calibration logic and the first stage residue amplifier. The remaining part of the converter was considered ideal. 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 non-linearity corresponding to different accuracy and power consumption levels, based on [24]. Performances are eval-

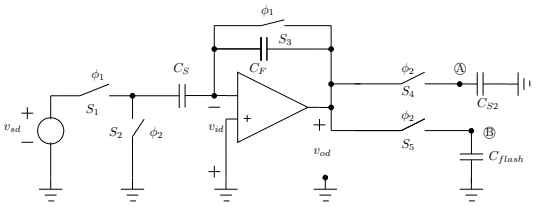

Figure 4: Single-ended equivalent (simplified) circuit of the switched capacitor SHA.

uated 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 (8):

$$\begin{array}{l} \min_\kappa \left(\prod_{j=1}^{4} \epsilon_j^{\theta_j}\right)^{-\frac{1}{4}} \left(\alpha+\delta\right)^{-\mu} \\ \text{s.t.} \quad \left\{ \begin{array}{l} \zeta = \mathcal{F}(\kappa) \\ P_{ADC} \leq 100 \cdot (1+\epsilon_1)^{-1}\,\text{mW} \\ DNL \leq 0.8 \cdot (1+\epsilon_2)^{-1}\,\text{LSB} \\ INL \leq 1 \cdot (1+\epsilon_3)^{-1}\,\text{LSB} \\ SNR \geq 76 \cdot (1+\epsilon_4)\,\text{dB} \\ \mathcal{P}_{SHA}(\kappa_{SHA}) = 1 \;\; \delta = \text{margin}(\mathcal{P}_{SHA}, \kappa_{SHA}) \\ \mathcal{P}_{GDEC}(\kappa_{GDEC}) = 1 \end{array} \right. \end{array} \tag{11}$$

where ϵ_1, ϵ_2, ϵ_3, 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 13h 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 [6]. Using a nominal optimization technique we obtained 52.5mW ADC power consumption with approximately 9% margins. This implies that obtaining reasonable architectural and system margins together with optimal performance was still possible in the past methodology by acting both on optimization constraints and feasible performance model generation constraints to get safety margins. However, we had not chances of quantitatively explore and efficiently control the involved performance/margin trade-offs as we have demonstrated here within the robust paradigm.

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	–

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

4.2 UMTS Front-end

In this subsection we proceed with robust optimization of the UMTS receiver front-end presented in [20]. 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 Fig. 5. The system level constraints (directly derived from UMTS specifications) are compactly formulated with

$$\left\{ \begin{array}{l} D_2 \triangleq \frac{1}{G_R^2}(P_2 + N + P_{rm}) \leq -99dBm \\ D_3 \triangleq \frac{1}{G_R^2}(P_3 + N + P_{rm}) \leq -96dBm \end{array} \right. \tag{12}$$

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 [20]. Exploiting the robust formulation (3) and the cost function template (8), the following robust optimization problem has been obtained

$$\begin{array}{l} \min_\kappa \quad \Big(\big(\alpha + \tanh(40 \cdot \min(\delta_L, \delta_M))\big) \cdot \\ \qquad \cdot \big(\epsilon_1^\theta \cdot \epsilon_2^2 \cdot \epsilon_3 \cdot \epsilon_4^{-2}\big)^{1/4}\Big)^{-1} \\ \text{s.t.} \quad \left\{ \begin{array}{l} \zeta = \mathcal{F}(\kappa) \\ P \leq 25mW \cdot (1+\epsilon_1)^{-1} \\ D_2 \leq -99 - \epsilon_2\; dBm \\ D_3 \leq -96 - \epsilon_3\; dBm \\ \frac{C_L}{C_M} = 1 + \epsilon_4 \\ \mathcal{P}_L(\kappa_L) \geq 1 \qquad \delta_L = \text{margin}(\mathcal{P}_L, \kappa_L) \\ \mathcal{P}_M(\kappa_M) \geq 1 \qquad \delta_M = \text{margin}(\mathcal{P}_M, \kappa_M) \\ \epsilon_i > 0 \end{array} \right. \end{array} \tag{13}$$

The parameter α has been used to control the amount of margin on δ and thus the architecture margin at optimum.

