Efficient Optimization Methodology for CT Functions Based on a Modified Bayesian Kriging Approach

Catalin-Adrian TUGUI, Romain BENASSI, Stefan APOSTOL, Philippe BENABES, *Senior Member*, *IEEE*
Department of Signal Processing and Electronic Systems
SUPELEC
91190 GIF-SUR-YVETTE, FRANCE
catalin.tugui@supelec.fr

***Abstract*—The conception of analog and mixed-signal functions requires great effort because the complex analog parts should be recursively optimized based not only on system-level requirements but also on technological limitations and imperfections. High-level behavioral models used for chip-level simulations can be employed using multi-domain hardware description languages (HDL), but they are usually manually written and lack technological characteristics. Moreover, automatic resizing and optimization at the transistor level are very limited, and the behavioral models cannot be re-adjusted to changes at the transistor level. In this paper, we present an efficient design methodology implying the automatic optimization of cells at the transistor level using a modified Bayesian Kriging approach and the extraction of robust analog macro-models, which can be directly regenerated during the optimization process. Coherent results were obtained when using the proposed methodology for the conception of a sixth-order continuous-time (CT) Sigma-Delta (ΣΔ) modulator.**

I. INTRODUCTION

Novel systems-on-a-chip (SoCs) and systems-in-a-package (SiPs) contain mixed analog and digital circuitry along with RF parts and even micro-mechanical and optical parts (oscillators, sensors, actuators), which are integrated in technologies at the nanometer scale. The conception of these complex systems imposes the existence of a unified methodology for the design and verification of all multi-domain components to achieve first-silicon success. Top-down design paradigms are preferred, aiming to ensure performance optimization directly at the system level; however, efficient analog design methodologies should still be refined for the automatic conception and optimization process and to balance existing high-performance digital design techniques [1]. Because transistor-level Spice/Spectre simulations are prohibitive in the case of large CT functions like Sigma-Delta modulators [2], the analog blocks are described as behavioral or functional models in mixed-signal multi-domain HDLs, such as VHDL-AMS, SystemC-AMS and VerilogAMS, allowing for fast chip-level simulations and improvement in the system architecture in early design stages.

We propose a refined top-down analog design methodology for linear CT functions and the corresponding tools with which the macro-models of circuits, including transimpedance amplifiers (TIA), transconductances amplifiers (Gm), current conveyors (CCII), and current mirrors, are automatically extracted from their transistor-level implementations. Prior to extraction, an online optimization process of the cell is conducted based on a Kriging metamodel, assuring fast convergence of the solution because analog functions are expensive to evaluate over large input space. The approach is innovative in many respects. Some features of our methodology and tools, which differentiate them from other procedures [3], include the following:

1. The entire process is automated and integrated in a dedicated MATLAB-CADENCE toolbox [4], from transistor-level simulations to optimization and macro-model extraction;
2. The optimization objectives are used along with 2 types of constraints: strong constraints (nonlinearities, distortions) and normal constraints (offsets, impedances, gains, bandwidths), which are adequate for analog characterization, ensuring robust optimal solutions.

Section II presents the analog design methodology. In section III, we present the optimization algorithms within the context of this methodology. Section IV provides the example of developing a fast TIA amplifier used in the architecture of a sixth-order CT Sigma-Delta modulator. System level results are also presented. Section V covers the conclusions.

II. ANALOG DESIGN METHODOLOGY

The conception of complex analog functions remains in many respects more challenging than the realization of digital functions because a multitude of figures of merit, performance and constraints are implied.

The use of macro-models can accelerate the design process; still, several problems should be addressed to create good analog behavior that can be exploited at the system level:

- The systems performance specifications are dependent on the technology employed and rapidly degrade for systems sensitive to dispersion, analog imperfections and mismatches; thus, high-level models should incorporate accurate transistor-level characteristics while enabling faster simulation;
- The models should be easily re-adjustable, function of changes at the system or transistor level [1];
- Efficient tools to automate the optimization of analog cells and the macro-model extraction process should be considered.

