INVITED PAPER

Hierarchical Modeling, Optimization, and Synthesis for System-Level Analog and RF Designs

Small models, that represent the overall functioning of portions of large or complex circuits, can be generated by algorithms and used for system design and verification.

By Rob A. Rutenbar, *Fellow IEEE*, Georges G. E. Gielen, *Fellow IEEE*, and Jaijeet Roychowdhury, *Senior Member IEEE*

ABSTRACT | The paper describes the recent state of the art in hierarchical analog synthesis, with a strong emphasis on associated techniques for computer-aided model generation and optimization. Over the past decade, analog design automation has progressed to the point where there are industrially useful and commercially available tools at the cell level—tools for analog components with 10–100 devices. Automated techniques for device sizing, for layout, and for basic statistical centering have been successfully deployed. However, successful component-level tools do not scale trivially to system-level applications. While a typical analog circuit may require only 100 devices, a typical system such as a phase-locked loop, data converter, or RF front-end might assemble a few hundred such circuits, and comprise 10 000 devices or more. And unlike purely digital systems, mixed-signal designs typically need to optimize dozens of competing continuous-valued performance specifications, which depend on the circuit designer's abilities to successfully exploit a range of nonlinear behaviors across levels of abstraction from devices to circuits to systems. For purposes of synthesis or verification, these designs are not tractable when considered "flat." These designs must be approached with hierarchical tools that deal with the system's intrinsic design hierarchy. This paper surveys recent advances in analog design tools that specifically deal with the hierarchical nature of practical analog and RF systems. We begin with a detailed survey of algorithmic techniques for automatically extracting a suitable nonlinear macromodel from a device-level circuit. Such techniques are critical to both verification and synthesis activities for complex systems. We then survey recent ideas in hierarchical synthesis for analog systems and focus in particular on numerical techniques for handling the large number of degrees of freedom in these designs and for exploring the space of performance tradeoffs early in the design process. Finally, we briefly touch on recent ideas for accommodating models of statistical manufacturing variations in these tools and flows.

KEYWORDS | Computer-aided design; integrated circuits; modeling; simulation

I. INTRODUCTION

Automated analog integrated circuit design is becoming a viable solution for increasing design productivity for critical analog components. Over the past decade, analog design automation has progressed to the point where there are industrially useful and commercially available tools at the cell level—tools for analog components with 10–100 devices. Automated techniques for device sizing, for layout, and for basic statistical centering have been successfully deployed. Gielen and Rutenbar [2] offer a fairly complete survey of the area. However, successful component-level tools do not scale trivially to system-level

Manuscript received May 3, 2006.
R. A. Rutenbar is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890 USA (e-mail: rutenbar@ece.cmu.edu).
G. G. E. Gielen is with ESAT-MICAS, Katholieke Universiteit Leuven, 3001 Leuven, Belgium (e-mail: gielen@esat.kuleuven.be).
J. Roychowdhury is with the Electrical and Computer Engineering Department, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: jr@umn.edu).
Digital Object Identifier: 10.1109/JPROC.2006.889371

applications. While a typical analog circuit may require only 100 devices, a typical system such as a phase-locked loop, a data converter, or an entire RF front-end might assemble a few hundred such circuits and comprise 10 000 devices or more. And unlike purely digital systems, mixed-signal designs typically need to optimize dozens of competing continuous-valued performance specifications, which depend on the circuit designer's abilities to successfully exploit a range of nonlinear behaviors across levels of abstraction from devices to circuits to systems. Such designs are also increasingly being integrated in large system-on-chip (SOC) environments, in challenging (i.e., highly scaled digital CMOS [1]) fabrication processes. Verifying, optimizing, and synthesizing such complex systems is generally not tractable when they are considered "flat." These designs must be approached with hierarchical tools that deal with the system's intrinsic design hierarchy. In any hierarchical approach macromodels or behavioral models are the bridge between the different levels in the hierarchy, both top-down and/or bottom-up.

This paper surveys recent advances in analog design tools that specifically deal with the hierarchical nature of practical analog and RF systems. We begin in Section II with a detailed survey of algorithmic techniques for automatically extracting a suitable nonlinear macromodel from a device-level circuit. Such techniques are critical to both verification and synthesis activities for complex systems; they allow us to simulate large systems, in practical amounts of time, with just enough fidelity for the behaviors we seek to model. We then survey in Section III recent ideas in hierarchical synthesis and optimization for analog systems. We focus in particular on numerical techniques for handling the large number of degrees of freedom in these designs and for exploring the space of performance tradeoffs early in the design process. Finally, in Section IV we briefly touch on recent ideas for accommodating models of statistical manufacturing variations in these tools and flows. Section V offers concluding remarks.

II. ALGORITHMIC MACROMODELING TECHNIQUES

Electronic systems today, especially those for communications and sensing, are typically composed of a complex mix of digital, analog, and RF circuit blocks. Simulating or verifying such systems is critical for discovering and correcting problems prior to fabrication, in order to avoid refabrication which is typically very expensive. Simulating entire systems to the extent needed to generate confidence in the correctness of the to-be-fabricated product is, however, also usually very challenging in terms of computation time. Full transistor-level verification of large systems can simply be prohibitive in terms of CPU time.

A common and useful approach towards verification in such situations, both during early system design and after detailed block design and layout, is to replace large and/or complex blocks by small macromodels that replicate their input–output functionality well and to verify the macromodeled system. The macromodeled system can be simulated rapidly in order to evaluate different choices of design-space parameters. Such a macromodel-based verification process affords circuit, system, and architecture designers considerable flexibility and convenience through the design process, especially if performed hierarchically using macromodels of differing sizes and fidelity.

The key issue in the above methodology is, of course, the creation of macromodels that represent the blocks of the system well. This is a challenging task for the wide variety of communication and other circuit blocks in use today. The most prevalent approach towards creating macromodels is manual abstraction. Macromodels are usually created by the same person who designs the original block, often aided by simulations. While this is the only feasible approach today for many complex blocks, it does have a number of disadvantages compared to the automated alternatives that are described in this paper. Simulation often does not provide abstracted parameters of interest directly (such as poles, residues, modulation factors, etc.); obtaining them by manual postprocessing of simulation results is inconvenient, computationally expensive, and error prone. Manual structural abstraction of a block can easily miss the very nonidealities or interactions that detailed verification is meant to discover. With semiconductor device dimensions shrinking below 100 nm and nonidealities (such as substrate/interconnect coupling, degraded device characteristics, etc.) becoming increasingly critical, the fidelity of manually generated macromodels to the real subsystems to be fabricated eventually is becoming increasingly suspect. Adequate incorporation of nonidealities into behavioral models, if at all possible by hand, is typically complex and laborious. Generally speaking, manual macromodeling is heuristic, time consuming, and highly reliant on detailed internal knowledge of the block under consideration, which is often unavailable when subsystems that are not designed in-house are utilized. As a result, the potential time-to-market improvement via macromodel-based verification can be substantially negated by the time and resources needed to first generate the macromodels.

It is in this context that there has been considerable interest in automated techniques for the creation of macromodels. Such techniques take a detailed description of a block—for example, a SPICE-level circuit netlist—and generate, via an automated computational procedure, a much smaller macromodel. The macromodel, fundamentally a small system of equations, is usually translated into Matlab/Simulink form for use at the system level. Such an automated approach, i.e., one that remains sustainable as devices shrink from deep submicron to nanoscale, is essential for realistic exploration of the design space in

current and future communication circuits and other system applications.

Several broad methodologies for automated macromodeling have been proposed. One is to generalize, abstract, and automate the manual macromodeling process. For example, common topological elements in a circuit are recognized, approximated, and conglomerated (e.g., [18] and [19]) to create a macromodel. Another class of approaches attempts to generate symbolic macromodels that capture the system's input–output relationship, e.g., [20]–[25]. Yet another class (e.g., [26]–[28]) employs a black-box methodology. Data is collected via many simulations or measurements of the full system and a regression-based model is created that can predict outputs from inputs. Various methods are available for the regression, including data mining, multidimensional tables, neural networks, genetic algorithms, etc.

In this paper, we focus on another methodology for macromodeling, often termed *algorithmic*. Algorithmic macromodeling methods approach the problem as the transformation of a large set of mathematical equations to a much smaller one. The principal advantage of these methods is generality—as long as the equations of the original system are available numerically (e.g., from within SPICE), knowledge of the circuit structure, operating principles, etc., is not critical. A single algorithmic method may therefore apply to entire classes of physical systems, encompassing circuits and functionalities that may be very disparate. Four such classes, namely *linear time-invariant* (LTI), *linear time-varying* (LTV), *nonlinear (nonoscillatory)*, and *oscillatory* are discussed in this section. Algorithmic methods also tend to be more rigorous about important issues such as fidelity and stability, and often provide better guarantees of such characteristics than other methods.

A. Macromodeling LTI Systems

Often referred to as *reduced-order modeling* (ROM) or *model-order reduction* (MOR), automated model generation methods for LTI systems are the most mature among algorithmic macromodeling methods. Any block composed of resistors, capacitors, inductors, linear controlled sources, and distributed interconnect models is LTI (often referred to simply as "linear"). The development of LTI MOR methods has been driven largely by the need to "compress" the huge interconnect networks, such as clock distribution nets, that arise in large digital circuits and systems. Replacing these networks by small macromodels makes it feasible to complete accurate timing simulations of digital systems at reasonable computational expense. Although interconnect-centric applications have been the main domain for LTI reduction, it is appropriate for any system that is linear and time invariant. For example, "linear amplifiers," i.e., linearizations of mixed-signal amplifier blocks, are good candidates for LTI MOR methods.

Fig. 1 depicts the basic structure of an LTI block. $u(t)$ represents the inputs to the system, and $y(t)$ the outputs in the time domain; in the Laplace (or frequency) domain their transforms are $U(s)$ and $Y(s)$, respectively. The definitive property of any LTI system [29] is that the input and output are related by convolution with an impulse response $h(t)$ in the time domain, i.e., $y(t) = x(t) * h(t)$. Equivalently, their transforms are related by multiplication with a system transfer function $H(s)$, i.e., $Y(s) = H(s)X(s)$. Note that there may be many internal nodes or variables within the block. The goal of LTI MOR methods is to replace the block by one with far fewer internal variables, yet with an acceptably similar impulse response or transfer function.

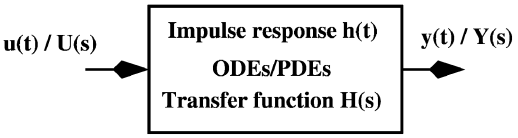


Fig. 1. *LTI block.*

In the majority of circuit applications, the LTI block is described to the MOR method as a set of differential equations, i.e.,

$$\begin{aligned} E\dot{x} &= Ax(t) + Bu(t) \\ y(t) &= C^T x(t) + Du(t). \end{aligned} \tag{1}$$

In (1), $u(t)$ represents the input waveforms to the block and $y(t)$ the outputs. Both are relatively few in number compared to the size of $x(t)$, the state of the internal variables of the block. A, B, C, D, and E are constant matrices. Such differential equations can be easily formed from SPICE netlists or AHDL descriptions, especially for interconnect applications, the dimension n of $x(t)$ can be very large.

The first issue in LTI ROM is to determine what aspect of the transfer function of the original system should be retained by the reduced system; in other words, what metric of fidelity is appropriate. In their seminal 1990 paper [30], Pileggi and Rohrer used moments of the transfer function as fidelity metrics, to be preserved by the model reduction process. The moments mi of an LTI transfer function $H(s)$ are related to its derivatives, i.e.,

$$m_1 = \left.\frac{dH(s)}{ds}\right|_{s=s_0}, \quad m_2 = \left.\frac{d^2H(s)}{ds^2}\right|_{s=s_0}, \cdots, \tag{2}$$

where s_0 is a frequency point of interest. Moments can be shown to be related to practically useful metrics, such as delay in interconnects.

In [30], Pileggi and Rohrer proposed a technique, asymptotic waveform evaluation (AWE) for constructing a reduced model for the system (2). AWE first computes a

number of moments of the full system (2), then uses these in another set of linear equations, the solution of which results in the reduced model. Such a procedure is termed explicit moment matching. The key property of AWE was that it could be shown to produce reduced models whose first several moments (at a given frequency point s_0) were identical to those of the full system. The computation involved in forming the reduced model was roughly linear in the size of the (large) original system. While explicit moment matching via AWE proved valuable and was quickly applied to interconnect reduction, it was also observed to become numerically inaccurate as the size of the reduced model increased beyond about ten. To alleviate these, variations based on matching moments at multiple frequency points were proposed [31] that improved numerical accuracy. Nevertheless, the fundamental issue of numerical inaccuracy, as reduced model sizes grew, remained.

In 1994, Gallivan *et al.* [32]–[34] identified the reason for this numerical inaccuracy. Computing the kth moment explicitly involves evaluating terms of the form $A^{-k}r$, i.e., the kth member of the Krylov subspace of A and r. If A has well-separated eigenvalues (as it typically does for circuit matrices), then for $k \sim 10$ and above, only the dominant eigenvalue contributes to these terms, with nondominant ones receding into numerical insignificance. Furthermore, even with the moments available accurately, the procedure of finding the reduced model is also poorly conditioned. Recognizing that these are not limitations fundamental to the goal of model reduction, the authors of [32] and [34] proposed alternatives. They showed that numerically robust procedures for computing Krylov subspaces, such as the Lanczos and Arnoldi (e.g., [35]) methods, could be used to produce reduced models that match any given number of moments of the full system. These approaches, called Krylov-subspace MOR techniques, do not compute the moments of the full system explicitly at any point, i.e., they perform implicit moment matching. In addition to matching moments in the spirit of AWE, Krylov-subspace methods were also shown to capture well the dominant poles and residues of the system. The Padé-via-Lanczos (PVL) technique [34] gained rapid acceptance within the MOR community by demonstrating its numerical robustness in reducing the DEC Alpha chip's clock distribution network.

Krylov-subspace methods are best viewed as reducing the system (1) via projection [36]. They produce two projection matrices, V and W^T, such that the reduced system is obtained as

$$\begin{aligned} \underbrace{W^T E}_{\hat{E}}\,\dot{x} &= \underbrace{W^T A V}_{\hat{A}}\,x(t) + \underbrace{W^T B}_{\hat{B}}\,u(t) \\ y(t) &= \underbrace{C^T V}_{\hat{C}^T}\,x(t) + Du(t). \end{aligned} \tag{3}$$

For the reduction to be practically meaningful, q, the size of the reduced system, must be much smaller than n, the size of the original. If the Lanczos process is used, then $W^T V \sim I$ (i.e., the two projection bases are bi-orthogonal). If the Arnoldi process is applied, then $W = V$ and $W^T V = I$.

The development of Krylov-subspace projection methods marked an important milestone in LTI macromodeling. However, reduced models produced by both AWE and Krylov methods retained the possibility of violating passivity or even being unstable. A system is passive if it cannot generate energy under any circumstances; it is stable if for any bounded inputs, its response remains bounded. In LTI circuit applications, passivity guarantees stability. Passivity is a natural characteristic of many LTI networks, especially interconnect networks. It is essential that reduced models of these networks also be passive, since the converse implies that under some situation of connectivity, the reduced system will become unstable and diverge unboundedly from the response of the original system.

The issue of stability of reduced models was recognized early in [32], and the superiority of Krylov-subspace methods over AWE in this regard also noted. Silveira *et al.* [37] proposed a coordinate transformed Arnoldi method that guaranteed stability, but not passivity. Kerns *et al.* [38] proposed reduction of admittance-matrix-based systems by applying a series of nonsquare congruence transformations. Such transformations preserve passivity properties while also retaining important poles of the system. However, this approach does not guarantee matching of system moments. A symmetric version of PVL with improved passivity and stability properties was proposed by Freund and Feldmann in 1996 [39].

The passivity-retaining properties of congruence transformations were incorporated within Arnoldi-based reduction methods for *RLC* networks by Odabasioglu *et al.* in 1997 [40], [41], resulting in an algorithm dubbed PRIMA (Passive Reduced-Order Interconnect Macromodeling algorithm). By exploiting the structure of *RLC* network matrices, PRIMA was able to preserve passivity and match moments. Methods for Lanczos-based passivity preservation [42], [43] followed.

All the above LTI MOR methods, based on Krylov-subspace computations, are efficient (i.e., approximately linear-time) for reducing large systems. The reduced models produced by Krylov-subspace reduction methods are not, however, optimal, i.e., they do not necessarily minimize the error for a macromodel of given size. The theory of balanced realizations, well known in the areas of linear systems and control, provides a framework in which this optimality can be evaluated. LTI reduced-order modeling methods based on truncated balanced realizations (TBR) (e.g., [44] and [45]) have been proposed. Balanced realizations are a canonical form for linear differential equation systems that "balance" controllability

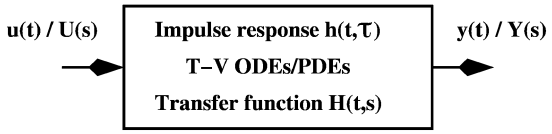


Fig. 2. ***LTV block.***

and observability properties. While balanced realizations are attractive in that they produce more compact macromodels for a given accuracy, the process of generating the macromodels is computationally very expensive, i.e., cubic in the size of the original system. However, recent methods [46] that combine Krylov-subspace techniques with TBR methods have been successful in approaching the improved compactness of TBR, while substantially retaining the attractive computational cost of Krylov methods.

B. Macromodeling LTV Systems

LTI macromodeling methods, while valuable tools in their domain, are inapplicable to many functional blocks in mixed-signal systems, which are usually nonlinear in nature. For example, distortion or clipping in amplifiers, switching, and sampling behavior, etc., cannot be captured by LTI models. In general, generating macromodels for nonlinear systems, as we shall see later, is a difficult task.

However, a class of nonlinear circuits (including RF mixing, switched-capacitor, and sampling circuits) can be usefully modelled as LTV systems. The key difference between LTV systems and LTI ones is that if the input to an LTV system is time shifted, it does not necessarily result in the same time shift of the output. The system remains linear, in the sense that if the input is scaled, the output scales similarly. This latter property holds, at least ideally, for the input-to-output relationship of circuits such as mixers or samplers. It is the effect of a separate local oscillator or clock signal in the circuit, independent of the signal input, that confers the time-varying property. This is intuitive for sampling circuits, where a time shift of the input, relative to the clock, can be easily seen not to result in the same time shift of the original output—simply because the clock edge samples a different time sample of the input signal. In the frequency domain, more appropriate for mixers, it is the time-varying nature that confers the key property of frequency shifting (upconversion or downconversion) of input signals. The time-varying nature of the system can be "strongly nonlinear," with devices switching on and off—this does not impact the linearity of the signal input-to-output path.