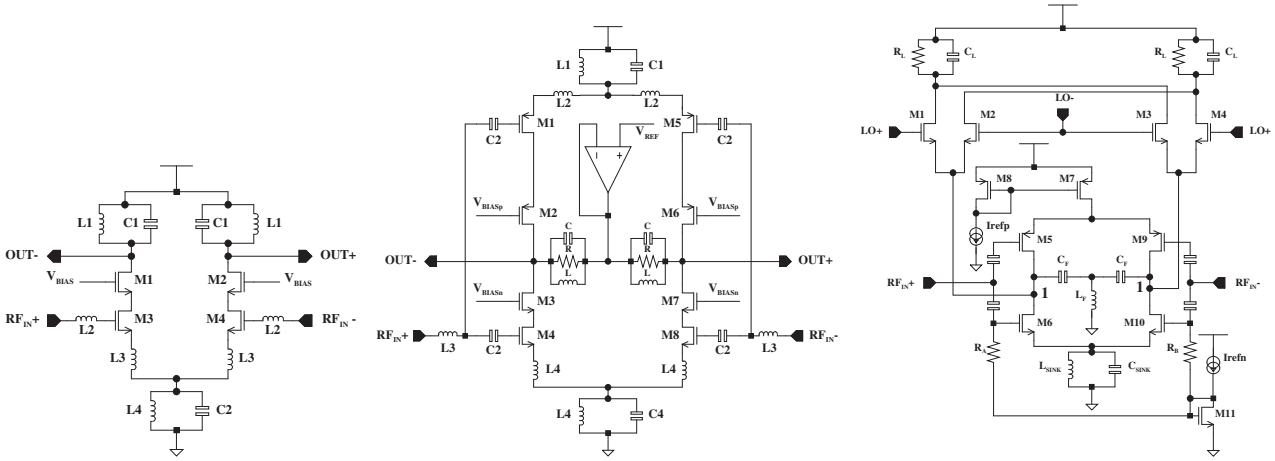

Figure 5: Mixer Schematic for the two LNA (with n and np input stages) and the mixer used in the UMTS receiver front-end.

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 [20] in order to guarantee correct platform composition.

An optimization trace projected onto the Power-NF plane for the LNA is reported in Fig. 6. The robust approach is able to perform architecture selection between the LNA topologies, as shown in Table 4.2. 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 [20]. In this example, area occupation is not directly traded with system robustness against variations. Tab. 4.2 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. As a final comment on the results, we could not perform a Monte Carlo analysis on the actual circuits for either 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.

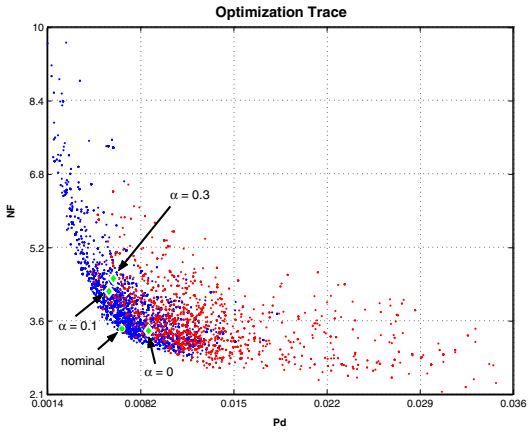

Figure 6: Optimization results compared with the nominal optimization trace (projections on the LNA NF-Power space). Circles correspond to npMOS instances, crosses to nMOS instances. Robust results do not lie on the Pareto optimal curve.

5. CONCLUSIONS

We extended analog platform-based design to offer a methodology that is robust with respect to both model and design uncertainties. The proposed approach allows robust hierarchical design without any assumption on the mathematical properties of the system models. Exploiting accurate performance models allows addressing even aggressive design with an automated robust approach, minimizing the performance overhead paid for robustness. The approach was tested on two case-studies: a mixed-signal pipeline ADC and an RF UMTS front-end. In both cases, significant improvements in terms of robustness were obtained, demonstrating the flexibility of the approach.

6. ACKNOWLEDGMENTS

This work has been partially funded by the Italian Research Program PRIN protocol 2005093524_002.

7. REFERENCES

[1] 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. 1334–62, October 1981.