To address these issues, the design flows are replaced by design chains or loops, where the top-down approaches are combined with bottom-up ones and the system structure and components can be adjusted according to low-level simulations. In this context, we propose a refined design methodology, which is presented in Figure 1.

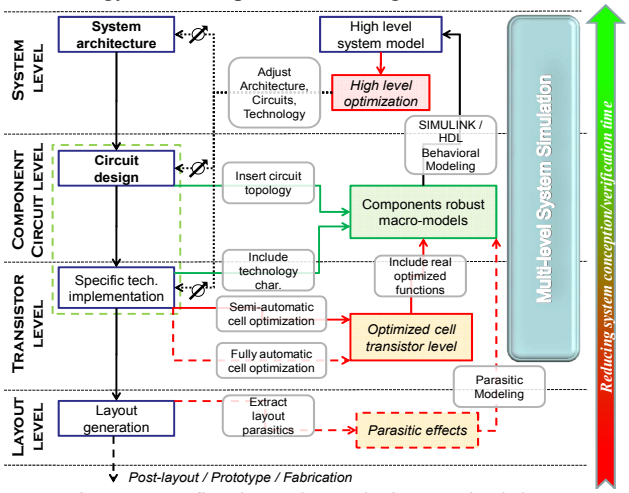


Figure 1: Refined top-down design methodology

Once the system architecture is defined, the circuit design and the transistor level on a specific technology are implemented. Then, an automatic optimization process is performed on each cell using the MATLAB-CADENCE application framework [5]. The optimization process can be fully automatic, where the designer will select the input variables from the circuit and their existing range, the objectives and constraints, the number of starting points and the number of optimization points. In addition, a semi-automatic optimization method based on gradient algorithms is available for early exploration and sensitivity analyses. Using the optimal performance solution (in terms of desired gains, supply, impedances, nonlineary, etc.) a robust macro-model of the Simulink or VHDL-AMS/VerilogA type is extracted for each cell and re-injected at the system level [6].

III. AUTOMATIC OPTIMIZATION

The problem considered here is an optimization problem subject to constraints. A Bayesian approach based on a Kriging probabilistic metamodel is used to spare a significant amount of evaluation time. Indeed, some heuristic methods such as genetic algorithms [7], Tabu search, simulated annealing can be used but are known to generally require more evaluations [8]. Kriging provides also an uncertainty measure easily than most of other metamodels (polynomial, radial basis functions…). As a first step, we explained the principle in an unconstrained case, and then we generalized it to a constrained one. Eventually, we described the specific treatment we used to address the nonlinearity constraint.

A. Bayesian optimization (unconstrained)

We consider here the problem of finding the minimum of a function f: $\boldsymbol{X} \rightarrow \boldsymbol{R}$, where $\boldsymbol{X} \subset \boldsymbol{R}^d$, using the *Expected Improvement* (EI) criterion [9]. The main idea of an EI-based algorithm is a Bayesian one: f is viewed as a sample path of a random process ξ defined on $\boldsymbol{R}^d$. We combine evaluation results and prior information about f to select, according to the EI criterion, new evaluation points efficiently, as long as the budget for evaluations is not exhausted.

After n evaluations, the available information corresponds to the previous evaluation points $(X_1, X_2, \ldots, X_n)$ and to the associated evaluation results $(\xi(X_1), \xi(X_2), \ldots, \xi(X_n))$. Let F_n denote the set $(X_1, \xi(X_1), X_2, \xi(X_2), \ldots, X_n, \xi(X_n))$ and m_n the current minimum, $\min(\xi(X_1), \xi(X_2), \ldots, \xi(X_n))$. Given a budget of N, the purpose of this algorithm is obviously to minimize the value m_N. The next evaluation point X_{n+1} is the maximizer of the EI criterion ρ_n,

$$\rho_n(x) = E_n((m_n - \xi(X_{n+1}))_+ | X_{n+1} = x), \qquad (1)$$

where E_n is the conditional expectation, given F_n and $x \in \boldsymbol{X}$.

