Automated Extraction of Broadly Applicable Nonlinear Analog Macromodels from SPICE-level Descriptions

Ning Dong, Jaijeet Roychowdhury
ECE Department, University of Minnesota
200 Union St. SE, Minneapolis, MN 55414
Email: {ningd, jr}@ece.umn.edu

Abstract— **Automated techniques for generating macromodels from SPICE-level circuit descriptions are rapidly gaining importance as a sustainable methodology for the design of large, complex mixed-signal SoCs and SiPs. In this paper, we demonstrate the efficacy of a novel macromodel extraction technique, dubbed PWP, for extracting broadly-applicable general-purpose macromodels from SPICE netlists. A key advantage of PWP over other methods is that it can generate a single macromodel that captures linear, weakly nonlinear and strongly nonlinear system dynamics. We demonstrate the application of PWP using a current-mirror op-amp, comparing simulations of the macromodel against those of the original SPICE circuit using DC, AC, harmonic balance and transient analyses. We also illustrate how PWP-generated macromodels can be used for system-level simulation using a simple analog-digital converter example. We confirm excellent accuracies, relative to full SPICE circuit simulation, while achieving order-of-magnitude speedups.**

I. Introduction

Dramatic increases in the complexity of modern mixed-signal integrated systems have exacerbated the gap between the capabilities that current CAD tools offer and the needs of circuit and system designers working to complete designs in highly competitive and time-limited environments. It has become imperative to verify in detail not only the performance of individual analog/mixed-signal/RF blocks during design, but also the end-to-end functionality of entire systems consisting of many such blocks. Using simple manually-created abstractions ("macromodels") of individual blocks for system-level verification, as has traditionally been the practice, is, however, breaking down as an effective system verification paradigm. Simple macromodels typically do not represent the effects of various undesired intra- and inter-block interactions and important device non-idealities, which have become of paramount concern especially in deep-submicron (DSM) technologies. The most serious interaction problems are usually difficult to quantify accurately during initial design, resulting as they do from a combination of process, technology, circuit and interconnect factors that become evident only after layout and parasitic extraction. Such problems are often felt most in cutting-edge on-chip mixed-signal/RF systems, where it is not uncommon for 5 or more design/fabrication/re-design cycles to be needed for a properly functioning product.

The most accurate way to verify functionality before fabrication would be to simulate the entire system with all blocks at the SPICE level; however, this is practically infeasible due to long simulation times even for systems with relatively few complex blocks. It is in this context that there has been much recent interest in CAD techniques for *extracting accurate, yet computationally inexpensive, macromodels of circuit blocks* directly from their (layout-extracted) SPICE-level descriptions (*e.g.*, [1], [4], [8], [10] and *etc.*). One of the attractions of employing appropriate algorithms to extract macromodels (instead of creating them manually) is that effects of non-idealities, parasitics, undesired interactions, *etc.*, at the transistor level can be captured well in the macromodels. Good macromodels that are automatically extracted therefore fill the gap between system-level and circuit/block-level design, by affording system designers the capability of accessing transistor-level details at much lower computational cost, and without necessarily knowing anything about the underlying operation or structure of the blocks.

Although the problem of extracting good macromodels that capture relevant nonlinearities is a very challenging one, there has been considerable recent activity in this area (*e.g.*, [1], [2], [4], [5], [7], [8], [10]). Without providing a detailed review of the available approaches here[1], we note that existing methods are usually targeted towards capturing only some properties of a given block, and do not produce general-purpose, drop-in replacement macromodels. For example, table-based methods capture DC input/output behaviour; small-signal linear approaches capture only the linear dynamics around a single DC operating point; some "large-signal" macromodelling approaches capture the relationship between, say, input sinusoids over certain amplitude/frequency ranges and output components and harmonics; other weakly nonlinear approaches can capture small-signal distortion/intermodulation effects, but not large-signal clipping or slewing; while yet others are better at capturing clipping/slewing but poor for distortion and intermodulation. However, there appear to be no CAD methods current available that extract *broadly applicable, general-use macromodels*, *i.e.*, macromodels that reproduce the original well under the variety of analyses conducted during a typical design cycle. These analyses include DC sweeps, AC analysis, small signal (often frequency-domain) distortion and intermodulation, large-signal transient analysis, as well as time-domain/frequency-domain steady-state and envelope-following simulations.