Fig. 2 depicts the basic structure of an LTV system block. Similar to LTI systems, LTV systems can also be completely characterized by impulse responses or transfer functions; however, these are now functions of two variables, the first capturing the time variation of the system and the second the changes of the input [29]. The detailed behavior of the system is described using time-varying differential equations, e.g.,

$$\begin{aligned} E(t)\dot{x} &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)^T x(t) + D(t)u(t). \end{aligned} \tag{4}$$

Time variation in the system is captured by the dependence of A, B, C, D, and E on t. In many cases of practical interest, this time variation is periodic. For example, in mixers, the local oscillator input is often a sine or a square wave; switched or clocked systems are driven by periodic clocks.

The goal of macromodeling LTV systems is similar to that for LTI ones: to replace (4) by a system identical in form, but with the state vector $x(t)$ much smaller in dimension than the original. Again, the key requirement is to retain meaningful correspondence between the transfer functions of the original and reduced systems.

Because of the time variation of the impulse response and transfer function, LTI MOR methods cannot directly be applied to LTV systems. However, Roychowdhury [47]–[49] showed that LTI model reduction techniques can be applied to LTV systems, by first reformulating (4) as an LTI system similar to (1), but with extra artificial inputs that capture the time variation. The reformulation first separates the input and system time variations explicitly using multiple time scales [50] in order to obtain an operator expression for $H(t, s)$. This expression is then evaluated using periodic steady-state methods [51]–[53] to obtain an LTI system with extra artificial inputs. Once this LTI system is reduced to a smaller one using any LTI MOR technique, the reduced LTI system is reformulated back into the LTV system form (4). The use of different LTI MOR methods within this framework has been demonstrated, including explicit moment matching [47] and Krylov-subspace methods [48], [49], [54]. Moreover, Phillips [54] showed that the LTV-to-LTI reformulation could be performed using standard linear system theory concepts [29], without the use of multiple time scales.

C. Macromodeling Nonoscillatory Nonlinear Systems

While wires, interconnect, and passive lumped elements are purely linear, any mixed-signal circuit block containing semiconductor devices is nonlinear. Nonlinearity is, in fact, a fundamental feature of any block that provides signal gain or performs any function more complex than linear filtering. Even though linear approximations of many nonlinear blocks are central to their design and intended operation, it is usually important to consider the impact of nonlinearities with a view to limiting their impact. For example, in "linear" amplifiers and mixers, distortion and intermodulation, caused solely

by nonlinearities, must typically be guaranteed not to exceed a very small fraction of the output of the linearized system. This is especially true for traditional RF and microwave designs. Such weakly nonlinear systems comprise an important class of blocks that can benefit from macromodeling. Additionally, many nonlinear blocks of interest are not designed to be approximately linear in operation. Examples include digital gates, switches, comparators, etc., which are intended to switch abruptly between two states. While such operation is obviously natural for purely digital systems, strongly nonlinear behavior is also exploited in analog blocks such as sampling circuits, switching mixers, analog-to-digital converters, etc. Furthermore, oscillators and PLLs, which are common and basic components in mixed-signal systems, exhibit complex dynamics which are fundamentally strongly nonlinear.

Unlike for the classes of linear systems considered in the previous sections, no technique currently exists that is capable, even in principle, of producing a macromodel that conforms to any reasonable fidelity metric for completely general nonlinear systems. The difficulty stems from the fact that nonlinear systems are richly varied, with extremely complex dynamical behavior possible that is very far from being exhaustively investigated or understood. This is in contrast to linear dynamical systems, for which comprehensive mathematical theories exist (see, e.g., [29]) that are universally applicable. In view of the diversity and complexity of nonlinear systems in general, it is difficult to conceive of a single overarching theory or method that can be employed for effective macromodeling of an arbitrary nonlinear block. It is not surprising, therefore, that macromodeling of nonlinear systems has tended to be manual, relying heavily on domain-specific knowledge for specialized circuit classes, such as ADCs, phase detectors, etc.

In recent years, however, linear macromodeling methods have been extended to handle weakly nonlinear systems. Other techniques based on piecewise approximations have also been devised that are applicable to some strongly nonlinear systems. As described below in more detail, these approaches start from a general nonlinear differential equation description of the full system, but first approximate it to a more restrictive form, which is then reduced to yield a macromodel of the same form. The starting point is a set of nonlinear differential-algebraic equations (DAEs) of the form

$$\begin{aligned} \dot{q}(x(t)) &= f(x(t)) + bu(t) \\ y(t) &= c^T x(t) \end{aligned} \tag{5}$$

where $f(\cdot)$ and $q(\cdot)$ are nonlinear vector functions.

1) Polynomial-Based Weakly Nonlinear Methods: To appreciate the basic principles behind weakly nonlinear macromodeling, it is first necessary to understand how the full system can be treated if the nonlinearities in (6) are approximated by low-order polynomials. The polynomial approximation concept is simply an extension of linearization, with $f(x)$ and $q(x)$ replaced by the first few terms of a Taylor series about an expansion point x_0 (typically the dc solution); for example

$$f(x) = f(x_0) + A_1(x - x_0) + A_2(x - x_0)^{②} + \cdots \tag{6}$$

where a^i represents the Kronecker product of a with itself i times. When (6) and its $q(\cdot)$ counterpart are used in (5), a system of polynomial differential equations results. If $q(x) = x$ (assumed for simplicity), these equations are of the form

$$\begin{aligned} \dot{x}(t) &= f(x_0) + A_1(x - x_0) + A_2(x - x_0)^{②} + \cdots + bu(t) \\ y(t) &= c^T x(t). \end{aligned} \tag{7}$$

The utility of this polynomial system is that it becomes possible to leverage an existing body of knowledge on weakly polynomial differential equation systems, i.e., systems where the higher order nonlinear terms in (6) are small compared to the linear term. In particular, Volterra series theory [55] and weakly nonlinear perturbation techniques [56] justify a relaxation-like approach for such systems, which proceeds as follows. First, the response of the linear system, ignoring higher order polynomial terms, is computed—denote this response by $x_1(t)$. Next, $x_1(t)$ is inserted into the quadratic term $A_2(x - x_0)^2$ (denoted a distortion input), the original input is substituted by this waveform, and the linear system solved again to obtain a perturbation due to the quadratic term—denote this by $x_2(t)$. The sum of x_1 and x_2 is then substituted into the cubic term to obtain another weak perturbation, the linear system solved again for $x_3(t)$, and so on. The final solution is the sum of x_1, x_2, x_3, and so on. An attractive feature of this approach is that the perturbations x_2, x_3, etc., which are available separately in this approach, correspond to quantities like distortion and intermodulation which are of interest in design. Note that at every stage, to compute the perturbation response, a linear system is solved—nonlinearities are captured via the distortion inputs to these systems.

The basic idea behind macromodeling of weakly nonlinear systems is to exploit this fact; in other words, to apply linear macromodeling techniques, appropriately modified to account for distortion inputs, to each stage of the relaxation process above. In the first such approach, proposed in 1999 by Roychowdhury [49], the linear system is first reduced by LTI MOR methods to a system of size q_1, as shown in Fig. 3, via a projection basis obtained using Krylov-subspace methods. The distortion inputs for the

quadratic perturbation system are then expressed in terms of the reduced state vector of the linear term, to obtain an input vector of size q_1^2. The quadratic perturbation system (which has the same linear system matrix, but a different input vector) is then again reduced via another projection basis, to size q_2. This process is continued for higher order terms. The overall reduced model is the union of the separate reduced models with outputs summed together, as depicted in Fig. 3. By tailoring the projection bases for each nonlinearly perturbed linear system, this approach focusses on accuracy; however, this is achieved at the cost of increased macromodel size $q1 + q2 + \cdots$. Recognizing the size issue, Phillips in 2000 [57], [58] proposed that a single projection basis be applied to the system (8) (analogous to LTI MOR systems) and also observed that Carlemann bilinearization [59] could be employed to obtain a canonical equation form. Intuitively, the use of a single projection basis consolidates the commonality in the three reduced models shown in Fig. 3, leading to smaller overall models.

In 2003, Li and Pileggi proposed the NORM method [60], which combines and extends the above two approaches. Similar to [49], NORM generates tailored projection bases for each perturbed linear system, but instead of retaining separate macromodels as in Fig. 3, it compresses these projection bases into a single projection basis. NORM then employs this single projection basis to reduce the system (6) as proposed in [57]. A particularly attractive property of NORM is that it produces a macromodel that matches a number of multidimensional moments of the Volterra series kernels [55] of the system—indeed, the distortion terms for each perturbed system are pruned to ensure matching of a specified number of moments. The authors of NORM also include a variant that matches moments at multiple frequency points. Fig. 4 shows an example of a NORM modeling result for the third-order intermodulation product for a double-balanced RF mixer.

2) Piecewise Approximation Methods: The polynomial approximations discussed above are excellent when the intended operation of the system exercises only weak nonlinearities, as in power amplifiers, "linear" mixers, etc. Outside a relatively small range of validity, however, polynomials are well known to be extremely poor global approximators. This limitation is illustrated in Fig. 5, where it can be seen that, outside a local region where there is a good match, even a sixth-degree Taylor-series approximation diverges dramatically from the function it is meant to represent.

It is for this reason that other ways of approximating (5) that have better global approximation properties than polynomials have been sought. One approach is to represent the nonlinear functions $f(\cdot)$ and $q(\cdot)$ in (5) by piecewise linear (PWL) segments. The state space is split into a number of disjoint regions, and within each region, a linear approximation is used that matches the nonlinear function approximately within the region. By using a sufficiently large number of regions, the nonlinear function can be represented accurately over the entire

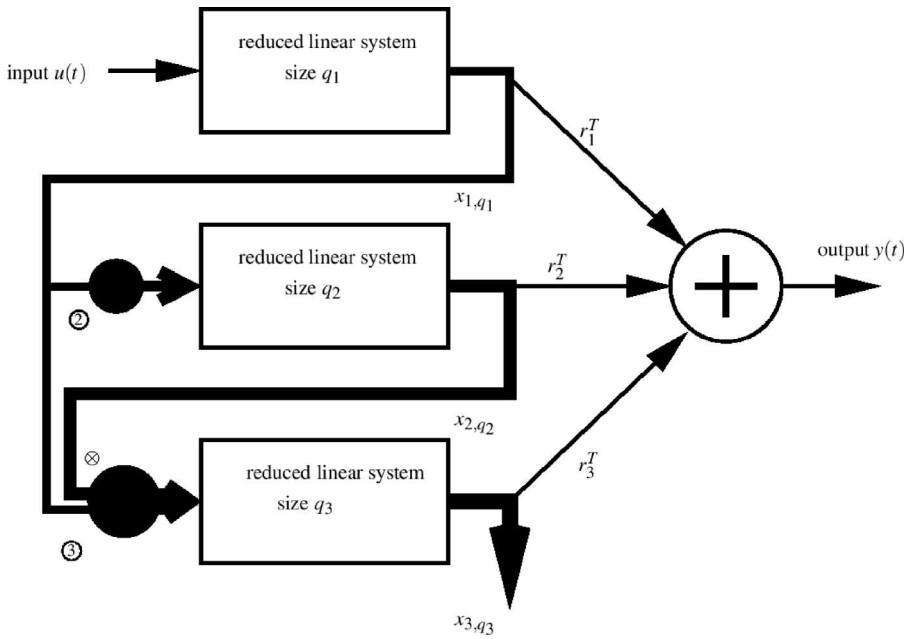


Fig. 3. ***Overall reduced-order model for a weakly nonlinear system.***

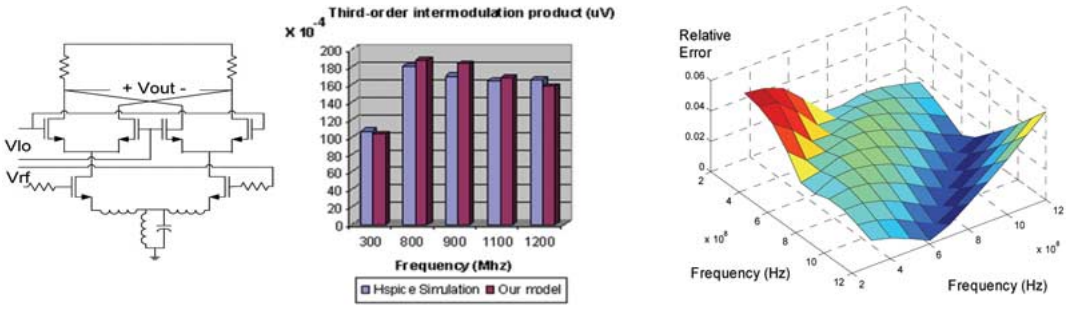


Fig. 4. ***NORM [60] modeling example. Double-balanced mixed (left) with extracted model for third-order intermodulation product (middle and right). Example shows a maximum error of 8% over all RF frequencies.***

domain of interest. From a macromodeling perspective, the motivation for PWL approximations is that since the system is linear within each region, linear macromodeling methods can be leveraged.

Piecewise linear approximations are not new in circuit simulation, having been employed in the past most notably in attempts to solve the dc operating point problem [61], [62]. One concern with these methods is a potential exponential explosion in the number of regions as the dimension of the state space grows. This is especially the case when each elemental device within the circuit is first represented in piecewise form, and the system of circuit equations is constructed from these piecewise elements. A combinatorial growth of polytope regions results via cross-products of the hyperplanes that demarcate piecewise regions within individual devices. To circumvent the explosion of regions, which would severely limit the simplicity of a small macromodel, Rewienski and White proposed the Trajectory PWL method (TPWL) [63] in 2001. In TPWL, a reasonable number of "center points" is first selected along a simulation trajectory in the state space, generated by exciting the circuit with a representative training input. Around each center point, system nonlinearities are approximated by linearization, with the region of validity of the linearization defined implicitly, as consisting of all points that are closer to the given center point than to any other. Thus, there are only as many piecewise regions as center points, and combinatorial explosion resulting from intersections of hyperplanes is avoided. The implicit piecewise regions in TPWL are in fact identical to the Voronoi regions defined by the collection of center points chosen. Within each piecewise region, the TPWL approach simply reduces the linear system using existing LTI MOR methods to obtain a reduced linear model. The reduced linear models of all the piecewise regions are finally stitched together using a scalar weight function to form a single-piece reduced model. The weight function identifies, using a closest-distance metric, whether a test point in the state space is within a particular piecewise region and weighs the corresponding reduced linear system appropriately.

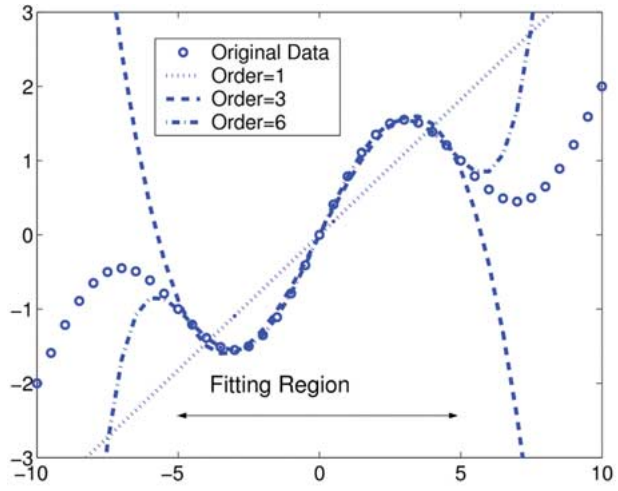


Fig. 5. ***Limitations of global polynomial approximations.***

The TPWL method, by virtue of its use of inherently better PWL global approximation, avoids the blow-up that occurs when polynomial-based methods are used with large inputs. It is thus better suited for circuits with strong nonlinearities, such as comparators, digital gates, etc. However, because PWL approximations do not capture higher order derivative information, TPWL's ability to reproduce small-signal distortion or intermodulation is limited.

To address this limitation, Dong and Roychowdhury proposed a piecewise polynomial (PWP) extension [64] of TPWL in 2003. PWP combines weakly nonlinear MOR techniques with the piecewise idea, by approximating the nonlinear function in each piecewise region by a polynomial, rather than a purely linear, Taylor expansion. Each piecewise polynomial region is reduced using one of the polynomial MOR methods outlined above, and the

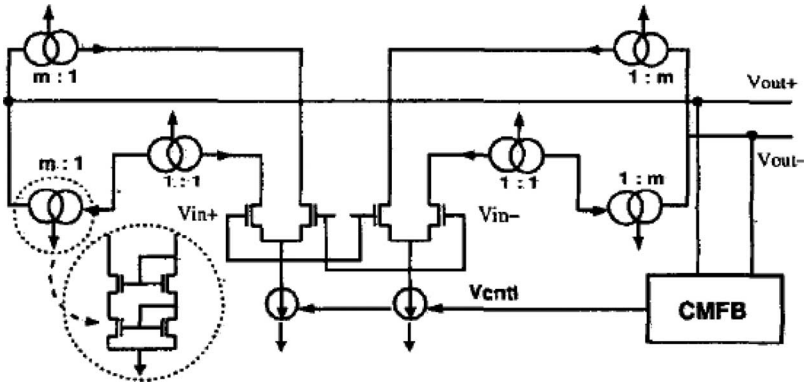


Fig. 6. *Current-mirror op-amp with 50 MOSFETs and 39 nodes.*

resulting polynomial reduced models are stitched together with a scalar weight function, similar to TPWL. Thanks to its piecewise nature, PWP is able to handle strong nonlinearities globally; because of its use of local Taylor expansions in each region, it is also able to capture small-signal distortion and intermodulation well. Thus, PWP expands the scope of applicability of nonlinear macromodeling to encompass blocks in which strong and weak nonlinearities both play an important functional role.

PWP is illustrated using the fully differential op-amp shown in Fig. 6. The circuit comprises 50 MOSFETs and 39 nodes. It was designed to provide about 70 dB of dc gain, with a slew rate of 20 V/μs and an open-loop 3-dB-bandwidth of $f_0 \sim 10$ kHz. The PWP-generated macromodel was of size 19, a number that refers to the size of the differential equation system describing the macromodel (i.e., roughly analogous to the number of nodes in the circuit). The macromodel is compared against the full SPICE-level op-amp using a number of analyses and performance metrics, representative of actual use in a real industrial design flow. Fig. 7 shows the results of performing dc sweep analyses of both the original circuit and the PWP-generated macromodel. Note the excellent match. Fig. 8 compares Bode plots obtained by ac analysis; two ac sweeps, obtained at different dc bias points, are shown. Note that PWP provides excellent matches around each bias point.

