Compact Modeling of Nonlinear Analog Circuits using System Identification via Semi-Definite Programming and Robustness Certification

Compact Modeling of Nonlinear Analog Circuits Using System Identification Via Semidefinite Programming and Incremental Stability Certification

Bradley N. Bond, Zohaib Mahmood, Yan Li, Ranko Sredojević, Alexandre Megretski, Vladimir Stojanović, Yehuda Avniel, and Luca Daniel

***Abstract*—This paper presents a system identification technique for generating stable compact models of typical analog circuit blocks in radio frequency systems. The identification procedure is based on minimizing the model error over a given training data set subject to an incremental stability constraint, which is formulated as a semidefinite optimization problem. Numerical results are presented for several analog circuits, including a distributed power amplifier, as well as a MEM device. It is also shown that our dynamical models can accurately predict important circuit performance metrics, and may thus, be useful for design optimization of analog systems.**

***Index Terms*—Analog macromodeling, model reduction, nonlinear systems, semidefinite programming, system identification.**

I. Introduction

AUTOMATIC generation of accurate compact models for nonlinear circuits (e.g., power amplifiers, or low-noise amplifiers) could enable very efficient simulation, design and optimization of complex integrated circuit systems. However, the only tool currently available to analog designers and system architects, is to manually generate analytical or semi-empirical behavioral models. Such a critical procedure mostly relies on the designers' experience and intuition, together with time consuming simulations. Circuit simulators (such as SPICE and SPECTRE) automatically construct large dynamical system models from schematics by combining conservation laws (e.g., Kirchhoff's current law) with the constitutive relations for each device in the system

$$\dot{q}(x) = f(x, u). \tag{1}$$

Here u is the input, x is the possibly huge state vector containing, for instance, all of the node voltages and inductor currents in the circuit, and $q(x)$ and $f(x, u)$ are the nonlinear vector functions defined by the circuit schematic and device models. Simulation of a complex analog system, such as a radio frequency (RF) receiver chain, is therefore computationally extremely expensive, as it may require solving thousands of coupled nonlinear ordinary differential equations (ODEs). Hence, during the recent years, a great effort has been dedicated by researchers to develop techniques for generating automatically accurate compact models of nonlinear system blocks.

The majority of existing compact modeling techniques involve "reducing" the large nonlinear systems produced by circuit schematics, or parasitic extractors. Some techniques have been proven on weakly nonlinear systems [1]–[5], while others can handle strongly nonlinear systems [6]–[13]. All these approaches typically employ a linear state-space projection, $x = V\hat{x}$ (where V is a "tall and skinny" change of basis matrix) and introduce low-complexity approximations $\hat{q}(\hat{x})$ and $\hat{f}(\hat{x}, u)$ in order to obtain a low-order system of nonlinear ODEs

$$\dot{\hat{q}}(\hat{x}) = \hat{f}(\hat{x}, u).$$

However, one shortcoming of such techniques is the extreme difficulty in preserving stability in the reduced model. Additionally, such model reduction approaches require knowledge of the original model expressions $q(x)$ and $f(x, u)$ from (1). This requires access to not only the schematic of the circuit, which is typically readily available, but also the exceedingly complicated transistor models, which are not always easily accessible.

In this paper, we present an alternative approach to achieve the same final goal, i.e., the automatic generation of accurate compact models, without "reducing" a given large system, but rather using a system identification approach to model reduction. The term system identification (SYSID) refers to the task of finding a stable dynamical model of low complexity that delivers the best match for a collection of dynamical input–output (or input-state-output) data. In classical control applications, the data is usually available in the form of actual physical measurements, and SYSID provides adequate models for systems for which no reliable first principles equations are available, either due to parameter uncertainty or system complexity. In integrated circuit applications, SYSID is the only viable option in generating compact models of circuits blocks when only input–output physical measurements are

Manuscript received June 14, 2009; revised September 17, 2009 and December 9, 2009. Date of current version July 21, 2010. This work was supported in part by the DARPA, under Grant N66001-09-1-2068, the Interconnect Focus Center, one of five research centers funded under the Focus Center Research Program, a Semiconductor Research Corporation and DARPA program, and by the Center for Integrated Circuits and Systems at Massachusetts Institute of Technology, Cambridge, MA. This paper was recommended by Associate Editor G. Gielen.

The authors are with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: bnbond@mit.edu; zohaib@mit.edu; liyan@mit.edu; rasha@mit.edu; ameg@mit.edu; vlada@mit.edu; avniel@mit.edu; luca@mit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2010.2049155

available. In addition, although it is true that in many cases circuit schematics of the original system are actually available, SYSID often still remains the most practical solution to compact modeling.

Within the control community, SYSID for linear-time-invariant (LTI) system is well understood and mature [14]. One can argue that also some of the approaches developed by the Electronic Design Automation community for LTI model order reduction could be interpreted as SYSID approaches, such as those based on transfer function fitting via least squares or optimization techniques [15]–[18].

Conversely, SYSID for nonlinear systems is still a problem that needs to be addressed on a case by case basis [19], [20]. Among the most general and used approaches in behavioral modeling one finds the Volterra series method [21]–[23]. In some more specific approaches, one assumes an internal structure (e.g., a Wiener [24], [25], or Wiener–Hammerstein [14], [19], [26], [27] or Wiener–Hammerstein with feedback structure [28]), and proceeds in identifying the coefficients for such structures [29], [30]. As a general observation, a significant difficulty in implementing any kind of SYSID based approach is caused by lack of efficient SYSID tools for generic nonlinear systems.

In this paper, we propose a new SYSID method for compact modeling of nonlinear circuit blocks and micro-electromechanical (MEM) components. Our approach is based on optimizing system coefficients to match given data while simultaneously enforcing stability. What distinguishes our SYSID method from existing "reduction" approaches is the ability to explicitly preserve the properties of nonlinear systems, such as stability, while controlling model accuracy. The efficiency issues encountered by past SYSID techniques are addressed in our approach by adopting recently developed semidefinite programming techniques for nonlinear system analysis [31]. Additionally, we have provided MATLAB code implementing our approach [32] to aid the reader in implementing our technique.

The remainder of the paper is organized as follows. In Section II, we summarize related background. In Section III, we develop the theoretical framework for our proposed SYSID approach to compact modeling of nonlinear systems and formulate the identification problem as a semidefinite program. In Section IV, we present one approach to solve efficiently the previously derived optimization problem by selecting a polynomial basis and rational model formulation, resulting in a sum of squares (SOS) problem. In Section V, we describe in detail how to implement the proposed procedure using freely available software. Finally, in Section VI we show the effectiveness of the proposed approach in modeling practical circuit blocks, such as low-noise amplifiers, power amplifiers, and MEM devices. The proposed approach is also compared to several existing SYSID techniques.

II. Background

A. Stability of Dynamical Systems

A difficult, yet crucial, aspect of model reduction is the preservation of stability in the reduced model. Most real physical systems behave in a stable manner, and it is therefore extremely important to preserve such behavior in reduced models. For example, stability might require that bounded inputs produce bounded outputs, or that the system does not generate energy.

One strong notion of stability is "incremental stability," which guarantees that perturbations to solutions decay to zero. As a result, incremental stability is an extremely important property for the purpose of simulation. Consider a nonlinear discrete time system implicitly defined as follows:

$$\begin{aligned} &F(v[t], v[t-1], \ldots, v[t-m], u[t], \ldots, u[t-k]) = 0 \\ &G(y[t], v[t]) = 0 \end{aligned} \tag{2}$$

where $v[t] \in \mathbb{R}^N$ is a vector of internal variables, $y[t] \in \mathbb{R}^{N_y}$ is the output, $u[t] \in \mathbb{R}^{N_u}$ is the input, $F \in \mathbb{R}^N$ is a dynamical relation between the internal variables and the input, and $G \in \mathbb{R}^{N_y}$ is a static relationship between the internal variables and the output.

Definition 1: System (2) is *well-posed* if given any arbitrary variables $v_1, \ldots, v_m \in \mathbb{R}^N$ and $u_0, \ldots, u_k \in \mathbb{R}^{N_u}$, there exist unique solutions $v_0 \in \mathbb{R}^N$ and $y \in \mathbb{R}^{N_y}$ to $F(v_0, v_1, \ldots, v_m, u_0, \ldots, u_k) = 0$ and $G(y, v_0) = 0$.

Definition 2: System (2) is *incrementally stable* if it is well-posed and, given any two sets of initial conditions $\bar{v}[t_0 - 1], \ldots, \bar{v}[t_0 - m]$ and $\hat{v}[t_0 - 1], \ldots, \hat{v}[t_0 - m]$, the resulting two solutions to (2) in response to the same input u satisfy

$$\sum_{t=t_0}^{\infty} \|\bar{y}[t] - \hat{y}[t]\|^2 < \infty \tag{3}$$

for all initial conditions and inputs.

Note that incremental stability implies traditional weaker notions of stability.

For the remainder of the paper, we shall use the following compact notation:

$$V = [v_0, \ldots, v_m], \quad U = [u_0, \ldots, u_k] \tag{4}$$

where $v_0, \ldots, v_m$ and $u_0, \ldots, u_k$ are arbitrary variables, not necessarily inputs and outputs satisfying (2)

$$V_+ = [v_0, \ldots, v_{m-1}], \quad V_- = [v_1, \ldots, v_m] \tag{5}$$

where V_+ contains the first m components of V and V_- contains the last m components of V

$$V[t] = [v[t], \ldots, v[t-m]], \quad U[t] = [u[t], \ldots, u[t-k]] \tag{6}$$

where $v[t]$ is the internal state of the identified model (2) in response to past inputs $U[t]$ and initial conditions $v[t-1], \ldots, v[t-m]$, i.e., $F(V[t], U[t]) = 0$, and

$$\tilde{V}[t] = [\tilde{v}[t], \ldots, \tilde{v}[t-m]], \quad \tilde{U}[t] = [\tilde{u}[t], \ldots, \tilde{u}[t-k]] \tag{7}$$

where $\tilde{v}[t]$ are training data state samples in response to training inputs $\tilde{u}[t]$. Similarly, y shall represent an arbitrary variable, $y[t]$ is the solution to $G(y[t], v[t]) = 0$, and $\tilde{y}[t]$ is a given training data output sample.

