Projection-Based Approaches for Model Reduction of Weakly Nonlinear, Time-Varying Systems

Joel R. Phillips

***Abstract*—The problem of automated macromodel generation is interesting from the viewpoint of system-level design because if small, accurate reduced-order models of system component blocks can be extracted, then much larger portions of a design, or more complicated systems, can be simulated or verified than if the analysis were to have to proceed at a detailed level. The prospect of generating the reduced model from a detailed analysis of component blocks is attractive because then the influence of second-order device effects or parasitic components on the overall system performance can be assessed. In this way overly conservative design specifications can be avoided.**

This paper reports on experiences with extending model reduction techniques to nonlinear systems of differential–algebraic equations, specifically, systems representative of RF circuit components. The discussion proceeds from linear time-varying, to weakly nonlinear, to nonlinear time-varying analysis, relying generally on perturbational techniques to handle deviations from the linear time-invariant case. The main intent is to explore which perturbational techniques work, which do not, and outline some problems that remain to be solved in developing robust, general nonlinear reduction methods.

***Index Terms*—Circuit noise, circuit simulation, nonlinear systems, reduced-order systems, time-varying circuits.**

I. Introduction

DESIGN of modern mixed-signal systems begins and ends at the system level. Particularly in communications applications, such as wireless systems described in signal-processing terms, correct operation of the mixed-signal components must be verified in the context of the overall system design. Operation of analog components, however, is tied closely to the lowest level details of implementation. In a mixed-technology context, it is thus necessary to provide efficient mechanisms for assessing the impact of the implementation details at various levels of abstraction. Most critical is final verification of system performance, including detailed effects of implementation tradeoffs, process limitations, and parasitic effects. Lack of effective verification can lead to costly redesigns and multiple silicon respins. The premise of this paper is that model-based approaches that can propagate circuit performance characteristics, in a bottom-up manner, to the highest levels of abstraction will provide the needed mechanism to assess low-level effects in system-level design and verification.

The implied need is the ability to generate reduced complexity models of the end-to-end behavior of circuit blocks, that incorporate transistor-level effects, but are suitable for simulation at system level. These macromodels can be used to perform rapid system-level simulation of engineering designs that are too complicated to analyze at the detailed component level. It is desirable that the model generation step be performed without requiring additional modeling expertise from designers, and without introducing major new steps into already tight schedules. The accuracy of the models must be predictable and controllable. These constraints motivate investigating automated model extraction procedures, in particular, model reduction strategies.

Model reduction refers to the procedure of automatic generation of system macromodels by direct operation on the lower-level, detailed descriptions. Compared to approaches that attempt to fit models to observed system behavior, model reduction approaches can easily obtain accurate, robust models, and usually with considerably less effort. Because the macromodels are obtained by operations on the original system itself, reduction can obtain and exploit information about the internal structure of the system to be approximated. For example, in a linear system, Krylov-subspace-based reduction methods can obtain very accurate information about pole locations and multiplicities in high-order, multiinput, multioutput systems. Such information is much more difficult to obtain by analyzing frequency response or time-series data. Second, because of access to the original system, more effective control of error in the model is possible. Finally, only by generating macromodels from detailed physical descriptions of components can the influence of complicated second-order physical effects ultimately be included at the system level. Thus an essential feature of reduction approaches is a relatively thorough control and assessment of approximation errors that is gained by formal analysis of the reduction algorithms.

Reduction algorithms have met with considerable success for modeling lumped, linear time-invariant (LTI) systems such as electrical interconnect. The most successful algorithms for reduction of large-scale linear systems have been *projection*-based approaches [1], [2]. Algorithms such as PVL [3], Arnoldi methods [4], and PRIMA [5] obtain reduced models by projecting the linear equations describing the LTI model system into a subspace of lower dimension. The subspace chosen determines the approximation properties of the reduced model. These algorithms exploit the connection between Krylov subspaces and rational approximation to develop algorithms that have a known relationship to the frequency-domain characteristics of the system, for example, matching the transfer function and some of its derivatives at various points in the complex plane. Because the algorithms rely only on simple operations

Manuscript received June 3, 2002; revised September 5, 2002. This paper was recommended by Guest Editor G. Gielen.

The author is with Cadence Berkeley Laboratories, San Jose, CA 95134 USA (e-mail: jrp@cadence.com).

Digital Object Identifier 10.1109/TCAD.2002.806605

0278-0070/03$17.00 © 2003 IEEE

such as matrix–vector products and linear system solutions, they can be applied to very large systems if efficient algorithms for products and solutions are available. And they usually are.

The motivation for the work described in this paper was examining the extent to which work on model reduction for LTI systems can be leveraged in analyzing a wider class of systems, as mixed-technology systems generally have parts that cannot be modeled as LTI. RF circuits form a particularly interesting application area as they illustrate frequency conversion and distortion effects that cannot be predicted by LTI models. In addition, an effective reduction algorithm must also provide a solution to the difficulty of modeling the multiple timescales that occur in RF circuits.

Many systems that are not LTI can be accurately modeled as linear time-varying (LTV). For example, if a nonlinear circuit model is linearized around a time-varying large signal, the resulting model is LTV. Clearly, the set of circuits that can be accurately modeled as LTV is much larger than the set that can be described as LTI. Many RF components (e.g., mixers and filters) are designed to have a near-linear response in the signal path, but may have a strongly nonlinear response to other excitations, such as the clock of a switched-capacitor filter, or a mixer's local oscillator. Such components are prime candidates for LTV model reduction, as several performance metrics, such as gain, noise, and bandwidth, can be predicted by analyzing an LTV description of the circuit.

Model reduction algorithms for time-varying systems are known [6], [7], but not as well-developed or as widely used as model reduction for LTI systems. More importantly, the known algorithms do not produce quite the right results that are needed for mixed-technology modeling. The first task of the paper will be to 1) show how models suitable for system simulation can be described from a transistor-level, LTV description, and 2) show how projection techniques suitable for reducing large-scale systems can be applied to generate the models.

The next step beyond modeling LTV systems is to show how model reduction algorithms can be applied to systems that exhibit weak nonlinear responses. Including effects of weak nonlinearities generally can extend the range of accuracy beyond that achievable with linear models. Weakly nonlinear analysis is particularly useful in RF communications circuits because the signal path is designed to be close to linear, and it is of great interest to characterize the lowest-order deviation from linearity. In RF systems are also an interesting test case because it is necessary to combine the weakly nonlinear and time-varying model reduction approaches into a single algorithm.

The approach of the paper will be to first review methods for reduction of LTI systems and leverage these techniques to produce a method for reduction of LTV systems. Next, approaches to nonlinear, but time-invariant systems will be discussed, and finally the methods will be combined into a technique for reduction of weakly nonlinear, time-varying systems. While not presenting a complete solution to the nonlinear model reduction problem, the paper will present some basic first results in the area, compare and contrast with similar methods developed concurrently [8]–[12], and speculate on the most productive avenues for future investigation.

II. Model Reduction for LTI Systems

In this section we will review model reduction techniques for LTI models. Consider the LTI state-space model

$$E\,\frac{dx}{dt} = Ax + Bu(t) \qquad y(t) = Cx(t) + Du(t) \tag{1}$$

where $E, A \in R^{n\times n}$, $B \in R^{n\times p}$, $C \in R^{q\times n}$, $D \in R^{q\times p}$ define the model dynamics, $x(t) \in R^n$, is the internal state, $u(t) \in R^p$ is the input, $y(t) \in R^q$ is the output, n is the system order, and p and q are the number of system inputs and outputs, respectively. If $E = I$, the system (1) describes a set of linear ordinary differential equations (ODEs) in normal form. If $E \neq I$ but is nonsingular, (1) describes an implicit set of ODEs, and if E is singular, (1) describes a set of differential–algebraic equations.

A projection-based reduction scheme involves selecting a matrix V whose columns span a "useful" subspace, and drawing an approximation $\hat{x} \simeq x$ from this subspace as $\hat{x} = Vz$. Equations for the reduced system are obtained by defining the residual $r \equiv A\hat{x} + Bu - Ed\hat{x}/dt$ and requiring the residual to be orthogonal to another space W, $W^T r = 0$. The state-space equations for the reduced model are then given by

$$\hat{E}\,\frac{dz}{dt} = \hat{A}z + \hat{B}u \qquad \hat{y} = \hat{C}z + \hat{D}u \tag{2}$$

where

$$\hat{A} \equiv W^T AV \quad \hat{B} \equiv W^T B \quad \hat{E} \equiv W^T EV \quad \hat{C} \equiv CV. \tag{3}$$

It is usually (but not always) the case that $\hat{D} = D$. If $W = V$, the projection is said to be *orthogonal*, as the residual will be orthogonal to the basis space spanned by V. We will mostly use orthogonal projection schemes in this paper.

Most reduction schemes can be cast into the projection formulation, the various techniques differing in the choice of the matrices V and W. Roughly speaking, and in increasing order of rigor, three main strategies for choosing V and W may be identified: heuristic schemes, methods based on rigorous system-theoretic analysis [13], [14], and parameter-matching techniques. Heuristic schemes include choosing W and V based on eigenvector analysis, which leads to reduced models that (exactly) preserve a subset of the poles and residue matrices of the larger model, but do not necessarily match the frequency response well. Techniques such as truncated balanced realization [13] have only recently been applied to large-scale systems, and rely on a Krylov (i.e., parameter-matching) technique as an initial step. As a result, the parameter-matching schemes are popular because it is often relatively easy to obtain an approximation space V of small dimension that results in accurate macromodels. The popular model reduction schemes are based on the theory of Krylov subspaces.

Definition 1—Krylov Subspace: The Krylov subspace $K_m(A, p)$ generated by a matrix A and vector p, of order m, is the space spanned by the set of vectors $\{p, Ap, A^2p, \ldots, A^{m-1}p\}$.

The essential elements of Krylov-subspace-based reduction are given by

Theorem 1: Suppose $K_m(A^{-1}, p) \subset \mathrm{colspan}(V)$, then $V(V^TAV)^{-k}V^TB = V\hat{A}^{-k}\hat{b} = A^{-k}b$, for $k < m$.

Proof: See [2]. □

To see why this is useful, consider writing the state-space model in the Laplace-transform domain

$$sEx(s) = Ax(s) + Bu(s) \qquad y(s) = Cx(s) \tag{4}$$

or $y(s) = H(s)u(s)$ where the transfer function $H(s)$ is given by

$$H(s) = C(sE - A)^{-1}B. \tag{5}$$

The transfer function $C(sE-A)^{-1}B$ is a rational function in s, so it seems logical to approximate it in turn with a rational function, such as a Padé approximant [15]. Pade approximants, and most of the other approximants used for model reduction, have the property that they match the transfer function and some of its derivatives with respect to s. That projection generates rational approximants is clear from the fact that the reduced transfer function $\hat{H}(s) = \hat{C}(s\hat{E} - \hat{A})^{-1}B$ is also a rational function, because the reduced model is also a linear state-space model. Theorem 1 connects the moments to the projection matrix V and is the key to the model reduction procedure. Note that the kth derivative, or *moment*, of the transfer function is given by $C(A^{-1}E)^k A^{-1}B$. Clearly the approximants we wish to generate are connected with powers of the matrix $A^{-1}E$ acting on B. These ideas are summarized by the following theorem.

Theorem 2—Krylov-Subspace Model Reduction: If the columns of V span $K_m(A^{-1}E,\, A^{-1}B)$, then the reduced-order transfer function $\hat{C}(s\hat{E} - \hat{A})^{-1}\hat{B}$ matches the first m moments of the original transfer function $C(sE - A)^{-1}B$.

Proof: Follows from the Taylor expansion of $(sE - A)^{-1}$ and Theorem 1 [1], [2]. □

Equally important is the following.

Theorem 3—Adjoint Krylov-Subspace Model Reduction: If the columns of W span $K_m(A^{-T}E^T,\, A^{-T}C^T)$, then the reduced-order transfer function $\hat{C}(s\hat{E} - \hat{A})^{-1}\hat{B}$ matches the first m moments of the original transfer function $C(sE - A)^{-1}B$.

For example, in the PVL algorithm [3], which forms a Pade approximation and is, thus, equivalent to the asymptotic waveform evaluation [16] technique because of the relation between Lanczos algorithm and Pade approximants [17], the choice of W and V is $W^T = W_l^T A^{-1}$, $V = V_l$, where W_l and V_l contain the biorthogonal Lanczos vectors, and in one variant of model reduction based on the Arnoldi method [4], $W^T = V_a^T A^{-1}$, $V = V_a$ where V_a is the orthonormal matrix generated by the Arnoldi process. PRIMA [5] uses $W = V = V_a$, where the columns of V_a span a Krylov space.