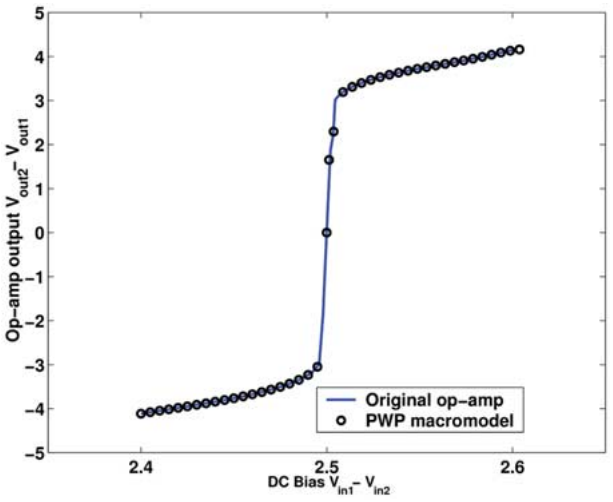


Fig. 7. *DC sweep of op-amp from Fig. 6.*

If the op-amp is used as a linear amplifier with small inputs, distortion and intermodulation are important performance metrics. As mentioned earlier, one of the strengths of PWP-generated macromodels is that weak nonlinearities, responsible for distortion and intermodulation, are captured well. Such weakly nonlinear effects are best simulated using frequency-domain harmonic balance (HB) analysis, for which we choose the one-tone sinusoidal input $V_{in1} - V_{in2} = A\sin(2\pi \times 100t)$ and $C_{load} = 10$ pF. The input magnitude A is swept over several decades and the first two harmonics plotted in Fig. 9. It can be seen that for the entire input range, there is an excellent match of the distortion component from the macromodel versus that of the full circuit. Note that the same macromodel is used for this harmonic balance simulation as for all the other analyses presented. Speedups of about 8.1× were obtained compared to the harmonic balance simulations.

Another strength of PWP is that it can capture the effects of strong nonlinearities excited by large-signal swings. To demonstrate this, a transient analysis was run with a large, rapidly rising input; the resulting waveforms are shown in Fig. 10. The slope of the input was chosen to excite slew-rate limiting, a dynamical phenomenon caused by strong nonlinearities (saturation of differential amplifier structures). Note the excellent match between the original circuit and the macromodel. The macromodel-based simulation ran about 8× faster.

Unfortunately, two problems are shared by all the piecewise trajectory-based modeling techniques, both the

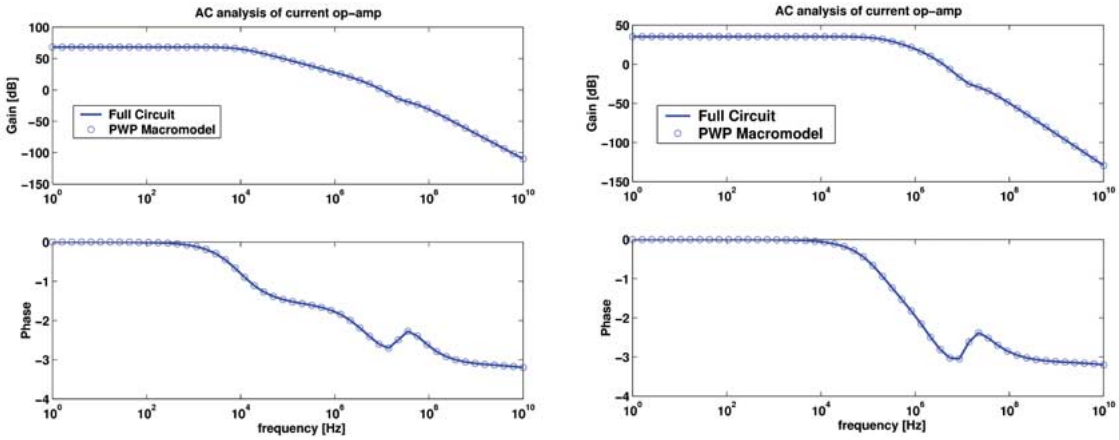


Fig. 8. *AC analysis of op-amp from Fig. 6: (left) AC sweep (at common-mode dc bias 2.5 V) and (right) AC sweep (at common-mode dc bias 2.0 V).*

original TPWL linear form [63] and the generalized quadratic PWP extension of [64]. As noted in Tiwary and Rutenbar [85], all the trajectory methods are fragile in the face of insufficient training waveforms to build the essential trajectory "coverage" of the state space. One can remedy this with a more rigorous training regime, of course. However, this creates a new problem: simulation inefficiency. Trajectory models interpolate from all the visited linearizations to estimate the behavior at any new point in the state space. Curing the undertraining problem can easily increase the number of trajectory samples from tens or hundreds to 10 000 or more and destroy any simulation speedup gains we sought to achieve. The same authors demonstrate a solution, the so-called *scalable* trajectory model [85]. Using ideas from data mining, they show how to cluster trajectory linearizations in the state space, prune away redundant linearizations, and synthesize estimated linearizations to represent each cluster of similar points. Then, instead of interpolating based on all visited trajectory points, they use a fast high-dimensional nearest neighbor lookup strategy in the trajectory space to select only the "relevant" linearizations to use to predict the local dynamics. The strategy allows the designer to train the model with a wide variety of potentially useful waveforms but enables the model to interpolate from a minimal set of the most locally relevant linearizations, preserving most of the speedup gains we originally sought. Fig. 11 shows an example from [85] of a simple common-mode feedback (CMFB) op-amp replaced in a sample-and-hold amplifier context with a scalable trajectory model.

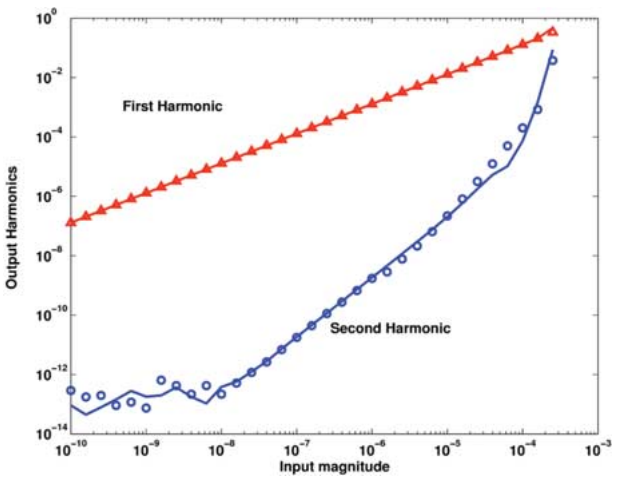


Fig. 9. *Harmonic analysis of current-mirror op-amp: full op-amp (solid line) and PWP model (discrete points).*

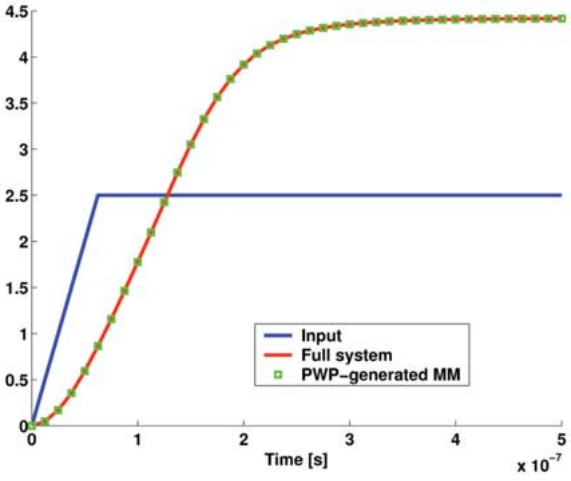


Fig. 10. *Transient analysis of current-mirror op-amp with fast step input.*

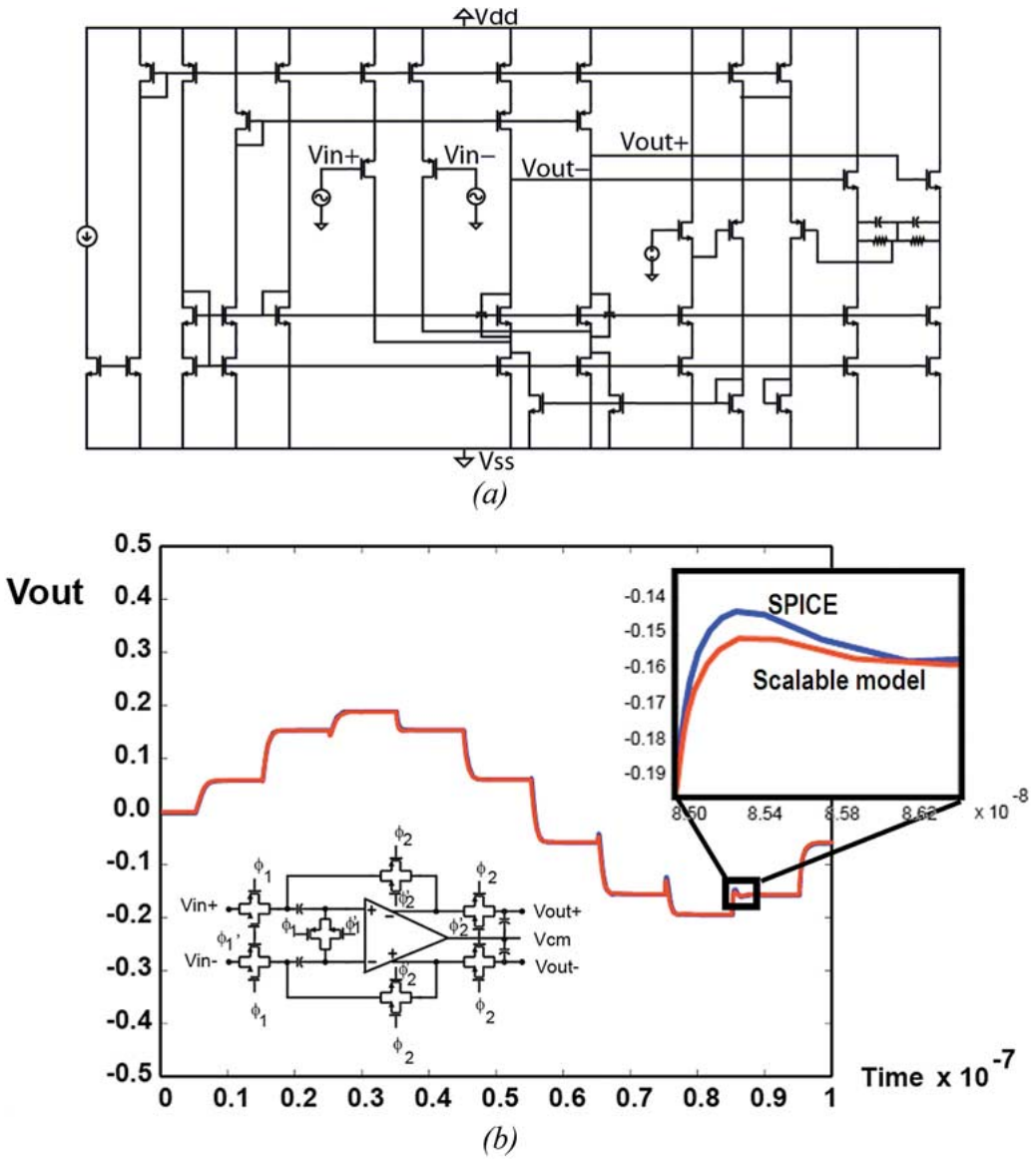


Fig. 11. *Example of scalable trajectory model from [85]. (a) CMFB op-amp with 40 MOSFETs, 24 nodes. (b) SPICE versus model accuracy for a scalable trajectory model of CMFB op-amp, hierarchically reinserted into sample-and-hold design.*

Additional work on the scalable model [86] showed how to output loading effects and apply additional model-order reductions to obtain faster models which can be successfully reinserted in an arbitrary system-level design.

D. Macromodeling Oscillatory Systems

Oscillators are ubiquitous in electronic systems. They generate periodic signals, typically sinusoidal or square-like waveforms, that are used for a variety of purposes. From the standpoint of both simulation and macromodeling, oscillators present special challenges. Traditional circuit simulators such as SPICE [65], [66] consume significant computer time to simulate the transient behavior of oscillators. As a result, specialized techniques based on using phase macromodels (e.g., [67]–[76]) have been developed for the simulation of oscillator-based systems. The most basic class of phase macromodels assumes a linear relationship between input perturbations and the output phase of an oscillator. A general time-varying expression for the phase $\phi(t)$ can be given by

$$\phi(t) = \sum_{k=1}^{n} \int_{-\infty}^{\infty} h_{\phi}^{k}(t,\tau) i_k(\tau) d\tau. \tag{8}$$

The summation is over all perturbations i_k to the circuit; $h_\phi^k(t,\tau)$ denotes a time-varying impulse response to the kth noise source. Very frequently, time-invariant simplifications of (8) are employed [77]. Linear models suffer, however, from a number of important deficiencies. In particular, they have been shown to be inadequate for capturing fundamentally nonlinear effects such as injection locking [82]. As a result, automatically extracted nonlinear phase models have recently been proposed [78], [79], [82]–[84] that are considerably more accurate than linear ones. The nonlinear phase macromodel has the form

$$\dot{\alpha}(t) = v_1^T(t + \alpha(t)) \cdot b(t). \tag{9}$$

In the above equation, $v_1(t)$ is called the perturbation projection vector (PPV) [79]; it is a periodic vector function of time, with the same period as that of the unperturbed oscillator. A key difference between the nonlinear phase model (9) and traditional linear phase models is the inclusion of the phase shift $\alpha(t)$ inside the perturbation projection vector $v_1(t)$. $\alpha(t)$ in the nonlinear phase model has units of time; the equivalent phase shift, in radians, can be obtained by multiplying $\alpha(t)$ by the free-running oscillation frequency ω_0.

We illustrate the use of (9) by applying it to model the VCO inside a simple PLL [80], shown in block form in Fig. 12. Using the nonlinear macromodel (9), we simulate the transient behavior of the PLL and compare the results with a full simulation and with linear models. The simulations encompass several important effects, including static phase offset, step response, and cycle slipping.

Fig. 13 depicts the static phase offset of the PLL when a reference signal of the same frequency as the VCO's free-running frequency is applied. The PLL is simulated to locked steady state and the phase difference between the reference and the VCO output is shown. The fact that the LPF is not a perfect one results in high-frequency ac components being fed to the VCO, affecting its static phase offset. Observe that both the full simulation and the nonlinear macromodel (9) predict identical static phase offsets of about 0.43 radians. Note also that the linear phase macromodel fails to capture this correctly, reporting a static phase offset of 0.

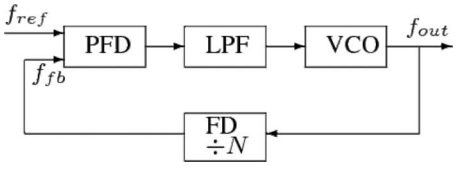


Fig. 12. ***Functional block diagram of PLL.***

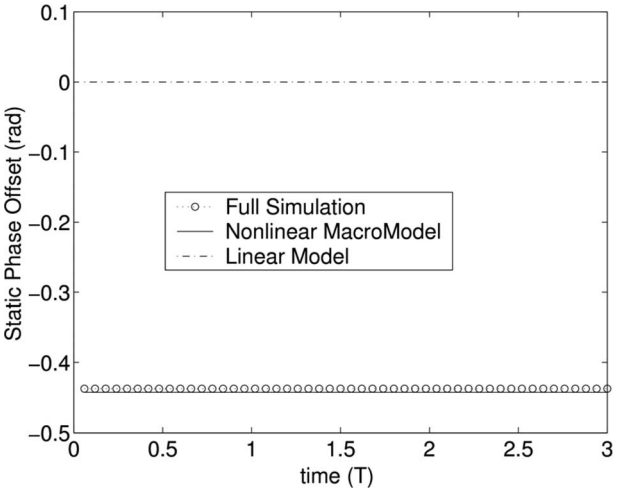


Fig. 13. ***PLL static phase offset when*** $f_{ref} = f_0$.

Fig. 14 depicts the step response of the PLL at different reference frequencies. Fig. 14(a) shows the step responses using the full simulation, the linear phase model, and the nonlinear macromodel when the reference frequency is $1.07f_0$. With this reference signal, both the linear and nonlinear macromodels track the reference frequency well, although, as expected, the nonlinear model provides a more accurate simulation than the linear one. When the reference frequency is increased to $1.074f_0$, however, the linear phase macromodel is unable to track the reference correctly, as shown in Fig. 14(b). However, the nonlinear macromodel remains accurate. The breakdown of the linear model is even more apparent when the reference frequency is increased to $1.083f_0$, at which the PLL is unable to achieve stable lock, as shown in Fig. 14(c). Note that the nonlinear macromodel remains accurate.

Finally, Fig. 15 illustrates the prediction of cycle slipping. A reference frequency $f_{ref} = 1.07f_0$ is provided and the PLL is brought to locked steady state. When a sinusoidal perturbation with amplitude 5 mA and duration 10 VCO periods is injected, the PLL loses lock. As shown in Fig. 15(a), the phase difference between the reference signal and the VCO output slips over many VCO cycles, until finally, lock is reachieved with a phase shift of -2π. Both nonlinear and linear macromodels predict the qualitative phenomenon correctly in this case, with the nonlinear macromodel matching the full simulation better than the linear one. When the injection amplitude is reduced to 3 mA, however, as shown in Fig. 15(b), the linear macromodel fails, still predicting a cycle slip. In reality, the PLL is able to recover without slipping a cycle, as predicted by both the nonlinear macromodel and the full simulation.