In general, stability (and more generally, dissipativity) can be proven through the use of storage functions [33].

Definition 3: System (2) is *dissipative* with respect to the supply rate $\sigma(u, v, y)$ if there exists a storage function $h(v) \geq 0$ such that

$$h(V_+) \leq h(V_-) + \sigma(U, V, y) \tag{8}$$

for all V, U, y satisfying (2).

Constraint (8) is referred to as a dissipation constraint, and different supply rates σ are used to prove different notions of stability. It has been shown in [31] that (2) (assuming $G = y - v_0$, i.e., an input–output system) is incrementally stable if the following dissipation constraint is satisfied:

$$\begin{aligned}(v_0 - \hat{v}_0)^T \left(F(V, U) - F(\hat{V}, U)\right) - |v_0 - \hat{v}_0|^2 \\ +h(V_-, \hat{V}_-) - h(V_+, \hat{V}_+) \geq 0\end{aligned} \tag{9}$$

for all V, U, and all $\hat{V} = [\hat{v}_0, \ldots, \hat{v}_m]$, where h is a nonnegative storage function such that $h(V_+, V_+) = 0$. Note that when V, U and $\hat{V}, U$ satisfy (2), dissipation constraint (9) simplifies to constraint (8) with $\sigma = -|v_0 - \hat{v}_0|$, which in turn implies (3).

Incremental stability can also be interpreted as the result of contraction behavior of the state-space [34]. Contraction analysis examines the stability of the differential system

$$\begin{aligned}F(V, U) + F_v(V, U)\Delta = 0 \\ G(y, v_0) + G_v(y, v_0)\delta_0 + G_y(y, v_0)\xi = 0\end{aligned} \tag{10}$$

where

$$\Delta = \begin{bmatrix}\delta_0 \\ \vdots \\ \delta_m\end{bmatrix} \qquad \Delta[t] = \begin{bmatrix}\delta[t] \\ \vdots \\ \delta[t-m]\end{bmatrix}$$

$$F_v = \left[\frac{\partial F}{\partial v_0}, \ldots, \frac{\partial F}{\partial v_m}\right] \qquad G_v = \frac{\partial G}{\partial v_0} G_y = \frac{\partial G}{\partial y}$$

with $\delta_0 \in \mathbb{R}^N$ and $\xi \in \mathbb{R}^{N_y}$. The system is said to be contracting if the increments Δ and ξ converge to zero exponentially. According to Theorem 3 in [34], if system (10) is well-posed and stable in the differential variable Δ, i.e., Δ converges exponentially to zero, for all y, $v_0, \ldots, v_m$, and $u_0, \ldots, u_k$ satisfying $F(v_0, \ldots, v_m, u_0, \ldots, u_k) = 0$ and $G(y, v_0) = 0$, then system (2) is incrementally stable. It is often easier to prove stability by examining the differential system (10) instead of the original system (2).

B. Robust Nonlinear Identification

In standard SYSID techniques for both discrete and continuous time systems, data is exclusively available in the form of a finite length vector of input-state-output $(\tilde{u}[t], \tilde{v}[t], \tilde{y}[t])$, or just input–output, sampled pairs. Such data can be generated either by physical measurements of a fabricated integrated circuit, or by simulation of an available circuit schematic. The objective of a SYSID algorithm is to generate automatically from training data, a dynamical system description, such as (2), such that the predicted output of the identified model minimizes the "output error," and it is "easy" to compute each new output sample when given previously computed past values of the input and output samples.

Definition 4: Given a sequence of inputs $\tilde{u}[0], \ldots, \tilde{u}[T]$, the corresponding states $\tilde{v}[0], \ldots, \tilde{v}[T]$, and outputs $\tilde{y}[0], \ldots, \tilde{y}[T]$, the *output error* of an identified model is defined as

$$E(F, G, \mathcal{X}) = \sum_t |y[t] - \tilde{y}[t]|^2 \tag{11}$$

where $y[t]$ are solutions to the identified model in response to training data inputs and initial conditions $\tilde{v}[t-1], \ldots, \tilde{v}[t-m]$, and $\mathcal{X}$ represents the training data set containing all given $\tilde{u}[t], \tilde{v}[t], \tilde{y}[t]$ pairs.

In general, minimization of the true output error is computationally extremely difficult as it is a highly nonconvex problem. Most approaches suggested by the classical literature in system identification [14] instead attempt to minimize the overall "equation error."

Definition 5: The *equation error* is defined as the sum of squared mismatches obtained from evaluating the identified model (2) over the training data samples $(\tilde{u}[t], \tilde{v}[t], \tilde{y}[t]) \in \mathcal{X}$

$$\tilde{E}(F, G, \mathcal{X}) = \sum_t |F(\tilde{V}[t], \tilde{U}[t])|^2 + |G(\tilde{y}[t], \tilde{v}[t])|^2. \tag{12}$$

It is, however, misleading to assume that a small equation error implies a small output error. It is possible to identify unstable models whose system equations are satisfied accurately by the given data, resulting in small equation error, but produce unstable outputs during simulation, resulting in large output error. It has been shown in [31] that if system (2) satisfies (9), then the equation error for the resulting system provides an upper bound for the model output error over the training data set. Minimization of this upper bound subject to incremental stability constraint can be cast as a semidefinite program, however, this approach typically produces overly conservative upper bounds for the output error due to the strong constraints imposed by (9).

III. SYSID Formulation

In this section, we present the theoretical development for our modeling framework that identifies systems of the form (2). In the event that state data $v[t]$ is not available, we may identify input–output models by selecting $v[t] = y[t]$ and defining $G(y, v_0) = y - v_0$.

A. Incremental Stability and Robustness

Minimization of the exact output error by enforcing dissipation constraint (9) is a computationally difficult problem and typically yields overly conservative fits. This is because enforcing incremental stability via constraint (9) imposes strong restrictions on the class of admissible models. Therefore we consider instead a different method for imposing incremental stability that leads to a bound on a reasonable alternative measure of output error, referred to as the "linearized output error." First, we define linearizations of (2) around y, V, U as

$$\begin{aligned}\bar{F}(V, U, \Delta) = F(V, U) + F_v(V, U)\Delta \\ \bar{G}(y, v_0, \delta_0, \xi) = G(y, v_0) + G_v(y, v_0)\delta_0 + G_y(y, v_0)\xi\end{aligned}$$

for F_v, G_v, G_y, Δ, and ξ as defined in Section II-A.

Definition 6: The *linearized output error* of identified model (2) is defined as

$$S(F, G, \mathcal{X}) = \sum_t |\xi[t]|^2$$

where $\xi[t]$ are solutions to

$$\bar{F}(\tilde{V}[t], \tilde{U}[t], \Delta[t]) = 0, \quad \bar{G}(\tilde{y}[t], \tilde{v}[t], \delta[t], \xi[t]) = 0 \quad (13)$$

in response to the zero initial condition $\delta[t-1], \ldots, \delta[t-m] = 0$ when evaluated on the training data.

Intuitively, this quantity is the result of the following procedure: linearize the identified model around *every* point in the training data set, compute the response of each linearized model after one time step in response to the corresponding training data input sample, and sum up the resulting output quantities over all time. In the case of linear systems, the true output error is exactly equal to the linearized output error.

In order to prove incremental stability of system (2), it is sufficient to show that linearizations of (2), as defined in (10), around all possible U, V, y satisfying (2) are stable, as was proposed in [34]. This can be proven with the following dissipation inequality:

$$\begin{aligned} h(V_+, \Delta_+) \leq h(V_-, \Delta_-) - |\xi|^2 - \epsilon|\Delta|^2 + 2\delta_0^T F_v(V, U)\Delta \\ +2\xi^T(G_v(y, v_0)\delta_0 + G_y(y, v_0)\xi) \quad \forall\ y, V, U, \Delta, \xi \end{aligned} \quad (14)$$

where h is a storage function, defined as

$$\begin{aligned} h(V_+, \Delta_+) &= \Delta_+^T H(V_+)\Delta_+ \\ h(V_-, \Delta_-) &= \Delta_-^T H(V_-)\Delta_- \end{aligned} \quad (15)$$

and $\epsilon > 0$. Since (10) is linear in Δ, it is sufficient to consider storage functions that are quadratic in Δ [34]. Note that for U, V, y, Δ, ξ satisfying (2) and (10), constraint (14) simplifies to (8) with supply rate $\sigma = -|\xi|^2 - \epsilon|\Delta|^2$. Inequality (14) can be thought of as a linearized version of inequality (9), and is less restrictive because although a stable system satisfying (9) also satisfies (14), there are many stable systems satisfying (14) that do not satisfy (9).

Definition 7: The *robust equation error*, $\hat{r}$, of system (2) over training data set $\mathcal{X}$ is defined as

$$\hat{r}(F, G, H, \mathcal{X}) = \sum_t r(\tilde{y}[t], \tilde{V}[t], \tilde{U}[t])$$

where

$$\begin{aligned} r(y, V, U) = \max_{\Delta,\xi}\{h(V_+, \Delta_+) - h(V_-, \Delta_-) \\ -2\delta_0^T \bar{F}(V, U, \Delta) - 2\xi^T \bar{G}(y, v_0, \delta_0, \xi) + |\xi|^2\}. \end{aligned} \quad (16)$$

The robust equation error serves as an upper bound for the linearized output error.

Theorem 1: If there exists a positive semidefinite function $H : \mathbf{R}^m \mapsto \mathbf{R}^{m\times m}$, positive scalars $\epsilon, \epsilon_1, \epsilon_2 > 0$ such that $\epsilon_1 I < H < \epsilon_2 I$ and (14) is satisfied for all $\tilde{y}, \tilde{V}, \tilde{U} \in \mathcal{X}$, and for all possible Δ, ξ, then system (2) is locally incrementally stable and the linearized output error on the training set is bounded from above by the robust equation error

$$S(F, G, \mathcal{X}) \leq \hat{r}(F, G, H, \mathcal{X}).$$

If, in addition, H is continuously differentiable and (14) is satisfied for all y, V, U, then system (2) is also globally incrementally stable.