In this paper, we demonstrate the utility of a recently-developed technique for automated nonlinear macromodelling, namely PWP [1], for creating broadly-applicable macromodels suitable for these and other analyses. PWP can extract, from a SPICE-level netlist, a series of macromodels that trade off accuracy against computational efficiency. A key feature of PWP is that it produces a single macromodel that captures important linear and nonlinear dynamics, including both weakly nonlinear effects (such as distortion/intermodulation) as well as strongly nonlinear ones (such as clipping and slewing). PWP is not limited to any particular topology or type of circuit; it is generally applicable. Macromodels produced by PWP are in the form of small systems of nonlinear differential equations that reproduce the input-output relationships of the original circuit well. As such, they are easily cast into any format convenient for use within system-level simulation tools, such as MATLAB/Simulink (used in this work), Verilog-A, VHDL-AMS, and even as SPICE subcircuits. As we demonstrate, the macromodels are equally well suited for time and frequency domain simulations, and also for mixed frequency/time analyses such as envelope-following simulations (*e.g.*, [11]).

We apply PWP to macromodel an op-amp, and provide comparisons of the original vs the extracted macromodel using several analyses, namely DC, AC, small-signal distortion (using harmonic balance), and transient simulations. We then compose a simple ADC circuit using the op-amp, and demonstrate that using the PWP-generated macromodel matches full simulations virtually exactly. We obtain speedups of an order of magnitude using the macromodels, and anticipate further speedup improvements (of about another order of magnitude) for larger blocks with the more complex device models used in industry.

Even in its current, relatively nascent state of development, we expect PWP to be of practical benefit in industrial design. With further refinement, we anticipate that the capabilities provided by PWP, and automated macromodelling methods in general, can result in significant benefits to mixed-signal design methodologies. Especially for the many time-consuming simulations carried out during system-level refinement along the way to a final design, we envisage dramatic reductions in simulation time by replacing

[1]We refer the interested reader to, *e.g.*, [12], for a survey of available approaches.

SPICE-level circuit blocks with PWP-generated macromodels, with very little given up in accuracy or predictive capability. Indeed, we expect extracted-macromodel based verification methodologies to be critically important in maintaining design productivity for the increasingly complex mixed-signal systems of the future.

The remainder of the paper is organized as follows. In Section II, we provide a brief overview of the PWP algorithm for automated macromodel extraction. In Section III, we apply PWP to a current-mirror op-amp circuit, and compare the resulting macromodel against the original using a variety of analyses. Finally, in Section IV, we use the op-amp macromodel to build a small ADC block and perform a small system-level simulation.

II. Overview of PWP

In this section, we present a brief review of the PWP method for extracting general-use macromodels from SPICE-like descriptions. Additional details are available in [1].

A difficulty in producing generally-applicable nonlinear macromodels stems from the requirement of reproducing faithfully *both* weak local nonlinearities (important for distortion and intermodulation), as well as large-signal global nonlinearities (important for clipping, slewing, *etc.*). The former can be captured well using low-order polynomial model reduction approaches (*e.g.*, [4], [6], [10]). In order to also capture large-signal global nonlinearities as well, the PWP approach extends the TPWL approach [8] by stitching separate polynomials together, in a roughly spline-line fashion, across separate regions in the system's state space [1].

In all SPICE-level simulators, electrical circuits are described by a set of differential algebraic equations (DAEs [10])

$$E\dot{x} = f(x) + Bu(t), \quad y(t) = Cx(t), \tag{1}$$

where $u(t)$ represents the input(s) to the circuit and $y(t)$ the output(s). $x(t)$ is the state vector of (many) internal node voltages and branch currents; E and $f(\cdot)$ capture the linear and nonlinear charge/flux and current terms from all the devices; C captures the output from the internal state. Extracting a macromodel from (1) involves finding a similar system of equations with a *state vector much smaller* than that of the original system. The macromodel's input-output characteristics must replicate that of the original within acceptable accuracy.