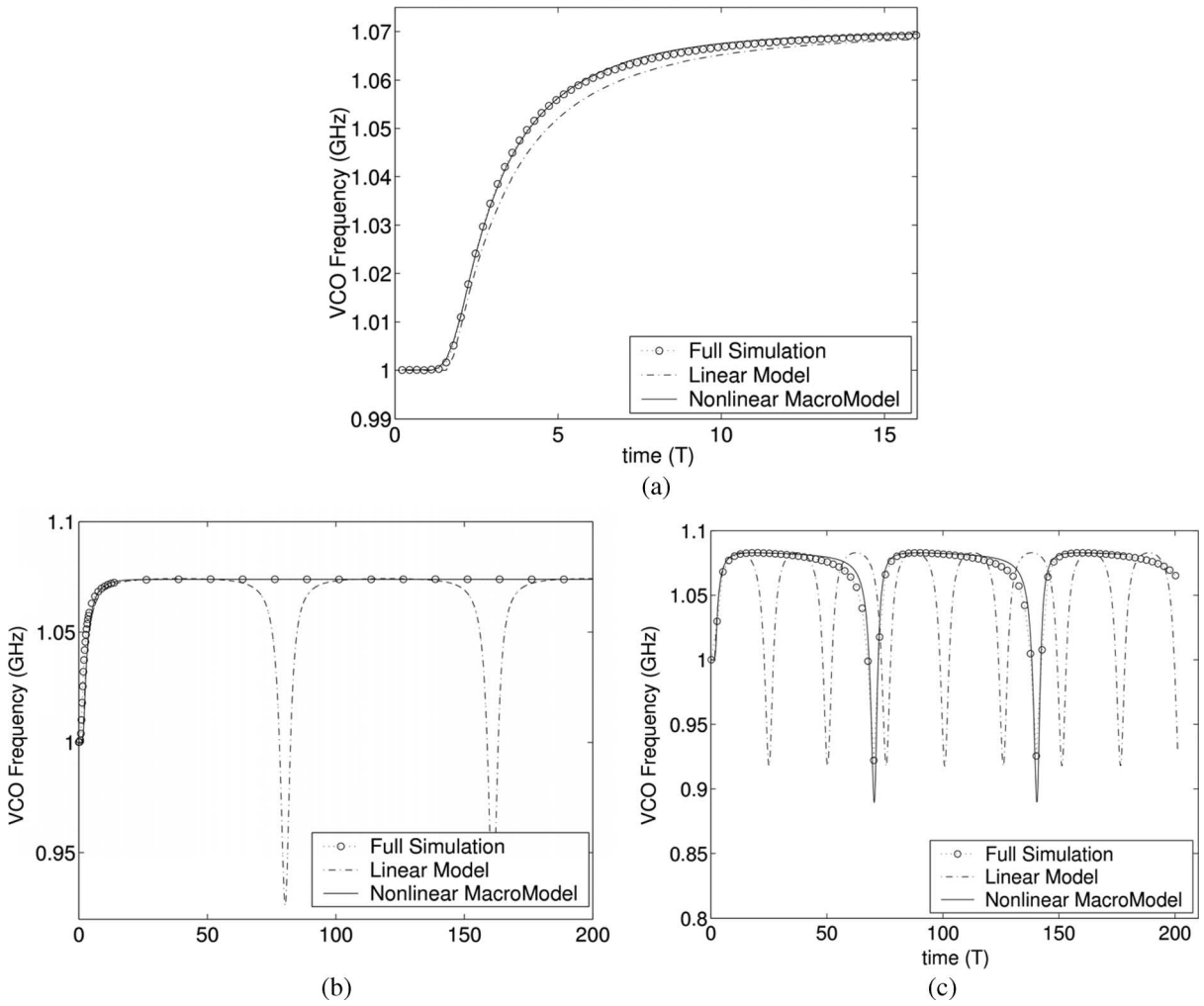


Fig. 14. ***Step response of PLL model with varying values of*** f_{ref}. ***(a) Response with*** $f_{ref} = 1.07f_O$. ***(b) Response with*** $f_{ref} = 1.074f_O$. ***(c) Response with*** $f_{ref} = 1.083f_O$.

E. Nonlinear Macromodelling: Current and Future Applicability

It should be noted that the problem of extracting nonlinear macromodels by algorithm is a very difficult problem. However, for future design sustainability, it is of great importance to solve this problem in practically useful ways. Of the methods surveyed above, the techniques for nonlinear oscillator macromodelling are the most mature at this point—the extraction techniques involved are being adopted by a number of CAD and semiconductor companies. The weakly nonlinear macromodelling methods discussed above are the most specialized, whereas the trajectory-piecewise strongly nonlinear macromodelling methods are perhaps the most broadly applicable. However, they are also, as already noted above, the most heuristically based technique of the three; it is expected that considerably more research will be required before these techniques achieve the reliability and robustness necessary for broad industrial deployment.

III. HIERARCHICAL SYNTHESIS TECHNIQUES

Recent years have seen the emergence of commercial CAD tool support for analog cell-level circuit and layout synthesis. Gielen and Rutenbar [2] offer a fairly complete survey of the area. Analog synthesis consists of two major steps: 1) circuit synthesis followed by 2) layout synthesis. Most of the basic techniques in both circuit and layout synthesis rely on powerful numerical optimization engines coupled to "evaluation engines" that qualify the merit of some evolving analog circuit or layout candidate. Fig. 16 shows the basic flow of most current tools. The goal of analog circuit synthesis is to create a sized circuit

schematic from given circuit specifications. It is a difficult and critical step for several reasons: 1) most analog circuits require a custom optimized design; 2) the design problem is typically underconstrained with many degrees of freedom; and 3) it is common that many (often conflicting) performance requirements must to be taken into account, and tradeoffs must be made that satisfy the designer using these tools.

We focus in this paper on the circuit, rather than layout synthesis. The former is a numerical problem that relies heavily on the macromodeling ideas of the previous section; the latter relies more on combinatorial optimization techniques (see [3]). Given a circuit schematic and the circuit's performance specifications, the sizes and biasing of all devices have to be determined such that the circuit meets the specifications at some optimal cost. It is the optimization engine that determines these optimal values, while the evaluation engine assesses the performance. In many cases, the initial sizing produces a near-optimal design that is further fine-tuned with a circuit optimization tool, e.g., to improve yield and design robustness. The performance of the resulting design is then verified using detailed circuit simulations with a simulator such as SPICE.

Although successful, these simulation-based circuit optimization methods still have to be used with care by designers because the run times (and therefore also the initial debug time) may still be long, especially since the optimizer may produce improper designs if the right design constraints are not added to the optimization problem. Reducing the CPU time, and the complexities of specifying the best set of design constraints, are active areas of research.

The flow of Fig. 16 presents several challenges when we try to scale these ideas from circuits to systems. Simulation-based synthesis (e.g., [87]–[90]) uses efficient numerical optimization to visit many circuit candidates and fully evaluates each candidate via detailed simulation. This methodology works very well for circuits having in the range of a few hundred devices. However, for larger circuits, the simulation time required for a single simulation is too large to do a practical simulator-in-the-loop circuit sizing. Also, due to the curse of dimensionality, the design space in which to search for optimal design points becomes too large for these circuits to be handled by these "flat" optimization-based tools.

We survey approaches to handling these larger system-level problems in this section. Targeting the explicitly hierarchical structure of these designs is of critical importance. Macromodeling also plays a central role, but it is perhaps surprising that the performance-oriented techniques of the previous section are a necessary but not yet sufficient response to these problems. In particular, we need a different sort of modeling abstraction, which builds *parameterized* tradeoff models, of the circuit components, for use in system-level analysis and synthesis.

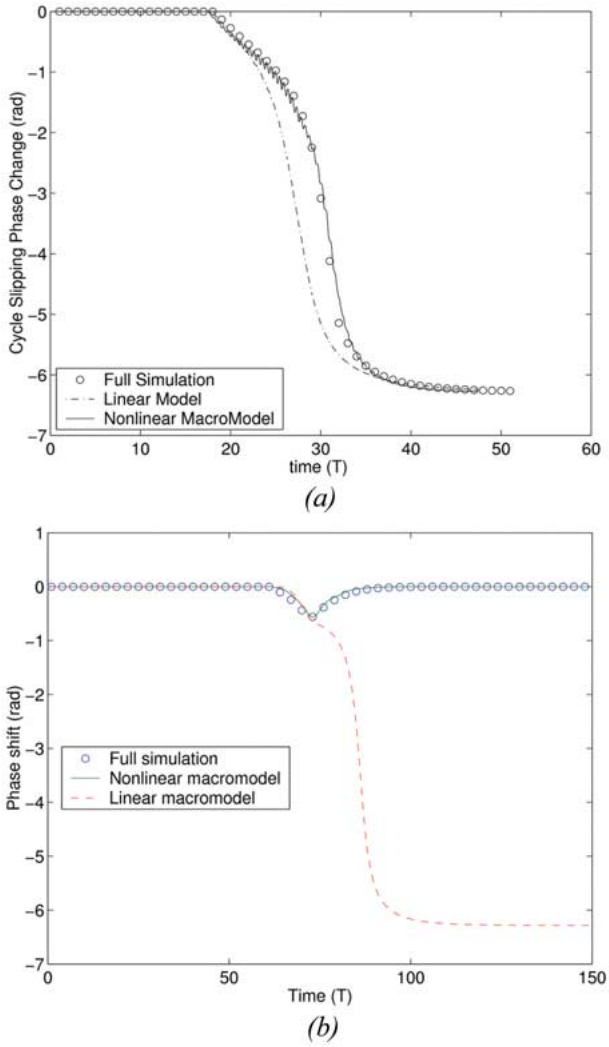


Fig. 15. *Cycle slipping of PLL model for different noise amplitudes. (a) PLL cycle slipping, for a noise amplitude of 5 mA. (b) PLL cycle slipping, for a noise amplitude of 3 mA.*

A. Macromodeling Revisited

One might be tempted to believe that the only problem with the optimization-based flow of Fig. 16 is the burdensome CPU time needed to evaluate each proposed solution candidate, since it relies on full device-level simulation for maximum fidelity and flexibility [87]. Thus, replacing some or all of the circuit level components with appropriate macromodels seems a good solution strategy. If we can make the necessary simulation evaluations run sufficiently fast, perhaps we can use these "flat" synthesis strategies without additional modification.

Unfortunately, the "instance oriented" model extraction techniques of Section II are a necessary but not sufficient solution to this problem. Let us consider a concrete example to explain the problem. Suppose we want to synthesize device-level sizing/biasing for a PLL.

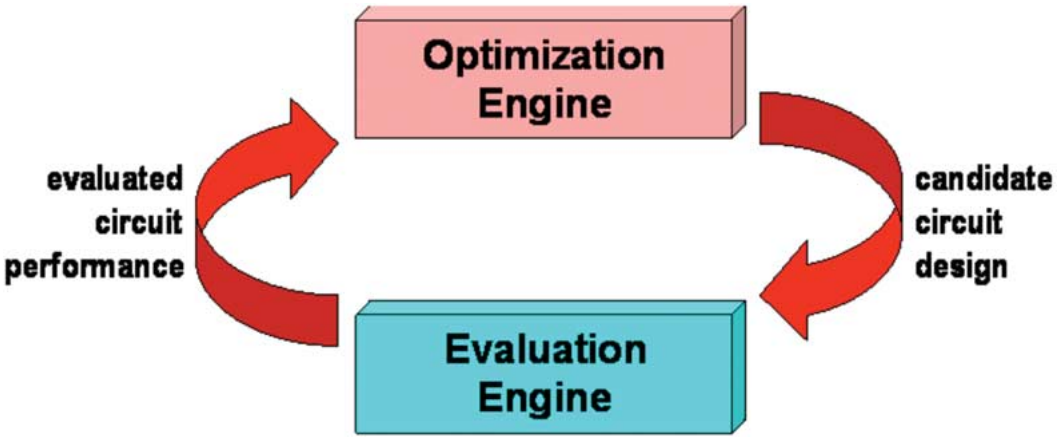


Fig. 16. *Basic flow of optimization-based analog circuit sizing.*

Fig. 12 shows a relatively generic system-level block diagram for a PLL. Synthesis in this context means determining all the device-level design details for all the individual blocks of this design: the phase frequency detector, the charge pump, loop filter, VCO, and divider. Simulating this design flat, at device level, in either the time domain or the frequency domain, may be expensive, i.e., hours not minutes of CPU time. We could mitigate this in several ways.

We could use an algorithmic modeling technique that targets this specific class of oscillatory systems, such as those we described in Section II-D. So, for example, we might create a fast model that is highly specific to the difficult VCO component in the PLL. We could also use any general algorithmic modeling technique and extract a general macromodel for any or all of the blocks. Fig. 17 shows an example from [86] in which a scalable trajectory model for a current-starved ring oscillator VCO is inserted into a simple 1 : 1 PLL.

If we are confident in our understanding of the essential behaviors of the PLL's constituent blocks (see again Fig. 11), we could use a nonalgorithmic model, i.e., a behavioral model that fits parameters to an expert-derived circuit "template" comprising a small set of equations which model only the essential dynamics of the circuit. For example, consider the requirements arising from a GSM-1800 design: frequency range around 1.8 GHz, phase noise -121 dBc/Hz @ 600 kHz frequency offset, and settling time of the loop for channel frequency changes below 1 ms within 1e-6 accuracy. Starting with a set of expert-derived top-down design models, using behavioral simulations with generic behavioral models for the subblocks, [8] shows how to derive the following characteristics for the PLL subblocks: $A_{\mathrm{LPF}} = 1$, $K_{\mathrm{VCO}} = 1e6$ Hz/V, $N_{\mathrm{div}} = 64$, $f_{\mathrm{LPF}} = 100$ kHz. These specifications are then the starting point for the device-level design of each of the subblocks.

For subsequent bottom-up system verification phase of a system, [8] also shows how to employ more detailed behavioral models that are tuned towards the actual circuit design. The authors suggest an accurate behavioral model for a designed VCO is given by the following equation set [8]:

$$\begin{aligned} v_{\mathrm{out}}(t) &= A_0(v_{\mathrm{in}}(t)) + \sum_{k=1}^{k=N} A_k(v_{\mathrm{in}}(t)).\sin(\Phi_k(t)) \\ \Phi_k(t) &= \varphi_k(v_{\mathrm{in}}(t)) \\ &\quad + 2\pi \int_{t_0}^{t} k.\left[h_{\mathrm{stat2dyn}}(\tau) \otimes f_{\mathrm{stat}}(v_{\mathrm{in}}(\tau))\right].d\tau \end{aligned} \quad (10)$$

where Φ_k is the phase of each harmonic k in the VCO output, A_k and φ_k characterize the (nonlinear) static characteristic of a VCO, and h_{stat2dyn} characterizes the dynamic voltage-phase behavior of a VCO, both as extracted from circuit-level simulations of the real circuit. Fig. 18 shows the resulting frequency response of both the original device-level circuit (red) and the extracted behavioral model (blue) for a low-frequency sinusoidal input signal. One can see that this input signal creates a side lobe near the carrier that is represented by the model within 0.25 dB accuracy compared to the original transistor-level circuit, while the gain in simulation time is more than 30× [8].

It would seem, then, that the only problem is which of these various macromodeling alternatives we should select. However, all the models we have presented so far have two fundamental characteristics.

1) *Instance oriented*: These models are extracted for *one* specific circuit that is fully designed at device

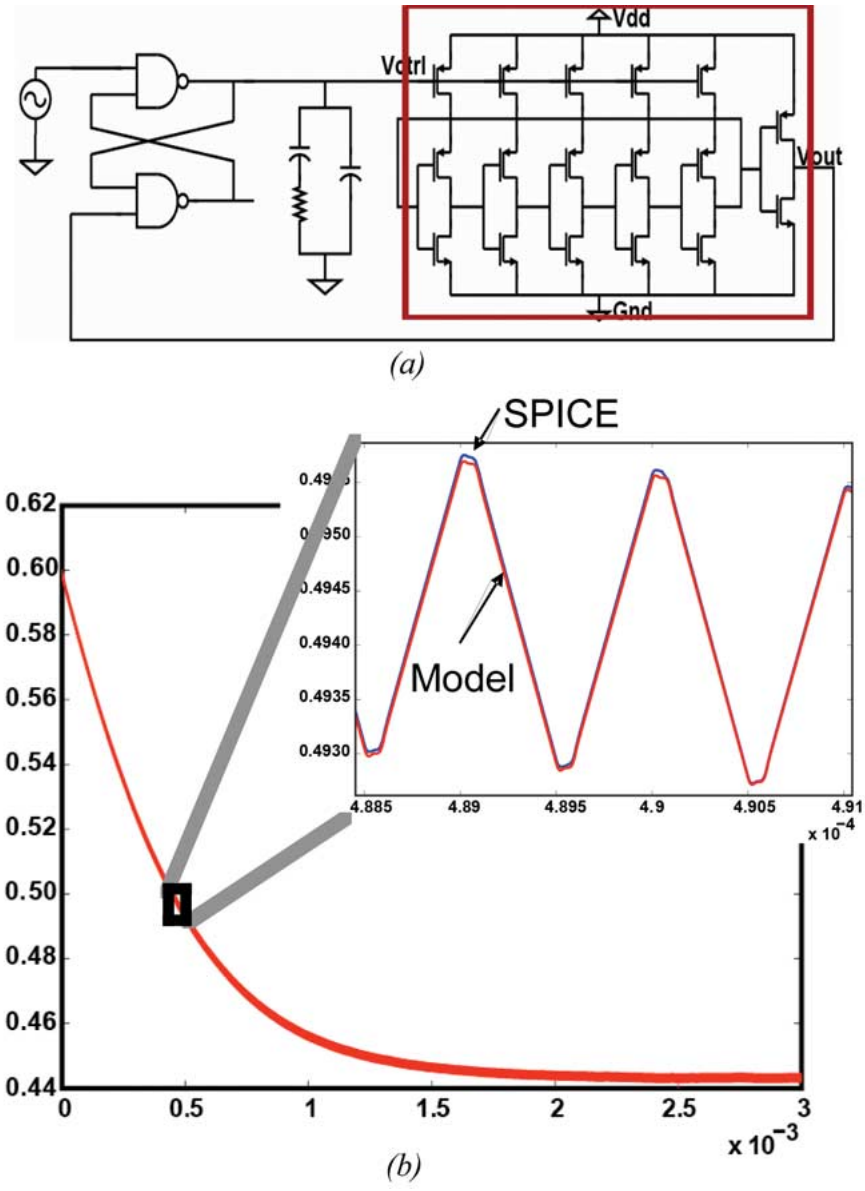


Fig. 17. ***Replacing the VCO in simple 1 : 1 PLL with scalable trajectory model. (a) Simple 1 : 1 PLL architecture with a current-starved ring-oscillator VCO (highlighted in bold). (b) SPICE versus scalable trajectory model simulation results; figure shows transient simulation result for PLL going into lock.***

level. If we change any of the device-level design variables, we must re-extract the model, which may be costly.

2) *Verification-oriented*: These instance-specific models are intended to be used in scenarios where we can amortize the cost of constructing the model over a large number of simulation runs that will be used to verify the correctness of the system-level design assembled from these models.

For synthesis tasks in which we plan to visit a large number of intermediate circuit design configurations, we really want a *parameterized* macromodel. This means a model that predicts the behavior of the circuit as a function of its designable parameters, e.g., transistor widths, lengths, biasing, etc. Since this is a much more challenging problem than extracting an instance-specific model, we usually are willing to accept some loss of model fidelity. For example, we may be willing to ignore some secondary or tertiary effects at this level, focus only on the essential nonidealities, and strive to repair these omissions when we do detailed circuit-level synthesis later, for each circuit.