Proof: Incremental stability is implied by (14) using a standard proof following the principles of [34]. It follows from (16) that

$$\begin{aligned} |\xi|^2 \leq r(y, V, U) + 2\delta_0^T \bar{F}(V, U, \Delta) \\ +2\xi^T \bar{G}(y, v_0, \delta_0, \xi) + h(V_-, \Delta_-) - h(V_+, \Delta_+) \end{aligned} \quad (17)$$

is satisfied for all y, V, U, Δ, ξ. To obtain the linearized output error, we sum (17) over all training data samples $\tilde{y}[t], \tilde{V}[t], \tilde{U}[t]$ and incremental variables $\Delta[t], \xi[t]$ satisfying (13), resulting in

$$S(X, F, G) = \sum_t |\xi[t]|^2 \quad (18)$$

$$\begin{aligned} \leq \textstyle\sum_t [r(\tilde{y}[t], \tilde{V}[t], \tilde{U}[t]) + h(\tilde{V}_-[t], \Delta_-[t]) - h(\tilde{V}_+[t], \Delta_+[t])] \\ \leq \textstyle\sum_t r(\tilde{y}[t], \tilde{V}[t], \tilde{U}[t]) = \hat{r}(F, G, H, \mathcal{X}). \end{aligned}$$

Here, we have also used the fact that

$$\sum_{t=0}^{T} \left[h(\tilde{V}_-[t]) - h(\tilde{V}_+[t])\right] = -h(\tilde{V}_+[T]) \leq 0$$

by definition of h and by the zero initial condition of Δ. Note that finiteness of $\hat{r}$ is guaranteed by (14). ■

In summary, for a given model F, G, if there exists a storage function h as defined in (15) that satisfies (14), then system (2) is incrementally stable. Furthermore, for such F, G, h, the robust equation error serves as an upper bound for the linearized output error over the training data set, as shown in (18).

B. Identification Procedure

The proposed system identification algorithm is based on minimization (with respect to F, G, H, and r) of the *linearized output error* upper bound, r, over the training data set $\mathcal{X}$ subject to a dissipation constraint

$$\min_{r,F,G,H} \sum_t r_t \qquad \text{subject to} \quad (19)$$

$$\begin{aligned} r_t + 2\delta_0^T \bar{F}_t(\Delta) + 2\xi^T \bar{G}_t(\delta_0, \xi) - |\xi|^2 \\ +h_{t-1}(\Delta_-) - h_t(\Delta_+) \geq 0 \quad \forall\ t, \Delta, \xi \end{aligned}$$

where $r_t = r(\tilde{y}[t], \tilde{V}[t], \tilde{U}[t])$, $\bar{F}_t(\Delta) = \bar{F}(\tilde{V}[t], \tilde{U}[t], \Delta)$, $\bar{G}_t = \bar{G}(\tilde{y}[t], \tilde{v}[t], \delta_0, \xi)$, $h_{t-1}(\Delta_-) = h(\tilde{V}_-, \Delta_-)$, and $h_t(\Delta_+) = h(\tilde{V}_-, \Delta_-)$. In this formulation, we simultaneously enforce accuracy by minimizing the linearized output error upper bound at the training data samples, and also enforce local incremental stability at each training sample through the constraint.

By construction, the robustness constraint is jointly *convex* with respect to the unknown functions F, G, H, r, and is a quadratic form in the incremental variables Δ, ξ. If the unknown functions are chosen among linear combinations of a finite set of basis functions Φ

$$F = \sum_{j\in\mathbb{N}_f} \alpha_j^F \phi_j^F(V, U), \quad G = \sum_{j\in\mathbb{N}_g} \alpha_j^G \phi_j^G(y, v_0) \quad (20)$$

$$H = \sum_{j\in\mathbb{N}_h} \alpha_j^H \phi_j^H(V), \quad r = \sum_{j\in\mathbb{N}_r} \alpha_j^r \phi_j^r(y, V, U)$$

where $\phi^F, \phi^G, \phi^H, \phi^r \in \Phi$, then $\alpha^F, \alpha^G, \alpha^H, \alpha^r$ become the free variables and the optimization problem becomes a semidefinite program (SDP). Additional details on semidefinite programming are given in Section V-A.

In order to obtain global incremental stability, it is necessary to additionally enforce constraint (14) globally for all y, V, U and to ensure that the storage function H is smooth with respect to all arguments. In this case, the complexity of the optimization problem depends heavily on the choice of basis functions ϕ for the unknown functions F, G, H, r. The basis must be chosen carefully to ensure that the inequalities in problem (19) can be easily verified numerically, and that feasible solutions exist. In Section IV, we describe one possible choice for the basis functions Φ that results in an optimization problem that can be efficiently solved.

C. Extension to Continuous Time Models

In this paper, we focus mainly on generating discrete time (DT) models for many typical circuit blocks in the signaling path that are also usable in high-level system simulation and design, using for instance Cadence analog mixed signal or Verilog A. In addition, it is possible to extend the previously developed dissipation-based identification approach to generate continuous time (CT) systems for greater compatibility with lower level circuit simulators. In this case, there are however, additional constraints on the choice of F to ensure that the system is uniquely solvable. For instance, F should not possess nonlinear dependence on derivatives of the input, otherwise the system may not be well-posed. Additionally, there are strong constraints on the relationship between the function F and the storage function H in order to guarantee existence of solutions to the optimization problem. To avoid excessive technicalities, we consider here only CT systems described in state-space form

$$F(\dot{v}(t), v(t), u(t)) = 0, \quad G(y(t), v(t)) = 0 \tag{21}$$

along with constant positive semi-definite (PSD) storage function matrices $H(v) = H$. As in the DT case, we define a robust dissipation inequality

$$\frac{\partial h(\Delta)}{\partial t} \le 2\delta^T F_v(v, u)\Delta + 2\xi^T G_y(y, v)\xi + 2\xi^T G_v(y, v)\delta - |\xi|^2 - |\delta|^2 \tag{22}$$

where $h(\Delta) = \Delta^T H \Delta$, such that system (21) is incrementally stable and the linearized output error is bounded from above by the robust equation error if there exists a storage function matrix H such that (22) holds for all y, v, u, Δ, ξ. Constraint (22) can then be used to formulate an optimization problem similar to (19). Results for CT modeling using this approach are presented in Section VI-B.

D. Identification of MIMO Models

The previously derived identification procedure is capable of identifying models with multiple inputs, multiple states, and multiple outputs. Multiport models can also be used to capture loading effects. If one of the ports is connected to a load, then varying the load will produce different input–output data for that port, which can then be used for training the model in order to capture loading effects. Our resulting multiport model can then be described for instance in Verilog-A and connected with other circuit blocks inside a commercial simulator. In Section VI, we present results for systems with multiple inputs (Section VI-E), multiple states (Section VI-B), and multiple outputs (Section VI-D).

E. Extension to Parameterized Models

Our approach can easily be extended to identify models parameterized by, for instance, device parameters or geometrical parameters. This is achieved by selecting the basis functions for F, G, H, r to possess dependence on design parameters Z, e.g., $\phi^F = \phi^F(V, U, Z)$, where $Z = [z_1, \ldots, z_p]$ is a vector of parameters. Conceptually this is equivalent to treating the parameters as constant inputs with no memory. Results using this parametrization approach are presented in Section VI-C.

IV. Identification of Rational Models in a Polynomial Basis

In this section, we present one possible choice of basis functions for representing the nonlinear functions in optimization problem (19).

A. Polynomial Basis

The complexity of optimization problem (19) with global stability constraint (14) depends on the choice of basis functions for the nonlinear function, robustness measure, and storage function. One possible choice resulting in a convenient formulation is a polynomial basis.

If we constrain F, G, H, r to be polynomial functions of the internal variables and inputs, i.e., define ϕ from Section III-B as

$$\phi(y[t], V[t], U[t]) = \prod_{i,j,k} v[t-\tau_i]^{p_i} u[t-\tau_j]^{p_j} y[t]^{p_k} \tag{23}$$

then we can formulate optimization problem (19) as a SOS problem. Proving global positivity of a multivariate polynomial is in general a hard problem (i.e., computationally challenging), however, SOS provides an efficient convex relaxation for such problem. Guaranteeing global positivity in the stability constraints is transformed to the task of solving for a PSD symmetric matrix $S = S^T \succ 0$ such that global stability constraint (14) is expressed as

$$h(V_-, \Delta_-) - h(V_+, \Delta_+) + 2\delta_0^T F_v(V, U)\Delta + 2\xi^T G_v(y, v_0)\delta_0 + 2\xi^T G_y(y, v_0)\xi - |\xi|^2 = \Psi^T S \Psi \quad \forall\ y, V, U, \Delta, \xi.$$

Here, Ψ is a vector of basis functions ψ such that all basis functions ϕ can be represented by the product $\Psi^T S \Psi$. That is, for every ϕ_i there exist ψ_j and ψ_k such that $\phi_i \propto \psi_j \psi_k$. Conceptually, the vector Ψ must contain the monomial terms present in the 'square root' of the dissipation constraint,

and for nonlinear systems these entries can be automatically selected from the Newton Polytope of the robustness constraint. See [35]–[37] for details on SOS programming and the Newton Polytope, and see [38] or [32] for our software implementation.

It is important to note that although we are using a polynomial basis, we are not identifying polynomial models. Specifically, the implicit representation of the nonlinear system (2) allows us to identify, for instance, rational models as described in the following section. In this way, we can represent highly nonlinear models in a much more compact form than is possible using traditional polynomial models such as Volterra expansions.