In this article, ξ is chosen Gaussian process. $\xi(x)$, conditionally with respect to F_n, is Gaussian with mean $\hat{\xi}_n(x)$ and variance $s_n(x)$, whose explicit expressions can be found in [10]. The main point of choosing ξ as a Gaussian process is the existence of an analytical form for the EI criterion expression (1):

$$s_n(x)\Phi'\left(\frac{m_n - \hat{\xi}_n(x)}{s_n(x)}\right) + \left(m_n - \hat{\xi}_n(x)\right)\Phi\left(\frac{m_n - \hat{\xi}_n(x)}{s_n(x)}\right), \qquad (2)$$

where Φ is the normal cumulative distribution. This expression shows that, given a set of evaluation points and a Gaussian process, the EI sampling criterion can be computed with a moderate amount of resources. Practically, to use this analytical expression, one needs to define an explicit mean and covariance functions for ξ. Experimentally, a Gaussian covariance function is used and the parameter values are estimated at each iteration by maximum likelihood (this approach corresponds exactly to the EGO algorithm [9]). It should be noted that both Kriging predictions and maximum likelihood estimations are performed with the DACE toolbox [11].

B. Bayesian optimization subject to constraints

In this part, we add some constraints to the previous minimization problem. Let us consider p functions $g_1, g_2, \ldots, g_p$, where the new optimization problem is the minimization, for $x \in \boldsymbol{X}$, of $f(x)$ subject to $a_i \leq g_i(x) \leq b_i$, with $(a_i, b_i) \in \boldsymbol{R}^2$, for $1 \leq i \leq p$. The general ideas are exactly the same as before. At step n, we choose the next evaluation point X_{n+1} as the maximizer of the expected improvement. However, if the constraints g_i are expensive to evaluate as well, we can also consider them as Gaussian random processes G_i. Let G_n^i denote the set $(X_1, G_i(X_1), X_2, G_i(X_2), \ldots, X_n, G_i(X_n))$. The only difference is that the expected improvement (EI) criterion is now subjected to the constraints. Let $\bar{\rho}_n$ denote this criterion, $\bar{\rho}_n(x) = E_n(I_n(x) | X_{n+1} = x)$, where $I_n(x)$ equals to $((m_n - \xi(X_{n+1}))_+$ if $a_i \leq G_i(x) \leq b_i$ is verified for all i values and equals 0 otherwise. Without going further into details (see [10]), it can be shown that if $f, g_1, g_2, \ldots, g_p$ are all statistically independent, this constrained EI criterion can be written as follows:

$$\bar{\rho}_n(x) = \rho_n(x) \prod_{i=1}^{p} P(a_i \le G_i(x) \le b_i | G_n^i), \qquad (3)$$

This criterion is simply the product of the unconstrained EI (2) with the probability that each constraint is verified. This probability is easy to compute because, for $x \in \boldsymbol{X}$, the random process $G_i(x) | G_n^i$ is a Gaussian process, i.e., this constrained EI criterion can also be computed with a limited amount of resources.

C. Modification of the criterion for the nonlinearity constraint

In practice, with respect to the application considered below, we notice that a modification of the algorithm can improve, from far away, the convergence to the minimum. Indeed, the nonlinearity constraint is a strong constraint; thus, the regions of the domain satisfying it are the most promising. In other words, most of the budget of the evaluations should be consumed in these specific parts of the domain. However, an important property of the EI criterion is avoiding convergence to a local minimum because of an exploration/exploitation tradeoff. This means that because of the exploration phase of the algorithm, many evaluation points are chosen in areas where the probability of verifying the nonlinearity constraint is very low. This problem could be solved by tuning the parameters of the Gaussian process (the mean and the parameters of the covariance function) in a more careful way (maximum likelihood estimation for the parameters is known for not always being relevant, see [12]). In this article, we used a different approach: for the nonlinearity constraint g_j, we decided to solve this problem by forcing the value of $\bar{\rho}_n(x)$ to 0 when $P\left(a_j \le G_j(x) \le b_j | G_n^j\right)$ is below a threshold α (usually $\alpha = 0.05$).