More generally, given a low-rank matrix B that maps state-space model inputs to the model internal state, by computing a matrix V whose columns span $K_m((s_0E - A)^{-1}E,\, (s_0E - A)^{-1}B)$, we may derive a reduced model such that the transfer function of the original and reduce system will match up to the mth derivative about the point s_0 in the complex plane. The essential computation in this procedure is the application of $(s_0E - A)^{-1}$ to a vector. This approach can be extended to the case of multipoint approximants where the transfer function and some of its derivatives are matched at several points in the complex plane [2], [18]–[21]. In this case W and V must contain a basis for the union of the Krylov subspaces constructed at the different expansion points.

III. Modeling LTV Systems

A. Time-Varying Systems in a Mixed-Technology Context

The time-varying analog to (1) is the state-space model

$$E(t)\,\frac{dx}{dt} = A(t)x + B(t)u(t) \qquad y(t) = C(t)x(t) + D(t)u(t). \tag{6}$$

In integrated circuit applications, the most common origin of LTV systems is by linearization of a nonlinear system of equations around a time-varying operating point. For example, to obtain an LTV circuit description, from, say, a set of circuit equations written using modified nodal analysis [22], first the differential equations describing the circuit are written as

$$f\left(v^{(T)}\right) + \frac{d}{dt}\,q\left(v^{(T)}\right) = Bu^{(T)}(t) \tag{7}$$

where u represents the input sources, $v^{(T)}$ describes the node voltages, f is the relation between voltages and currents, and the function q relates voltages to charges (or fluxes) internals. We have written the voltage v and input variable u with the superscript T to indicate that they are total quantities that we will split into two parts, a large-signal part and a small signal part, in order to obtain an LTV model

$$u^{(T)} = u^{(L)} + u \qquad v^{(T)} = v^{(L)} + v. \tag{8}$$

By linearizing around $v^{(L)}$, an LTV system of the form

$$\underline{G}(t)v + \frac{d}{dt}\left(\underline{C}(t)v\right) = bu(t) \tag{9}$$

where $\underline{G}(t) = \partial f(v^{(L)})(t)/\partial v$ and $\underline{C}(t) = \partial q(v^{(L)})(t)/\partial v$ are the time-varying conductance and capacitance matrices, is obtained for the small response v. To relate to the standard notation, we may make the identification $E(t) = \underline{C}(t)$, $A(t) = -(\underline{G}(t) + \dot{\underline{C}}(t))$.

Formally speaking, we could consider obtaining a reduced model in similar form by applying a projection operation with matrices W and V just as in the time-invariant case

$$\begin{aligned} \hat{A}(t) &\equiv W^T(t)A(t)V(t)\\ \hat{B}(t) &\equiv W^T(t)B(t)\\ \hat{E}(t) &\equiv W^T(t)E(t)V(t)\\ \hat{C}(t) &\equiv C(t)V(t). \end{aligned} \tag{10}$$

This model is of potentially smaller dimension, and thus lower computational cost, than the original model (6), but it is not in a form suitable for use in higher level simulation.

To motivate the development of a reduction algorithm, let us first consider two special cases of interest in integrated circuit applications. Then we will present a general purpose modeling algorithm. Both example problems are periodic time-varying systems, where $E(t+T) = E(t)$, $A(t+T) = A(t)$, $B(t+T) = B(t)$, $C(t+T) = C(t)$ for some fundamental period T. The first example relates to systems with a time-varying operating point that are sampled at the output, the second to systems, such as RF transceivers, that convert narrowband communications signals between well defined carrier frequencies.

First, consider a switched-capacitor filter problem. A five-pole low-pass switched capacitor filter, containing 71 MOSFETs was simulated and the LTV response to a 1 kHz-sinusoid computed. The results are shown in Fig. 1. The jagged

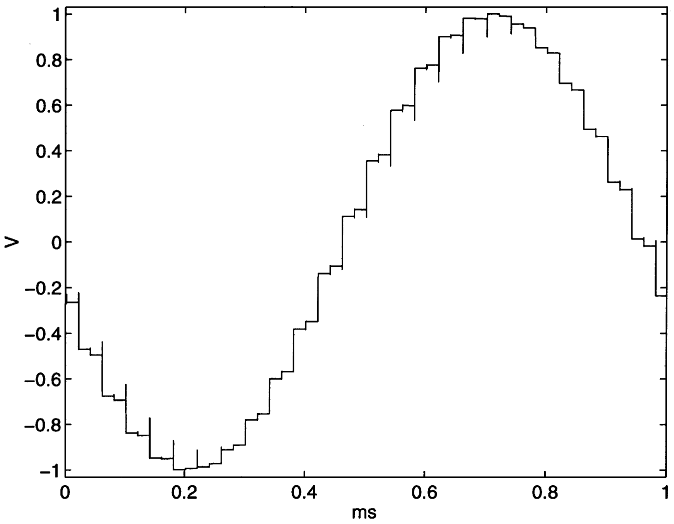

Fig. 1. Response of the switched-capacitor filter to a small 1-kHz sinusoid.

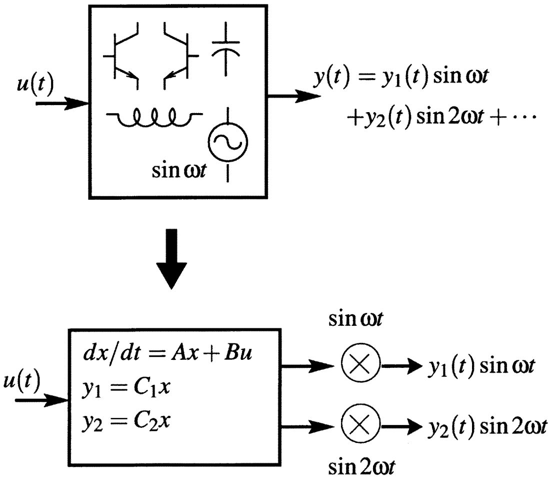

Fig. 2. Pictorial representation of models for LTV systems. Top: The original circuit contains periodic time-varying sources and nonlinear devices linearized around a time-varying operating point. Input $u(t)$ generates outputs at all harmonics of the fundamental period. The outputs contain fast time-varying behavior even if the inputs are simple. Bottom: The LTI-reduced model, which is smaller, and contains no details of time-varying behavior and no time-variant elements. Time-varying behavior (sampling, a fast carrier, etc.) is abstracted away. The original outputs could be recovered by introducing time-varying components external to the reduced model, but this is not desirable in high-level modeling.

waveform shape is a result of the strong nonlinearity of the filter with respect to the clock. Incorporation of such effects is the point-of-time-varying modeling. An LTI system would produce a smooth sinusoid, shifted in phase and scaled in magnitude, as a response to this input. The output of the filter, however, is usually followed by a function (e.g., the sample/hold of an A/D converter) that discards the filter output outside some small sample time window. To model this system at a higher level, we would need a reduced-order model that relates continuous, sinusoidal inputs to the window of output that the following circuitry (the sample and hold) needs. This can be achieved by choosing $B(t)$ to be a constant matrix B, and $C(t)$ to be the sampling function. The output $y(t)$ can then be interpreted in a discrete time sense. In the final model, the internal rapid time-variation of the operating point will not appear. It has been abstracted away, further increasing the efficiency of later high-level simulation.

A similar abstraction is desirable when modeling RF systems. Here, because of the presence of a high-frequency carrier, the time-variation of the system can be at a very fast rate, much faster than the data signals passing through the transceiver. Often, it is desirable to extract a model than represents only the "baseband-equivalent" behavior of the model. The time-varying carrier will be abstracted away. In this view, the spectrum of the signals of interest is concentrated in a narrow band around the harmonics of the fundamental frequency ω_0 ($\omega_0 = 2\pi/T$ if T is the period of variation) or frequencies. We desire models that represent the transfer functions between inputs in one frequency band, and outputs possibly in the same or another frequency band. These transfer functions can be interpreted, as in the LTI case, as the system response when a single frequency sinusoid is applied to the inputs. In circuit problems, since the physical inputs and outputs are fixed, the transfer-function model can be achieved, as in [23], [24], and in most harmonic balance codes, by choosing the columns of $B(t)$ and/or rows of $C(t)$ to have sinusoidal time-variation. If we can construct reduced models that represent these transfer functions, we will have a modeling algorithm that not only reduces the size of the internal systems, but also abstracts away the fast time-varying behavior (Fig. 2). Therefore now we proceed to see how these transfer functions may be calculated.

B. Model Reduction for LTV Systems

In model reduction of LTI systems, most progress has been made in exploiting rational approximations to the frequency-domain transfer functions. Thus motivated, we adopt the formalism of Zadeh's variable transfer functions [25] that were developed to describe time-varying systems. In this formalism, the response $v(t)$ can be written as an inverse Fourier transform of the product of a time-varying transfer function and the Fourier transform of $u(t)$ and $u(w)$. That is,

$$v(t) = \int_{\infty}^{-\infty} h(i\omega', t) u(\omega') e^{i\omega' t} \, d\omega'. \qquad (11)$$

To obtain the frequency-by-frequency response, we let u be a single-frequency input, $u_{\omega'} = u_\omega \delta(\omega - \omega')$, and see that

$$v(t) = h(i\omega, t) u(\omega) e^{i\omega t}. \qquad (12)$$

Writing $s = i\omega$ and substituting into (9), an equation for $h(s, t)$ (which is a matrix with as many columns as B has) is obtained as

$$\underline{G}(t)h(s, t) + \frac{d}{dt}\left(\underline{C}(t)h(s, t)\right) + s\underline{C}(t)h(s, t) = B(t). \qquad (13)$$

These equations may also be obtained from the multivariate-partial-differential-equation formalism [26], [27]. Note that, since $h(s, t)$ comes from a lumped linear system, it will be a rational function with an infinite number of poles [28]. For example, in a periodically time-varying system with fundamental frequency ω_0, if η is a pole (in particular, a Floquet multiplier) of the system, then $\eta + k\omega_0$, k an integer, will be a pole of $h(s, t)$. This is because signals can be converted by harmonic k in the time-varying description from a frequency $\eta - k\omega_0$ to the pole at η.

Because, for the lumped time-varying systems considered here, the time-varying transfer functions are rational functions, it seems reasonable to believe that reduced models can be obtained from the same sorts of rational approximation paths that have been so profitable for reduction of LTI systems. Therefore, we first seek a representation of the transfer functions in terms of finite-dimensional matrices.

To perform actual computation of the transfer functions, we discretize the (13). Because the focus of this paper is on periodic LTV systems that occur in RF applications, at this point we also introduce explicit assumptions about the time-variation of the system, and explain how the input–output mappings are incorporated into the model reduction procedure.

Following [29], [23], and [24], in the example case of a backward-Euler discretization, (13) becomes

$$(sE_{TV} - A_{TV})h(s) = B_{TV} \tag{14}$$

with

$$A_{TV} = -\begin{bmatrix} \frac{\underline{C}_1}{\Delta_1} + \underline{G}_1 & & & -\frac{\underline{C}_M}{\Delta_1} \\ -\frac{\underline{C}_1}{\Delta_2} & \frac{\underline{C}_2}{\Delta_2} + \underline{G}_2 & & \\ & \ddots & \ddots & \\ & & -\frac{\underline{C}_{M-1}}{\Delta_M} & \frac{\underline{C}_M}{\Delta_M} + \underline{G}_M \end{bmatrix} \tag{15}$$

$$E_{TV} = \begin{bmatrix} \underline{C}_1 & & & \\ & \underline{C}_2 & & \\ & & \ddots & \\ & & & \underline{C}_M \end{bmatrix} \tag{16}$$

$$h(s) = [\, h_1(s)^T \quad h_2(s)^T \quad \cdots \quad h_M(s)^T \,]^T \tag{17}$$

and

$$B_{TV} = [\, B_1^T \quad B_2^T \quad \cdots \quad b_M^T \,]^T \tag{18}$$

where $\underline{G}_j = \underline{G}(t_j)$, $\underline{C}_j = \underline{C}(t_j)$, $B_j = B(t_j)$, $h_j(s) = h(s, t_j)$, and Δ_j is the jth timestep.

With additionally

$$C_{TV} = [\, C_1 \quad C_2 \quad \cdots \quad C_M \,]^T \tag{19}$$

where $C_j = C(t_j)$, the matrix of baseband-equivalent transfer functions $H_{TV}(s)$ is given by

$$H_{TV}(s) = C_{TV}h(s) = C_{TV}(sE_{TV} - A_{TV})^{-1}B_{TV}. \tag{20}$$

The discretization procedure has converted the time-varying system of (13) to an equivalent LTI system, of dimension larger by a factor equal to the number of timepoints in the discretization. At this point, any of the algorithms developed for reduction of lumped LTI systems can be applied to the matrices and vectors defined in (15)–(20). The result will be an LTI state-space model that represents the baseband-equivalent transfer functions in $H_{TV}(s)$.

