General-Purpose Nonlinear Model-Order Reduction Using Piecewise-Polynomial Representations

Ning Dong, *Student Member, IEEE*, and Jaijeet Roychowdhury, *Senior Member, IEEE*

***Abstract*—We present algorithms for automated macromodeling of nonlinear mixed-signal system blocks. A key feature of our methods is that they automate the generation of general-purpose macromodels that are suitable for a wide range of time- and frequency-domain analyses important in mixed-signal design flows. In our approach, a nonlinear circuit or system is approximated using piecewise-polynomial (PWP) representations. Each polynomial system is reduced to a smaller one via weakly nonlinear polynomial model-reduction methods. Our approach, dubbed PWP, generalizes recent trajectory-based piecewise-linear approaches and ties them with polynomial-based model-order reduction, which inherently captures stronger nonlinearities within each region. PWP-generated macromodels not only reproduce small-signal distortion and intermodulation properties well but also retain fidelity in large-signal transient analyses. The reduced models can be used as drop-in replacements for large subsystems to achieve fast system-level simulation using a variety of time- and frequency-domain analyses (such as dc, ac, transient, harmonic balance, etc.). For the polynomial reduction step within PWP, we also present a novel technique [dubbed multiple pseudoinput (MPI)] that combines concepts from proper orthogonal decomposition with Krylov-subspace projection. We illustrate the use of PWP and MPI with several examples (including op-amps and I/O buffers) and provide important implementation details. Our experiments indicate that it is easy to obtain speedups of about an order of magnitude with push-button nonlinear macromodel-generation algorithms.**

***Index Terms*—Nonlinear macromodeling, piecewise polynomial (PWP).**

I. Introduction

CIRCUIT simulators and other electronic-design-automation tools today are being increasingly challenged by the ever-growing size and complexity of mixed-signal integrated systems. To enable an effective verification with reasonable computation, it has become commonplace to replace large system blocks with smaller ones or, in other words, to use macromodels to speed up simulations of interconnected system blocks. The emergence of Verilog-AMS [1] language provides powerful modeling capabilities such that designers can encapsulate high-level behavioral descriptions as well as structural descriptions of systems and components. However, a key bottleneck for any methodologies, including Verilog-AMS, remains the bottom-up generation of good macromodels.

Conventionally, macromodeling of mixed-signal nonlinear-system blocks has been accomplished manually, relying on the designers' understanding of the specific block being macromodeled for its proper abstraction. While manually generating macromodel continues to be the only option for classes of nonlinear systems for which no alternatives exist (general-purpose nonlinear macromodeling being a very difficult problem), it is not an effective methodology for several obvious reasons. The manually generated macromodels often fail to capture critical effects stemming from unanticipated interactions between blocks or from second-order phenomena in device models. Moreover, the fidelity and the performance of these macromodels are heavily skill-dependent, requiring considerable insights into the detailed design and operation of the underlying circuits. Moreover, important transistor-level nonidealities, particularly for deep-submicrometer technologies, usually become available only after layout and parasitic extraction and are difficult to prequantify prior to this stage.

It is in this context that there has been recent interest in automated computer-aided-design techniques for extracting accurate yet computationally inexpensive macromodels of circuit blocks directly from their (layout-extracted) SPICE-level descriptions. One of the attractions of such automated techniques is that the effects of nonidealities, parasitics, and undesired interactions at transistor level can be captured from the ground up in the generated macromodels. One can generate macromodel "on demand" with great convenience and efficiency, which is typically a matter of minutes at the push of a button as opposed to weeks or months of manual efforts.

There have been many well-established automated macromodeling techniques that apply mainly to relatively simple classes of circuits—linear time-invariant (LTI) systems, such as large $R/L/C$ interconnect networks (AWE [2], PVL [3]–[5], PRIMA [6], truncated balanced realization (TBR) [7]–[9], etc.), and linear time-varying (LTV) systems, such as mixers, switching-capacitor filters [10], [11], etc. However, many important effects in mixed-signal applications, such as distortions, intermodulations, clipping, slewing, etc., cannot be captured at all by the LTI or LTV systems. These phenomena are due to fundamental nonlinear behaviors in circuit equations and device models, which are discarded during linear approximations.