IV. Application: Transimpedance Amplifier

The presented methodology is applied to the design of a 6th order CT Sigma-Delta modulator based on the architecture presented in [2]. A simple TIA application is considered, but all other amplifiers were treated using the proposed approach.

A differential TIA is used in the ΣΔ filter, composed of two single-ended amplifiers. The schematic of the single-ended structure is shown in Figure 2. It is implemented using a STMicroelectronics 65 nm LPGP CMOS technology process.

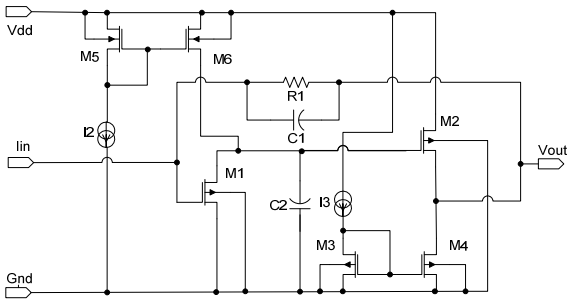


Figure 2: Single-ended transimpedance amplifier

Transistors M1 and M2 and resistor R1 act as a current-controlled voltage source. The source's gain is equal to R1. The pairs M3-M4 and M5-M6 and the current generators I2 and I3 are real current sources used to bias the main transistors M1 and M2. When this amplifier is connected to a high capacitance, it may become unstable, and capacitances C1 and C2 help stabilize the amplifier.

The optimization objectives and constraints for this circuit are summarized in Table 1.

Parameter	Type	Desired Value(s)	Others
Input impedance (Z_{in})	Objective	Minimum	-
Output impedance (Z_{out})	Constraint	[5 ; 10] [Ω]	Normal
Direct gain (g^d)≈ R_1(5000 Ω)	Constraint	[4900 ; 5100] [Ω]	Normal
Input Nonlinearity (nl^{in})	Constraint	<1%	Strong
Output Nonlinearity (nl^{out})	Constraint	<1%	Strong

Table 1: Transimpedance amplifier optimization goals

The initial current sources values are determined by manual DC considerations, while the initial transistor sizes are sufficiently large to minimize the saturation voltages drain-source. Then, the automatic optimization process is launched. Table 2 lists the variation range of the design variables.

Design Variable	Variation range	Max. number of points
I2	[20; 80] µA	25
I3	[20; 160] µA	60
w_i/w_{min}	[1; 60] ; w_i→[0.065; 3.9] µm	60
l_i/l_{min}	[1; 30] ; l_i→[0.065; 1.95] µm	30
w_i/w_{min}	[1; 60] ; w_i→[0.065; 3.9] µm	30
l_i/l_{min}	[1; 30] ; l_i→[0.065; 1.95] µm	15

Table 2: Variables variation; w_{min}= l_{min}=65 nm; i=1,2; j=3-6

The starting points used to compute the metamodel are sampled to create a Latin Hypercube sample (LHS) in order to benefit from its specific space-filling proprieties (see [9] and references therein). Then, the algorithm starts choosing optimization points according to the constrained EI criterion. In the case of the TIA amplifier, a total of 80 evaluation points are selected (40 starting points and 40 optimization points).

An interesting measure of the algorithm's performance is the temporal evolution of the optimization goals. Figure 3 depicts the variation of three figures of merit with the advancement of the optimization: the optimization objective (Z_{in}), a normal constraint (Z_{out}) and a strong constraint (nl^{in}). For the starting points domain, because the points are arbitrarily chosen, the goals also have arbitrary values. As soon as the metamodel is designed and optimization points are considered, the goals start to converge towards the desired values (e.g. the *Objective minimum evolution* reaches points where smaller values are obtained with the optimization advancement). The non-uniform variation of the goals results from the algorithm exploring different regions of the design variables domain, where the probability of finding "better" candidate points is higher.