C. Two-Stage Approximate-Solve Computational Scheme

In [24], the transfer functions from a small-signal sinusoidal input to sinusoids at harmonics of the output were obtained by solving the finite-difference equations

$$\begin{bmatrix} \frac{\underline{C}_1}{\Delta_1} + \underline{G}_1 & & & -\frac{\underline{C}_M}{\Delta_1} \cdot \alpha(s) \\ -\frac{\underline{C}_1}{\Delta_2} & \frac{\underline{C}_2}{\Delta_2} + \underline{G}_2 & & \\ & \ddots & \ddots & \\ & & -\frac{\underline{C}_{M-1}}{\Delta_M} & \frac{\underline{C}_M}{\Delta_M} + \underline{G}_M \end{bmatrix} \begin{bmatrix} \tilde{v}(t_1) \\ \tilde{v}(t_2) \\ \vdots \\ \tilde{v}(t_M) \end{bmatrix} = \begin{bmatrix} \tilde{B}(s, t_1) \\ \tilde{B}(s, t_2) \\ \vdots \\ \tilde{B}(s, t_M) \end{bmatrix} \tag{21}$$

where $\alpha(s) \equiv e^{-sT}$, T is the fundamental period, and $\tilde{B}(s, t_k) = e^{st_k}B$. The transfer function $h(s, t)$ is then given by $h(s, t) = e^{-st}\tilde{v}(t)$.

It is now convenient to decompose A_{TV} into a lower triangular and an upper triangular piece, $A_{TV} = L + U$. Using the expressions for L and U, (21) becomes

$$(L + \alpha(s)U)\tilde{v} = \tilde{B}(s). \tag{22}$$

If we define a small-signal modulation operator $\Omega(s)$,

$$\Omega(s) \equiv \begin{bmatrix} Ie^{st_1} & & & \\ 0 & Ie^{st_2} & & \\ & \ddots & \ddots & \\ & & 0 & Ie^{st_M} \end{bmatrix} \tag{23}$$

then we can make the identification

$$h(s) = \Omega^H(s)\tilde{v}(s) \qquad \tilde{B}(s) = \Omega(s)B_{TV} \tag{24}$$

and more importantly we make the identification that

$$sE_{TV} - A_{TV} \approx \Omega^H(s)[L + \alpha(s)U]\Omega(s). \tag{25}$$

The left- and right-hand sides of (25) differ because of the way the small applied test signal was treated. The left-hand side corresponds to a spectral discretization, and the right-hand side to a finite-difference discretization.

Suppose we need to solve (22) for some right-hand side $\tilde{B}$. Again following [24], consider preconditioning with the matrix L. Because L is lower triangular, with a small block bandwidth, block Gaussian elimination is very efficient at computing the inverse acting on a vector. In this procedure, once the M diagonal blocks have been factored, an operation that must be performed exactly once, then every application of the inverse is an M step procedure, at each step needing a backsolve with the factored diagonal matrices and multiplication by the blocks off the diagonal. There is one off-diagonal block in each row for the simple backward Euler discretization. The preconditioned system can be written as

$$\left(I + \alpha(s)L^{-1}U\right)\tilde{v} = L^{-1}\tilde{B}(s). \tag{26}$$

Suppose that (26) is solved by a Krylov-subspace-based iterative method such as GMRES [30].[1] Because the Krylov sub-

[1] These are different Krylov subspaces than the ones used for model reduction.

space of a matrix A is invariant to shifts [31], [24] of the form $A \to A + \beta I$, the same Krylov subspace may be used to solve (26) at multiple frequency points. This "recycled Krylov subspace" algorithm is made even more efficient [24] by exploiting the special structural properties of $L^{-1}U$, and because the spectrum of $L^{-1}U$, being related to the Floquet multipliers of the LTV system, is usually clustered. The recycling of the Krylov space also accelerates solution with different right-hand sides. The final point is that systems of the form (22) may be solved very efficiently for different frequencies and right-hand side vectors.

To see how the finite-difference form can be used to derive an appropriate basis for projecting the matrices A_{TV}, E_{TV}, B_{TV}, and C_{TV} in order to give the final reduced model, consider what would happen if the matrix V used for the projection was not a basis for the Krylov subspace of $(sE_{TV} - A_{TV})^{-1}$, but instead a nearby matrix. The reduced model would still be a projection of the original, the difference being that interpolation conditions would be only approximately satisfied. If the approximation of the interpolation conditions (or matched moments) was comparable to or less than the accuracy desired in the final model, these additional errors would be negligible. This suggests that the basis for the projector in the model-reduction procedure be obtained by using the finite-difference equations. Because the basis will be a good approximation to the Krylov subspaces of the spectral operator, good reduced models will be obtained.

The overall algorithm can be seen as an example of a two-stage model reduction scheme. In the first stage, Krylov subspaces $K(L^{-1}U, \tilde{B}(s_k))$ are accumulated for several s_k. A single basis V_1 is constructed to span the union of these spaces. V_1 is constructed such that (22) can be solved to an appropriate tolerance at each s_k, by adding in to the overall basis the Krylov space $K(L^{-1}U, \tilde{B}(s_k))$ of minimal necessary size. The solutions to (22) are used to construct a second basis V_2. This basis V_2 is used to form the reduced model of H_{TV} via the projection equations. In addition, because of the recycled Krylov scheme, obtaining projectors from expansions about multiple frequency points is essentially no more expensive than single-frequency-point expansions.

IV. Nonlinear Systems Modeling

A. Formal Projection Operations

To illustrate some of the difficulties in developing nonlinear reduction algorithms, let us contrast the behavior of generic linear and nonlinear state-space models under projection-based reduction. The Krylov-subspace-based projection schemes work for large-scale linear systems for four reasons. First, good choices for the subspace defined by V exist. Rational approximation paradigms suggest matching properties of the transfer function. Second, the columns of the matrix V can be efficiently obtained. Only products with A or solution of linear systems $Ax = b$ are required, and thus any sparsity or special structure of the underlying linear system can be exploited. Therefore, reduced models can be efficiently obtained as typically only a small number of matrix–vector products are needed. Finally, the reduced models can be efficiently simulated. This is guaranteed by construction, since the reduced models have the same form as the original equations, i.e., linear systems of differential equations, but of much lower dimensionality.

In contrast, consider a system with a nonlinear state-evolution function

$$E\,\frac{dx}{dt} = f(x) + Bu \qquad y = Cx \tag{27}$$

where $f\colon R^n \to R^n$. Circuit equations can always be put into this form, possibly by introducing additional algebraic variables. We may formally apply the projection recipe to this system of equations to obtain a "reduced" model

$$\hat{E}\,\frac{dz}{dt} = V^T f(Vz) + \hat{B}u \qquad \hat{y} = \hat{C}\hat{x} \tag{28}$$

where $\hat{E}$, $\hat{B}$, $\hat{C}$ are as before. Several difficulties with this approach are apparent.

First, it is not at all clear how to choose V, and even less is known about efficient computation. Approaches based on analysis of the linearized model have been proposed [32], [9], but by definition such approaches do not include information about nonlinear model properties. Heuristic approaches based on the singular value decomposition of a statistically representative sampling of the state-space $x(t)$ have met with some success [33]–[35], but the computations are extensive and little control over model accuracy is available. Balancing based procedures exist in theory [36], but it is not clear how they may be computed. More importantly, in the general case, interpreting the term $V^T f(Vz)$ as a *reduced* model is problematic.

Since f is a nonlinear function, V may not be generally passed through the parentheses. It is always possible to evaluate $V^T f(Vz)$ by explicitly constructing $\hat{x} = Vz$, evaluating the nonlinearity $\hat{f}(z) \equiv f(\hat{x})$ and finally explicitly projecting onto the reduced space by multiplication with V^T. However, except in special situations (e.g., large linear subnetworks), if there are N degrees of freedom in the detailed circuit description, then evaluation of the model will require $O(N)$ operations. As a result, efficient simulation is not guaranteed. For example, in nonlinear circuit simulation, even for circuits with tens of thousands of nodes, roughly half the simulation time is spent in evaluation of the nonlinear function f. Thus, regardless of the reduction in the size of the state space, if the original function f must be evaluated, the efficiency gain in moving from the detailed to reduced model will be at most a factor of two or three.

For a reduced model, we seek a model evaluation cost that is related only to the complexity of the system input–output relationship, not to the size of the underlying system. Suppose that the detailed system description arises from discretization of a partial differential equation. As the discretization is refined, the dimension of the unreduced state-space model grows. However, for a fixed set of inputs and outputs, and a fixed accuracy requirement, past a certain point, increasingly fine discretization has little effect on the transfer function in a bounded frequency range. Therefore, we expect that if the system transfer function can be approximated by, say, a fourth-order rational function, a reduction algorithm can generally obtain a reduced model with number of states not greatly exceeding four, regardless of the size of the underlying detailed description. In a circuit context, the reduction algorithm should be able to discard states that arise from overly detailed representation of parasitic elements, and increasing the number of parasitics beyond this point should not affect the final model size. These properties must be retained in effective nonlinear reduction algorithms.

As a first step to such a viable general-purpose algorithm, we will first show how to extend the rational approximation paradigm to the nonlinear case, thereby providing a sound theoretical basis for selection of the subspace spanned by V. Our emphasis is not so much on proposing a specific algorithm as in showing that, for sufficiently well-restricted classes of nonlinear systems, algorithms with provable approximation properties and finite computation time exist, and that these algorithms generate models with reduced dimensionality and model complexity.

Since to obtain reduced descriptions of nonlinear systems, it seems necessary to reduce the complexity of the nonlinear function, we begin by examining how to approximate nonlinear systems input/output behavior.

B. Functional Series Expansions

Consider again the system of differential equations in (1). This state-space model defines a linear functional L: $[0, t]^p \rightarrow R^q$ that maps $u(t)$ on the past time interval $[0, t]$ to the output $y(t)$ at time t. Specifically, the output $y(t)$ of our LTI system may be expressed in the time domain using the convolution representation

$$y(t) = \int_0^t h(\sigma) u(t-\sigma)\, d\sigma \tag{29}$$

where $h(\sigma)$ is the kernel, or in the frequency domain, as $Y(s) = H(s)U(s)$ where $H(s)$ is the Laplace transform of $h(t)$. The nonlinear system of differential equations (27) also defines a functional K: $[0, t]^p \rightarrow R^q$ that maps $u(t)$ on the past time interval $[0, t]$ to the output $y(t)$ at time t, but K is *nonlinear* because $f(x)$ is not a linear function.

For a nonlinear system, quantities analogous to the convolution operator may be obtained by performing a functional series expansion of the output y that has the general form [37], [38]

$$y(t) = \sum_{n=1}^{\infty} y_n(t) \tag{30}$$

where $y_n(t)$ is the nth-order response

$$y_n(t) = \int_0^t \int_0^{\sigma_n} \cdots \int_0^{\sigma_2} h_n(\sigma_1, \ldots, \sigma_n) u(t-\sigma_1) \cdots u(t-\sigma_n)\, d\sigma_1 \cdots d\sigma_n. \tag{31}$$

The top system shown in Fig. 3 represents this sort of term-by-term decomposition of the response. The nth term in (31) represents an n-dimensional convolution of n products of the input u with an n-dimensional kernel $h_n(\sigma_1, \ldots, \sigma_n)$. In a circuit context, the nth order term gives rise to the nth order distortion products. Series expansions of the form (31) can be shown to exist for a broad class of nonlinear systems [39]. Volterra series, for example, may be considered a Taylor series expansion of the nonlinear functional K: $[0, t]^p \rightarrow R^q$ that maps $u(t)$ on the past time interval $[0, t]$ to the output $y(t)$ at time t.

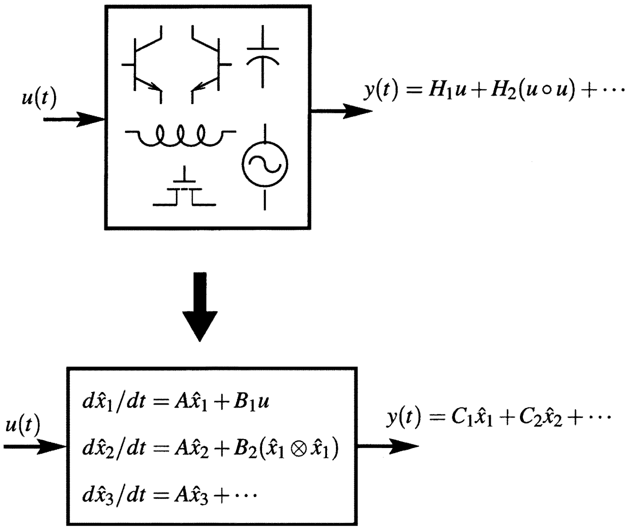
$u(t)$
$y(t) = H_1 u + H_2(u \circ u) + \cdots$
$u(t)$
$d\hat{x}_1/dt = A\hat{x}_1 + B_1 u$
$d\hat{x}_2/dt = A\hat{x}_2 + B_2(\hat{x}_1 \otimes \hat{x}_1)$
$d\hat{x}_3/dt = A\hat{x}_3 + \cdots$
$y(t) = C_1\hat{x}_1 + C_2\hat{x}_2 + \cdots$

Fig. 3. Pictorial representation of models and model reduction of weakly nonlinear systems. Top figure: The original circuit response is expanded in a series of powers of the input *waveform* $u(t)$ and the first few terms retained as an approximation. Bottom: The variational-based reduced model (Section VI-B) has one set of linear-state-space equations for each order in the truncated functional series expansion of the circuit response.