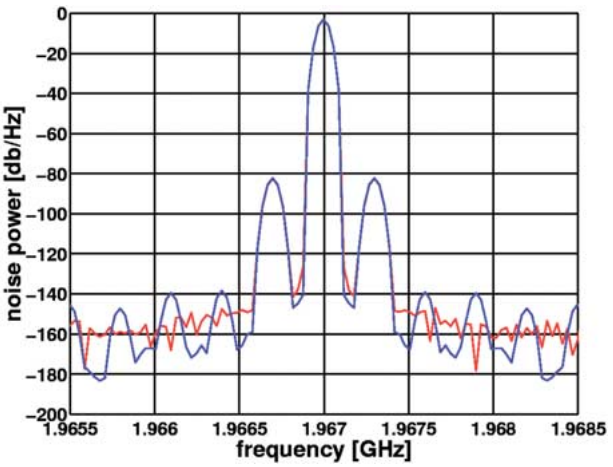


Fig. 18. *Frequency response of an extracted behavioral VCO model (blue) compared to the underlying device-level circuit response (red) [8].*

B. Parametric Circuit Performance Modeling Techniques

Parametric performance models—in contrast to the instance models described in earlier sections—relate the achievable performances of a circuit (e.g., gain, bandwidth, slew rate, or phase margin) to its design variables (e.g., device sizes and biasing). These are also sometimes called *performance space* or *design space models*. Fig. 19 for example shows part of such a parametric model, displaying the phase margin as a function of two design variables for a CMOS operational amplifier [9], [10]. Such performance models are used to speed up circuit sizing: in every iteration of the synthesis procedure of Fig. 16, calls to the transistor-level simulator are replaced by evaluations of a suitable constructed parametric model. This not only results in substantial speedups, once the performance models have been created and calibrated, but in many cases is the difference between a tractable and an intractable system-level synthesis process. The model building process is a one-time up-front investment that has to be done only once for each circuit in each technology.

Most approaches for performance model generation are based on fitting or regression methods where the coeffients of a prespecified model—often referred to as a *model template*—are fitted to have the model match as closely as possible a sample set of simulated data points. As a concrete example, consider the flow of Fig. 20. First, a large set of data samples is generated by simulating well chosen design points with SPICE. For instance, a design-of-experiments (DOE) scheme can be used to sample the design space. The use of SPICE simulations allows the modeling of any nonlinear circuits and circuit characteristics, as opposed to symbolic analysis techniques that are restricted to rather linear circuit characteristics only. Next, a model is fitted through these data points. If the modeling error is too large, then additional data points can be generated, or a more sophisticated model template has to be chosen.

A recent example of such a fitting approach is the automatic generation of posynomial performance models for analog circuits, that are created by fitting a pre-assumed posynomial equation template to simulation data created according to a design of experiments scheme [9]. Such a posynomial model could then, for instance, be used in the very efficient sizing of analog circuits through convex circuit optimization.

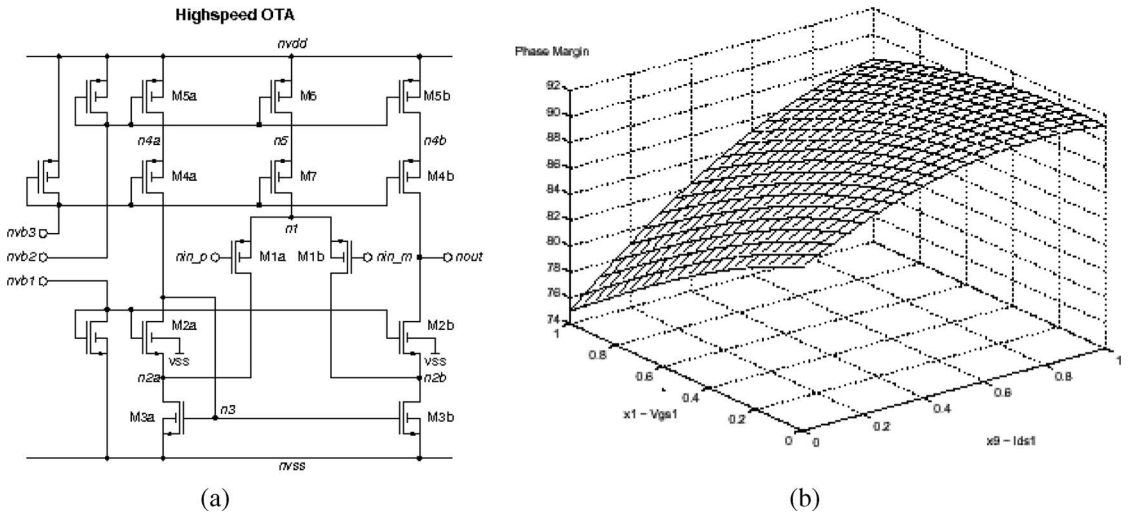


Fig. 19. *High-speed CMOS OTA (left) and performance model of phase margin as a function of two design variables (right). Note that this is just a subset of the actual multidimensional parametric performance model.*

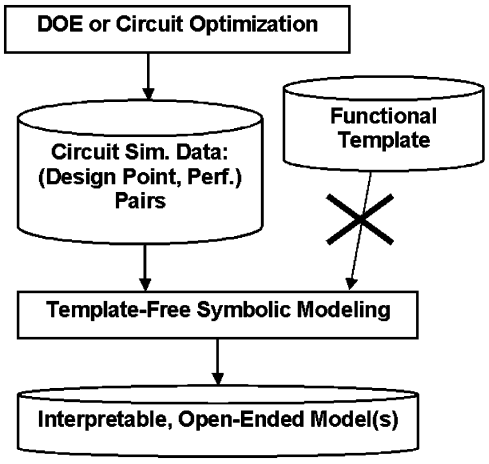


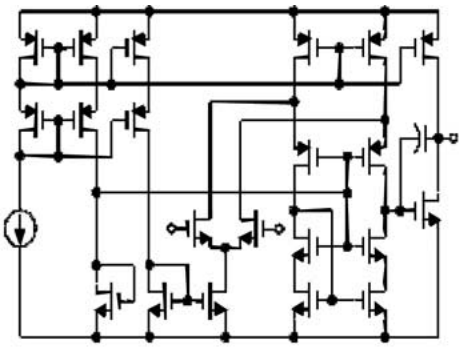
Fig. 20. *Flow for template-based and template-free performance modeling.*

One of the core problems with all regression style parameterized models is the need to balance the goodness-of-fit of the model against the complexity (i.e., numerical difficulty) of fitting the template coefficients necessary to complete the model. For example, a linear model of the circuit response is quite easy to fit, even for many independent variables. But, unless we really expect the behavior to be linear, it will likely be a poor fit to the circuit's actual behavior. The quadratic style posynomial models of [9] are one response to this problem. This particular model offers a wider range of nonlinearity, along with a workable heuristic, to calculate numerically the essential fitting parameters. The work in [91] takes this a step further and develops a very efficient fitting strategy for models of this type. One problem with higher dimensional nonlinear templates is that they create numerically challenging problems to solve for all the necessary fitting coefficients. Both [9] and [91] use a quadratic performance template for circuit performance $f(X)$

$$f(X) = X^T AX + B^T X + C \tag{11}$$

where $X = [x_1 \ \cdots \ x_N]^T$ is the vector of designable circuit parameters, A is an $N \times N$ matrix of coefficients, B is a N-vector of unknown coefficients, and C is a single unknown scalar coefficient. Our goal is find A, B, and C so that the quadratic function of vector X closely matches a large set of training data, e.g., sampled from many SPICE simulations. For large N, i.e., for a design with many degrees of freedom, we have $O(N^2)$ coefficients to solve for, which can be daunting. The methodology in [91], called ROAD, replaces the large unknown A matrix with a carefully chosen low-rank approximation; the technique employs the rank-one projection which can be solved for numerically via an efficient, implicit power iteration in $O(N)$ steps. Fig. 21 shows one example of parametric fitting using ROAD.

Another class of regression techniques borrows ideas from *data mining*, which focuses on extracting meaningful patterns (e.g., fitting predictive models) to large amounts of high-dimensional data (e.g., training a model by means of many SPICE runs). There is a range of useful techniques from this domain. One of the first large-scale applications of these ideas is the work of Liu *et al.* in [11]. A perennial problem in the area of parametric modeling is how one chooses the nonlinear template to which one will try to fit the desired circuit behavior. References [9] and [91] choose an explicit analytical form, a higher order quadratic of (11). Reference [11] supports fitting to a more nonlinear form by using a so-called *boosted community of regressors*. The idea is to iteratively fit a sequence of regression



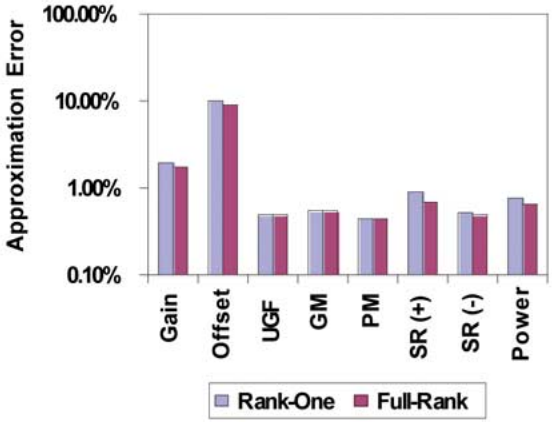


Fig. 21. *Applying ROAD [91] projection-based quadratic fitting procedure to 0.25-μm CMOS op-amp. Results show that low-rank approximation strategy is extremely accurate and also reduces fitting complexity from $O(N^2)$ to $O(N)$.*

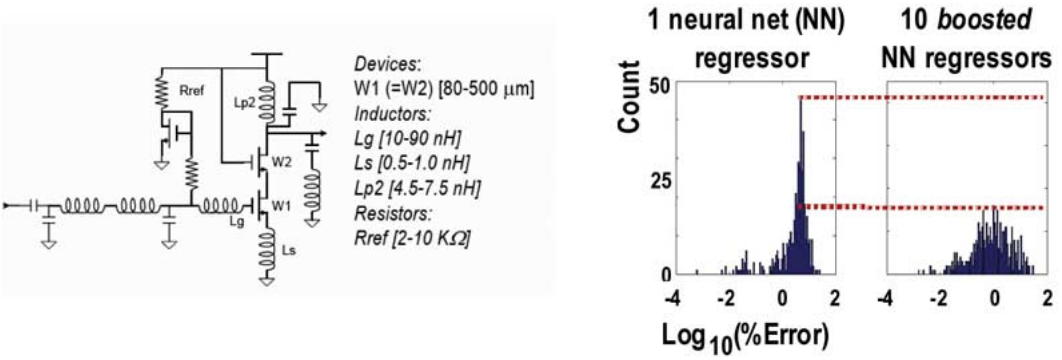


Fig. 22. ***Example data mining ideas for parametric circuit modeling. RF LNA at left has five designable variables, varying over ranges shown. Histograms at right show fitting error for predicting IIP3, for one simple neural network, and ten boosted neural net based regressors, for fitting 2000 simulated samples of IIP3, across uniform samples of the five design variables. Boosted result makes fewer and much smaller errors [11].***

models; later models in the sequence fit well in regions of the design space where earlier models fit poorly. An elegant numerical formulation, called *boosting* [93], efficiently combines the predictions of the many individual models into a single final numerical value. The technique has the attractive feature that any specific nonlinear regressor may be used in the overall fitting process; Liu *et al.* used a set of relatively small neural networks [11]. Fig. 22 shows a relatively challenging parametric fitting example, in which these ideas were used to predict IIP3 as the design variables for an RF LNA were varied uniformly, randomly over quite wide ranges of values. Other applications of ideas from the data mining world include [10] and [94].

Another recent approach is the CAFFEINE method (Canonical Functional Form Expressions in Evolution) which presents the first *template-free* model generation approach, i.e., the designer does not have to specify *a priori* a model template, but the model itself evolves as part of the genetic-programming optimization process [12]. This corresponds to the flow without functional template in Fig. 20. Genetic programming is applied as a means of traversing the space of possible symbolic expressions. A grammar is specially designed to constrain the search to a canonical form for functions. Novel evolutionary search operators are designed to exploit the structure of the grammar. Using multi-objective optimization, the approach generates a set of symbolic models which collectively provide a tradeoff between error and complexity. For the circuit of Fig. 19, Table 1 shows the CAFFEINE-generated performance models for six different performance characteristics that each have less than 10% training and test error. Table 2 then shows a possible set of alternative models generated by CAFFEINE with different tradeoff between error and complexity for the phase margin of the circuit of Fig. 19.

Note that an operating-point driven formulation has been used where the performance characteristics are modeled as a function of bias currents and bias voltages. In [13], it was shown that this method generates the most accurate models of ten different methods that have been compared, including regression with spline functions and support vector machines. The results are repeated in Fig. 23. Very recently, the method has been sped up significantly by the use of implicit canonical form functions and introns [14].

Table 1 CAFFEINE-Generated Performance Models for Six Different Characteristics of the Circuit of Fig. 19, Which Each Have Less Than 10% Training and Testing Error

Perf.	Target (%)		Expression
	q_{wc}	q_{oc}	
A_{LF}	10	10	- 10.3 + 7.08e-5 / id1 + 1.87 * ln(-1.95e+9 + 1.00e+10 / (vsg1*vsg3)+ 1.42e+09 *(vds2*vsd5) /(vsg1*vgs2*vsg5*id2))
f_u	10	10	10^(5.68 - 0.03 * vsg1 / vds2 - 55.43 * id1+ 5.63e-6 / id1)
PM	10	10	90.5 + 190.6 * id1 / vsg1 + 22.2 * id2 / vds2
v_{offset}	10	10	- 2.00e-3
SR_p	10	10	2.36e+7 + 1.95e+4 * id2 / id1 - 104.69 / id2 + 2.15e+9 * id2 + 4.63e+8 * id1
SR_n	10	10	- 5.72e+7 - 2.50e+11 * (id1*id2) / vgs2 + 5.53e+6 * vds2 / vgs2 + 109.72 / id1

Table 2 CAFFEINE-Generated Performance Models of Phase Margin (PM), in Order of Decreasing Error and Increasing Complexity, for the Circuit of Fig. 19

Test error (%)	Train error (%)	Num basis funcs	PM Expression
3.98	15.4	0	90.2
3.71	10.6	2	90.5 + 186.6 * id1 + 22.1 * id2 / vds2
3.68	10.0	2	90.5 + 190.6 * id1 / vsg1 + 22.2 * id2 / vds2
3.42	9.2	2	90.3 + 153.0 * id1 / vsg1 - 5.78e-07 * vsg1 / (vds2*vsd5*id1)
3.39	8.8	3	90.1 + 156.85 * id1 / vsg1 - 2.06e-03 * id2 / id1 + 0.04 * vgs2 / vds2
3.31	8.0	3	91.1 - 2.05e-3 * id2 / id1 + 145.8 * id1 + 0.04 * vgs2 / vds2 - 1.14 / vsg1
3.20	7.7	4	90.7 - 2.13e-3 * id2 / id1 + 144.2 * id1 + 0.04 * vgs2 / vds2 - 1.00 / (vsg1*vsg3)
2.65	6.7	5	90.8 - 2.08e-3 * id2 / id1 + 136.2 * id1 + 0.04 * vgs2 / vds2 -1.14 / vsg1 + 0.04 * vsg3 / vsd5
2.58	5.0	8	90.9 – 7.56e-4 * (vsg1*id2) / id1 + 143.2 * id1 + 0.03 * vgs2 / vds2 - 0.74 / vsg1 + 0.03 * vsg1 / vsd5 - 3.00e-7 / (vds2*vsd5*id1) + 22.8 * (vgs2*id2) - 0.62 / vsg3
2.41	3.9	11	91.11 - 5.91e-4 * (vsg1*id2) / id1 + 119.79 * id1 + 0.03 * vgs2 / vds2 - 0.78 / vsg1 + 0.03 * vsg1 / vsd5 - 2.72e-7 / (vds2*vsd5*id1) + 7.11 * (vgs2*vsg4*id2) - 0.37 / vsg5 - 0.58 / vsg3 - 3.75e-6 / id2 - 5.52e-6 / id1

A final class of important parametric modeling technique are the *Pareto* methods [4], [5], [95], [96]. We briefly review the fundamental idea of Pareto optimality, following the discussion of [97]. The capability of any analog circuit is defined using a set of performance metrics (e.g., dc gain, bandwidth, power, slew rate for an operational amplifier). Like any physical system, there are limits on how good these metrics can be for any circuit topology. The set of all possible performance metric values achievable by any circuit topology defines the *performance feasibility region* of the topology. These performance metrics are often competing (e.g., gain and bandwidth), and certain portions of the feasible region boundary define the tradeoff relationship while trying to achieve the optimal values for these metrics. These tradeoff surfaces are said to be *Pareto optimal*. These are referred to equivalently as Pareto *curves*, Pareto *fronts*, and Pareto *tradeoffs*.

Suppose that $\mathbf{x} \in X \subset R^n$ is the vector of n design variables. $\mathbf{p}(\mathbf{x}) \in P \subset R^m$ is the vector of circuit performances. These performances can be categorized into *constrained* performances p_c (which must meet certain specifications for acceptable circuit performance) and *objective* performances p_o (which are to be optimized: we assume minimized without loss of generality). $\mathbf{c}(\mathbf{x}) \in C \subset R^l$ is the vector of constraint variables that are needed to guarantee correct circuit operation. $\mathbf{g}(\mathbf{c}, \mathbf{p}_c) \in F$ is the vector of real-valued constraint functions to guarantee correct circuit behavior and minimum acceptable performance ($\mathbf{g} \leq 0$). These include constraints like minimum UGF specification, dc biasing conditions, etc.

The performance feasibility region of a circuit is the subset of P over which the constraints $\mathbf{g}$ are met. Here, we define the *domination* operator (a dominates b)

$$a \prec b \Leftrightarrow \forall_{i \in 1,\ldots,k} a_i \leq b_i \wedge \exists_{i \in 1,\ldots,k} a_i < b_i. \qquad (12)$$

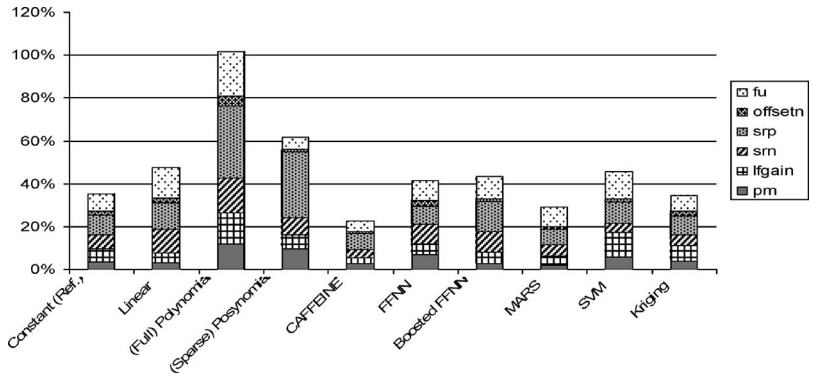


Fig. 23. ***Comparison of accumulated modeling error for six different performance characteristics of circuit of Fig. 19 for ten different performance modeling methods, starting from the same original set of SPICE data points.***

The Pareto-optimal front is the set of points on the feasible region boundary that are not dominated by any other point in the feasible region. Computing the performance feasibility region of a circuit topology will imply computing the set of "best" possible points in terms of the objective performances and this is the Pareto-optimal front. This front defines the capability limit of the topology in terms of the objective performance metrics, assuming all the constraints $\mathbf{g}$ are met.