A. Polynomial-based nonlinear macromodelling

In polynomial model reduction, a Taylor series expansion of the nonlinear function $f(\cdot)$ is first obtained to yield (say) a quadratic model

$$E\dot{x} = f(x^*) + J(x - x^*) + H(x - x^*)^{(2)} + Bu(t), \tag{2}$$

where J and H are the $1st$ and $2nd$ derivatives of $f(x)$ evaluated at the operating point x^*. $(x - x^*)^{(2)} = (x - x^*) \otimes (x - x^*)$ is vector Kronecker (tensor) product. Reduction of (2) to a similar form but with a much smaller internal size is based on Krylov-subspace projection (*e.g.* [3]), which involves a *projection basis* V, obtained using techniques described in [4], [6], [9]. V, a rectangular matrix with far fewer columns than rows, is used to project the original (large) state-space x to a much smaller one z via $x = Vz$, resulting in the *macromodelled system*

$$\begin{cases} \hat{E}\dot{z} = \hat{f}(x^*) + \hat{J}(z - z^*) + \hat{H}(z - z^*)^{(2)} + \hat{B}u(t), \\ y = \hat{C}z. \end{cases} \tag{3}$$

The reduced matrices are: $\hat{E} = V^T EV$, $\hat{B} = V^T B$ and $\hat{C} = CV$, *etc.*. It has been shown (*e.g.*, [4], [6], [9]) that such a reduced polynomial model capture small signal distortions and intermodulations well.

B. Piecewise polynomial (PWP) macromodelling

When deviations from the operating point x^* become large, the error of the single-piece polynomial model (3) increases rapidly, due to limitations of Taylor series representations. To maintain good *global* fidelity, the PWP approach extends TPWL [8] by dividing the nonlinear state-space into different regions, each of which is fitted with a polynomial model around the center expansion point. These points can be selected either from "training simulation" or from DC sweeps. The resulting macromodel is refined incrementally by adding new piecewise regions until a desired accuracy is met. The separate regions are stitched together with smooth scalar weight functions to obtain the PWP-reduced macromodel

$$\begin{aligned} \hat{E}\dot{z} &= \sum_{i=1}^{s} w_i(z)(\hat{f}(x^*) + \hat{J}(z - z_i^*) + \cdots + \hat{B}u(t)), \\ y &= \hat{C}z, \end{aligned} \tag{4}$$

where s is the number of regions, $\hat{f}(x_i^*) = V_c^T f(x_i^*)$, $\hat{E} = V_c^T EV_c$, $\hat{B} = V_c^T B$ and $\hat{C} = CV_c$. $w_i(z)$ are the weight functions, which "choose" the piecewise region closest to the current point z being considered by multiplying the contribution of each region with a scalar weight that varies between 0 and 1.

III. Current-mirror Op-amp: Macromodel Generation and Accuracy Assessment

In this section, we conduct an in-depth evaluation of PWP-generated macromodels using a current mirror op-amp (Fig. 1) as a representative test-case[2] for validation. The fully differential op-amp, which includes a common-mode feedback circuit, has 50 MOSFETs and 39 nodes. It was designed to provide about 70dB of DC gain, with a slew rate of $20V/\mu s$ and an open-loop 3dB-bandwidth of $f_0 \approx 10kHz$. The MOS devices were modelled using a smooth, bulk-referred version of the Schichman-Hodges (MOS Level 1) equations[3]. It should be noted that PWP-generated macromodels automatically abstract relevant features of all underlying device models in the original circuit, no matter they are simple or complex. We compare the macromodel 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. Runtime and speedup numbers are presented.

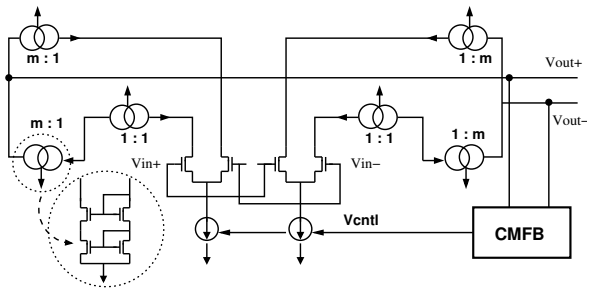


Fig. 1. Current-mirror op-amp with 50 MOSFETs and 39 nodes

To generate the PWP macromodel, we first obtained a "training trajectory" by transient simulation with the input

$$V_{in}^+ = V_{dd}\frac{t}{10^{-5}}, \quad V_{in}^- = \frac{V_{dd}}{2}, \quad V_{dd} = 5V, t = 0 \sim 10^{-5}s. \tag{5}$$

The choice of this input was dictated by a desire to exercise the circuit through all its important nonlinear and dynamical behaviours. As mentioned in Section II and elaborated in [1], expansion points are chosen partly along the trajectory and partly from a DC sweep of the full circuit. The PWP-generated macromodel has 47 piecewise

[2]We emphasize that PWP is not limited to macromodelling op-amps – it is applicable to any kind of circuit, or even multi-physics systems, whose equations can be cast in the form of (1).