The Laplace transform $Y(s)$ of the output $y(t)$ is likewise given by a sum of terms

$$Y(s) = \sum_{n=0}^{\infty} Y_n(s) \tag{32}$$

where each Y_n is related to a multidimensional response function by

$$Y_n(s) = \frac{1}{(2\pi i)^{(n-1)}} \int_{\sigma_1 - i\infty}^{\sigma_1 + i\infty} \cdots \int_{\sigma_n - i\infty}^{\sigma_n + i\infty} \cdot Y_n(s - s_1 - \cdots s_{n-1}, s_1, \ldots, s_{n-1})\, ds_{n-1} \cdots ds_1 \tag{33}$$

with the responses related to the input by the frequency-domain kernels H_n

$$Y_n(s_1, \ldots, s_n) = H_n(s_1, \ldots, s_n) U(s_1) \cdots U(s_n) \tag{34}$$

and we have defined the multidimensional Laplace transform $H_n(s_1, \ldots, s_n) \equiv \mathcal{L}(h_n(t_1, \ldots, t_n))$ as

$$\mathcal{L}(h_n(t_1, \ldots, t_n)) \int_0^{\infty} \cdots \int_0^{\infty} h(t_1, \ldots, t_n) \cdot e^{-s_1 t_1 + \cdots - s_n t_n}\, dt_1 \cdots dt_n. \tag{35}$$

From (33) it is clear than the nth kernel H_n represents nth order distortion products that result in inputs at frequencies $s_1, \ldots, s_n$ generating an output response at the frequency $s = s_1 + \cdots s_n$. If we construct a reduced model whose nth kernel $\hat{H}_n(s_1, \ldots, s_n)$ matches the original system kernel $H_n(s_1, \ldots, s_n)$, then $\hat{y}$ will approximate y for (at least) terms in the nth order. If the nonlinearity of the system is sufficiently weak and/or the input sufficiently small, matching the first few kernels will allow the reduced order system to approximate the full nonlinear system.

V. Nonlinear Reduction Procedure Using Bilinear Forms

It is instructive to begin by examining a special class of nonlinear systems, bilinear state-space models, where the kernels in the functional series expansion are easy to calculate. After showing how to calculate the kernels for bilinear systems, we will show how to apply Krylov-subspace-based model reduction to obtain reduced models with accuracy guarantees on the reduced kernels. Next, we will show how general weakly nonlinear systems can be embedded in a bilinear system of higher dimensionality, which leads to a general weakly nonlinear model reduction procedure.

A. Bilinear State-Space Models

Bilinear state-space models are nonlinear systems whose state-space equations have a special form, bilinear in the state x and the inputs u. By "bilinear" it is meant linear in x and u individually, but not jointly. A p-input bilinear system has the form (where we will take $E = I$, for now, to simplify notation)

$$\frac{dx}{dt} = Ax + \sum_{j=1}^{p} N_j x u_j + Bu \qquad y = Cx \tag{36}$$

where $N_i \in R^{n\times n}$ and $u_j(t)$ denotes the jth component of $u(t)$. The terms $N_j x u_j$ are responsible for the nonlinear response of the system.

For the moment, in order to simplify notation, let us restrict the analysis to single-input, single-output systems, and further impose the restriction that $x(0) = 0$. A nonzero initial state will lead to an additional series in what follows. To analyze bilinear systems we construct a functional series representation of the state response $x(t)$. For simplicity, for the moment we restruct to the single-input, single-output (SISO) case, $p = 1$, and let $N = N_1$.

Theorem 4: The Laplace transform of the nth order kernel describing the response $x(t)$ of the SISO bilinear system to input $u(t)$ is given by

$$\begin{aligned} &H_n(s_1, \ldots, s_n) \\ &\quad = C((s_n + \cdots s_1)I - A)^{-1} N \cdots N (s_1 I - A)^{-1} B. \end{aligned} \tag{37}$$

Proof: See [37]. □

It will be convenient to express the frequency-domain kernels in what is called the regular form [37]

$$\begin{aligned} &H_n(s_1, \ldots, s_n) \\ &\qquad = H_n^{(reg)}(s_1 + s_2 + \cdots s_n, s_2 + \cdots s_n, \ldots, s_1) \end{aligned} \tag{38}$$

so we have

$$\begin{aligned} &H_n^{(reg)}(s_1, \ldots, s_n) \\ &\quad = C(s_n I - A)^{-1} N (s_{n-1} I - A)^{-1} N \cdots N (s_1 I - A)^{-1} B. \end{aligned} \tag{39}$$

B. Reduction of Bilinear Forms

Assuming that an appropriate q-dimensional subspace has been identified as the range of $\mathrm{colsp}(V)$ with $V \in R^{n\times q}$, reduction by orthogonal projection proceeds precisely as in the linear case, by first identifying the approximate state $\hat{x} = Vz$, and then defining the residual $R \equiv A\hat{x} + N\hat{x}u + Bu - (d\hat{x}/dt)$ and imposing the Galerkin condition $V^T R = 0$. We obtain a reduced model

$$\hat{E}\,\frac{dz}{dt} = \hat{A}z + \hat{N}zu + \hat{B}u \qquad \hat{y} = \hat{C}\hat{x} \tag{40}$$

where

$$\begin{aligned} \hat{A} &\equiv V^T A V \\ \hat{N} &\equiv V^T N V \\ \hat{B} &\equiv V^T B \\ \hat{E} &\equiv V^T V \\ \hat{C} &\equiv CV. \end{aligned} \tag{41}$$

The reduced bilinear system is also bilinear, but of dimension q, and if $q \ll n$, substantial computational savings can be achieved. The key point is that, in contrast to general nonlinear functions, the nonlinear part of the system, as represented by the matrix N, can be compressed by the projection operation. Now we address the choice of the matrix V.

To see how to extend the rational approximation properties of the Krylov-subspace methods to bilinear systems, consider the second-order regular kernel, given by

$$H_2^{(reg)}(s_1, s_2) = C(s_2 I - A)^{-1} N (s_1 I - A)^{-1} B. \tag{42}$$

Similarly, the kernel for the reduced system is

$$\hat{H}_2^{(reg)}(s_1, s_2) = \hat{C}\left(s_2 I - \hat{A}\right)^{-1} \hat{N} \left(s_1 I - \hat{A}\right)^{-1} \hat{B}. \tag{43}$$

The natural generalization of moment-matching is to require that $\hat{H}(s_1, s_2)$ and $H(s_1, s_2)$ agree to terms in $s_1^{q_1} s_2^{q_2}$, i.e., that $\hat{H}$ be a multidimensional partial Padé approximation of H. The two-dimensional Taylor series expansion of $H_2^{(reg)}(s_1, s_2)$ is

$$H_2^{(reg)}(s_1, s_2) = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} CA^{-l} N A^{-k} B s_1^k s_2^l. \tag{44}$$

We call terms of the form $CA^{-l_n} N \cdots NA^{-l_1} B$ a *multimoment*. We now show how Krylov-subspace algorithms lead to reduced models that match multimoments.

Suppose we construct a V_1 such that $A^{-k}B \in \mathrm{colsp}(V_1)$ for $k = 0 \cdots q_1$, i.e., $R(V_1) \supset K_{q_1}(A^{-1}, B)$. This is the condition for the *first*-order kernels, i.e., the transfer function of the linearized model, to match up to terms in s^{q_1} (see Theorems 1 and 2), or in other words, for $A^{-k}B = V_1 \hat{A}_1^{-k} \hat{B}_1$, $k \le q_1$ where $\hat{A}_1 = V_1^T A V_1$, $\hat{B}_1 = V_1^T B$. Now construct V_2 such that $K_{q_2}(A^{-1}, NV_1) \subset R(V_2)$, and construct $V \in R^{n\times q}$ such that $R(V) \supset R(V_1) \cup R(V_2)$. Then we must have

$$A^{-k}B = V\hat{A}^{-k}\hat{B}, \qquad k \le q_1 \tag{45}$$

where now $\hat{A} = V^T A V$, $\hat{B} = V^T B$, because $R(V_1) \subset R(V)$. In addition, if we take any $x \in R(V_1)$, so that $x = V_1 z_1 = Vz$ for some $z \in R^q$, we have

$$A^{-l}Nx = V\hat{A}^{-l}V^T N V z = V\hat{A}^{-l}\hat{N}z, \qquad l \le q_2. \tag{46}$$

The key point is that, by construction, $A^{-k}B \in R(V_1)$ for $k \le q_1$. Thus,

$$A^{-l}NA^{-k}B = V\hat{A}^{-l}\hat{N}\hat{A}^{-k}\hat{B} \qquad k \le q_1,\ l \le q_2 \tag{47}$$

so we may conclude that

$$H_2^{(reg)}(s_1, s_2) - \hat{H}_2^{(reg)}(s_1, s_2) = O\left(s_1^{q_1} s_2^{q_2}\right) \tag{48}$$

and that $\hat{A}$, $\hat{N}$, etc., is the desired model.

This result is easily generalized.

Theorem 5: Given

$$\frac{dx}{dt} = Ax + \sum_{j=1}^{p} N_j x u_j + Bu \tag{49}$$

$$y = Cx \tag{50}$$

where $A \in R^{n\times n}$; $x(t) \in R^n$, $B \in R^{n\times p}$, $N_i \in R^{n\times n}$, $u(t) \in R^p$, $C \in R^r$, suppose

$$R(V_1) \supset K_{q_1}\left(A^{-1}, B\right) \tag{51}$$

$$R(V_j) \supset \bigcup_{k=1}^{p} K_{q_j}\left(A^{-1}, N_k V_{j-1}\right), \qquad j > 1 \tag{52}$$

$$R(V) \supset \bigcup_{j=1}^{J} V_j. \tag{53}$$

Then if

$$\begin{aligned} \hat{A} &\equiv V^T A V \\ \hat{N}_j &\equiv V^T N_j V \\ \hat{B} &\equiv V^T B \\ \hat{E} &\equiv V^T V \\ \hat{C} &\equiv CV, \end{aligned} \tag{54}$$

and $H_n^{(reg)}$ are the order-n regular kernels of the models (A, N, B, C) and $(\hat{A}, \hat{N}, \hat{B}, \hat{C})$ respectively,

$$H_n^{(reg)}(s_1, \ldots, s_n) - \hat{H}_n^{(reg)}(s_1, \ldots, s_n) = O(s_1^{q_1} \cdots s_n^{q_n}) \tag{55}$$

for $n \leq J$.

Proof: Follows directly from Theorems 1 and 2 via the procedure above. □

Clearly, generating models that match many moments of high order kernels could be numerically expensive, particularly for systems with many inputs, and practical implementations will require careful selection of the minimum number of moments to be matched at each nonlinear order, deflation procedures, principal component selection of the spaces V_j [1], and so forth.

C. Bilinearization of General Nonlinear Forms

Again consider the system of (27), and for simplicity, assume $E = I$. Assume $f(x)$ may be expanded in a multidimensional polynomial series

$$f(x) = \sum_{k=1}^{\infty} \phi_k(x, \ldots, x) \tag{56}$$

where each ϕ_k is a k-multilinear form. For example $\phi_1(x)$ is linear in the argument x, and may be written as a matrix $\phi_1(x) = A_1 x$. $\phi_2(x, x)$ is bilinear in each argument, that is, $\phi_2(\alpha x + \beta y, z) = \alpha\phi_2(x, z) + \beta\phi_2(y, z)$ and similarly for the second argument. For now the analysis will proceed by constructing a bilinear representation of the system $(f(\cdot), B, C)$ [37]. Our goal is to guarantee accurate nonlinear representation by including information about the higher order nonlinear terms explicitly in the reduction process. Later in the paper we will consider the prospect [9], [11], [8] of forming reduced models by applying a projection formalism directly to $\hat{\phi}$.

A concrete representation of the ϕ_k may be obtained by using Kronecker forms. In particular, define

$$x^{(1)} \equiv x \qquad x^{(2)} \equiv x \otimes x \qquad x^{(3)} \equiv x \otimes x \otimes x, \text{ etc.} \tag{57}$$

The polynomial series expansion for $f(x)$ may be written as

$$f(x) = A_1 x^{(1)} + A_2 x^{(2)} + A_3 x^{(3)} + \cdots \tag{58}$$

so that

$$\frac{dx}{dt} = A_1 x^{(1)} + A_2 x^{(2)} + A_3 x^{(3)} + \cdots + Bu \tag{59}$$

where $A_k \in R^{n\times n^k}$. The coefficients A_k may be obtained via several means. They may be Taylor series coefficients, obtained either analytically, via numerical differentiation,[2] or through automatic differention of the computer program describing the function $f(x)$. Or, the coefficients A_k may represent more general polynomial fits, either to available device model behavior, or to data tables.