The optimum point is found at $I2$=27.5µA, $I3$=145µA, w_1/w_{min}=28, l_1/l_{min}=27, w_2/w_{min}=19, l_2/l_{min}=6, w_3/w_{min}=w_4/w_{min} =24, l_3/l_{min}=l_4/l_{min}=10, w_5/w_{min}=w_6/w_{min}=4, l_5/l_{min}= l_6/l_{min}=2. For this, we obtained the following optimization results: Z_{in}=66.42Ω, Z_{out}=6.5Ω, g^d=4926Ω, nl^{in}=0.15%, nl^{out}=0.1%.

In terms of computation time, the optimization of an amplifier can take up to 5-10 hours, depending on the machine speed, the design complexity and the restrictions on the goals, but all is automatic and independent of the designer effort.

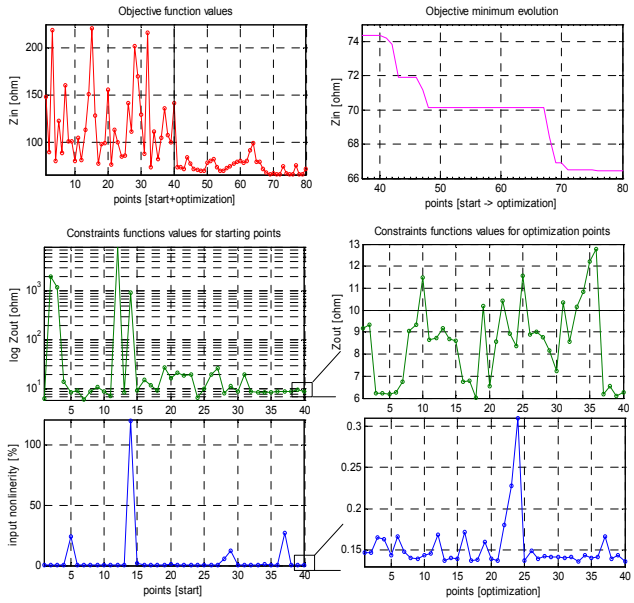


Figure 3: Temporal evolution of the optimization goals

At the end of the optimization, the model of the amplifier can be extracted as a SIMULINK (Figure 4) or VHDL-AMS/VerilogA module. This includes the direct and reverse transfer functions, the input and output impedances, computed as minimum-order stable s-functions and all offsets [6].

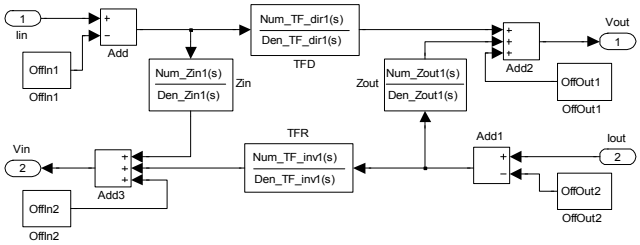


Figure 4: SIMULINK linear macro-model of the amplifier

Using the extracted macro-models, high-level ΣΔ implementations were conducted for different orders of models (2-6). MATLAB-SIMULINK and Dolphin Integration SMASH were used as simulators. System-level performance characteristics were evaluated to validate the methodology. As an example, in Figure 5 the direct transfer function of the ΣΔ filter is compared between the transistor-level implementation (CADENCE Spectre AC simulation) and the high-level ones (SIMULINK and VHDL-AMS order 2 macro-models simulations). The maximum total relative errors normalized to the number of points between the transistor-level and macro-models characteristics were used to predict the accuracy of the models. We obtained errors in $[10^{-8}; 10^{-6}]$ for 2^{nd}-6^{th} order models [6]. An important aspect of the methodology resides in the speed improvement for transient analyses: factors of 10×-30× were obtained for specific components and the whole ΣΔ modulator. Table 3 shows the comparative computation times needed for 1000 output samples in the two cases.