The Pareto-optimal front generation problem can be formulated as the following multi-objective optimization problem:

$$\begin{aligned} \text{Minimize} \quad & \mathbf{p}_o(\mathbf{x}) = \{p_{o1}(\mathbf{x}), p_{o2}(\mathbf{x}), \ldots, p_{ok}(\mathbf{x})\} \\ s.t. \quad & \mathbf{g}(\mathbf{c}(\mathbf{x}), \mathbf{p}_c(\mathbf{x})) \leq 0, \mathbf{x} \in X. \end{aligned} \tag{13}$$

A multi-objective optimization algorithm to solve this problem will generate points on the Pareto-optimal front, while satisfying two requirements. First, the solution set should approximate the real Pareto front as closely as possible and the algorithm should generate the solution set as quickly as possible. Second, the solution points should be uniformly spread out over a large extent of the Pareto front. This is to enable reliable model building.

There are several approaches to generating the Pareto front. An interesting characteristic of all such techniques is that they leverage the core infrastructure of existing circuit-level analog synthesis tools. In other words, one can extend the engines that synthesize sizing/biasing for an individual circuit with a specific set of performance targets and use these instead to trace out the Pareto front. For example, stochastic approaches such as the Watson tool [4] exploit ideas from genetic algorithms and try to evolve populations of sized circuits whose "fitness" is rewarded proportionally to how well each candidate fill in gaps in the evolving Pareto front and how "dominated" the point is by other members of the population.

It is also possible to use more direct methods to find these curves. Of these, the Normal Boundary Intersection (NBI) method is most well known [96]. Suppose, via circuit synthesis, we can find circuits that define the best possible values of each individual performance objectives $\mathbf{p}_o(\mathbf{x}) = \{p_{o1}(\mathbf{x}), p_{o2}(\mathbf{x}), \ldots, p_{ok}(\mathbf{x})\}$. Roughly speaking, we build the convex hull of these individual points and then search in a direction normal to this hull, toward the Pareto-optimal points, from a set of suitably uniformly spaced points on this hull. Fig. 24 illustrates the idea for the geometrically simplest case of a two-dimensional Pareto front; in this case, the convex hull is simply the chord between the points defining the two best values of the two performance objectives. We search from the points on the convex chord, adding an additional constraint that we seek a circuit solution that: 1) lies on this normal and 2) is maximally distant from the chord. Such points will comprise the Pareto points we seek. A version of the WiCKed synthesis tool uses NBI ideas for automatically tracing Pareto points [5]. One challenge with this formulation is the need to add the additional points-on-the-normal constraint, which takes the possibly numerically more difficult form of an equality constraint. It is possible to relax this and formulate an NBI-like method using only inequalities, as shown in [95]. The attractive feature of this version is that the problem of search along the normal is transformed into a problem of search near the normal, which involves only inequalities constraints, which are robustly handled by most industrial synthesis engines.

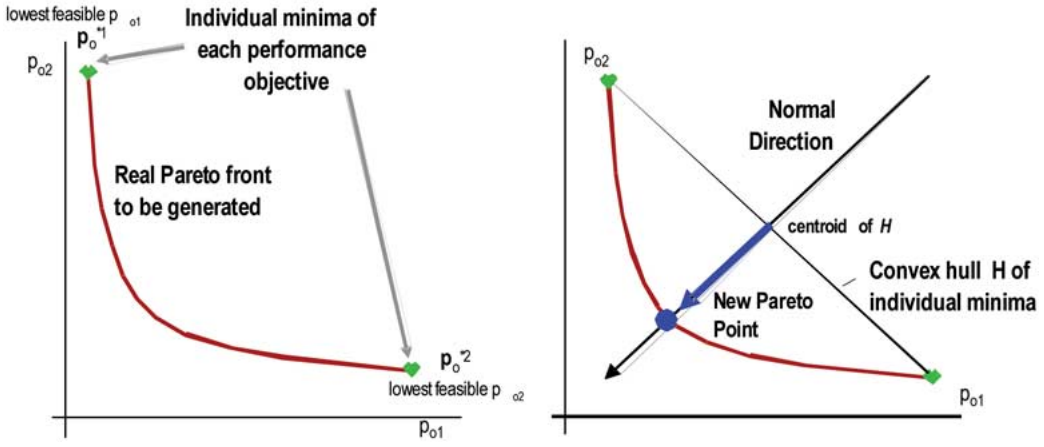


Fig. 24. *Illustrating NBI method [96] for tracing Pareto front in two dimensions. From individual minima for two performance objectives (left) the convex hull H (chord, right) of these points is formed, then from the centroid of H a search on the normal is performed to find the new Pareto point.*

Finally, it is worth noting that another significant benefit of the Pareto-based model is dimensionality reduction. That is, we replace the potentially large number of degrees of freedom of a cell-level circuit with the relatively small number of degrees of freedom of the Pareto curve. To be concrete, suppose we have an amplifier which has 20 CMOS devices; we need perhaps 20 widths and 20 lengths to design this circuit. However, suppose that at system level, all we really want to know is the best point on the gain versus bandwidth Pareto curve. Then, we have really reduced this to a single degree of freedom: when we select a value for one of these dimensions, our Pareto model uniquely specifies the other. This dimensionality reduction and the fact that these sorts of tradeoff analyses are extremely familiar to working circuit designers accounts for recent interest in the Pareto models. Of course, these do become quite unwieldy in more than a few dimensions; this is the main drawback of the approach.

Despite the progress made so far, still more research in the area of automatic performance model generation is needed to reduce analog synthesis times, especially for hierarchical synthesis of complex analog blocks. The field is a very active research area at the moment.

C. Hierarchical Synthesis Flows

Recall that our original motivations for the various forms of modeling in the previous sections were twofold: first, for full-system design verification, we needed efficient instance-based macromodels that allowed use to simulate large designs in tractable amounts of time; second, for optimization-based synthesis flows which visit many intermediate circuit solutions and evaluate each with full simulation, we needed efficient parametric models that let us quickly explore how changes circuit-level changes affect system-level outcomes. In this section, we briefly review how synthesis can be organized, using all these modeling ideas.

The earliest approaches used top-down strategies. Chang *et al.* [105], [106] described a constraint-driven top-down design methodology where constraints are propagated down the hierarchy for complex mixed-signal systems. The methodology is demonstrated for some practical design examples. The performance constraints are also used to control the layout synthesis tools (placement, routing). A second early approach that can be mentioned here is the OASYS tool of [6]: at each level in the design hierarchy, the performance specifications of the subblocks within the selected block architecture have to be determined based on the block's overall specifications. This can be done in a top-down manner using optimization methods at each level, until the device level is reached and all devices have been sized. The complete design flow is then an alternation of topology selection (i.e., picking the right circuit-level architecture) and specification translation (i.e., determining how specifications at one level decompose into specification at the next). Reference [6] suggested a set of equation-based strategies for this purpose.

Unfortunately, hierarchical techniques based on *ad hoc* sets of equations have not proven to be very practical: they require excessive amounts of effort to redirect for new circuits and new technologies. One interesting workaround strategy is to restrict the form of the equations used to model devices, circuits, and system—level arrangements of blocks. For example, if one restricts all performance objectives to a posynomial form, one can apply powerful, efficient convex optimization techniques to find solutions. These ideas have been applied at circuit level (e.g., for op-amps [98] and *LC* oscillators [99]) and for a few system-level designs (e.g., simple converters [100] and PLLs [101]). However, the effort to transform performance goals into the restrictive convex form can itself be challenging, the quality of the fit may be problematic, and each new circuit requires a new set of burdensome hand-developed analytical equations.

In light of this, we can see why there is such interest in parameterized macromodels. These potentially let us keep the simulator-in-the-loop optimization flow in the style of Fig. 16 but provide the necessary twin reductions in per-candidate simulation times and number of degrees of freedom being optimized. Phelps *et al.* [88] was one notable step in this direction who showed how some simple macromodels for the essential blocks in an industrial ADSL design could allow simulation-based synthesis to be performed. The experiment showed how to resynthesize a multiple-week industrial design overnight.

Today, the Pareto front techniques seem to be the most powerful methods we have for reducing simulation times and dimensionality. Given this, it is worth stepping through a small but concrete example to show exactly how many of the modeling ideas of the previous sections come together. We use the small PLL example of [95] here.

The design scenario is outlined in Fig. 25. We want to build a 1 : 2 PLL; the technology is 0.18-μm CMOS. The main block of interest is the VCO, which uses a current starved ring oscillator similar to the one highlighted in Fig. 17(a). Evaluating the phase noise specification for the overall PLL makes the problem too expensive to synthesize by flattening the PLL; for example, simulating 3 μs of lock-in requires roughly 2 h of CPU time. We resolve this by macromodeling each of the blocks as follows.

1) The phase frequency detector, charge pump, and frequency divider are replaced with standard behavioral models, using ideas from [102]. These will not be synthesized.
2) The loop filter is a simple passive *RC* filter and is simply flattened and simulated directly. We design the *RC* values as part of this synthesis task.
3) The VCO is replaced with a behavioral model which embeds a Pareto tradeoff curve. Since meeting the PLL-level jitter specification while

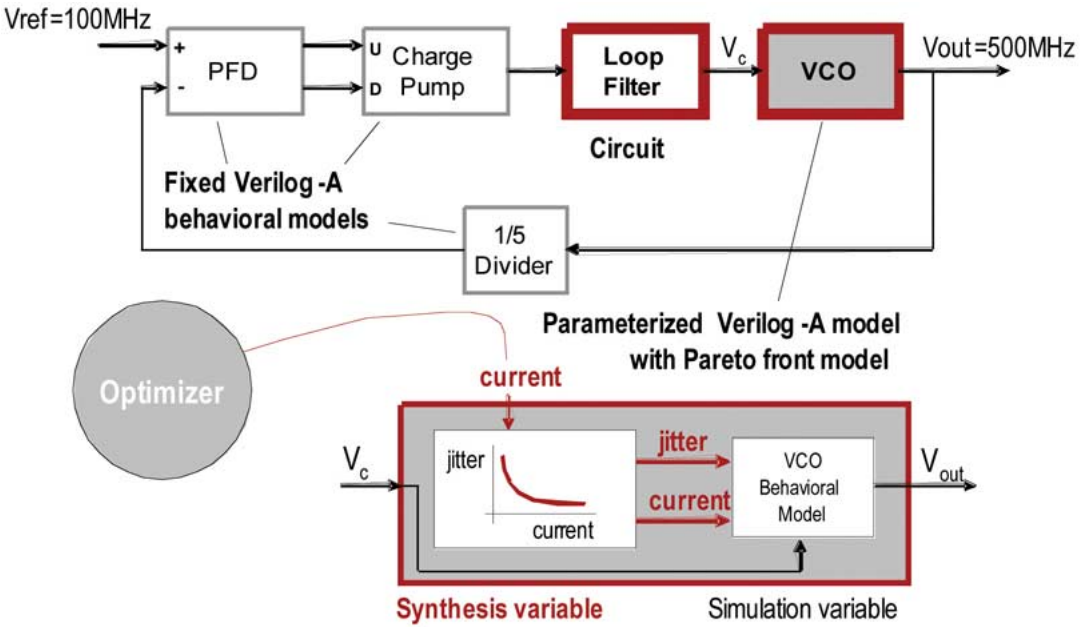


Fig. 25. *2 : 1 PLL hierarchical design scenario from [95].*

minimizing power is the real challenge, we use an industrial analog synthesis tool and ideas from the NBI method and trace a jitter versus bias current (i.e., power) for the VCO [95]. This appears in Fig. 26. However, a Pareto curve is not a simulation model: we need to embed this information somehow in a simulation model. Fig. 27 shows the solution. For the voltage versus frequency nonlinearity, and dynamics of the VCO, we fit a simple, conventional behavioral model with one pole (for dynamics) and a third-order polynomial nonlinearity. We then add the jitter versus current Pareto tradeoff as a bias-current-dependent frequency perturbation, again using ideas from [102]. In this way, we create a fully simulateable model which has also one degree of freedom—the VCO

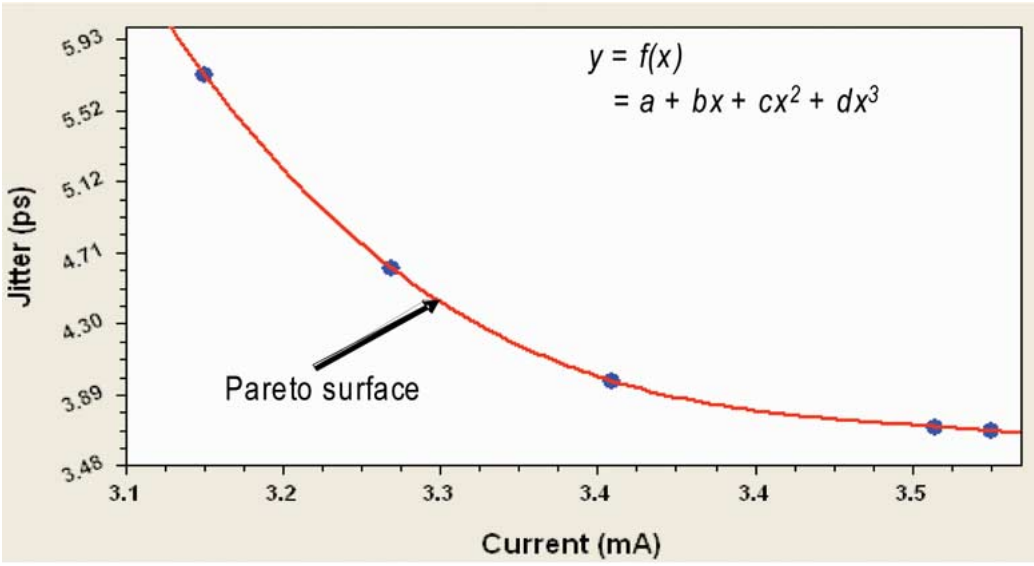
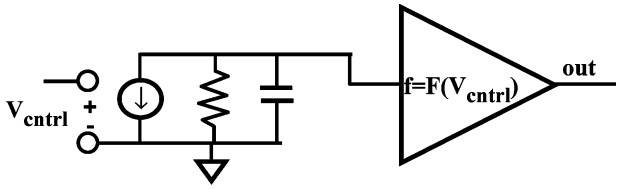


Fig. 26. *Extracted jitter versus bias current Pareto front for VCO of Fig. 25.*



(a)

```
module vcoModel (out, currentOut, in);
…
analog begin
  @(initial_step) begin
     // Pareto equation for jitter trade-off with power
     jitter = (1.03-(.87*cur)+(.24*cur^2)-(.02*cur^3))*1p;
  end

  //Output freq. of the VCO as function of control volt.
  freq = (-1.27 + 9.99*cos(1.10*V(in) + 4.29))*1e8;
…
  // Adding phase noise to the output
  dT = 1.414*jitter*$dist_normal(seed,0, 1);
  freq = freq/(1 + dT*freq);
…
  V(out) <+ transition(Vout, 0, tt);
I(currentOut) <+ cur;
```

(b)

Fig. 27. ***Overall nonlinear behavior model for VCO: (a) topology for modeling dynamics and nonlinear voltage-to-frequency behavior and (b) verilog-A pseudo-code for adding phase noise from the Pareto front of Fig. 26.***

bias current is an independent parameter—which will be manipulated by the circuit synthesis process to optimize the overall PLL.

The PLL was simulated for 4000 cycles at each synthesis point and it took about 4 h to synthesize the optimal circuit solution on a single CPU. The optimized variable values for the PLL include the *RC* values for the loop filter and the jitter and current for the VCO. Note that this means that the overall synthesis has not given us the device level sizing for the VCO, but rather, the best point on the jitter versus bias current Pareto at which to operate it. To finish this design, we thus need to search for the sizing solution for the VCO that has these specified jitter and current specifications. Recall that our Pareto curve is it itself a regression fit based on a moderate number a synthesis-derived circuits, so we do not simply have this detailed VCO solution stored. However, it is easy to obtain, since we know a good estimate of the circuit's Pareto tradoffs. Thus, one more synthesis completes the task. Final PLL specifications appear in Table 3. To verify that our final design does indeed meet the specs for the PLL, we simulated the final transistor level circuit and its equivalent macromodel. Fig. 28 shows the waveforms at the input control voltage of the VCO for both the macromodel version and the flat, SPICE-level simulation. As can be seen from figure, the responses of the two circuits match each other quite closely which confirms the accuracy of our models.

The ability to use optimization-based synthesis to extract Pareto fronts has led to a variety of more sophisticated approaches. For example, an alternative presented recently is to use a multi-objective bottom-up (MOBU) synthesis methodology [7]. Rather than traversing the design hierarchy in a top-down manner, the hierarchy is traversed in a bottom-up way. The core ideas are: 1) to determine just tradeoffs among the performance objectives, not whole feasibility regions, and 2) to directly use designed circuits rather than models. We now discuss this in detail. These cell-level Pareto optimal sets returned by most analog multi-objective-based sizing tools today can be directly exploited for system-level design, in what amounts to a "Multi-Objective Bottom-Up" (MUBU) methodology.

The design space for the next level up is the "selection" of a design for each of the subblocks. A "selected" subblock design is actually pointing to a specific design from the lower level tradeoff Pareto-hypersurface of that subblock. The hierarchy traversal proceeds in an upwards fashion, in the end providing an optimal system-level tradeoff. Fig. 29 illustrates this MOBU process. The example considered here is an analog to digital converter, which contains a number of op-amps as subblocks. First, using SPICE simulations the Pareto tradeoff fronts of the op-amp are constructed. Fig. 29 shows the slew rate versus gain-bandwidth as example tradeoff. Following the MOBU methodology, the Pareto-front data points of the op-amps and other subblocks are then used to generate the Pareto tradeoffs of the entire converter, using behavioral simulations. Fig. 29 shows the example of the converter-level tradeoff between the signal-to-noise ratio (SNR) versus power consumption.