In any event, the bilinear model is obtained by defining a new state variable, $x^{\otimes}$

$$x^{\otimes} \equiv \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \\ \vdots \end{bmatrix}. \tag{60}$$

To do this, note that the time-derivative of a term $x^{(k)}$ is related to the time-derivative of the original state, $\dot{x}$, and a term of one order lower, $x^{(k-1)}$. For example,

$$\begin{aligned} \frac{d}{dt} x^{(2)} &= \frac{d}{dt}\left[x^{(1)} \otimes x^{(1)}\right] \\ &= \dot{x} \otimes x^{(1)} + x^{(1)} \otimes \dot{x} \\ &= \left[A_1 x^{(1)} + A_2 x^{(2)} + \cdots + Bu\right] \otimes x^{(1)} \\ &\quad + x^{(1)} \otimes \left[A_1 x^{(1)} + A_2 x^{(2)} + \cdots + Bu\right] \\ &= \left[(A_1 \otimes I)\left(x^{(1)} \otimes x^{(1)}\right) + \cdots\right] + (B \otimes I) x^{(1)} u \\ &\quad + \left[(I \otimes A_1)\left(x^{(1)} \otimes x^{(1)}\right) + \cdots\right] + (I \otimes B) x^{(1)} u \\ &= \left[(A_1 \otimes I + I \otimes A_1) x^{(2)} + \cdots\right] + (B \otimes I) x^{(1)} u \\ &= A_{21} x^{(2)} + \cdots + B_{20} x^{(1)} \end{aligned} \tag{61}$$

where $A_{21} = (A_1 \otimes I + I \otimes A_1)$ and $B_{20} = (B \otimes I)$. Continuing this process, we will obtain a bilinear realization $(A^{\otimes}, N^{\otimes}, B^{\otimes}, C^{\otimes})$

$$\frac{d}{dt} x^{\otimes} = A^{\otimes} x^{\otimes} + N^{\otimes} x^{\otimes} u + B^{\otimes} u \tag{62}$$

$$y = C^{\otimes} x^{\otimes} \tag{63}$$

[2] Accurate numerical differentiation will be limited to relatively low orders, about two orders more than the highest order expression available analytically. For example, if the Jacobian is available analytically, a fairly accurate approximation of the third-order coefficients can reasonably be expected.

for the original nonlinear system, where

$$A^{\otimes} = \begin{bmatrix} A_{11} & A_{12} & & \cdots & \\ & A_{21} & A_{22} & & \\ & & A_{31} & A_{32} & \\ & & & \ddots & \end{bmatrix} \tag{64}$$

$$N^{\otimes} = \begin{bmatrix} 0 & 0 & & \cdots \\ B_{20} & 0 & 0 & \\ & B_{30} & 0 & \\ & & & \ddots \end{bmatrix} \tag{65}$$

$$B^{\otimes} = \begin{bmatrix} B \\ 0 \\ \vdots \end{bmatrix}$$

$$C^{\otimes} = [C \quad 0 \quad \cdots]. \tag{66}$$

$A_{1,k} = A_k$, and $A_{j,k} = A_k \otimes I \otimes \cdots \otimes I + I \otimes A_k \otimes \cdots \otimes I + \cdots + I \otimes I \otimes \cdots \otimes A_k$ where there are $j-1$ Kronecker products in each term, and j terms. Similarly B_{j0} is given by $B_{j0} = B \otimes I \otimes \cdots \otimes I + I \otimes B \otimes \cdots \otimes I + I \otimes I \otimes \cdots \otimes B$.

D. Multimoment-Matching

The multimoments of the nonlinear system can be expressed in terms of the bilinear system quantities as

$$m_k^{(1)} \equiv C^{\otimes}(A^{\otimes})^{-k}B,\ m_{lk}^{(2)} = C(A^{\otimes})^{-l}N^{\otimes}(A^{\otimes})^{-k}B,\ \ldots. \tag{67}$$

Even though the matrix $A^{\otimes}$ has infinite dimension, due to its special structure, multimoment calculation is feasible. Because $A^{\otimes}$ is block-upper-triangular, and the matrix $N^{\otimes}$ is nonzero only on the first lower block subdiagonal, multimoments $m_{k_1 \cdots k_r}^{(r)}$ depend only on the submatrices in the first $k \times k$ size blocks. In particular, note that the kth kernel H_k depends only on the first k terms in the series expansion, so a bilinear approximate system obtained by dropping terms of order higher than k in the expansion of $f(x)$ will agree with the original system in the first k kernels H_k. It is important to keep the distinction between the order of the kernels and the order of the system in mind, however. For example, we have seen that a general bilinear system will possess kernels H_k of all orders and thus even if $f(x)$ can be expressed exactly as, say, a second-order multinomial, powers of the input u to every order may be significant in the output.

Computing the multimoments of the kth kernel will involve 1) inversion of the diagonal blocks A_{k1}, and 2) products with the off-diagonal matrices A_{kj} for $j > 1$. Since the second operation essentially involves products with the tensor series terms, terms that are sparse in many applications of interest, we expect the bulk of the numerical effort to be in the first step.

It is not so hard to see that the primary difficulty with the bilinear forms is that the dimension of the state space grows exponentially with the order of nonlinear approximation. Solving linear systems of equations whose size grows exponentially is generally considered difficult. However, it is not actually necessary for the model reduction procedure to compute the exact matrix solutions, only the subspaces needed for projection. The linear equations involved in the proposed paradigm are highly structured, and efficient algorithms can be developed to solve them. For example, consider inverting the second diagonal block in $A^{\otimes}$. This block has the form $A \otimes I + I \otimes A$. Solving this equation is the same as solving the Lyapunov equation $AX + XA^T = C$ for some C [40]. It has recently been shown how to construct low-rank solutions of Lyapunov equations [41] for low-rank right-hand sides, which are the sort that occur in the model-reduction procedure. The virtue of the low-rank solutions is that if they can be constructed for all the diagonal blocks, then the projection vectors can be represented with a number of degrees of freedom that is $O(QN)$, where Q is the total number of multimoments matched and N the dimension of the original nonlinear system. It is still true that Q will grow exponentially with the degree of nonlinear approximation order, but this is unavoidable for functional series representations.

VI. Nonlinear Reduction Procedure Using Polynomial Forms

The bilinear system realizations have the advantage that the Volterra kernels are particularly easy to compute once the system is put in this form. By using the bilinear form, we were able to see that the Krylov projection methods used for reduction of LTI systems can also be applied to nonlinear systems, and that it is possible to make rigorous statements about the approximation properties. However, computationally, the procedures that rely on bilinear realization are computationally complicated. In this section, simpler approaches will be discussed.

A. Polynomial Approximations and Variational Analysis

Consider again the system of (27). Assume $f(x)$ may be expanded in a series of multidimensional polynomials (such as, but not necessarily, a multidimensional Taylor series) as in (56)–(59), so that

$$\frac{dx}{dt} = A_1 x^{(1)} + A_2 x^{(2)} + A_3 x^{(3)} + \cdots + Bu \tag{68}$$

where $A_k \in R^{n \times n^k}$. The matrices A_k, representing the k-dimensional tensors needed to expand $f(x)$ in multidimensional series, are usually extremely sparse and in that case a product of one of the A_k and a vector $x^{(k)}$ can be computed in $O(N)$ operations if N is the dimension of the vector $x^{(1)}$.

The key to reduction of the nonlinear system in polynomial series form is to note that with this realization of the multilinear forms, the reduced multilinear forms $\hat{\phi}(z, \ldots, z)$ can be expressed as the matrices

$$\hat{A}_k = V^T A_k (V \otimes V \otimes \cdots V) \tag{69}$$

because of the Kronecker product identity [40]

$$\begin{aligned} &A_k(Vz \otimes Vz \otimes \cdots Vz) \\ &\qquad = A_k(V \otimes V \otimes \cdots V)(z \otimes z \otimes \cdots z). \end{aligned} \tag{70}$$

Mechanically speaking, given any projection matrix V, (69) tells us how to perform the reduction. To see how V may be computed with accuracy comparable to that achievable for Krylov methods acting on linear systems, we adopt a variational procedure commonly used for computing Volterra kernels [37]. Suppose we introduce α as a variational parameter and calculate the response of the system $\dot{x} = f(x) + B(\alpha u)$ as a function of α. We may expand the response in a power series in α

$$x(t) = \alpha x_1(t) + \alpha^2 x_2(t) + \cdots. \tag{71}$$

It is clear that

$$x^{(2)}(t) = \alpha^2 x_1^{(2)} + \alpha^3[x_1 \otimes x_2 + x_2 \otimes x_1] + \cdots \tag{72}$$

$$x^{(3)}(t) = \alpha^3 x_1^{(3)} + \cdots \tag{73}$$

and so

$$\begin{aligned}\alpha\dot{x}_1(t) + \alpha^2\dot{x}_2(t) + \cdots \\ = \alpha A_1 x_1 + \alpha^2[A_1 x_2 + A_2(x_1 \otimes x_1)] + \cdots.\end{aligned} \tag{74}$$

By comparing terms in the variational parameter α, we may obtain a set of differential equations, each N-dimensional, that describe the time-evolution of each of the x_k

$$\dot{x}_1 = A_1 x_1 + Bu \tag{75}$$

$$\dot{x}_2 = A_1 x_2 + A_2(x_1 \otimes x_1) \tag{76}$$

$$\begin{aligned}\dot{x}_3 &= A_1 x_3 + A_2[x_1 \otimes x_2 + x_2 \otimes x_1] \\ &\quad + A_3[x_1 \otimes x_1 \otimes x_1]\end{aligned} \tag{77}$$

and so on. Each n-dimensional set describes the time-evolution of an x_k that represents the kth order nonlinear response terms. The above variational analysis, though not widely familiar, is used in the SPICE distortion (.DISTO) analysis.

B. *Reduction of Polynomial Forms*

The system describing the first-order response is a standard linear state-space system. To obtain a model of this system, we would need to compute a Krylov subspace with starting vectors given by the column space of B. One approach [11], [8] to computing the matrix V that defines the basis for projection is to note that the system describing the second-order response is *also* a linear state-space system with the same system matrix A_1; it only has different inputs. We may write more suggestively

$$\dot{x}_2 = A_1 x_2 + A_2 u_2 \tag{78}$$

with the identification $u_2 = x_1 \otimes x_1$ where x_1 is the state vector from the first-order system. The input to the second-order system is A_2 times the "squared" first-order response. To obtain a reduced model that will match the frequency response of the second-order component of the response, we must span the Krylov space of the inputs to the second-order system, described by A_2.

Now suppose that the first-order system was reduced by orthogonal projection, with projection matrix V_1. The state vector is approximated as $\hat{x}_1 = V_1 z_1$, where z_1 is the state of the reduced first-order system. The second-order system becomes

$$\dot{x}_2 = A_1 x_2 + A_2(\hat{x}_1 \otimes \hat{x}_1) \tag{79}$$

or, by using the identity $(V_1 z_1) \otimes (V_1 z_1) = (V_1 \otimes V_1)(z_1 \otimes z_1)$, and with the identification $\hat{u}_2 = z_1 \otimes z_1$

$$\dot{x}_2 = A_1 x_2 + A_2(V_1 \otimes V_1)\hat{u}_2 \tag{80}$$

or more transparently as

$$\dot{x}_2 = A_1 x_2 + B_2 \hat{u}_2 \tag{81}$$

where $B_2 = A_2(V_1 \otimes V_1)$. The key point is that if the first-order system was adequately described by a reduced model obtained from a projector matrix V_1, then the inputs to the second-order system must lie in the column space of $B_2 = A_2(V_1 \otimes V_1)$. Thus to obtain a space V_2 that will give a model for the second-order response, we first compute V_1 whose columns are a basis for $K_q((s_0E - A)^{-1}E, (s_0E - A)^{-1}B)$. Then we compute V_2 whose columns span $K_q((s_0E - A)^{-1}E, (s_0E - A)^{-1}B_2)$. The procedure for higher orders of the nonlinear expansion follows analogously. Note that the bilinear reduction scheme gave rise to a similar nested series of projection spaces to compute.

This observation also gives us some insight into when reduced models based on strictly linear information are useful. If a space V, perhaps generated from the Krylov $K(A_1^{-1}, B)$, has small projection onto the range of one of the multilinear forms $A_k(V_j \otimes V_j \cdots \otimes V_j)$, then the resulting reduced model will not likely be a good approximation to the kth order response. The only way to guarantee an accurate nonlinear representation is to include information about the higher order nonlinear terms explicitly in the reduction process as suggested above. Conversely, if the span of $A_k(V_j \otimes V_j \cdots \otimes V_j)$ is already contained in the span of V, then it is advantageous to "deflate" the spaces V_k obtained at higher order by performing a singular value decomposition of $[V_1 V_2 \cdots]$ and using the resulting single V to perform the projection. Using this approach, aggressive deflation is essential to obtaining a reasonable size model when matching the frequency domain response of the higher order nonlinearities, since, strictly speaking, the number of required terms grows exponentially with order of the polynomial expansion (just as the number of degrees of freedom of Volterra kernels does).