[3]The choice of Schichman-Hodges, rather than a full-featured MOS model like BSIM, was dictated by a lack of comprehensive device model support in the MATLAB-based protyping tool we use for implementing the PWP algorithm. We emphasize strongly that the choice of model makes *no difference* to the efficacy of PWP – indeed, with more expensive-to-compute models like BSIM in the original circuit, we expect order-of-magnitude increases in speedup (over those reported here) of PWP-generated macromodels vs the full SPICE circuits. We are currently in the process of implementing BSIM support in our prototyping framework.

regions, each approximated by a polynomial model with a state-space of size 24[4].

A. DC and AC analyses

We first perform DC sweep analysis and compare the results of full op-amp with that of PWP-generated macromodel. As shown in Fig. 2, two models are precisely matched.

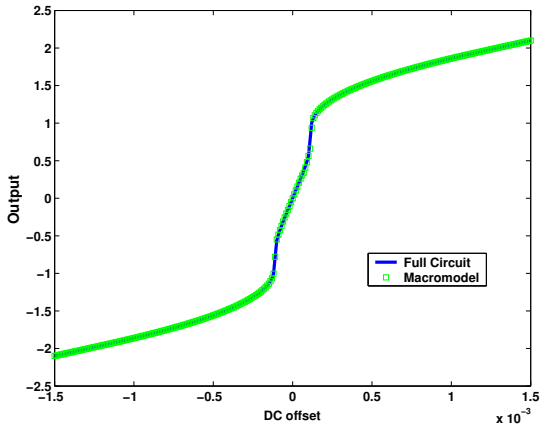
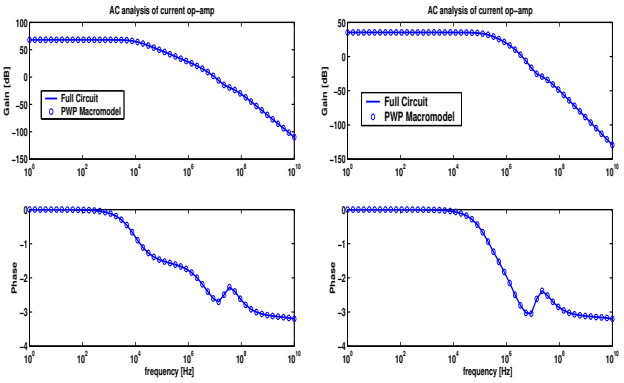


Fig. 2. DC sweep

Next, we compare Bode plots, obtained by AC analysis, of the PWP-generated macromodel against those of the full op-amp. Two AC sweeps, obtained at different DC biases, are shown in Fig. 3. Note that PWP provides excellent matches around *each* bias point.



(a) $V_{in}^{+} = V_{in}^{-} = 2.5V$ (b) $V_{in}^{+} = V_{in}^{-} = 2.0V$

Fig. 3. AC analysis with different DC bias

B. Distortion via frequency-domain (Harmonic Balance) simulations

When the op-amp is used as a linear amplifier with small inputs, distortion and intermodulation are important performance metrics. 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

$$V_{in}^{+} = A\sin(2\pi \times 100t), \quad V_{in}^{-} = 2.5. \tag{6}$$

[4]Even though the reduction in state-space size is not dramatic in this case, we nevertheless obtain order-of-magnitude speedups. With further improvements in the PWP algorithm, leading to greater state-space compression, this speedup is expected to become considerably larger.

The input magnitude A is swept over several decades, and the first two harmonics plotted in Fig. 4. It can be seen that for the entire

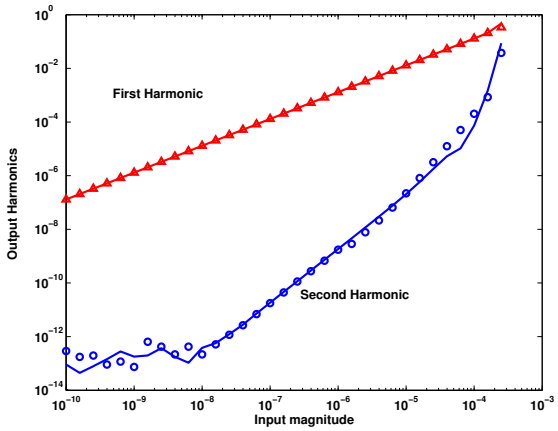


Fig. 4. Harmonic analysis of current-mirror op-amp: solid line – full op-amp; discrete point – PWP model

input range, there is an excellent match of the distortion component from the macromodel vs that of the full circuit (at very small input magnitudes, the distortion component of both is dominated by numerical noise). Note that the *same* macromodel (essentially a subcircuit) is used for this harmonic balance simulation as for all the other analyses presented.