B. Rational Model Description

In general, the identified implicit nonlinear model (2) can be extremely expensive to simulate. To ensure that the resulting DT model can be simulated in an efficient manner, we consider only models that are linear in the unknowns $v[t]$. For example, consider the model

$$\begin{aligned} F(V[t], U[t]) &= Q(V_-[t], U[t])v[t] - p(V_-[t], U[t]) = 0 \\ G(y[t], v[t]) &= g_q(v[t])y[t] - g_p(v[t]) = 0 \end{aligned} \tag{24}$$

where $Q \in \mathbb{R}^{N\times N}$ is a matrix of nonlinear functions, $p \in \mathbb{R}^N$ is a vector of nonlinear functions, and $V_-[t] = [v[t-1], \ldots, v[t-m]]$. Although F is defined implicitly, the system is linear in the unknowns, making the simulation of this discrete time system equivalent to linear system solves when all previous values of the state, $v[t-1], \ldots, v[t-m]$, and input, $u[t], \ldots, u[t-k]$, are known

$$v[t] = Q(V_-[t], U[t])^{-1} p(V_-[t], U[t]).$$

The presence of the nonlinear matrix function $Q(V_-, U)$ is extremely important, as it allows the model to capture nonlinear effects that are significantly stronger than those that would be captured by considering the case where $Q = I$, without significantly increasing the complexity of the optimization problem and of simulation.

C. Existence of Solutions

Given a nonlinear function F, the existence of solutions to (27) depends on the ability of the storage function h to certify stability for that particular nonlinear function. For models without feedback, such as the Volterra model

$$y_t = p(u_t, u_{t-1}, \ldots, u_{t-k}) \tag{25}$$

a storage function is not required to prove stability, and solutions always exist. One implication of this is that Volterra models are a strict subset of the stable models identifiable by our approach. When feedback is present in the model, for certain functions F, storage functions are available to prove stability. For example, for a linear system, it is always possible to certify stability with a constant matrix H. As a result, if the polynomial basis contains linear terms, then there always exists a globally stable solution described by a linear function F and constant matrix H. Additionally, since the storage function and stability of the resulting model do not depend strictly on the inputs u to the system, a constant matrix H can certify stability for a system that is linear in v and highly nonlinear in u. Thus, it is always possible to identify models highly nonlinear in the input even if high degrees of nonlinearity in the state cannot be achieved due to stability constraints.

D. Reduction of States Through Projection

In the event where data is available for a large number of internal states (i.e., N is large), it is not practical to fit a model with N states because it is both computationally expensive to identify the model, and simulation of the large model may be slow. However, it is possible to identify a low-order space in which the system states are well-approximated, and fit by projection to a set of reduced vectors, $\hat{v}[t] \in \mathbb{R}^{\hat{N}}$, where $\hat{N} < N$.

For example, given a collection of training samples, $X = [\tilde{v}[t_1], \tilde{v}[t_2], \ldots, \tilde{v}[t_T]] \in \mathbb{R}^{N\times T}$, it is possible to identify a low-order basis $\Theta \in \mathbb{R}^{N\times\hat{N}}$ such that $X \approx \Theta\hat{X}$, where $\hat{X} = \Theta^T X$ is a projection of the training data onto the reduced space. The projection matrix Θ can be computed using any standard projection technique, such as POD [39], [40] using training data X. The system identification is then performed using the reduced-order training data set $\hat{X}$, resulting in a model with $\hat{N}$ states.

This approach is similar to traditional model reduction techniques utilizing projection in the sense that we approximate the solution in a low-dimensional space spanned by Θ. However, the key difference of our approach is that instead of constructing the reduced model by projecting the system equations explicitly, we instead identify the reduced equations through an optimization procedure to optimally fit the given training data. Numerical results obtained from this projection approach are presented in Section VI-B.

E. Reduction of Polynomial Basis Through Fitting

In addition to the number of state variables, the cost of identifying and simulating the models also depends on the number of delays (memory) of the system. To decrease this cost without reducing the polynomial order of the desired function, it is useful to consider only important polynomial basis terms for identification. Let $\tilde{\Phi}_{[0,n]}$ denote a nominal set of basis functions comprised of variables u, v with up to n delays. Important basis terms $\hat{\phi}$ may be selected as linear combinations of the nominal basis components

$$\hat{\phi}_i = \sum_j \beta_{j,i}\tilde{\phi}_j, \quad \tilde{\phi}_j \in \tilde{\Phi}_{[0,n]}. \tag{26}$$

The coefficients $\beta_{j,i}$ can be identified by fitting a linear model with memory n to the training data with basis $\tilde{\Phi}$.

For example, suppose the nominal basis functions are selected to be input samples, i.e., $\tilde{\Phi}_{[0,k]} = [u[t], \ldots, u[t-k]]$. The training data $(\tilde{y}[t], \tilde{u}[t])$ can be used to identify a linear

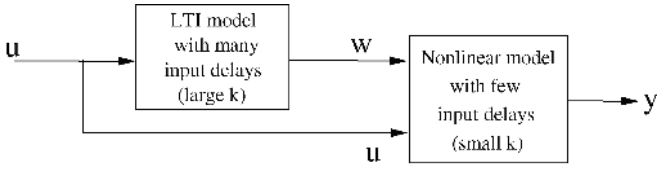

Fig. 1. Block diagram illustrating one approach to implementing the reduced basis selection technique.

model with memory $\hat{k}$ and output $w[t] \approx y[t]$

$$w[t] = \sum_{j=0}^{\hat{k}} b_j u[t-j].$$

This identified linear model now defines a linear transformation of the nominal basis vectors to the reduced basis vector if we define $\beta_{j,1} = b_j$ for the new basis vector set $\hat{\Phi}_{[0,\hat{k}]} = [w[t], \ldots, w[t-\hat{k}]]$ for some $\hat{k} < k$. This new basis vector can then be used for identification of a nonlinear model with low memory. Conceptually, this is equivalent to treating $w[t]$ as an additional input to a new nonlinear model

$$Q(V_-[t], W[t], U[t])v[t] = p(V_-[t], W[t], U[t])$$

as depicted in Fig. 1.

Since the identification of linear systems is cheap, even when m and k are large, this approach can be very useful for reducing the complexity of the final nonlinear model by automatically selecting important combinations of basis vectors. Numerical examples using this reduced basis identification approach are presented in Section VI-E.

V. Implementation

The optimization problem (19) derived in Section III, along with global stability constraint (14), can be expressed generically as the following

$$\min_{r,F,G,H} \sum_t r_t \qquad \text{subject to} \tag{27}$$

$$r_t + 2\delta_0^T \bar{F}_t(\Delta) + 2\xi^T \bar{G}_t(\delta_0, \xi) - |\xi|^2 + h_{t-1}(\Delta_-) - h_t(\Delta_+) \geq 0 \qquad \forall\ t, \Delta, \xi \tag{27a}$$

$$h(V_-, \Delta_-) - h(V_+, \Delta_+) + 2\delta_0^T F_v(V, U)\Delta - |\xi|^2 + 2\xi^T G_v(y, v_0)\delta_0 + 2\xi^T G_y(y, v_0)\xi \geq 0 \quad \forall\ y, V, U, \Delta, \xi. \tag{27b}$$

In this section, we describe how to formulate (27) as a SDP when using a polynomial basis and how to solve the resulting SDP.

A. Implementation as a Semidefinite Program

The benefit of formulating (27) as an SDP is that it can be solved efficiently using readily available software routines. Roughly speaking, a semidefinite program is one whose objective function is linear, and whose constraints can be expressed as requiring matrices to be PSD.

Algorithm 1 Implementation as SDP using SPOT

1: Given symbolic functions F, G, H, r defined as in (20) and training data set χ
2: Initialize optimization problem `pr`
`pr=mssprog`
3: Assign free variables
`pr.free=`$\{\alpha^F, \alpha^G, \alpha^H, \alpha^R\}$
4: **for** t=1:T **do**
5: Compute $M_t = M(\tilde{U}[t], \tilde{V}[t], \tilde{y}[t])$ as defined in (28)
6: Assign local robustness constraint (27a)
`pr.PSD=`M_t
7: **end for**
8: Assign global stability constraint (27b)
`pr.SOS= (27b)`
9: Call solver to minimize $\sum_t r_t$ subject to given constraints
`pr.min=`$\sum_t r_t$
10: Output is coefficients $\{\alpha^F, \alpha^G, \alpha^H, \alpha^R\}$

By construction, constraint (27a) is a quadratic form in the variable $\zeta = [1,\ \Delta^T, \xi^T]^T$, and can therefore be expressed as

$$r(y, V, U) + 2\delta_0^T \bar{F}(V, U, \Delta) + 2\xi^T \bar{G}(y, v_0, \delta_0, \xi) - |\xi|^2 + h(V_-, \Delta_-) - h(V_+, \Delta_+) = \zeta^T M(U, V, y)\zeta \tag{28}$$

for some symmetric matrix M. Thus, global positivity of (28) is satisfied if the matrix M is PSD. In (27), we are not requiring $M(U, V, y)$ to be PSD for all U, V, y, but rather only when evaluated at the given training data samples. That is, $M_t = M(\tilde{U}[t], \tilde{V}[t], \tilde{y}[t])$ is PSD for all t.

On the other hand, the global stability constraint (27b) must be satisfied for all possible U, V, y. This can be achieved using the SOS relaxation described in Section IV-A, which transforms constraint (27b) into a single semidefinite matrix constraint. While it is easy to construct the $M(U, V, y)$ matrix explicitly, and possible to construct S from Section IV-A by hand, these tasks can be performed automatically by the freely available software systems polynomial optimization toolbox (SPOT) [38], when given symbolic constraints in the form of (27a) and (27b).

In Algorithm 1, we outline how optimization problem (27) can be defined and solved in MATLAB using the freely available software SPOT [38] and SeDuMi [41]. SPOT is a 'parser', which takes as input a high-level symbolic description of the optimization problem and reformulates it in such a manner that it can be solved by an optimization 'solver' (in this case, SeDuMi). For additional details and a sample implementation of this approach, see [32] and [38].