In MOBU, any design that is selected on any level is already fully sized. An analog designer just has to choose a solution at the system level according to the performance specifications, and immediately all the design variables of the complete system are set. MOBU simultaneously provides both full sizings and flexibility in specifications. In addition, once a given block has had its Pareto optimal set generated in a given technology, it can be reused. In top-down design a good feasibility-modeling approach is necessary for success to make sure that feasible specifications are chosen for the subblocks. In contrast, MOBU

Table 3 PLL Performance Specification After Final Synthesis

Performance Specification	Achieved Value
PLL Settling Time	0.822 us
VCO bias current	3.37 mA
PLL jitter	0.45% (9.1ps)

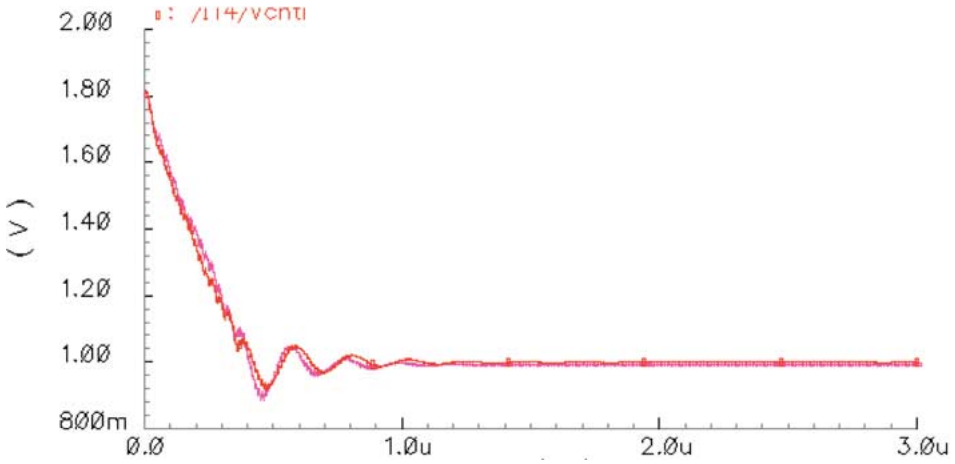


Fig. 28. *Comparison plot of the control voltage of the VCO for complete PLL simulation showing both the transistor-level circuit and behavioral model.*

completely avoids explicit modeling of any performance surfaces. The closest thing it has to a "model" is the Pareto-optimal set, which collectively approximates a performance surface with only "optimal" design points. Of course, one could build a regression model, for example by extending [4] for use in hierarchical design. Fig. 30 illustrates the differences between top-down and bottom-up hierarchical design.

IV. STATISTICAL OPTIMIZATION TECHNIQUES

A complete survey of statistical optimization techniques, targeting yield and robustness for hierarchically specified designs, is beyond the scope of this paper. We merely outline the problem and note a few recent promising efforts in this very challenging area.

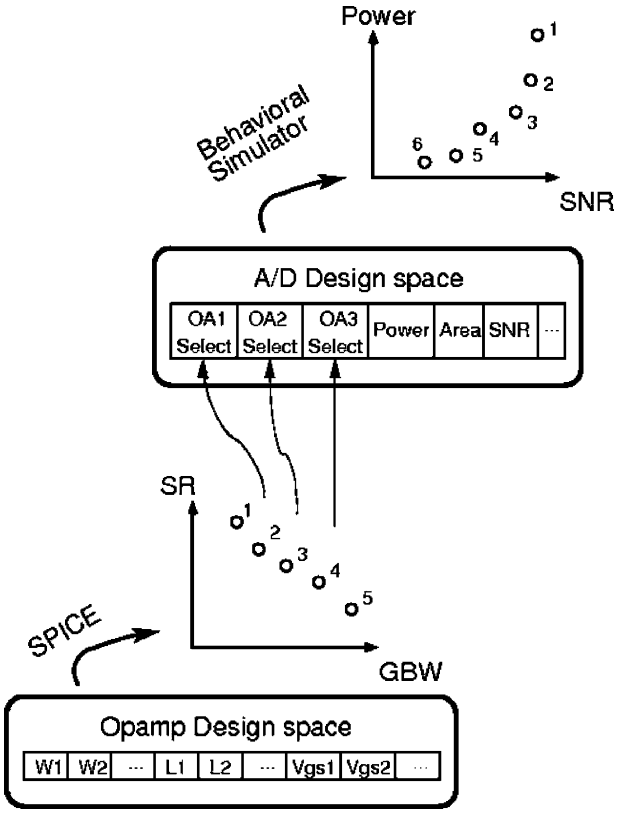


Fig. 29. *MOBU maps design points to performance points via performance simulation as usual. It finds and keeps just the tradeoff via multi-objective optimization. It then propagates tradeoffs upwards where they combine to make the next level's optimal design space.*

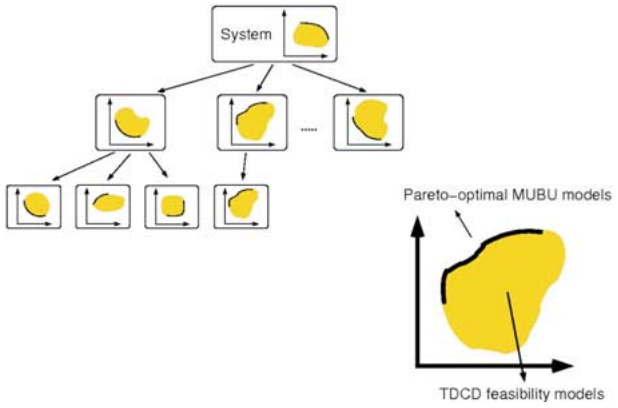


Fig. 30. *Top-down design with feasibility modeling versus MOBU. Shaded-out space for each block represents its performance feasibility region, which feasibility modeling generates bottom-up. Boundary curve line on one edge of the region represents the Pareto-optimal set that MOBU generates bottom-up.*

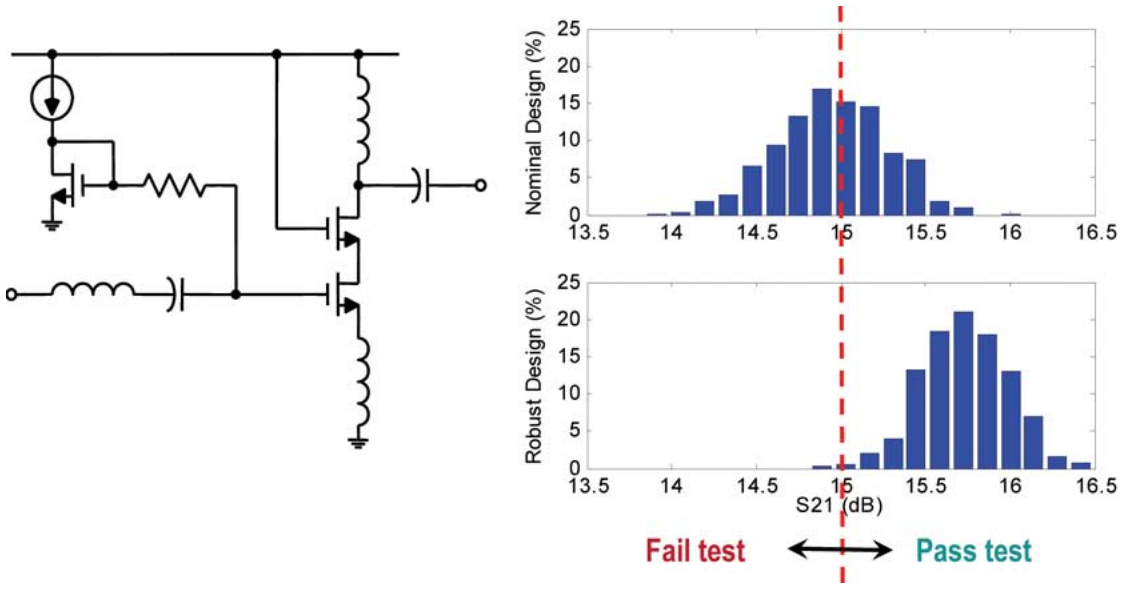


Fig. 31. *Applying ROAD methodology for post-nominal statistical robustness optimization on RF LNA in 0.25-μm SiGe process [91].*

Industrial design practice not only calls for fully optimized *nominal* design solutions but also expects high robustness and yield in the light of varying operating conditions (supply voltage or temperature variations) and statistical manufacturing tolerances and mismatches [15], [16]. Due to these fluctuations, the device parameters and consequently also the circuit performance characteristics will show fluctuations. The corresponding parametric yield is the ratio of the number of acceptable (i.e., functional and meeting all specifications) to all fabricated IC samples. The yield of course depends on the nominal design point chosen for the circuit. Unfortunately, the relation between the (fluctuating) device parameters and the circuit performances is in general a nonlinear transformation that is not known explicitly but has to be simulated. All this makes yield estimation a time-consuming task, which in practice is often obtained by Monte Carlo simulations. An overview of more efficient techniques that trade off accuracy versus CPU time can be found in [2]. Note that in practice not only the yield, but in general the robustness of the design against variations of both technological and environmental parameters, has to be maximized. This implies techniques for variability minimization and design centering.

The most straightforward—and, sadly, still the most practically encountered solution—is to evaluate the circuit performance not only in the nominal design point, but also in a set of predefined worst case process, voltage, and temperature (P,V,T) corners. The circuit then needs to satisfy the specifications in all corners. The problem with this approach is that the CPU time increases with the number of corners that needs to be simulated and that the number of corners becomes intractably large in nanometer technologies. In addition, for any reasonably complex system level design, we do know not in advance which of these intractable number of design corners is actually the worst corner for each individual circuit specification. Corner-based design also easily leads to overly pessimistic worst case design. Hence, the current trend is to move towards true statistical yield optimization, where the statistical distribution of the parameter variations is considered.

Handling difficult statistics, with complex correlations, for complex circuits with difficult-to-simulate analog behaviors, remains a challenge. We simply mention a few notable efforts here. In the area of simulation-based synthesis, the WiCked approach [17] uses worst case parameter distances as robustness objectives to obtain a nominal design that satisfies all specifications with as much safety margin as possible for process variations. The resulting formulation is the same as for design centering and can be solved efficiently using the generalized boundary curve. Design centering, however, still remains a second step after the nominal design. The efficient projection-based quadratic modeling ideas of the ROAD approach [91] actually apply very effectively to the statistical case, since one can build statistical response surfaces with this idea. In combination with efficient numerical techniques like APEX [103] to estimate the pdfs of nonlinear circuits, ROAD can be used as an effective post-nominal robust optimizer. An example appears in Fig. 31, which shows a successful post-nominal robustness optimization for an RF LNA. Finally, the convex modeling ideas mentioned in Section III-B can also be applied in this context, e.g., [104] formulates a novel system-level exploration methodology based on geometric centering ideas that can be rendered in a convex form.

V. CONCLUSION

In this paper, we have attempted to review the recent state of the art in hierarchical analog synthesis, with a strong emphasis on associated techniques for computer-aided model generation and optimization. There are industrially useful and commercially available tools at the cell level—tools for analog components with 10–100 devices. However, successful component-level tools do not scale trivially to system-level applications. These hierarchically specified designs need to optimize many competing continuous-valued performance specifications, which depend on the circuit designer's abilities to successfully exploit a range of nonlinear behaviors across levels of abstraction from devices to circuits to systems. For purposes of synthesis or verification, these designs are not tractable when considered "flat." These designs must be approached with hierarchical tools that deal with the system's intrinsic design hierarchy. We reviewed recent ideas in analog modeling and synthesis that specifically deal with the hierarchical nature of practical mixed-signal and RF systems: algorithmic techniques for automatically extracting a suitable nonlinear macromodel from a device-level circuit; parametric modeling techniques; ideas for hierarchical synthesis, in particular, numerical techniques for handling the large number of degrees of freedom in these designs, and for exploring the space of performance tradeoffs early in the design process. Finally, we briefly touched on emerging ideas for accommodating models of statistical manufacturing variations in these tools and flows.

As more analog and RF designs migrate onto difficult SOC-style digital platforms, we expect to see a continuing demand for these sorts of hierarchically oriented modeling and synthesis tools and, in particular, tools to help designs deal with the challenges to circuit robustness posed by nanometer technologies. ■

Acknowledgment

The authors would like to thank A. Singhee of Carnegie Mellon University, Pittsburgh, PA, who provided valuable comments and proofreading at various early stages of the development of this manuscript and detailed material on Pareto-optimal fronts used in Section III. J. Roychowdhury would like to thank N. Dong (TI), X. Lai (University of Minnesota), and T. Mei (University of Minnesota), on whose research much of the material here on macromodelling and simulation relies. G. Gielen would like to thank Ph.D. students T. Eeckelaert, E. Martens, and T. McConaghy, who have contributed largely to the work on hierarchical synthesis and performance modeling.

REFERENCES

[1] International Technology Roadmap for Semiconductors 2003. [Online]. Available: http://www.public.itrs.net

[2] G. Gielen and R. Rutenbar, "Computer-aided design of analog and mixed-signal integrated circuits," *Proc. IEEE*, vol. 88, no. 12, pp. 1825–1854, Dec. 2000.

[3] R. A. Rutenbar, G. G. E. Gielen, and B. Antao, Eds., *Computer-Aided Design of Analog Integrated Circuits and Systems*. Hoboken, NJ: Wiley-IEEE, Apr. 2002.

[4] B. De Smedt and G. Gielen, "WATSON: Design space boundary exploration and model generation for analog and RF IC design," *IEEE Trans. Computer-Aided Design*, vol. 22, no. 2, pp. 213–224, Feb. 2003.

[5] G. Stehr, H. Graeb, and K. Antreich, "Performance trade-off analysis of analog circuits by normal-boundary intersection," in *Proc. IEEE/ACM Design Automation Conf. (DAC)*, 2003, pp. 958–963.

[6] R. Harjani, R. Rutenbar, and L. R. Carley, "OASYS: A framework for analog circuit synthesis," *IEEE Trans. Computer-Aided Design*, vol. 8, no. 12, pp. 1247–1265, Dec. 1989.

[7] G. Gielen, T. McConaghy, and T. Eeckelaert, "Performance space modeling for hierarchical synthesis of analog integrated circuits," in *Proc. IEEE Design Automation Conf. (DAC)*, Jun. 2005, pp. 881–886.

[8] B. De Smedt and G. Gielen, "Models for systematic design and verification of frequency synthesizers," *IEEE Trans. Circuits Systems, Part II: Analog Digital Signal Processing*, vol. 46, no. 10, pp. 1301–1308, Oct. 1999.

[9] W. Daems, G. Gielen, and W. Sansen, "Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits," *IEEE Trans. Computer-Aided Design*, vol. 22, no. 5, pp. 517–534, May 2003.

[10] T. Kiely and G. Gielen, "Performance modeling of analog integrated circuits using least-squares support vector machines," in *Proc. Design, Automation Test Eur. (DATE) Conf.*, Feb. 2004, pp. 448–453.

[11] H. Liu, A. Singhee, R. Rutenbar, and L. R. Carley, "Remembrance of circuits past: Macromodeling by data mining in large analog design spaces," in *Proc. IEEE/ACM Design Automation Conf. (DAC)*, Jun. 2002, pp. 437–442.

[12] T. McConaghy, T. Eeckelaert, and G. Gielen, "CAFFEINE: Template-free symbolic model generation of analog circuits via canonical form functions and genetic programming," in *Proc. Design, Automation Test Eur. Conf. (DATE)*, Mar. 2005, vol. 2, pp. 1082–1087.

[13] T. McConaghy and G. Gielen, "Analysis of simulation-driven numerical performance modeling techniques for application to analog circuit optimization," in *Proc. IEEE Int. Symp. Circuits Systems (ISCAS)*, May 2005, vol. 2, pp. 1298–1301.

[14] T. McConaghy, T. Eeckelaert, and G. Gielen, "Fast-acting CAFFEINE for template-free symbolic model generation of analog circuits via implicit canonical form functions and explicit introns," in *Proc. Design, Automation Test Eur. Conf. (DATE)*, Mar. 2006.

[15] S. Director, W. Maly, and A. Strojwas, *VLSI Design for Manufacturing: Yield Enhancement*. Boston, MA: Kluwer, 1990.

[16] J. Zhang and M. Styblinski, *Yield and Variability Optimization of Integrated Circuits*. Boston, MA: Kluwer, 1995.

[17] K. Antreich, H. Graeb, and C. Wieser, "Circuit analysis and optimization driven by worst-case distances," *IEEE Trans. Computer-Aided Design*, vol. 13, no. 1, pp. 57–71, Jan. 1994.

[18] X. Huang, C. S. Gathercole, and H. A. Mantooth, "Modeling nonlinear dynamics in analog circuits via root localization," *IEEE Trans. Computer-Aided Design*, vol. 22, no. 7, pp. 895–907, Jul. 2003.

[19] K. Francken, M. Vogels, E. Martens, and G. Gielen, "A behavioral simulation tool for continuous-time/spl Delta//spl Sigma/ modulators," in *Proc. ICCAD*, Nov. 2002, pp. 229–233.

[20] S. X.-D. Tan and C. J.-R. Shi, "Efficient DDD-based term generation algorithm for analog circuit behavioral modeling," in *Proc. IEEE ASP-DAC*, Jan. 2003, pp. 789–794.

[21] ——, "Efficient DDD-based term generation algorithm for analog circuit behavioral modeling," in *Proc. IEEE DATE Conf.*, Mar. 2003, pp. 1009–1108.

[22] ——, "Efficient very large scale integration power/ground network sizing based on equivalent circuit modeling," *IEEE Trans. Computer-Aided Design*, vol. 22, no. 3, pp. 277–284, Mar. 2003.

[23] Y. Qicheng and C. Sechen, "A unified approach to the approximate symbolic analysis of large analog integrated circuits," *IEEE Trans. Cts. Syst.—I: Fund. Th. Appl.*, pp. 656–669, Aug. 1996.

[24] W. Daems, G. Gielen, and W. Sansen, "A fitting approach to generate symbolic expressions for linear and nonlinear analog circuit performance characteristics," in *Proc. IEEE DATE Conf.*, Mar. 2002, pp. 268–273.

[25] P. Vanassche, G. Gielen, and W. Sansen, "Constructing symbolic models for the input/output behavior of periodically time-varying systems using harmonic

transfer matrices," in *Proc. IEEE DATE Conf.*, Mar. 2002, pp. 279–284.

[26] L. Hongzhou, A. Singhee, R. Rutenbar, and L. R. Carley, "Remembrance of circuits past: Macromodeling by data mining in large analog design spaces," in *Proc. IEEE DAC*, Jun. 2002, pp. 437–442.

[27] D. Schreurs, J. Wood, N. Tufillaro, D. Usikov, L. Barford, and D. E. Root, "The construction and evaluation of behavioral models for microwave devices based on time-domain large-signal measurements," in *Proc. IEEE IEDM*, Dec. 2000, pp. 819–822.

[28] J. Wood and D. E. Root, "The behavioral modeling of microwave/RF ICs using non-linear time series analysis," in *IEEE MTT-S Dig.*, Jun. 2003, pp. 791–794.

[29] L. A. Zadeh and C. A. Desoer, *Linear System Theory: The State-Space Approach,* ser. System Science. New York: McGraw-Hill, 1963.

[30] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer-Aided Design,* vol. 9, pp. 352–366, Apr. 1990.