[2] S. Director, "The Simplicial Approximation Approach to Design Centering," *IEEE Transactions on Circuits and Sys-*

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
Power_{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
Power_{MIX} (mW)	5.1	7.1	6.8	5.9
Margin	0.9 %	15.8%	6.1%	4.8%

Table 2: UMTS receiver robust optimization results as a function of α. Larger values of α decrease the sensitivity on δ in (8). Note that the LNA topology is also affected by robust optimizations.

tems, vol. 24, no. 7, pp. 363–72, July 1977.

[3] K. Low and S. Director, "A New Methodology for the Design Centering of IC Fabrication Processes," *IEEE Trans. on CAD*, vol. 10, no. 7, pp. 895–903, 1991.

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

[5] G. Box and N. Draper, *Empirical Model-Building and Response Surfaces*. John Wiley, 1987.

[6] F. De Bernardinis, P. Nuzzo, and A. Sangiovanni Vincentelli, "Mixed Signal Design Space Exploration through Analog Platforms," in *Proc. IEEE/ACM DAC*, 2005.

[7] Y. Xu, K. Hsiung, X. Li, I. Nuasieda, S. Boyd, and L. Pileggi, "OPERA: OPtimization with Ellispoidal uncertainty for Robust Analog IC design," in *Proc. IEEE/ACM DAC*, 2005, pp. 632–637.

[8] X. Li, J. Wang, L. Pileggi, T. Chen, and W. Chiang, "Performance-Centering optimization for System-Level Analog Design Exploration," in *Proc. IEEE/ACM International Conference on CAD*, 2005, pp. 421–428.

[9] H. Chang *et al.*, *A Top-Down Constraint-Driven Design Methodology for Analog Integrated Circuits*. Kluwer Academic Publishers, 1997.

[10] ——, "Top-Down, Constraint-Driven Methodology Based Generation of n-bit Interpolative Current Source D/A," in *Proc. IEEE CICC*, May 1994, pp. 369–372.

[11] I. Vassiliou *et al.*, "A Video Driver System designed using a Top-Down, Constraint-Driven Methodology," in *Proc. IEEE/ACM International Conference on CAD*, November 1996, pp. 463–468.

[12] G. Van Der Plas, G. Gielen, and W. Sansen, *A Computer-Aided Design and Synthesis Environment*. Kluwer Academic Publishers, 2002.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univesity Press, 2004.

[14] X. Li, P. Gopalakrishnan, Y. Xu, and L. Pileggi, "Robust Analog/RF Design with Projection-Based Posynomial Modeling," in *Proc. IEEE/ACM International Conference on CAD*, 2004, pp. 855–862.

[15] H. Liu, A. Singhee, R. Rutenbar, and L. R. Carley, "Remembrance of Circuits Past: Macromodeling by Data Mining in Large Analog Design Spaces," in *Proceedings of DAC*, 2002.

[16] T. Kiely and G. Gielen, "Performance Modeling of Analog Integrated Circuits using Least-Squares support vector machines," in *Proc. of DATE*, 2004.

[17] F. De Bernardinis, M. Jordan, and A. Sangiovanni Vincentelli, "Support Vector Machines for Analog Circuit Performance Representation," in *Proc. of DAC*, June 2003.

[18] G. Stehr, H. Graeb, and K. Antreich, "Analog Performance Space Exploration by Fourier-Motzkion Elimination with Application to Hierarchical Sizing," in *Proc. of ICCAD*, 2004.

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

[20] 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 *Proc. IEEE/ACM International Conference on CAD*, November 2004.

[21] I. Galton, "Digital Cancellation of D/A Converter Noise in Pipelined A/D Converters," *IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing*, vol. 47, no. 3, pp. 185–196, March 2000.

[22] 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, December 2003.

[23] F. De Bernardinis, P. Nuzzo, and A. Sangiovanni Vincentelli, "Platform-based mixed signal design: Optimizing a high-performance pipelined ADC," *Analog Integrated Circuit and Signal Processing*, to be published.

[24] P. Nuzzo, F. De Bernardinis, and P. Terreni, "Efficient polynomial inversion for the linearization of pipeline ADC," in *Proc. of PRIME*, 2006.