C. *Time-Varying Weakly Nonlinear Systems*

Finally, it is worth noting that the above analysis applies for a general operating point. In particular, by adopting the procedures of Section III, a time-varying weakly nonlinear model reduction procedure can be derived. We omit the details as they are straightforward. Essentially the proposed approach matches "moments" of time-varying Volterra kernels. The models can thus be used to predict intermodulation distortion and other nonlinear effects in RF circuits operating under periodic or quasiperiodic bias, as will now be demonstrated.

VII. Examples

A. *LTV Systems*

To test the time-varying model reduction procedure, the proposed algorithms were implemented in a time-domain RF circuit simulator. The large-signal periodic steady state is calculated using a shooting method [23]. The LTV system was discretized using variable-timestep second-order backward-difference formulas.

The first example considered was the switched-capacitor filter previously discussed, running at a clock frequency of 25 kHz. This example generated 58 equations in the circuit simulator, and 453 timesteps were needed to describe the steady-state waveform. For the model reduction procedure, the input function $B(t)$ [see (14)] was a constant column vector, corresponding to the continuous small-signal input present at

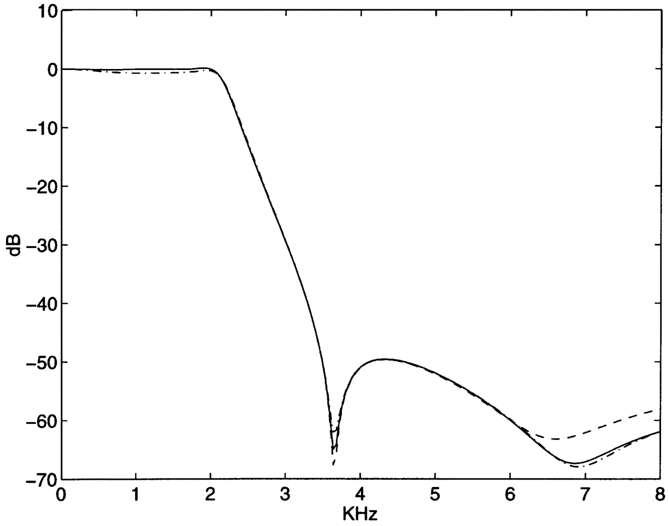

Fig. 4. Time-sampled switched-capacitor filter transfer function.

the filter input. To specify an output function, we took a sample function that was constant over a 200 ns-period 1 μs before the clock edge at the start of the cycle. Essentially, the final model is a real LTI system that represents a transfer function between the continuous analog input and the sampled digital output. The amplitude of the transfer function, as a function of input frequency, of the reduced model is shown in Fig. 4.

Two nine-state models are shown in Fig. 4. The model shown with a dashed line was generated by matching nine real moments at the origin. The dash-dot line, virtually identical to the actual transfer function, was generated from matching three real moments at the origin, and one moment at 200, 400, and 800 kHz on the imaginary axis. As these expansion points were off the real axis, each complex moment in the Krylov space generates two states in the final real model, corresponding to the Krylov vector and its complex conjugate. The multipoint approximation is seen to be a better match.

Another application of the time-varying model reduction machinery is to obtain reduced models for cyclostationary noise [42], [43]. This can be done by computing a reduced model, based on the adjoint system, from the point where noise is to be analyzed to all the noise sources. Once this low-rank reduced model is available, equivalent noise sources of the same rank can be computed using the standard singular value decomposition or QR procedure. Note that because they are based on the adjoint procedure, unlike the approach described in [44], the models suggested here preserve information about noise sources. In addition, just as with the large-signal models, the noise models can be constructed in time or frequency domain. Fig. 7 shows the time-variation of the noise power given by a macromodel of the cyclostationary noise from the same switched capacitor circuit.

The second example is the complex image-rejection receiver studied in [45]. This receiver is a complicated circuit with several functional component blocks (a low-noise amplifier, a splitting network, two double-balanced mixers, and two broad-band Hilbert transform output filters). The entire circuit has 167 bipolar transistors and generates 986 equations in the circuit simulator. 200 timesteps were needed for the time-domain analysis, so that the matrix K has a rank of almost 200 000.

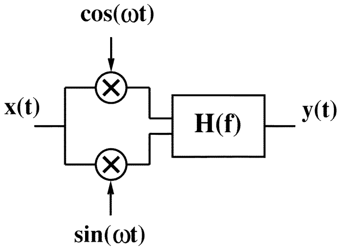

Fig. 5. Receiver macromodel. The frequency of the sin and cosine elements is the mixer LO frequency, 780 MHz.

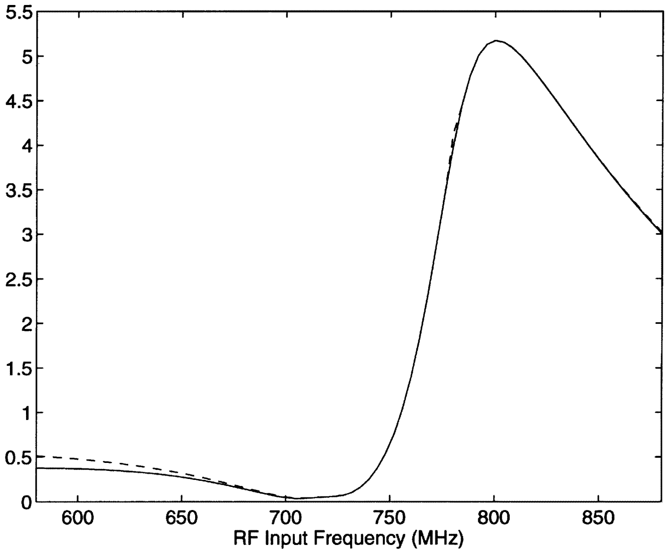

Fig. 6. Transfer function of the mixer (solid) and the 15-state mixer model (dash), from RF input to mixer output.

A fifteenth order real-valued time-varying model was generated to represent the receiver conversion path from RF to output. Since a model from multiple sidebands to mixer output was desired, the adjoint matrix was used to generate the model reduction. In this case the time-varying elements appear before an LTI filter in the final model, as is shown schematically in Fig. 5. The mixing elements shift the input from the RF frequency by 780 MHz, the mixer LO frequency. Following these elements is a multiinput LTI filter whose response is shown in Fig. 6. The lower sideband rejection characteristic of this mixer is evident in the model. If the model were used in a suppressed-carrier DSP simulator, the mixing elements would simply be omitted.

Table I shows the statistics for the computational costs for the reduced model extraction and evaluation. 200 frequency points were considered in the filter example and fifty in the mixer example. In both examples, the reduced model took less time to extract than the single frequency sweep, and the evaluation was vastly more efficient (in fact the overhead in the code was sufficient that it was difficult to determine exactly how much time was consumed in the actual model evaluation). Note the efficiency in particular of the reduction of the switched-capacitor example. The time-varying model has a rank of 26 274, yet the reduced model was generated in only 7 CPU seconds. Table I also demonstrates the efficiency

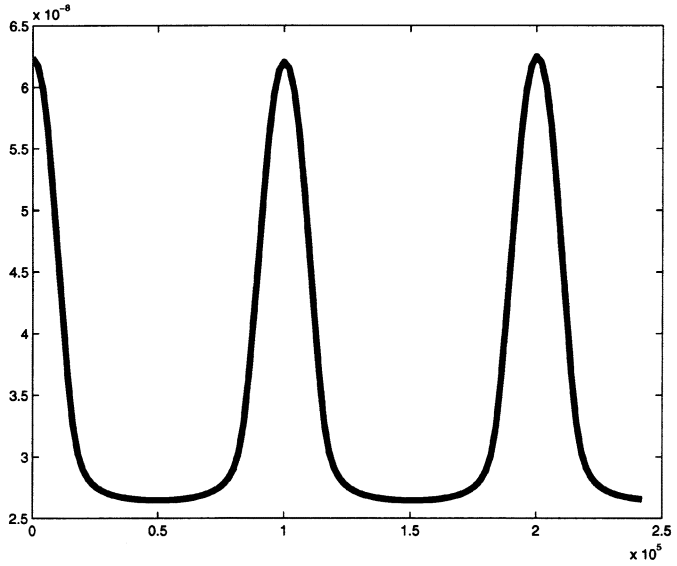

Fig. 7. Time-variation of sampled noise in the switched-capacitor circuit reduced model.

TABLE I
COMPARISON OF TIME-VARYING MODEL REDUCTION PROCEDURES AND POINTWISE FREQUENCY SWEEPS. THE FREQUENCY SWEEPS WERE ACCELERATED BY THE RECYCLED GMRES ALGORITHM. MVP REFERS TO EQUIVALENT REAL–REAL MATRIX VECTOR SOLVES WITH THE MATRIX L. "REDUCE" IS THE MODEL REDUCTION TIME IN SECONDS OR MINUTES, AND "SOLVE" THE CPU TIME REQUIRED TO OBTAIN THE FREQUENCY RESPONSE

Circuit		MOR/Recycle	MOR/Std	Pointwise
SCF	MVP	45	99	410
	Reduce	7s	12s	–
	Solve	1s	1s	1.5m
Receiver	MVP	166	1980	280
	Reduce	4m	48m	–
	Solve	2s	2s	8.5m

of the two-stage ("recycled") procedure. In the case of the filter, for an order-nine real model, only 18 applications of L^{-1} were required in the recycled GMRES procedure that obtained the moments. The remaining matrix vector products (or backsolves) were needed to perform the projection or for the initial preconditioning steps. Even in the mixer example, where GMRES had much more difficulty converging, 124 backsolves were needed for the reduction, about eight per model order, which is still good.

B. Nonlinear Systems

The first nonlinear system example is an artificial one, constructed to demonstrate how in some cases information from the linear part of the model alone [10], [9] is insufficient in constructing an accurate projection subspace, but that including nonlinear information in the manner suggested in this paper can improve accuracy.

To accomplish this, a random second-order polynomial system, of order sixty, was generated. The poles of the linear part of the system were constrained to lie on an oval separated from the origin, as show in Fig. 8. This pole distribution was chosen because linear reduction methods that are based on eigenanalysis alone perform poorly for this type of pole distribution, because many of the poles contribute more or less equally to the transfer functions at low frequencies. Moment-matching methods, on the other hand, can still capture the transfer function accurately. In other words, Krylov subspaces for different B, C vectors will be very different, so the reduced models are sensitive to the particular subspace chosen. Because of this, we expect the model accuracy to depend critically on deliberately matching the higher order multimoments. This is indeed observed in Fig. 9, which shows the second-order Volterra kernel $H_2(s, s)$, evaluated using the variational procedure, along the diagonal in the two-dimensional frequency plane. A reduced model was computed using the polynomial-based procedure, to match two moments of the first (linear) and second (quadratic) order subsystems. This gives a final model of order ten. In addition, another reduced model, of the same size but matching moments of only the linear subsystem, was also generated. As expected, the model incorporating nonlinear information is a much better match at low frequency.

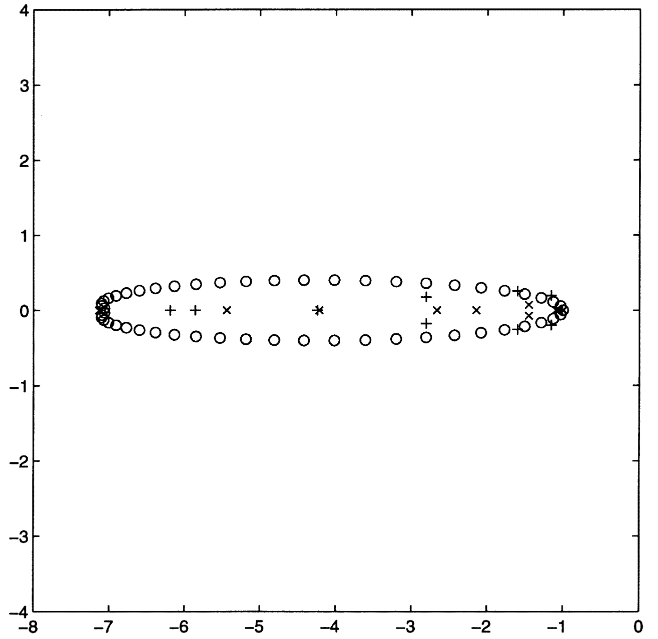

Fig. 8. Poles of linearized polynomial system and reduced models. (o): Poles of unreduced system. (x):pPoles of linear-based reduced model. (+): poles of nonlinear-based reduced model.