[31] E. Chiprout and M. S. Nakhla, *Asymptotic Waveform Evaluation*. Norwell, MA: Kluwer, 1994.

[32] K. Gallivan, E. Grimme, and P. Van Dooren, "Asymptotic waveform evaluation via a Lanczos method," *Appl. Math. Lett.*, vol. 7, pp. 75–80, 1994.

[33] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Pade approximation via the Lanczos process," *IEEE Trans. Computer-Aided Design,* vol. 14, no. 5, pp. 639–649, May 1995.

[34] R. W. Freund and P. Feldmann, "Efficient small-signal circuit analysis and sensitivity computations with the Pvl algorithm," in *Proc. ICCAD*, Nov. 1995, pp. 404–411.

[35] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Boston, MA: PWS, 1996.

[36] E. J. Grimme, "Krylov projection methods for model reduction," Ph.D. dissertation, EE Dep., Univ. Illinois, Urbana-Champaign, 1997.

[37] L. M. Silveira, M. Kamon, I. Elfadel, and J. White, "A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of *RLC* circuits," in *Proc. ICCAD*, Nov. 1996, pp. 288–294.

[38] K. J. Kerns, I. L. Wemple, and A. T. Yang, "Stable and efficient reduction of substrate model networks using congruence transforms," in *Proc. ICCAD*, Nov. 1995, pp. 207–214.

[39] R. Freund and P. Feldmann, "Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm," in *Proc. ICCAD*, Nov. 1996, pp. 280–287.

[40] Odabasioglu, M. Celik, and L. T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodelling algorithm," in *Proc. ICCAD*, Nov. 1997, pp. 58–65.

[41] ——, "PRIMA: Passive reduced-order interconnect macromodelling algorithm," *IEEE Trans. Computer-Aided Design*, pp. 645–654, Aug. 1998.

[42] Z. Bai, R. Freund, and P. Feldmann, "How to make theoretically passive reduced-order models passive in practice," in *Proc. IEEE CICC*, May 1998, pp. 207–210.

[43] R. Freund, "Passive reduced-order models for interconnect simulation and their computation via Krylov-subspace algorithms," in *Proc. IEEE DAC*, Jun. 1999, pp. 195–200.

[44] J.-R. Li and J. White, "Efficient model reduction of interconnect via approximate system gramians," in *Proc. ICCAD*, Nov. 1999, pp. 380–383.

[45] J. Phillips, L. Daniel, and L. M. Silveira, "Guaranteed passive balancing transformations for model order reduction," in *Proc. IEEE DAC*, Jun. 2002, pp. 52–57.

[46] M. Kamon, F. Wang, and J. White, "Generating nearly optimally compact models from Krylov-subspace based reduced-order models," *IEEE Trans. Cts. Syst.—II: Sig. Proc.*, pp. 239–248, Apr. 2000.

[47] J. Roychowdhury, "MPDE methods for efficient analysis of wireless systems," in *Proc. IEEE CICC*, May 1998.

[48] ——, "Reduced-order modelling of linear time-varying systems," in *Proc. ICCAD*, Nov. 1998.

[49] ——, "Reduced-order modelling of time-varying systems," *IEEE Trans. Cts. Syst.—II: Sig. Proc.,* vol. 46, no. 10, Nov. 1999.

[50] ——, "Analysing circuits with widely-separated time scales using numerical PDE methods," *IEEE Trans. Cts. Syst.—I: Fund. Th. Appl.*, May 2001.

[51] K. S. Kundert, J. K. White, and A. Sangiovanni-Vincentelli, *Steady-State Methods for Simulating Analog and Microwave Circuits*. Boston, MA: Kluwer, 1990.

[52] M. Rösch and K. J. Antreich, "Schnell stationäre simulation nichtlinearer schaltungen im frequenzbereich," *AEÜ,* vol. 46, no. 3, pp. 168–176, 1992.

[53] R. Telichevesky, K. Kundert, and J. White, "Efficient steady-state analysis based on matrix-free Krylov subspace methods," in *Proc. IEEE DAC,* 1995, pp. 480–484.

[54] J. Phillips, "Model reduction of time-varying linear systems using approximate multipoint Krylov-subspace projectors," in *Proc. ICCAD*, Nov. 1998.

[55] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. New York: Wiley, 1980.

[56] Nayfeh and B. Balachandran, *Applied Nonlinear Dynamics*. New York: Wiley, 1995.

[57] J. Phillips, "Projection frameworks for model reduction of weakly nonlinear systems," in *Proc. IEEE DAC*, Jun. 2000.

[58] ——, "Projection-based approaches for model reduction of weakly nonlinear, time-varying systems," *IEEE Trans. Computer-Aided Design,* vol. 22, no. 2, pp. 171–187, Feb. 2000.

[59] W. Rugh, *Nonlinear System Theory—The Volterra-Wiener Approach*. Baltimore, MD: Johns Hopkins Univ. Press, 1981.

[60] P. Li and L. Pileggi, "NORM: Compact model order reduction of weakly nonlinear systems," in *Proc. IEEE DAC*, Jun. 2003, pp. 472–477.

[61] J. Katzenelson and L. H. Seitelman, "An iterative method for solution of nonlinear resistive networks," AT&T Bell Laboratories, Tech. Rep. TM65-1375-3.

[62] F. Ohtsuki and Kumagai, "Existence theorems and a solution algorithm for piecewise linear resistor networks," *SIAM J. Math. Anal.*, vol. 8, Feb. 1977.

[63] M. Rewienski and J. White, "A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices," in *Proc. ICCAD*, Nov. 2001.

[64] N. Dong and J. Roychowdhury, "Piecewise polynomial model order reduction," in *Proc. IEEE DAC*, Jun. 2003, pp. 484–489.

[65] L. W. Nagel, "SPICE2: A computer program to simulate semiconductor circuits," Ph.D. dissertation, EECS Dep., Univ. Calif. Berkeley, Elec. Res. Lab., 1975, Memo. ERL-M520.

[66] T. L. Quarles, "Analysis of performance and convergence issues for circuit simulation," Ph.D. dissertation, EECS Dep., Univ. Calif. Berkeley, Elec. Res. Lab., Apr. 1989, Memo. UCB/ERL M89/42.

[67] M. Schwab, "Determination of the steady state of an oscillator by a combined time-frequency method," *IEEE Trans. Microwave Theory Tech.*, vol. 39, pp. 1391–1402, Aug. 1991.

[68] O. Narayan and J. Roychowdhury, "Analysing oscillators using multitime PDEs," *IEEE Trans. Cts. Syst.—I: Fund. Th. Appl.,* vol. 50, no. 7, pp. 894–903, Jul. 2003.

[69] E. Ngoya and R. Larchevèque, "Envelop transient analysis: A new method for the transient and steady state analysis of microwave communication circuits and systems," in *Proc. IEEE MTT Symp.*, 1996.

[70] W. Gardner, *Introduction to Random Processes*. New York: McGraw-Hill, 1986.

[71] Hajimiri and T. H. Lee, "A general theory of phase noise in electrical oscillators," *IEEE J. Solid-State Cts.*, vol. 33, pp. 179–194, Feb. 1998.

[72] Demir, E. Liu, A. L. Sangiovanni-Vincentelli, and I. Vassiliou, "Behavioral simulation techniques for phase/delay-locked systems," in *Proc. Custom Integrated Circuits Conf.*, May 1994, pp. 453–456.

[73] Costantini, C. Florian, and G. Vannini, "VCO behavioral modeling based on the nonlinear integral approach," in *IEEE Int. Symp. Circuits and Syst.*, May 2002, vol. 2, pp. 137–140.

[74] L. Wu, H. W. Jin, and W. C. Black, "Nonlinear behavioral modeling and simulation of phase-locked and delay-locked systems," in *Proc. IEEE CICC*, May 2000, pp. 447–450.

[75] M. F. Mar, "An event-driven pll behavioral model with applications to design driven noise modeling," in *Proc. Behav. Model Simul. (BMAS)*, 1999.

[76] P. Vanassche, G. G. E. Gielen, and W. Sansen, "Behavioral modeling of coupled harmonic oscillators," *IEEE Trans. Computer-Aided Design Integrated Circuits Syst.*, vol. 22, no. 8, pp. 1017–1026, Aug. 2003.

[77] K. Kundert. (2002). *Predicting the phase noise and jitter of PLL-based frequency synthesizers*. [Online]. Available: www.designers-guide.com

[78] Demir, A. Mehrotra, and J. Roychowdhury, "Phase noise in oscillators: A unifying theory and numerical methods for characterization," *IEEE Trans. Cts. Syst.—I: Fund. Th. Appl.,* vol. 47, pp. 655–674, May 2000.

[79] Demir and J. Roychowdhury, "A reliable and efficient procedure for oscillator PPV computation, with phase noise macromodelling applications," *IEEE Trans. Cts. Syst.—I: Fund. Th. Appl.*, pp. 188–197, Feb. 2003.

[80] J. L. Stensby, *Phase-Locked Loops: Theory and Applications*. Boca Raton, FL: CRC, 1997.

[81] N. Dong and J. Roychowdhury, "Automated extraction of broadly-applicable nonlinear

analog macromodels from SPICE-level descriptions," in *Proc. IEEE CICC*, Oct. 2004.

[82] X. Lai and J. Roychowdhury, "Capturing injection locking via nonlinear phase domain macromodels," *IEEE MTT*, vol. 52, no. 9, pp. 2251–2261, Sep. 2004.

[83] ——, "Fast, accurate prediction of PLL jitter induced by power grid noise," in *Proc. IEEE Custom Integrated Circuits Conf.*, May 2004.

[84] X. Lai, Y. Wan, and J. Roychowdhury, "Fast PLL simulation using nonlinear VCO macromodels for accurate prediction of jitter and cycle-slipping due to loop non-idealities and supply noise," in *Proc ASP-DAC*, Jan. 2005.

[85] S. K. Tiwary and R. A. Rutenbar, "Scalable trajectory methods for on-demand analog macromodel extraction," in *Proc. ACM/IEEE Design Automation Conf.*, Jun. 2005.

[86] S. K. Tiwary, "Scalable trajectory methods for on-demand analog macromodel extraction," Ph.D. dissertation, Dep. Electrical Computer Eng., Carnegie Mellon Univ., Pittsburgh, PA, May 2006.

[87] M. Krasnicki, R. Phelps, R. A. Rutenbar, and L. R. Carley, "MAELSTROM: Efficient simulation-based synthesis for custom analog cells," in *Proc. DAC*, 1999, pp. 945–950.

[88] R. Phelps, M. Krasnicki, R. A. Rutenbar, L. R. Carley, and J. R. Hellums, "A case study of synthesis for industrial-scale analog IP: Redesign of the equalizer/filter frontend for an ADSL CODEC," in *Proc ACM/IEEE DAC*, Jun. 2000, pp. 1–6.

[89] G. Van der Plas, G. Debyser, F. Leyn, K. Lampaert, J. Vandenbussche, G. Gielen, W. Sansen, P. Veselinovic, and D. Leenaerts, "AMGIE: A synthesis environment for CMOS analog integrated circuits," *IEEE Trans. Computer-Aided Design*, vol. 20, no. 9, pp. 1037–1058, Sep. 2001.

[90] K. Antreich, J. Eckmller, H. Grb, M. Pronath, F. Schenkel, R. Schwencker, and S. Zizala, "Wicked: Analog circuit synthesis incorporating mismatch," in *IEEE Custom Integrated Circuits Conf. (CICC)*, May 2000.

[91] X. Li and L. Pileggi, "Robust analog/RF circuit design with projection-based posynomial," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2004.

[92] X. Li, J. Le, L. Pileggi, and A. J. Strojwas, "Projection-based performance modeling for inter/intra-die variations," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2005.

[93] Y. Freund, "Boosting a weak learning algorithm by majority," *Information Computation*, vol. 121, no. 2, pp. 256–285, 1995.

[94] F. De Bernardinis, M. I. Jordan, and A. Sangiovanni Vincentelli, "Support vector machines for analog circuit performance representation," in *Proc. ACM/IEEE DAC*, Jun. 2003.

[95] S. K. Tiwary, S. Velu, R. A. Rutenbar, and T. Mukherjee, "Pareto optimal modeling for efficient PLL optimization," in *Proc. Int. Conf. Modeling Simulation in Microsystems*, Mar. 2004.

[96] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, Aug. 1998.

[97] A. Singhee, "Performance feasibility regions of analog circuits: Computation and application," Dep. Electrical and Computer Eng., Carnegie Mellon Univ., Pittsburgh, PA, May 2005, Ph.D. Qualifying Exam Paper.

[98] M. Hershenson, S. Boyd, and T. H. Lee, "GPCAD: A tool for CMOS opamp synthesis," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, San Jose, CA, 1998, pp. 296–303.

[99] M. Hershenson, A. Hajimiri, S. S. Mohan, S. P. Boyd, and T. H. Lee, "Design and optimization of *LC* oscillators," in *Proc. IEEE/ACM Int. Conf. Computer Aided Design*, San Jose, CA, Nov. 1999.

[100] M. Hershenson, "Design of pipeline analog-to-digital converters via geometric programming," in *Proc. ACM/IEEE ICCAD*, Nov. 2002.

[101] D. M. Colleran *et al.*, "Optimization of phase-locked loop circuits via geometric programming," in *Proc. IEEE CICC*, May 2003.

[102] T. K. Ogawa and K. Kundert, "VCO jitter simulation and its comparison with measurement," in *Proc. ASP-DAC*, Jan. 1999.

[103] X. Li and L. Pileggi, "Asymptotic probability extraction for non-normal distributions of circuit performance," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2004.

[104] X. Li, J. Wang, W. Chiang, and L. Pileggi, "Performance-centering optimization for system-level analog design exploration," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2005.

[105] H. Chang *et al.*, "A top-down, constraint-driven design methodology for analog integrated circuits," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, 1992, pp. 8.4.1–8.4.6.

[106] H. Chang, E. Charbon, U. Choudhury, A. Demir, E. Felt, E. Liu, E. Malavasi, A. Sangiovanni-Vincentelli, and I. Vassiliou, *A Top-Down, Constraint-Driven Design Methodology for Analog Integrated Circuits.* Boston, MA: Kluwer, 1997.

ABOUT THE AUTHORS

Rob A. Rutenbar (Fellow, IEEE) received the Ph.D. degree in computer engineering from the University of Michigan, Ann Arbor, in 1984.

He subsequently joined the faculty at Carnegie Mellon University (CMU), Pittsburgh, PA, where he is currently the Stephen J. Jatras Professor of Electrical and Computer Engineering. His research focuses on optimization-based methods for custom circuit synthesis and modeling, for statistical analysis of deeply scaled circuits, and for automatic layout of complex analog and digital designs. He has published over 100 papers in the field of CAD for circuits and systems In 1998, on a leave of absence from CMU, he cofounded Neolinear, Inc., and served as its Chief Scientist until its acquisition by Cadence in 2004. He chaired Cadence Design Systems' Analog Technical Advisory Board from 1992 to 1996.

Dr. Rutenbar is the founding Director of the MARCO/DARPA Focus Center for Circuit & System Solutions (called 'C2S2'), a consortium of 17 major U.S. universities, supporting roughly 50 faculty investigators, funded by the U.S. semiconductor community and U.S. DOD (DARPA) to address future circuit design challenges. He is a 2001 winner of the Semiconductor Research Corporation (SRC) Aristotle Award for excellence in education. In 2002 he was honored with a University of Michigan Alumni Society Merit Award for Electrical Engineering. He is also a 2004 winner of an SRC Technical Excellence Award for his contributions to "Electronic Design Automation for Analog/Mixed-Signal Design" and recipient of numerous Best Paper awards (e.g., ACM/IEEE Design Automation Conference, in 1987 and again in 2002). He was the General Chair of the 1996 ACM/IEEE International Conference on CAD. He is a member of the ACM and Eta Kappa Nu.

Georges G. E. Gielen (Fellow, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from the Katholieke Universiteit Leuven, Belgium, in 1986 and 1990, respectively.

In 1990, he was appointed as a Postdoctoral Research Assistant and Visiting Lecturer at the Department of Electrical Engineering and Computer Science of the University of California, Berkeley. From 1991 to 1993, he was a Postdoctoral Research Assistant of the Belgian National Fund of Scientific Research at the ESAT Laboratory of the Katholieke Universiteit Leuven. In 1993, he was appointed Assistant Professor at the Katholieke Universiteit Leuven, where he was promoted to full Professor in 2000. His research interests are in the design of analog and mixed-signal integrated circuits, especially in analog and mixed-signal CAD tools and design automation (modeling, simulation and symbolic analysis, analog synthesis, analog layout generation, analog and mixed-signal testing). He is a coordinator or partner of several (industrial) research projects in this area. He has authored or coauthored two books and more than 300 papers in edited books, international journals, and conference proceedings.

Dr. Gielen serves regularly as a member of the program committees of international conferences (DAC, ICCAD, ISCAS, DATE, CICC), and served as General Chair of the DATE conference in 2006. He has been a member of editorial boards of international journals including IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, *Springer International Journal on Analog Integrated Circuits and Signal Processing*, and *Elsevier Integration*. He received the 1995 Best Paper Award in the *John Wiley International Journal on Circuit Theory and Applications*, and was the 1997 Laureate of the Belgian Royal Academy on Sciences, Literature and Arts in the discipline of Engineering. He received the 2000 Alcatel Award from the Belgian National Fund of Scientific Research for his innovative research in telecommunications and won the DATE 2004 Best Paper Award. He served as and elected member of the Board of Governors of the IEEE Circuits And Systems (CAS) society and as Chairman of the IEEE Benelux CAS chapter. He served as the President of the IEEE Circuits And Systems (CAS) Society in 2005.

Jaijeet Roychowdhury (Senior Member, IEEE) received the B.S. degree in electrical engineering from the Indian Institute of Technology, Kanpur, in 1987, and the Ph.D degree in electrical engineering and computer science from the University of California, Berkeley, in 1993.

From 1993 to 1995, he was with the CAD Lab of AT&T's Bell Laboratories, Allentown, PA. From 1995 to 2000, he was with the Communication Sciences Research Division of Lucent's Bell Laboratories, Murray Hill, NJ, and from 2000 to 2001, with CeLight, Inc., an optical networking startup, Silver Spring, MD. In 2001, he joined the ECE Department and the Digital Technology Center of the University of Minnesota, Minneapolis, as an Associate Professor. His professional interests include the design, analysis, and simulation of electronic, electro-optical and mixed-domain systems, particularly for high-speed and high-frequency communications. He holds ten patents.

Dr. Roychowdhury has received Distinguished or Best Paper Awards at ICCAD 1991, DAC 1997, ASP-DAC 1997 and ASP-DAC 1999, was cited for Extraordinary Achievement by Bell Laboratories, and has served on the Technical Program Committees of DAC, ICCAD, BMAS and SCEE.