B. Complete Algorithm

Our entire identification process is summarized in Algorithm 2. The first step in identifying a model in the form of (24) is to select the number of states (N), the number of state delays (m), the number of input delays (k), the maximum polynomial degree for Q (ρ_Q), the maximum polynomial degree for p (ρ_p), and the maximum polynomial degree for storage function matrix H (ρ_H). These parameters generally depend on the behavior of the system being modeled, and can

Algorithm 2 SOS Identification of Robust Compact Models

1: Generate training data sample set $\mathcal{X} = \{\tilde{U}[t], \tilde{V}[t], \tilde{y}[t]\}$ from simulation or measurement of the original system
2: Select the model parameters $m, k, N, \rho_Q, \rho_p, \rho_H$ as defined in Section V-B
3: **if** State data is available and N is large **then**
4: Use SVD to identify low-order basis for states, $\Theta \in \mathbb{R}^{N\times\hat{N}}$, $\hat{N} < N$, as described in Section IV-D
5: Project data samples: $\tilde{v}[t] \leftarrow \Theta^T \tilde{v}[t]$
6: **end if**
7: Select nominal set of basis functions $\tilde{\Phi}_{[0,k]}$
8: **if** Large delay between input and output **then**
9: Identify linear model defining coefficients $\beta_{j,i}$ of new basis functions $\hat{\phi}$ defined in (26)
10: $\Phi \leftarrow [\tilde{\Phi}_{[0,\hat{k}]}, \ \hat{\Phi}_{[0,\hat{k}]}]$ for $\hat{k} < k$.
11: **else**
12: $\Phi \leftarrow \tilde{\Phi}_{[0,k]}$
13: **end if**
14: Use Algorithm 1 to solve (27) for coefficients α_i
15: Define F, G, H, r, as in (20), resulting in the model

$$Q(V_-[t], U[t])v[t] = p(V_-[t], U[t])$$
$$g_q(v[t])y[t] = g_p(v[t])$$

certified stable by Theorem 1 for matrix function H, and with $\sum_t r_t$ serving as a measure of the model's accuracy on the training data.

be selected either by intuition (based on the system's expected behavior) or through experiment. One approach that we have found to be effective is described below in Section V-C. For CT models, it is often possible to obtain derivatives of states and outputs directly from the simulator, as they are typically required internally for simulation. For systems with a large delay between input and output, the reduced basis technique described in Section IV-E should be used at step 2 to reduce the required number of basis functions. Typically we have found that selecting $\tilde{\Phi}$ as containing the past $15-20$ input samples can produce good results for such systems. The basis set Φ for the final model can then be selected at step 2 as a small number of delayed samples of the true input u and the delayed input w, as well as delays of the state and output. Finally, when considering only local stability for the identified model, it is only necessary to enforce the first constraint (27a) in optimization problem (27).

C. Selecting the Model Parameters

For a given set of parameters $N, m, k, \rho_Q, \rho_p, \rho_H$, as defined in Section V-B, let $\Upsilon = \{N, m, k, \rho_Q, \rho_p\}$ denote the set of all possible models with these parameters, and let $\mathcal{H}$ denote the set of all storage functions of maximum polynomial degree ρ_H. The goal of the identification procedure is to find a stable model in Υ that accurately fits the training data χ and is certified stable by a storage function in $\mathcal{H}$.

It is difficult to accurately determine Υ and $\mathcal{H}$ a priori, but we have found the following procedure to be quite effective. First, we select a set Υ and attempt to fit a model *with no stability constraints*. This can be achieved, for instance, by using a least-squares solve to minimize equation error (which is computationally cheap). Varying Υ through experiment, it is possible to identify a model that accurately fits χ.

Next, it is necessary to determine whether there exists a *stable* model in Υ that is certifiable by $\mathcal{H}$. To determine this, we select ρ_H and solve (27) using Algorithm 2 while first enforcing *only local stability* constraint (27a) in Algorithm 1. If no accurate locally stable model is found, then ρ_H should be increased. If, for large ρ_H, no accurate stable model is found, then Υ should be increased (i.e., increase any of N, m, k, ρ).

Once an accurate locally stable model is found, then (27) should be solved using Algorithm 2, this time also enforcing global stability constraint (27b). If no accurate globally stable model is found, then ρ_H and Υ should be increased, as described above. If stability constraint (27b) is not enforced, then the robust equation error is not guaranteed to be an upper bound for the linearized output error, meaning that simulation of the resulting model, even over the training data set, could lead to inaccurate results.

D. Constructing Basis Functions

For a given set of parameters $N, m, k, \rho_Q, \rho_p, \rho_H$, the basis functions for F, G and H can be constructed as defined in (23), where $p_i + p_j + p_k \leq \rho$, $\tau_i < m$, and $\tau_j < k$. In general, ρ_Q should be an even integer to ensure that matrix function Q is always invertible. For the robustness measure r, we typically use a piecewise-constant function, resulting in one unknown parameter for each training data point.

VI. Examples

A. Testing Procedure

Our approach was tested on several nonlinear systems, including a CT model, a DT parameterized model, and a DT single-input multiple-output (SIMO) model. For each example, training data was generated from simulations of the full system in response to a series of periodic inputs, using SPECTRE circuit simulator for the circuit examples and a MATLAB simulator for the MEM system (MEMS) example. The training inputs must be carefully chosen in order to excite all possible behavior of interest in the system, while avoiding driving the system to regions of the space that will not be excited by typical inputs of interest. Attempting to model dynamics not encountered by testing inputs could greatly increase the complexity of the identified model. In order to maximize robustness while minimizing complexity, in our experience, the best approach is to train with inputs having the same form (e.g., sum of sinusoids) as the inputs to be used for testing. In this case, the amplitudes, frequencies, and phases of the training inputs may be varied over a wide range containing all possible values to be used for testing. For all examples, the models are identified in a polynomial basis with a rational description as described in Section IV.

All of the model generation and simulation times reported were obtained using a desktop PC with a dual core 3.33 GHz processor and 4 GB of RAM. The SOS problem (27) was solved using the SPOT [38], which uses SeDuMi [41].

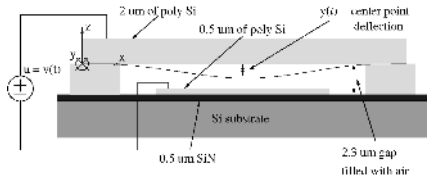

Fig. 2. Micromachined switch MEM device [7].

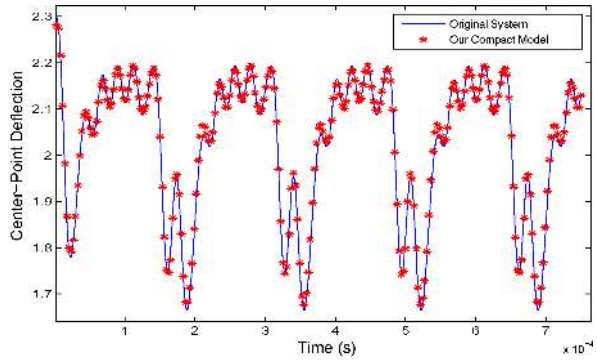

Fig. 3. Output of order 400 original system (solid line) and our order 4 model (stars) tested on a periodic input of amplitude and frequency different from the training inputs.

B. MEM Device

In our first example, we identify a CT model of a MEMS device to show that our preliminary CT approach from Section III-C and projection approach from Section IV-D are feasible. The MEMS device [7], shown in Fig. 2, is described by a pair of nonlinear PDEs that can be discretized along the surface of the device to obtain a system of nonlinear ODEs. A detailed analysis of the example can be found in [42].

For this example, the training data was generated from inputs of the form

$$u(t) = [A_1 \sin(\omega_1 t) + A_2 \sin(\omega_2 t) + A_3 \sin(\omega_3 t)]^2 \tag{29}$$

where A_i vary between 4 Volts and 7 Volts, and $f_i = \frac{2\pi}{\omega_i}$ vary between 1.5 kHz and 240 kHz. From this data, we identified a 4th order nonlinear CT model suitable for usage in any ODE integrator and in particular a low-level circuit simulator

$$Q_2(v,u)\dot{v} = p_7(v,u), \quad y = C^T v$$

where $v \in \mathbb{R}^4$, $Q_2 \in \mathbb{R}^{4\times 4}$ is a matrix of second order polynomials, $p_7 \in \mathbb{R}^4$ is a vector of seventh order polynomials, and $C \in \mathbb{R}^4$ is a constant vector, all identified using the projection technique described in Section IV-D and the reduced basis technique from Section IV-E, resulting in only 52 parameters in the reduced model. For this model, the identification procedure took less than two minutes.

The identified model was tested on an input of the form (29) with A_i and f_i different from the training set, and the resulting output is compared to the output of the full nonlinear system in Fig. 3. To make the comparison fair, both full and reduced models were simulated using the same MATLAB built in ODE solver. Simulation of the full 400th order nonlinear system for this example required approximately 400 s to integrate for 5000 time points, while the reduced model was simulated in response to the same input for the same number of time steps in just 10 s, resulting in a speedup of about 40 times.

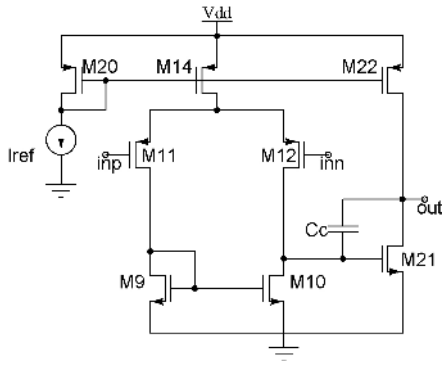

Fig. 4. Schematic of operational amplifier.

C. Operational Amplifier

In our second example, we identify a parameterized model, using the approach described in Section III-E, for a two-stage operational amplifier. The opamp, shown in Fig. 4, is designed with a 90 nm predictive model and nominal reference current as 10μ, has an open-loop DC gain of 260, and unity-gain bandwidth 125 MHz. For the parameterized model, the reference current is considered as a circuit parameter and varies from 7μA to 19μA.

Training data was generated using inputs of the form

$$u(t) = \text{inp-inn} = \sum_{i=1}^{5} A_i \sin(2\pi f_i t + \phi_i) \tag{30}$$

where A_i are chosen randomly, but large enough to saturate the opamp, f_i are randomly sampled between DC to unity-gain frequency, and ϕ_i are randomly sampled in $[0^o, 360^o]$. The resulting model was a parameterized input–output model of the form

$$y[t] = \frac{p(y[t-1], u[t], u[t-1], z)}{q(y[t-1], u[t], u[t-1])} \tag{31}$$

where p is cubic in u, y and quadratic in z, q is a fourth order polynomial of u, y, and the model contains 97 terms.