The next step is to compare the bilinear- and polynomial-based procedures on more practical examples. One of the difficulties in testing nonlinear reduction schemes is demonstrating that the algorithm genuinely reduces nonlinear complexity, not just a linear piece of the problem. A network with many linear elements can be reduced much more easily than if every element is nonlinear. Ultimately, we are interested in studying problems such as RF circuits (mixers, power amplifiers, etc.) under time-varying bias conditions [11], but to illustrate the concepts in this paper, we adapt an example familiar from the linear model reduction literature, the *RC* line. We introduce strong global nonlinearity by connecting a diode in parallel with each resistor. This example is motivated by a problem in [9]. We drive the circuit with a sine wave at one end and observed the transmitted signal. By measuring distortion,

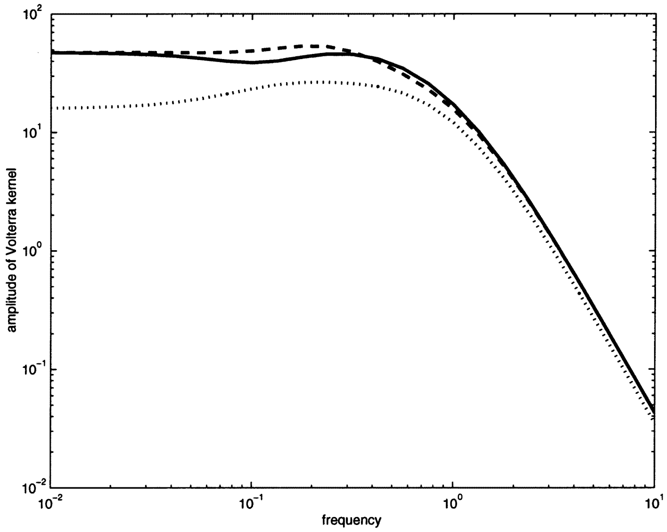

Fig. 9. Amplitude of second-order volterra kernel of polynomial system. Solid line: original system. Dashed line: polynomial-based model matching two order-one moments and two order-two moments. Dotted line: model based on purely linear information.

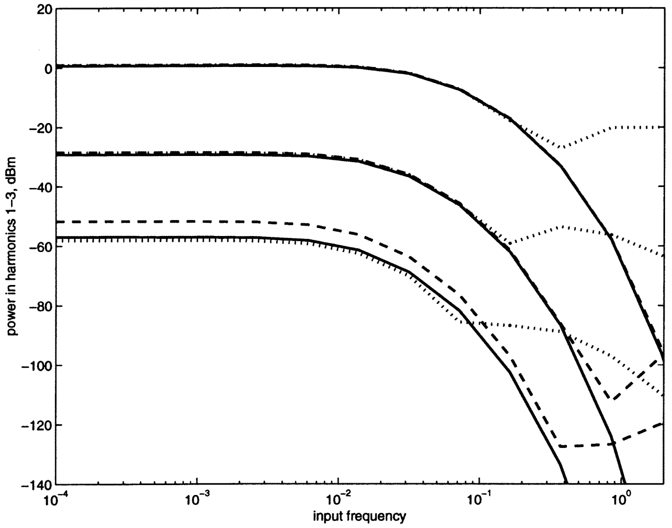

Fig. 10. First, second, and third harmonics, in dBm, of nonlinear *RC* line. The full system response is shown as the solid curves, and an order 11 bilinear reduced model as the dotted lines, and an order 11 polynomial-based reduced model as the dashed curves.

an intrinsically nonlinear phenomenon, in the frequency-domain, we can separate the effects of the linear and nonlinear model contributions more cleanly than by observing time-domain simulations. Note that in our problem, nonlinearity and capacitance is distributed throughout the network and so the distortion will vary with frequency in a way that may be difficult to approximate by approaches that do not explicitly consider the nonlinearity in the model-reduction procedure. Harmonic balance was used to calculate the response of the original and reduced system. The original line had thirty *RC* sections, and was driven sufficiently hard to produce second-order distortion terms 20 dB down from the primary signal, and −40 dB third-order terms. This is enough distortion to indicate that the approach presented here is capable of treating nonlinear effects that occur in practical examples.

Fig. 10 shows computed results for two reduced models as well as the original system. The first reduced model, of order 11, was computed based on the bilinear form, designed to match four linear moments and two second-order multimoments. The second reduced model, also of order 11 and similar construction, was derived via the polynomial/variational analysis. As expected, at low frequencies, the first- and second-order harmonics are reasonably well approximated. In addition, reasonable approximation of the third order term is also observed. This is possible because at lower signal levels, the second-order term in the polynomial series expansion is the dominant contributor to many higher orders of the system output response. Finally, note that the model generated directly from the polynomial system is generally more accurate. This is not surprising, because the bilinear representation of a given polynomial system is always of larger order. For a fixed model order, we thus expect the polynomial form to give more accurate models. Conversely, for a fixed degree of approximation to the original nonlinearities, we expect the bilinear form to be somewhat more expensive. For example, for a second-order polynomial expansion, the variational representation has $2q$ states, where q is the rank of the matrix A_1, but the bilinear representation has $q + q^2$ states.

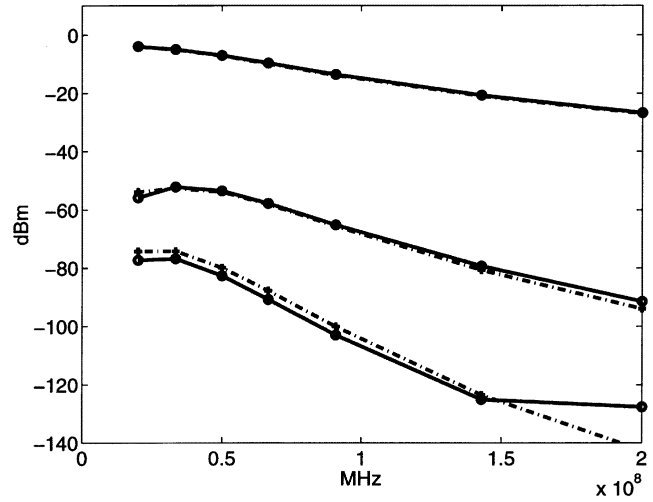

Fig. 11. First- second- and third-order response of RF mixer as a function of signal input frequency. The full system response is shown as the solid (o) curves and the reduced model as the dashed (+) curves.

Note also, that the bilinear procedure, while conceptually more transparent, is more computationally demanding. Without using special techniques to invert the diagonal blocks in (64), for the small system here, the bilinear reduction procedure was about sixty times slower than the polynomial method, due the large increase in the size of the original state space. However, by exploiting structure (64) in specifically treating the inversion of the diagonal blocks as a Lyapunov equation solution, the penalty dropped to about a factor of three.

Next, we demonstrate reduction of weakly nonlinear, time-varying systems by computing a nonlinear reduced model of an RF mixer. We compute the primary upconversion response as well as the first two terms generated by distortion (second- and third-order responses in the formulation above). Fig. 11

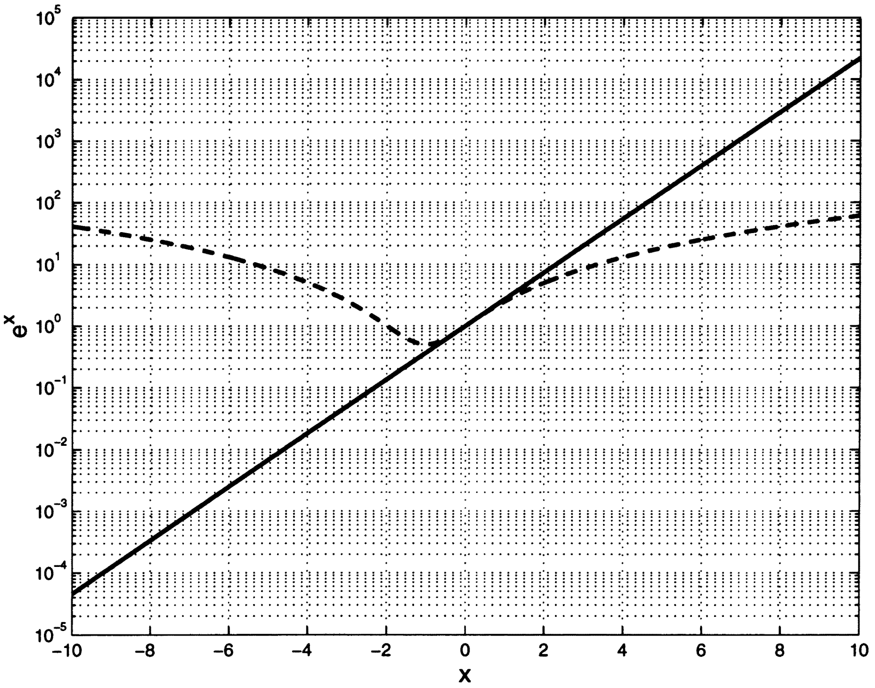

Fig. 12. Solid line: e^x. Dashed line: Second-order Taylor approximation of e^x. Note that the Taylor series crosses above unity at $x = -2$.

shows comparisons of the original system and a dimension-38 reduced model needed to match the frequency response of the first- and second-order nonlinearities, respectively, up to fourth and second order, respectively, in the frequency domain series expansion. Harmonic balance was used to calculate the response of the original and reduced system. The generation of the reduced model required about four times as much computation time as generation of the initial operating point and less computation time than an LTV (i.e., first-order small-signal) analysis performed on the detailed circuit at the same set of frequency points. Good agreement is observed for the first- and second-order terms and the third-order response is, interestingly, fairly well-captured, also. Even for this relatively small circuit, compared to detailed simulation approaches such as multifrequency harmonic balance (or envelope-based simulations) needed to capture the distortion induced by the signal tones in the presence of the periodic local oscillator, the model reduction procedure results in more than an order of magnitude reduction in computational complexity.

Finally, we discuss one of the main drawbacks of methods based on local polynomial approximations. It is generally appreciated that functional series approximations are only effective for weak nonlinearities, or, when the system is driven by small inputs. Polynomial-based systems have the additional drawback that they may create pathological simulation behavior when operated outside the range of accuracy of the polynomial approximation. To see why this is the case, consider examining a second-order polynomial approximation to an exponential nonlinearity, as occurs in the diode models for the nonlinear *RC* line previously discussed. Fig. 12 shows the exponential function and a quadratic Taylor approximation. Near zero, it is a good approximation, and becomes progressively worse far away. Of particular interest is the behavior of large negative arguments.

For $x < -2$, the quadratic approximation is positive and larger than unity. This means that when using the quadratic approximation in the expression $I = I_s(e^{v/v_t} - 1)$, for $v < -2v_t$, the approximate model will predict a (possibly very large) *forward* current under reverse bias. A nonlinear device with a negative current–voltage product corresponds to an operating condition where the approximated diode is generating energy; it is not *passive*, as all physical devices, including nonlinear ones, must be. This is a general property of polynomials. They are not well behaved for large arguments, though the particular pathology varies according to the approximation chosen. Neither, then, are the reduced models likely to be well behaved, e.g., passive, except by chance.

Time-domain simulation of the nonlinear *RC* line illustrates the behavior more concretely. Fig. 13 shows the original *RC* line and a quadratic polynomial-based reduced model with a medium-strength input sinusoid. The input sinusoid is strong enough to illustrate weak nonlinearity, as evidenced by the slight degree of asymmetry in the response. Fig. 14 shows the same line with larger input signal. The deviation of the reduced model from the original is clearly visible. More interestingly, if the input amplitude is increased by a small amount, the time-domain response of the polynomial system diverges at about $t = 0.8$. The time-domain simulation is unable to proceed beyond this point. Neither the polynomial approximate system, nor the reduced model derived from it, can be used in time-domain simulation for large inputs.

VIII. Conclusion

In this paper we have demonstrated extraction of simple and compact macromodels from nonlinear, time-varying transistor-level circuit descriptions. As the switched-capacitor and receiver examples demonstrate, the methods presented are

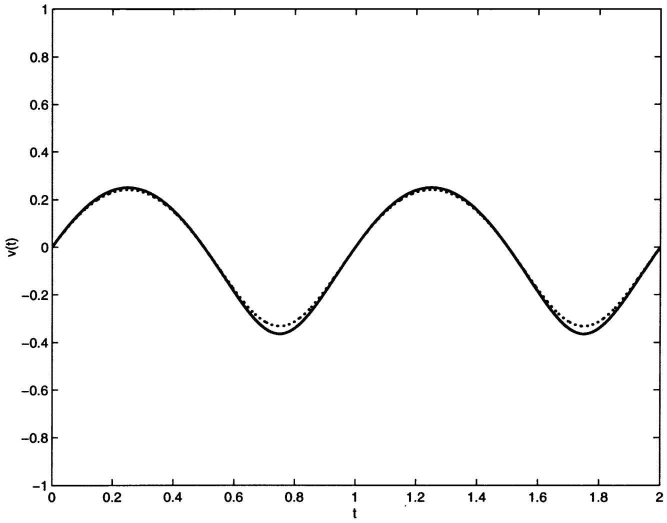

Fig. 13. Weakly driven line, time-domain response. Dotted line: original system. Solid line: reduced model.