Simulation / Design	Transistor-level (CADENCE Spectre)	Macro-models (SIMULINK)	Behavioral (SMASH)
TIA	15.7s	1.5s	1.2s
ΣΔ 6^{th} order	3h37m1s	7m8s	6m41s

Table 3: Computation time for transient analyses (TIA, ΣΔ)

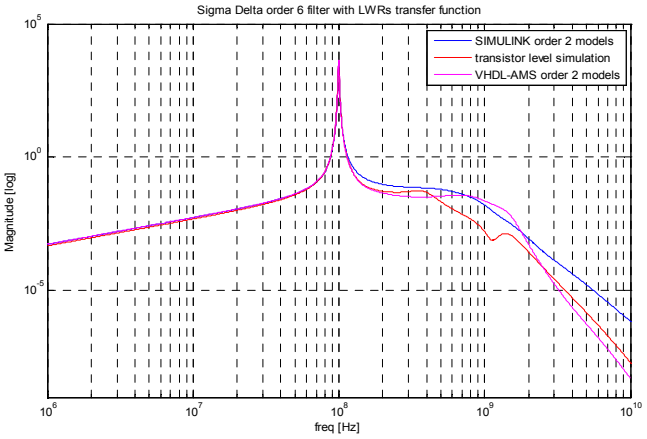


Figure 5: Sigma-Delta filter direct transfer function

V. CONCLUSIONS

We proposed a refined design methodology for weakly non-linear CT functions. This methodology is based on the optimization process of transistor-level cells using a Kriging metamodel and the synthesis of robust high-level analog behavioral models, which can be used for fast system-level simulations. Automatic optimizations were performed for the components of a sixth-order CT Sigma-Delta modulator, reducing the effort required for development and assuring coherent results at the system level. The proposed algorithms were integrated in a more general MATLAB-CADENCE framework, which can be used for the design and optimization of complex CT functions [4].

REFERENCES

[1] Gielen, G.G.E., « Design methodologies and tools for circuit design in CMOS nanometer technologies », *The 32nd European Solid-State Circuits Conference, 2006 (ESSCIRC 2006).*

[2] Javidan, M., « Design of high-order sigma-delta modulators for parallel analog-to-digital converters », PhD thesis, SUPELEC, France, 2009.

[3] Bernardo, M.C.; Buck, R.; Liu, L.; Nazaret, W.A.; Sacks, J.; Welch, W.J., « Integrated circuit design optimization using a sequential strategy », IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 3:361–372, Mar 1992.

[4] *http://www.mathworks.com/matlabcentral/fileexchange/26525*

[5] Benabes, P.; Tugui, C.-A., « A high-level modeling framework for the design and optimization of complex CT functions », *NEWCAS 2011,* Bordeaux, France, Jun 2011.

[6] Benabes, P.; Tugui, C.-A., « Effective Modeling of CT Functions for Fast Simulations Using MATLAB-Simulink and VHDL-AMS Applied to Sigma-Delta Architectures », *IEEE International Symposium on Circuits and Systems (ISCAS 2011).*

[7] Rogenmoser, R..; Kaeslin, H.; Blickle, T., « Stochastic Methods for Transistor Size Optimization of CMOS VLSI Circuits », Proceedings of *PPSN IV 1996,* Springer-Verlag London, UK.

[8] Egea, J. A.; « New heuristics for global optimization of complex bioprocesses », PhD thesis, University of Vigo, 2008

[9] Jones, D.R.; Schonlau, M.; William, J, « Efficient global optimization of expensive black-box functions», J. Global Optim., 13:455-492, 1998.

[10] Schonlau, M., «Computer Experiments and Global Optimization», PhD thesis, University of Waterloo, 1997.

[11] Lophaven, S. N.; Nielsen, H. B.; Sondergaard, J., « DACE: a MATLAB Kriging toolbox », Technical Report IMM-REP-2002-12, Technical University of Denmark, 2002.

[12] Jones, D. R.; « A taxonomy of global optimization methods based on response surfaces ». J. Global Optim., 21(4):345-383, 2001.