The identified model was tested on 140 randomly generated inputs of the form (30) with parameter values randomly selected between 7μA and 19μA. Fig. 5 plots the model output and output error, defined as

$$e[t] = \frac{|y[t] - \tilde{y}[t]|}{\max_t |\tilde{Y}|} \times 100 \tag{32}$$

for one of these testing signals, while Fig. 6 plots the maximum error over the entire signal for each testing set, defined as

$$e_m = \max_t e[t] \tag{33}$$

where $y[t]$ is the output of our model at time t, $\tilde{y}[t]$ is the output of SPECTRE at time t, and $\tilde{Y}$ is the full waveform of SPECTRE outputs over one period.

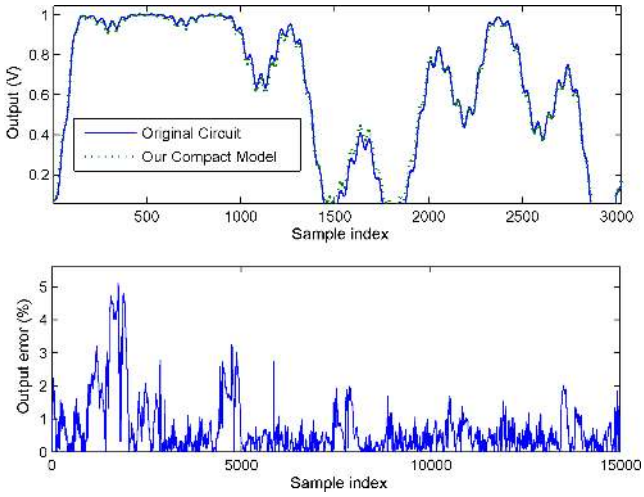

Fig. 5. Time-domain output and error, as defined in (32), for our identified model in response to a random input of the form (30) and random parameter value between 7μA and 19μA.

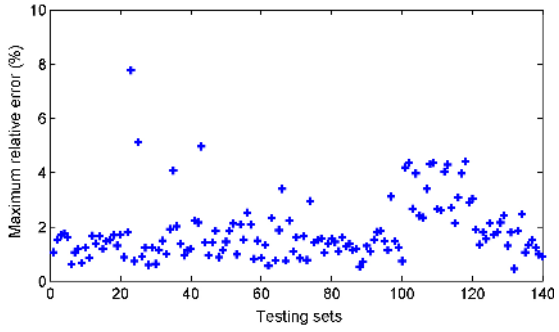

Fig. 6. Maximum output error, as defined in (33), of our model tested on 140 random input signals (30) and parameter values ranging from 7μA to 19μA.

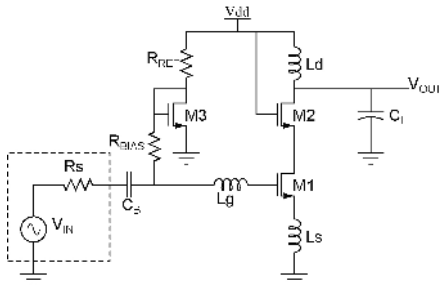

Fig. 7. Schematic of LNA [44].

D. Low-Noise Amplifier

In our third example, we identify a SIMO model of a single ended low-noise amplifier (LNA) designed in 0.5μm complementary metal-oxide semiconductor (CMOS) technology [44], shown in Fig. 7. The designed LNA has a gain of approximately 13 dB centered around 1.5 GHz.

For this example, we wish to capture the nonlinear behavior of both the amplifier output V_{out} and the supply current in response to a modulated input signal with an added jamming signal. The overall input to the system is

$$\mathrm{V_{IN}} = A_j \cos(2\pi f_j t) + \sum_{n=0,1,3,5} A\cos(2\pi n f_0 t)\cos(2\pi f_c t) \quad (34)$$

with carrier frequency $f_c = 1.5$ GHz, sideband frequency $f_0 = 5$ MHz, and jamming frequency $f_j = 1$ GHz. The system was trained by varying the amplitude A between 15 mV and 85 mV, and the jamming amplitude between 0 mV and 250 mV.

The identified model in this example is a DT multiple-input multiple-output model, usable for instance by a Verilog-A or higher level simulator, described by the rational model

$$Q_2(y[t-1], U[t])y[t] = p_3(y[t-1], U[t]) \quad (35)$$

where $U[t] = [u[t], u[t-1], u[t-2]]$, $Q_2 \in \mathbb{R}^{2\times 2}$ is a matrix of second order polynomials, $p_3 \in \mathbb{R}^{2\times 1}$ is a vector of third order polynomials, $y \in \mathbb{R}^{2\times 1}$, and $u \in \mathbb{R}^{2\times 1}$. The rational nonlinearity is sufficient to capture the highly nonlinear behavior resulting from the large jamming signal, and the total number of parameters describing the identified model is 102. The entire identification procedure took less than two minutes, and the resulting model can be simulated in MATLAB for 15 000 time steps in under 3 s.

To test the model, it was simulated over a full period with six pairs of amplitudes, A and A_j, differing from the training data amplitudes, producing outputs with approximately 4% maximum error from the outputs of the original circuit. Fig. 8 compares the two time domain outputs, over a small portion of the period, of the model identified by our procedure (dots) with the outputs of the full original circuit (solid lines) in response to an input with $A = 50$ mV and $A_j = 150$ mV.

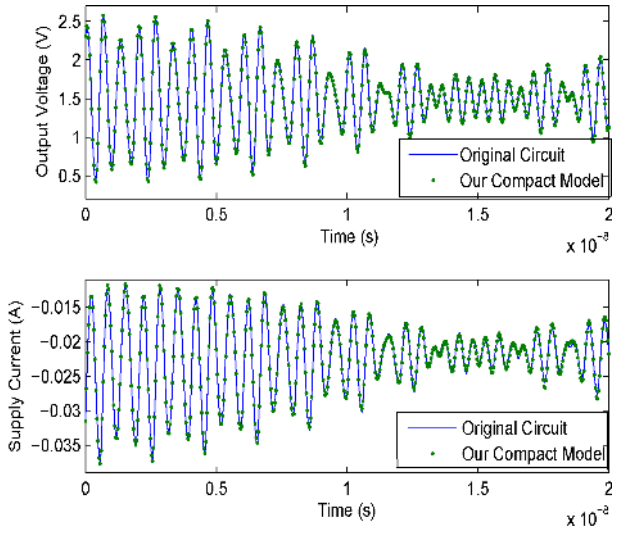

Fig. 8. Time domain outputs, over a small portion of the period, of the original LNA circuit (solid line) and the compact model identified by our procedure (dots) in response to an input signal different from the training signals.

E. Distributed Power Amplifier

The final example considered is a distributed power amplifier designed in 90 nm CMOS technology, with a distributed

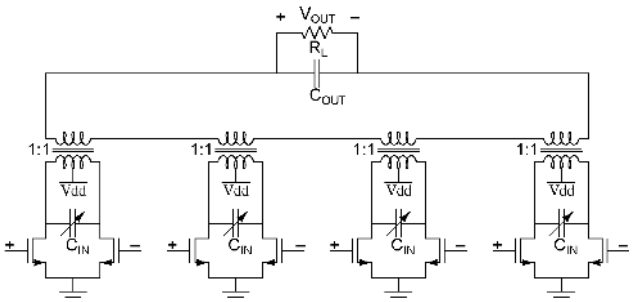

Fig. 9. Transformer-based power amplifier with distributed architecture [43].

architecture and transformer-based power combiner as proposed in [43]. The amplifier delivers 24 dBm power with a gain of 6 dB at 5.8 GHz. A simplified schematic of the amplifier is shown in Fig. 9. Transistors are biased to operate in differential Class-B configuration to improve efficiency, however, at the cost of linearity. Nonlinearities also arise because of parasitic inductance introduced by supply and ground bond wires.

Power combining is achieved by using 1 : 1 transformers, as shown in Fig. 9. Losses in primary and secondary inductors are modeled by using quality factor of 12.5 and coupling coefficient of 0.7, based on which optimum values of inductances were selected as Lp= Ls= 157 nH [43]. Similarly, the following parameters were selected based on optimized performance of the amplifier at 5.8 GHz [43]: Vdd= 1.0V, W/L of transistors = 1.2 mm/90 nm, C_{IN} = 2.6 pF, C_{OUT} = 610 fF, R_L = 50Ω, $L_{Vdd} = L_{GND}$ = 1 nH, C_B = 20 pF, Rg = 18Ω.

For this example, training data samples were generated in response to periodic inputs of the form

$$V_{IN} = V_{DC} + \sum_{n=0,1,3,5} A\cos(2\pi n f_0 t)\cos(2\pi f_c t) \qquad (36)$$

with carrier frequency f_c = 5.8 GHz, $f_0 \in \{25, 50\}$ MHz, and amplitude $A \in \{30, 90\}$ mV. The simulation was performed with SPECTRE, whose model for the power amplifier contained 284 equations, 95 internal variables, and 8 transistors modeled with 90 nm predictive technology models.

The identification procedure, using the reduced basis technique from Section IV-E, identified a DT input–output model

$$y[t] = \frac{p_3\left(y[t-1], u[t], u[t-1], w[t], w[t-1]\right)}{q_4\left(u[t], u[t-1], w[t], w[t-1]\right)} \qquad (37)$$

where

$$w[t] = \sum_{j=0}^{19} b_j u[t-j]$$

is a linear transformation of the input $u[t]$ with coefficients b_j determined by first fitting a linear system as described in Section IV-E. Here p_3 indicates a third order polynomial, q_4 represents a fourth order polynomial, and the total number of parameters in the model is 106. The entire identification procedure took approximately 12 minutes.