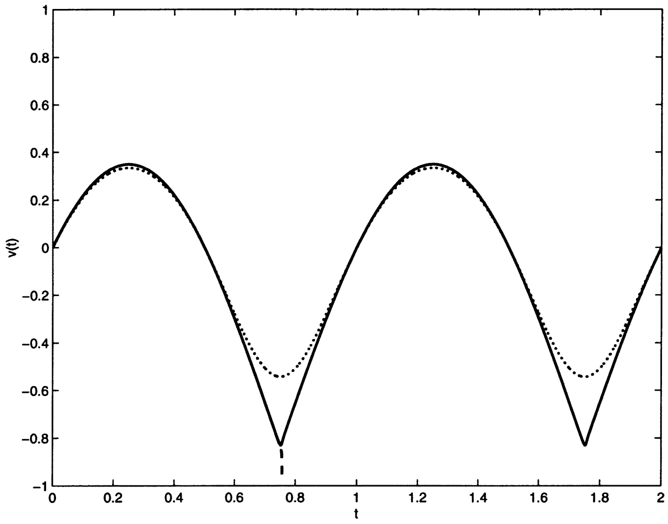

Fig. 14. Strongly driven line, time-domain response. Dotted line: original system. Solid line: reduced model. Dashed line: reduced model at 1% large input amplitude.

capable of extracting compact models from large circuits with complicated underlying dynamics. We investigated in some detail the potential, and drawbacks, of two approaches for reduction of weakly nonlinear systems. The bilinear-based approach is attractive for analytic work, but the polynomial-based approach seems to be computationally more desirable.

After this investigation, it seems that most of the advantages of the Krylov-subspace-based schemes, in being able to approximate system dynamics on a large scale, will be transportable to the time-varying and the nonlinear setting. The major challenges appear to be reduced representation of nonlinear functions. The first challenge is efficient representation of strong nonlinearities. The size of the terms in the polynomial expansion grow exponentially with expansion order, in practical terms, limiting their usefulness to order two or three. The second challenge is to devise a method, with global approximation properties that are well behaved in the sense of not representing nonphysical objects. Much work in reduction of linear systems has been concerned with preservation of system passivity. In the nonlinear context, an approximation scheme that would preserve the passivity of networks containing nonlinear devices would be very useful.

Appendix
Considerations for Descriptor Systems

While developing most of the nonlinear material, the assumption $E = I$ was made. Procedures become more complicated when it is not desirable or possible to put a system into this form (for example, E is not invertible, or nonlinearities appear under the time-derivative). For example, in the bilinearization procedure, by following precisely the procedure in Section V-C, we can obtain the bilinear representation

$$\begin{aligned}\frac{d}{dt}\begin{bmatrix} E_{11} & & & \\ & E_{22} & & \\ & & E_{33} & \\ & & & \ddots \end{bmatrix} x^{\otimes} &= \begin{bmatrix} A_{11} & A_{12} & & \cdots \\ & A_{21} & A_{22} & \\ & & A_{31} & A_{32} \\ & & & \ddots \end{bmatrix} x^{\otimes} \\ &+ \begin{bmatrix} 0 & 0 & & \cdots \\ B_{20} & 0 & 0 & \\ & B_{30} & 0 & \\ & & & \ddots \end{bmatrix} x^{\otimes} u + \begin{bmatrix} B \\ 0 \\ \vdots \end{bmatrix} u \end{aligned} \tag{82}$$

$$y = [C \quad 0 \quad \cdots] x^{\otimes} \tag{83}$$

where

$$E_{kk} = E \otimes E \otimes \cdots \otimes E \tag{84}$$

($k-1$ Kronecker products) and now

$$\begin{aligned} A_{j,k} = &A_k \otimes E \otimes \cdots \otimes E + E \otimes A_k \otimes \cdots \otimes E \\ &+ \cdots + E \otimes E \otimes \cdots \otimes A_k \end{aligned} \tag{85}$$

and

$$\begin{aligned} B_{j0} = &B \otimes E \otimes \cdots \otimes E + E \otimes B \otimes \cdots \otimes E \\ &+ E \otimes E \otimes \cdots \otimes B. \end{aligned} \tag{86}$$

If a nonlinearity appears under the time derivative, as in

$$\frac{d}{dt} g(x) = f(x) + Bu \tag{87}$$

$$y = Cx \tag{88}$$

it, too, must be expanded in series

$$g(x) = G_1 x^{(1)} + G_2 x^{(2)} + G_3 x^{(3)} + \cdots \tag{89}$$

(again we assume expansion around an equilibrium point) and the additional terms under the time derivative appear in all following expressions, such as the state equation,

$$\begin{aligned} \frac{d}{dt}\left[G_1 x^{(1)} + G_2 x^{(2)} + G_3 x^{(3)} + \cdots\right] \\ = A_1 x^{(1)} + A_2 x^{(2)} + A_3 x^{(3)} + \cdots + Bu \end{aligned} \tag{90}$$

and the equation for the second-order subsystem of the variational equations

$$\frac{d}{dt}\left[x_2 + G_2(x_1 \otimes x_1)\right] = A_1 x_2 + A_2(x_1 \otimes x_1). \qquad (91)$$

References

[1] C. de Villemagne and R. E. Skelton, "Model reduction using a projection formulation," *Int. J. Control*, vol. 46, pp. 2141–2169, 1987.

[2] E. Grimme, "Krylov projection methods for model reduction," Ph.D. dissertation, Coordinated-Sci. Lab., Univ. Illinois, Urbana-Champaign, IL, 1997.

[3] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 639–649, May 1995.

[4] L. M. Silveira, M. Kamon, and J. K. White, "Efficient reduced-order modeling of frequency-dependent coupling inductances associated with 3-d interconnect structures," in *Proc. 32nd Design Automation Conf.*, San Francisco, CA, June 1995, pp. 376–380.

[5] A. Odabasioglu, M. Celik, and L. T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 645–654, Aug. 1998.

[6] E. I. Verriest and T. Kailath, "On generalized balanced realizations," *IEEE Trans. Automat. Contr.*, vol. 28, pp. 833–844, Aug., 1983.

[7] P. DeWilde and A. Jan van de Veen, *Time-Varying Systems and Computations*. Boston, MA: Kluwer, 1998.

[8] J. Roychowdhury, "Reduced-order modeling of time-varying systems," *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 1273–1288, Oct. 1999.

[9] Y. Chen, "Model order reduction for nonlinear systems," M.S. thesis, Dept. of Elec. Eng. Comput. Sci., Mass. Inst. Tech., Cambridge, MA, Sept. 1999.

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

[11] J. R. Phillips, "Automated extraction of nonlinear circuit macromodels," in *Proc. Custom Integrated Circuit Conf.*, Orlando, FL, May 2000, pp. 451–454.

[12] ——, "Projection frameworks for model reduction of weakly nonlinear systems," in *Proc. 37th ACM/IEEE Design Automation Conf.*, 2000, pp. 184–189.

[13] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 17–32, Feb. 1981.

[14] K. Glover, "All optimal Hankel-norm approximations of linear multivariable systems and their l^∞-error bounds," *Int. J. Control*, vol. 39, no. 6, pp. 1115–1193, June 1984.

[15] G. A. Baker, Jr. and P. Graves-Morris, "Padé approximants Part I: Basic theory," in *Encyclopedia of Mathematics and Its Applications*, first ed. Reading, MA: Addison-Wesley, 1981.

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

[17] W. B. Gragg, "Matrix interpretations and applications of the continued fraction algorithm," *Rocky Mountain J. Math.*, vol. 4, no. 2, pp. 213–225, 1974.

[18] A. Ruhe, "The rational Krylov algorithm for nonsymmetric eigenvalue problems III: Complex shifts for real matrices," *BIT*, vol. 34, pp. 165–176, 1994.

[19] I. M. Elfadel and D. D. Ling, "Zeros and passivity of Arnoldi-reduced-order models for interconnect networks," in *Proc. 34th ACM/IEEE Design Automation Conf.*, Anaheim, CA, June 1997, pp. 28–33.

[20] E. Chiprout and M. S. Nakhla, "Analysis of interconnect networks using complex frequency hopping (CFH)," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 186–200, Feb. 1995.

[21] K. Gallivan, E. Grimme, and P. Van Dooren, "Multi-point Padé approximants of large-scale systems via a two-sided rational krylov algorithm," in *Proc. 33rd IEEE Conf. Decision and Control*, Lake Buena Vista, FL, Dec. 1994, pp. 14–16.

[22] C.-W. Ho, A. E. Ruehli, and P. A. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits Syst.*, vol. 22, pp. 504–509, June 1975.

[23] R. Telichevesky, J. White, and K. Kundert, "Efficient steady-state analysis based on matrix-free Krylov-subspace methods," in *Proc. 32rd Design Automation Conf.*, June 1995, pp. 480–484.

[24] ——, "Efficient AC and noise analysis of two-tone RF circuits," in *Proc. 33rd Design Automation Conf.*, June 1996, pp. 292–297.

[25] L. Zadeh, "Frequency analysis of variable networks," *Proc. I.R.E.*, pp. 291–299, Mar. 1950.

[26] J. Roychowdhury, "MPDE methods for efficient analysis of wireless systems," in *Proc. Custom Integrated Circuits Conf.*, Santa Clara, CA, May 1998, pp. 451–454.

[27] ——, "Efficient methods for simulating highly nonlinear multirate circuits," in *Proc. Design Automation Conf.*, Anaheim, CA, June 1997, pp. 269–274.

[28] H. D'Angelo, *Linear Time-Varying Systems*. Boston, MA: Allyn & Bacon, 1970.

[29] H. Tanimoto, M. Okumura, T. Itakura, and T. Sugawara, "Numerical noise analysis for nonlinear circuits with a periodic large signal excitation including cyclostationary noise sources," *IEEE Trans. Circuits Syst. I*, vol. 40, pp. 581–590, Sept. 1993.

[30] Y. Saad and M. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, vol. 7, no. 3, pp. 856–869, July 1986.

[31] B. N. Parlett, *The Symmetric Eigenvalue Problem*. Englewood Cliffs, NJ: Prentice-Hall, 1980.

[32] X. Ma and J. A. de Abreau-Garcia, "On the computation of reduced order models of nonlinear systems using balancing technique," in *Proc. 27th Conf. Decision and Control*, Dec. 1988, pp. 1165–1166.

[33] E. S. Hung and S. D. Senturia, "Generating efficient dynamical models for microelectromechanical systems from a few finite-element simulation runs," *J. Microelectromech. Syst.*, vol. 8, pp. 280–289, Sept. 1999.

[34] L. Sirovich, "Turbulence and the dynamics of coherent structures," *Q. Appl. Math.*, vol. 45, pp. 561–571, 1987.

[35] G. Berkooz, P. Holmes, and J. L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," *Annu. Rev. Fluid Mech.*, vol. 25, pp. 539–575, 1993.

[36] J. M. A. Scherpen, "Balancing for nonlinear systems," *Syst. Control Lett.*, vol. 21, pp. 143–153, 1993.

[37] W. J. Rugh, *Nonlinear System Theory*. Baltimore, MD: The Johns Hopkins Univ. Press, 1981.

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

[39] S. Boyd and L. O. Chua, "Fading memory and the problem of approximating nonlinear operators with Volterra series," *IEEE Trans. Circuits Syst.*, vol. 32, pp. 1150–1161, Nov. 1985.

[40] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1991.

[41] J. Li and J. White, "An efficient Lyapunov equation base approach for generating reduced-order models of interconnect," in *Proc. 36th ACM/IEEE Design Automation Conf.*, New Orleans, LA, June 1999, pp. 1–6.

[42] J. Roychowdhury, D. Long, and P. Feldmann, "Cyclostationary noise analysis of large RF circuits with multitone excitations," *IEEE J. Solid-State Circuits*, vol. 33, pp. 324–336, Mar. 1998.

[43] J. R. Phillips and K. S. Kundert, "Noise in mixers, oscillators, samplers, and logic: An introduction to cyclostationary noise," in *Proc. Custom Integrated Circuit Conf.*, Orlando, FL, May 2000, pp. 431–439.

[44] P. Feldmann and R. Freund, "Circuit noise evaluation by Padé approximation based model-reduction techniques," in *Proc. Int. Conf. Computer Aided-Design*, San Jose, CA, Nov. 1997, pp. 132–138.

[45] R. Telichevesky, J. White, and K. Kundert, "Receiver characterization using periodic small-signal analysis," in *Proc. Int. Custom Integrated Circuits Conf.*, May 1996, pp. 449–452.

Joel R. Phillips received the S.B., M.S., and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1991, 1993, and 1997, respectively.

He joined Cadence Design Systems in 1997 and is currently a Research Scientist with Cadence Berkeley Laboratories, San Jose, CA. His research interests are in the application of numerical simulation techniques to problems in electronic design automation.