C. Slewing/clipping via time-domain (transient) simulations

Another strength of PWP is that it can capture the effects of strong nonlinearities, excited by large signal swings, well. To demonstrate this, a transient analysis was run with the large input

$$V_{in}^{+} = 0.1\sin(2\pi \times 10^{5}t), \quad V_{in}^{-} = 2.5. \tag{7}$$

The input frequency was chosen to excite slew-rate limiting, a dynamical phenomenon caused by strong nonlinearities (saturation of differential amplifier structures); hard limiting due to the power and ground rails is also present. Comparisons of the macromodel vs the original are shown in Fig. 5. The excellent match between

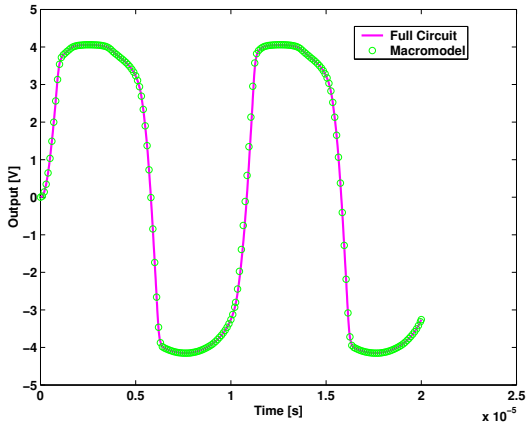


Fig. 5. Transient analysis of current-mirror op-amp with clear evidence of slew rate limitation

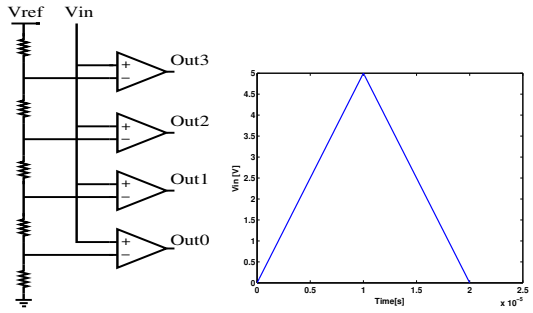
the PWP-generated macromodel and the original circuit is evident. This simulation, the most expensive of all the above, took 351.7s[5] for the full system, vs 39.05s for the PWP-generated macromodel,

[5]Simulations represent apples-to-apples comparisons in the same MATLAB prototyping environment running on a 1.8GHz Pentium 4 CPU under Linux.

representing a 9× speedup. Slightly smaller speedups were obtained for the harmonic balance simulations above (8.1× – 125.9s for the full simulation vs 15.58s for the macromodel).

IV. Evaluation of Macromodel Utility: Simulation of a Simple 4-bit ADC

The main purpose of generating macromodels is to use them to speed simulation of bigger ("system-level") blocks. Here, we construct a simple 4-bit ADC (Fig. 6(a)) using the current-mirror op-amp from the previous section (Fig. 1) as comparators. The input V_{in} is ramped from 0 to $V_{dd} = 5V$, and back again, as shown in Fig. 6(b).



(a) Simple 4-bit ADC using comparators.

(b) Input waveform to ADC.

Fig. 6. A simple ADC with swapping input

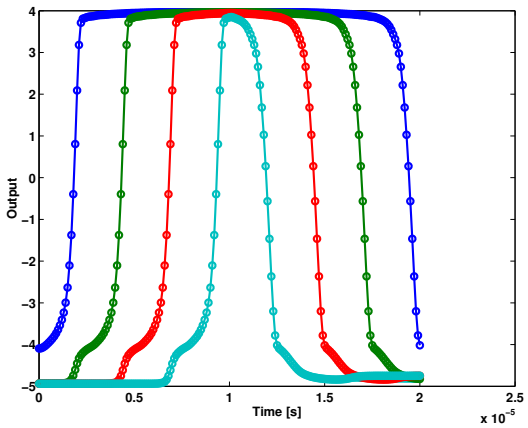


Fig. 7. Transient simulation of analog-digital converter: solid line – full system; discrete dot – macromodelled system. Waveforms from left to right: MSB through LSB.