The identified model was able to reproduce the training data with less than 4% maximum error in the time-domain, and was able to be simulated for 10, 000 time steps in under 2 s. When tested with nontraining inputs of the form (36) with parameters

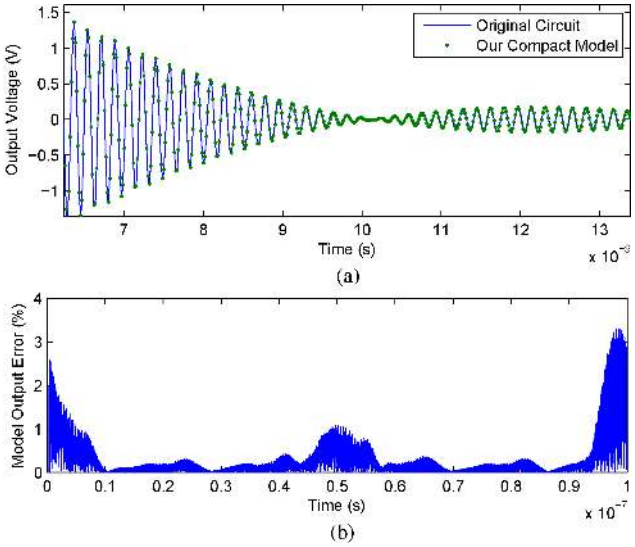

Fig. 10. (a) Time domain output of the original circuit (solid line) and the compact model identified by our procedure (dots) in response to a testing input with amplitude and frequency different from the training inputs. (b) Output error e_t, as defined in (32), of our model over the full signal from (a).

$A \in \{10, 30, 60, 90\}$ mV and $f_0 \in \{10, 25, 40, 50\}$ MHz, our model reproduces the outputs of the original circuit with an average error of less than 1% for each testing input. Fig. 10 compares the output of the identified model with the output of the original power amplifier circuit in response to a testing input with A = 60 mV and f_0 = 10 MHz, both differing from the training data set. For clarity, the top plot in Fig. 10 shows a small portion of the output signals, while the bottom plot shows the model output error, as defined in (32), over a full period of the signal. To show that our model, trained only with with sinusoids of fixed amplitude, is capable of capturing the circuit behavior in response to also *different classes of inputs*, Fig. 11 plots the constellation diagram for the output from our model in response to a 16-quadrature amplitude modulation (QAM) input signal, which is a nonsmooth input.

A Volterra model with approximately the same number of parameters identified with our procedure for this example produced over three times the average error on the training data set compared to model (37). With our current testing setup, it was not possible to obtain a pure Volterra model that is as accurate as model (35) due to memory constraints in our computer (4 GB). This is a result of the large number of parameters that would be required in the Volterra model of high order and with many delays.

In addition to matching input–output behavior, it is important that our identified models can also accurately predict the performance curves of the circuits being modeled. The top plot in Fig. 12 plots output power versus input power (compression curve) at 5.8 GHz for the original circuit (solid line) and our identified model (circles), while the bottom plot show the drain efficiency (defined as the ratio of output RF power to input DC power) versus output power for the original circuit (solid line) and our identified model (circles), also at 5.8 GHz. This model was identified by training with sinusoids at 5.8 GHz

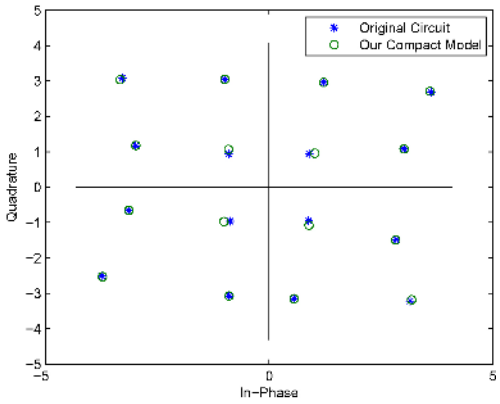

Fig. 11. Constellation diagram for output of the power amplifier model (trained using only amplitude-modulated sinusoids) in response to a 16-QAM input signal.

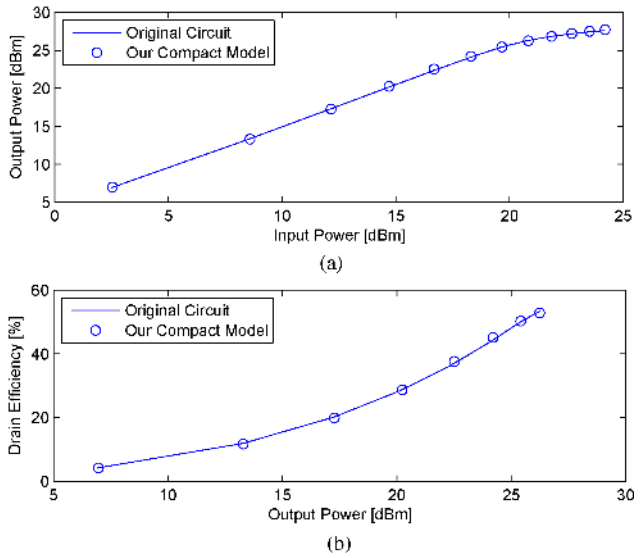

Fig. 12. (a) Compression curve, plotting input power versus output power, at 5.8 GHz for the original circuit (solid line) and our compact model (circles). (b) Drain efficiency versus output power at 5.8 GHz for the original circuit (solid line) and our compact model (circles).

with amplitudes $A \in [100, 400, 800, 1200]$ mV, and was tested at 12 amplitudes evenly spaced between 100 mV and 1200 mV. We want to emphasize that these performance curves were obtained from simulation of our identified *dynamical models*, and not by simply fitting performance curves.

F. Comparison to Existing SYSID Techniques

Finally, we compare our proposed approach to several existing SYSID techniques from literature. Traditional identification techniques suffer from several shortcomings. Some techniques, such as the Hammerstein–Wiener (H–W) model [14] (a cascade connection of an LTI system between two memoryless nonlinearities), forces a specific block-structure on the model which restricts the types of systems that can be accurately modeled. Volterra models, as defined in (25), do not force a specific block structure, but require many parameters to represent complex systems due to a lack of feedback and polynomial nonlinearities. More general nonlinear models,

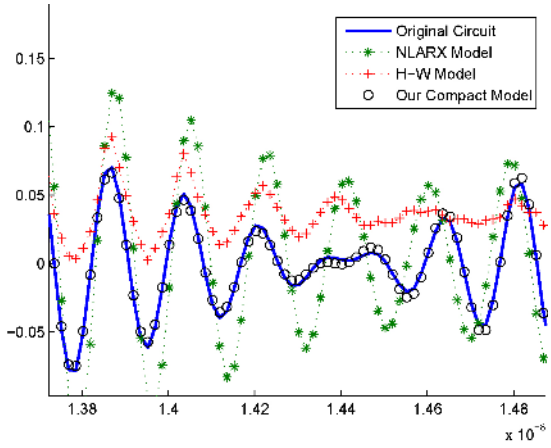

Fig. 13. Outputs of our compact model (circles), a Hammerstein–Wiener (H–W) model (pluses), and a NLARX model (stars) all generated from the four training inputs used in Section VI-E, compared to the output of the original circuit (solid line) in response to a training input with $f_0 = 10$ MHz and $A = 60$ mV.

such as nonlinear autoregressive model with exogenous inputs (NLARX) [14], have the more general structure

$$y[t] = f(y[t-1], \ldots, y[t-m], u[t], \ldots, u[t-k])$$

which is similar to the DT models identified by our proposed procedure, and do incorporate feedback, but they typically do not explicitly enforce stability during identification. For both the H–W and NLARX models, the nonlinearities are typically identified as a linear combination of nonlinear basis functions.

The same training data sets from Section VI-E were used to identify models of the distributed power amplifier in the form of a H–W model and a NLARX model. These models were generated using the MATLAB system identification toolbox, which uses techniques described in [14]. In general, both types of models were found to be less accurate than our proposed approach, with the H–W models producing average errors between 5% and 10%, and the NLARX models producing average errors between 3% and 5%, compared to average errors of 1% from model (37) identified by our technique. Additionally, *the NLARX models were often unstable*, and as a result, the testing inputs often produced unbounded outputs. Fig. 13 plots the output response of our compact model (circles), a H–W model (pluses), and a NLARX model (stars), all generated from the four training inputs from Section VI-E, compared to the output of the original circuit (solid line) in response to one testing input with frequency and amplitude different from the training data. For this example, all three identified models contain approximately the same number of parameters, and the nonlinearities in both the HW and NLARX models were described by sigmoidnet functions.

VII. Conclusion

In this paper, a specialized system identification technique has been developed and has been shown to be an effective alternative technique to model reduction of typical nonlinear analog circuit blocks, such as low-noise amplifiers and power amplifiers. The proposed identification technique requires only

input–output data, eliminating the need for extensive knowledge of the internal system description required by existing nonlinear model reduction techniques, but is also capable of utilizing internal state data when it is available. Furthermore, it has been shown that the identification of stable nonlinear models described by rational functions can be cast as a sum-of-squares program, which is a specific case of semidefinite programming. By enforcing incremental stability as a constraint of the identification procedure, we are able to obtain a certificate of robustness for the model, which quantifies the model accuracy on a given set of training data. Our approach has been shown to compare favorably to existing identification techniques that either impose restrictive structure on the model description, or are incapable of guaranteeing stability when feedback is present.

ACKNOWLEDGMENT

The authors would like to acknowledge the help and support of the late Dr. Dennis Healy, DARPA, MTO.

REFERENCES

[1] M. Celik, A. Atalar, and M. A. Tan, "Transient analysis of nonlinear circuits by combining asymptotic waveform evaluation with Volterra series," *IEEE Trans. Circuits Syst. I: Fundam. Theory Appl.*, vol. 42, no. 8, pp. 470–473, Aug. 1995.

[2] J. R. Phillips, "Projection frameworks for model reduction of weakly nonlinear systems," in *Proc. IEEE/ACM Design Autom. Conf.*, Jun. 2000, pp. 184–189.

[3] J. Chen and S. M. Kang, "An algorithm for automatic model-order reduction of nonlinear MEMS devices," in *Proc. ISCAS*, 2000, pp. 445–448.

[4] J. R. Phillips, "Projection-based approaches for model reduction of weakly nonlinear, time-varying systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 2, pp. 171–87, Feb. 2003.

[5] P. Li and L. T. Pileggi, "Norm: Compact model order reduction of weakly nonlinear systems," in *Proc. IEEE/ACM Design Autom. Conf.*, Jun. 2003, pp. 249–264.

[6] Y. Chen, "Model order reduction for nonlinear systems," M.S. Thesis, Dept. Elect. Eng. Comput. Sci., Massachusetts Instit. Technol., Cambridge, MA, Sep. 1999.

[7] M. Rewienski and J. White, "A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 2, pp. 155–170, Feb. 2003.

[8] D. Vasilyev, M. Rewienski, and J. White, "A TBR-based trajectory piecewise-linear algorithm for generating accurate low-order models for nonlinear analog circuits and MEMS," in *Proc. IEEE/ACM Design Autom. Conf.*, Jun. 2003, pp. 490–495.

[9] S. Tiwary and R. Rutenbar, "Scalable trajectory methods for on-demand analog macromodel extraction," in *Proc. IEEE/ACM Design Autom. Conf.*, Jun. 2005, pp. 403–408.

[10] Y. Wan and J. Roychowdhury, "Operator-based model-order reduction of linear periodically time-varying systems," in *Proc. IEEE/ACM Design Autom. Conf.*, Jun. 2005, pp. 391–396.

[11] B. Bond and L. Daniel, "A piecewise-linear moment-matching approach to parameterized model-order reduction for highly nonlinear systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 12, pp. 2116–2129, Dec. 2007.

[12] B. Bond and L. Daniel, "Stabilizing schemes for piecewise-linear reduced order models via projection and weighting functions," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2007, pp. 860–866.

[13] N. Dong and J. Roychowdhury, "General purpose nonlinear model-order reduction using piecewise-polynomial representations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 2, pp. 249–264, Feb. 2008.

[14] L. Ljung, *System Identification, Theory for the User*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1999.

[15] W. Beyene and J. E. Schutt-Aine, "Efficient transient simulation of high-speed interconnects characterized by sampled data," *IEEE Trans. Compon. Packag. Manuf. Technol. B*, vol. 21, no. 1, pp. 105–114, Feb. 1998.

[16] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Trans. Power Del.*, vol. 14, no. 3, pp. 1052–1061, Jul. 1999.

[17] C. Coelho, J. Phillips, and L. Silveira, "Optimization-based passive constrained fitting," in *Proc. Int. Conf. Comput.-Aided Design*, Nov. 2002, pp. 10–14.

[18] K. C. Sou, A. Megretski, and L. Daniel, "A quasi-convex optimization approach to parameterized model order reduction," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 3, pp. 456–469, Mar. 2008.

[19] R. Haber and L. Keviczky, *Nonlinear System Identification: Input–Output Modeling Approach: Volume 1: Nonlinear System Parameter Identification*. New York: Springer, 1999.

[20] L. Ljung, "Identification of nonlinear systems," Dept. Electr. Eng., Linköping Univ., Linköping, Sweden, Tech. Rep. LiTH-ISY-R-2784, Jun. 2007.

[21] S. Boyd, "Volterra series: Engineering fundamentals," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, CA, 1985.

[22] S. Maas, "How to model intermodulation distortion," in *Proc. IEEE MTT-S Int. Microw. Symp. Dig.*, 1991, pp. 149–151.

[23] A. Soury, E. Ngoya, and J. Rousset, "Behavioral modeling of RF and microwave circuit blocks for hierarchical simulation of modern transceivers," in *Proc. IEEE MTT-S Int. Microw. Symp. Dig.*, 2005, pp. 975–978.

[24] S. Billings and S. Fakhouri, "Identification of a class of nonlinear systems using correlation analysis," in *Proc. Inst. Elect. Engineers*, vol. 125. Jul. 1978, pp. 691–697.

[25] A. Hagenblad, L. Ljung, and A. Wills, Maximum likelihood identification of Wiener models, *Automatica*, Nov. 2007, submitted for publication.

[26] A. H. Tan and K. Godfrey, "Identification of Wiener–Hammerstein models using linear interpolation in the frequency domain (LIFRED)," *IEEE Trans. Instrum. Meas.*, vol. 51, no. 3, pp. 509–521, Jun. 2002.

[27] M. Kozek and C. Hametner, "Block-oriented identification of Hammerstein–Wiener-models using the RLS-algorithm," *Int. J. Appl. Electromagnetics Mechanics*, vol. 25, no. 10, pp. 529–535, May 2007.

[28] K. C. Sou, A. Megretski, and L. Daniel, "Convex relaxation approach to the identification of the Wiener-Hammerstein model," in *Proc. IEEE Conf. Decision Control*, Dec. 2008, pp. 1375–1382.

[29] R. Smith and J. Doyle, "Model validation: A connection between robust control and identification," *IEEE Trans. Automat. Control*, vol. 37, no. 7, pp. 942–952, Jul. 1992.

[30] K. Poolla, P. Khargonekar, A. Tikku, J. Krause, and K. Nagpal, "A time-domain approach to model validation," *IEEE Trans. Automat. Control*, vol. 39, no. 5, pp. 951–959, May 1994.

[31] A. Megretski, "Convex optimization in robust identification of nonlinear feedback," in *Proc. IEEE Conf. Decision Control*, Dec. 2008, pp. 1370–1374.

[32] *STINS: A MATLAB tool for stable identification of nonlinear systems via semidefinite prgoramming* [Online]. Available: http://www.rle.mit.edu/cpg/codes/stins

[33] J. C. Willems, "Dissipative dynamical systems, part I: General theory," *Arch. Rational Mechanics Anal.*, vol. 45, no. 5, pp. 321–393, 1972.

[34] W. Lohmiller and J. Slotline, "On contraction analysis for nonlinear systems," *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.

[35] P. Parillo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, Dept. Control Dynamical Syst., California Instit. Technol., Pasadena, CA, 2000.

[36] B. Sturmfels, "Polynomial equations and convex polytopes," *The Am. Math. Monthly*, vol. 105, no. 5, pp. 907–922, 1998.

[37] B. Reznick, "Extremal PSD forms with few terms," *Duke Math. J.*, vol. 45, no. 2, pp. 363–374, 1978.

[38] *SPOT: Syst. Polynomial Optimization Toolbox* [Online]. Available: http://web.mit.edu/ameg/www/

[39] J. Lumley, "The structures of inhomogeneous turbulent flow," in *Atmospheric Turbulence and Radio Wave Propagation*, A. M. Yaglow and V. I. Tatarski, Eds. Moscow, Russia: Nauka, 1967, pp. 166–178.

[40] K. Willcox and J. Peraire, "Balanced model reduction via the proper orthogonal decomposition," in *Proc. 15th AIAA Comput. Fluid Dynamics Conf.*, Jun. 2001, pp. 2323–2330.

[41] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Methods Softw.*, vols. 11–12, nos. 1–4, pp. 625–653, 1999.
[42] M. Rewienski, "A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems," Ph.D. disertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Instit. Technol., Cambridge, MA, 2003.
[43] P. Haldi, D. Chowdhury, P. Reynaert, G. Liu, and A. M. Niknejad, "A 5.8 ghz 1 v linear power amplifier using a novel on-chip transformer power combiner in standard 90 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 43, no. 5, pp. 1054–1063, May 2008.
[44] T. H. Lee, *The Design of CMOS Radio-Frequency Integr. Circuits*. Cambridge, U.K.: Cambridge University Press, 1998.

Bradley N. Bond received the B.S. degree in engineering science and mechanics from Penn State University, University Park, PA, in 2004, and the M.S. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, MA, in 2006 and 2010, respectively.

He is currently a Post-Doctoral Researcher with the Computational Prototyping Group, Department of Electrical Engineering and Computer Science, MIT. His current research interests include numerical simulation, dynamical system theory, and nonlinear modeling.

Zohaib Mahmood received the B.S. degree in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan. He is currently pursuing the M.S. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA.

His current research interests include modeling of dynamical nonlinear systems, numerical simulation, and analog circuit design.

Yan Li received the B.E. degree in electrical engineering from the University of Science and Technology of China, Hefei, in 2004, and the M.A.Sc. degree from McMaster University, Hamilton, ON, Canada, in 2006. Currently, she is pursuing the Ph.D. degree from the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, under the supervision of Professor Vladimir Stojanovic in the Integrated Systems Group.

Her current research interests include convex optimization and robust optimization for integrated systems.

Ranko Sredojević received the B.S. degree in electrical engineering from the Department of Electronics, University of Belgrade, Belgrade, Serbia, in 2004. He is currently pursuing the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, working on circuit and system optimization as a member of the Integrated Systems Group.

His current research interests include optimization, convex analysis, data fitting and modeling, power and analog electronics, and control systems.

Alexandre Megretski is currently a Professor of Electrical Engineering and Computer Science with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA. He was a Researcher with both the Royal Institute of Technology, Stockholm, Sweden, and the University of Newcastle, NSW, Australia, and a Faculty Member with Iowa State University, Ames.

His current research interests include nonlinear dynamical systems (identification, analysis, and design), validation of hybrid control algorithms, optimization, applications to analog circuits, control of animated objects, and relay systems.

Vladimir Stojanović received the Dipl.Ing. degree from the University of Belgrade, Belgrade, Serbia, in 1998, the M.S.E.E. and Ph.D. degrees from Stanford University, Stanford, CA, in 2000 and 2005, respectively.

From 2001 to 2004, he was a Principal Engineer with the Logic Interface Division of Rambus, Inc., Los Altos, CA. Currently, he is a Principal Investigator with the Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA. His current research interests include optimization of integrated circuits and systems, application of convex optimization to digital communications, analog and VLSI circuits, communications and signal processing architectures, and high-speed digital and mixed-signal IC design.

Yehuda Avniel received the Ph.D. degree from the Massachusetts Institute of Technology (MIT), Cambridge, MA, in 1985, in the field of control.

He is currently a Research Affiliate with the Research Laboratory of Electronics, MIT.

Luca Daniel received the Laurea degree summa cum laude in electronic engineering from the Universita' di Padova, Padua, Italy, in 1996, and the Ph.D. degree in electrical engineering from the University of California, Berkeley, in 2003.

He is an Associate Professor with the Department of Electrical Engineering and Computer Science and a Principal Investigator of the RLE Computational Prototyping Group, Massachusetts Institute of Technology, Cambridge, MA. His current research interests include parameterized model order reduction of linear and nonlinear dynamical systems, mixed-signal, RF and mm-wave circuit design and robust optimization, power electronics, microelectromechanical systems design and fabrication, parasitic extraction, and accelerated integral equation solvers.