To test the accuracy provided by the PWP-generated macromodel in a system, we compared simulations of the ADC implemented using full transistor-level op-amp blocks against the same system with the op-amp blocks replaced by the macromodel of the previous section. Transient simulation results[6] are shown in Fig. 7. It can be seen that the macromodel provides excellent fidelity, capturing the sharp slope and the clipped corner. A speedup of 9.4× resulted from using the macromodels (1453.9s for the full simulation vs 147.79s using the macromodels). The total model generation time is 411.5s, including 154.9s for transient analysis, 45.3s for a number of DC analysis and the rest 211.3s for calculating projection bases and *etc.*.

[6]The negative value is from differential output $V_{out}^{+} - V_{out}^{-}$.

Thus it is evident that PWP-generated macromodels can be profitably employed as general-purpose drop-in replacements in system-level simulation, resulting in attractive speedups with little or no loss of accuracy. As mentioned earlier, this can have a significant impact on methodologies for mixed-signal design, with simulation times during the many intermediate steps of design refinement reduced dramatically through the use of macromodels for most system blocks. We repeat that PWP-generated macromodels are easily targeted to a variety of model-description languages, including MATLAB/Simulink blocks (used in this work), Verilog-A, VHDL-AMS, and even SPICE subcircuits. We note that it is prudent towards the final stages of a system design to always perform full simulations, to the extent feasible with detailed SPICE-level blocks, and confirm the predictions of prior macromodel-based simulations.

V. Conclusions

In this paper, we have demonstrated that piecewise polynomial (PWP) macromodelling is an effective approach for automated nonlinear macromodel extraction. With one single macromodel, PWP is able to capture important linear and nonlinear dynamical features of SPICE-level circuits, including distortion, slewing and clipping. We have demonstrated the utility of PWP by macromodelling an op-amp and verifying it with DC, AC, Harmonic Balance, and transient analyses. We have also put together a simple 4-bit ADC using the PWP-generated macromodel to demonstrate its value for system-level simulation. We have obtained speedups of about one order of magnitude, with further improvements expected for large circuit blocks employing expensive SPICE-level device models.

Acknowledgments

It is a pleasure to acknowledge a number of individuals who have had important influences on this work: Gaurav Chandra, Preetam Charan, Ankit Sreedhar, Peter Fang, Vinod Gupta, Lawrence Arledge, and David Yeh (Texas Instruments); Juan-Antonio Carballo (IBM Austin Research Laboratory); Jacob White (MIT), Joel Phillips (Cadence Berkeley Labs) and Rob Rutenbar (CMU). Support from the NSF (awards CCR-0312079 and CCR-0204278), the SRC and DARPA is gratefully acknowledged. Computational resources used include those from the Digital Technology Center and the Supercomputing Institute of the University of Minnesota.

References

[1] N. Dong and J. Roychowdhury. Piecewise Polynomial Nonlinear Model Reduction. *Proc. IEEE DAC*, 2003.

[2] T. Eeckelaert, W. Dames, G. Gielen, and W. Sansen. Generalized posynomial performance modeling. *Design, Automation and Test in Europe Conference and Exhibition*, 2003.

[3] R. W. Freund. Krylov-subspace methods for reduced-order modeling in circuit simulation. *Journal of Computational and Applied Mathematics*, 123:395–421, 2000.

[4] P. Li and L. T. Pileggi. NORM: Compact Model Order Reduction of Weakly Nonlinear Systems. *Proc. IEEE DAC*, 2003.

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

[6] J. Phillips. Projection frameworks for model reduction of weakly nonlinear systems. In *Proc. IEEE DAC*, June 2000.

[7] J. Phillips. Analog Macromodeling Using Kernel Methods. *Proc. ICCAD*, 2003.

[8] 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.

[9] J. Roychowdhury. Reduced-order modelling of linear time-varying systems. In *Proc. ICCAD*, Nov. 1998.

[10] J. Roychowdhury. Reduced-order modelling of time-varying systems. *IEEE Trans. Ckts. Syst. – II: Sig. Proc.*, 46(10), Nov. 1999.

[11] J. Roychowdhury. Making Fourier-envelope simulation robust. In *Proc. ICCAD*, Nov. 2002.

[12] J. Roychowdhury. An overview of automated macromodelling techniques for mixed-signal systems. *Proc. IEEE CICC*, 2004.