Recently, several techniques have emerged to address the automated macromodeling of certain nonlinear circuits. For an important class of nonlinear circuits whose nonlinearities can be adequately represented as polynomials (e.g., power amplifiers and sampling systems), a technique based on Volterra

Manuscript received May 31, 2006; revised March 30, 2007. This paper was recommended by Associate Editor J. R. Phillips.

N. Dong is with Texas Instruments Incorporated, Dallas, TX 75243 USA (e-mail: ningd@ti.com).

J. Roychowdhury is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: jr@umn.edu).

Digital Object Identifier 10.1109/TCAD.2007.907272

series expansions was first proposed in [12], followed by several important extensions [13]–[15]. These approaches essentially generate macromodels of the linearized system using Krylov-subspace methods and incorporate higher order nonlinearities via distortion inputs.

For general nonlinear systems with strong nonlinearities, a trajectory-based piecewise-linear (TPWL) approach was first proposed in [16] and has become increasingly popular [17]–[24]. These methods divide the state space of the nonlinear system into piecewise regions and reduce each, depending on the order of approximation, with linear or weakly nonlinear model-order-reduction (MOR) techniques. There are also techniques that build macromodels directly from simulation data using data-mining methods (e.g., [25]–[29]), but many of them are targeting performance verifications, for which the detailed discussion is beyond the scope of this paper.

In this paper, we present a method that combines the trajectory-based techniques and the weakly nonlinear MOR algorithms, as they have complementary advantages and disadvantages. The weakly polynomial macromodels capture small-signal distortions well around the expansion point, but they become rapidly inaccurate for large input excitations. The TPWL, on the other hand, captures strong nonlinearities well in wider range; however, its accuracy in representing weak nonlinearities within each region is limited since any higher order nonlinearities are due only to the smoothing function, which is not necessarily consistent with the local Volterra expansions. As a result, the TPWL-generated macromodels can be useful for large-signal transient simulations but are often not suited for, e.g., small-signal distortion analysis. Our method, dubbed **PWP** because of its reliance on **P**iece**W**ise **P**olynomials, is to follow the TPWL methodology but approximate each region with higher order (tensor) polynomials instead of using purely linear representations. The PWP strives to deliver one macromodel that can capture strong and weakly nonlinearities simultaneously, thus remedying the limitations of previous approaches. The generated macromodels are targeted for general nonlinear circuits and can be used as drop-in replacements for virtually any kind of analyses.

Many unique features have been incorporated into PWP in order to generate broadly applicable macromodels. To improve its validity, we merge the piecewise regions from multiple training trajectories to enlarge the state-space coverage. A novel smoothing function, which is used to achieve superior smoothness and better convergence, enhances the robustness of PWP. Our implementations use vector inputs and outputs to account for loading effects, which is important for system-level usage of the generated macromodels. Heuristics critical for macromodel accuracy and efficiency, such as choosing expansion points along trajectories, generation of linear projection basis, selection of appropriate training inputs, etc., are all explored in depth in this paper. From an implementation and modularity standpoint, the PWP can make use of any existing polynomial MOR technique (e.g., [10], [14], and [15]) to perform the weakly nonlinear reduction for each piecewise region. In particular, we present an alternative novel technique, which is termed the multiple pseudoinput (MPI), which exploits the idea of combining proper orthogonal decomposition (POD) within Krylov-based reduction framework. The implementation of MPI is very easy and straightforward, and its performance is comparable with the existing methods.

We validate the PWP using a current-mirror op-amp and two high-speed digital I/O buffers with various types of analysis, including dc, ac, large-signal transient, and harmonic balance (HB). The experimental results confirm that the PWP-generated macromodels can indeed be employed as general-purpose drop-in replacements in typical mixed-signal design environments. The macromodels capture strongly nonlinear phenomena (clipping, slewing, etc.), as well as weakly nonlinear ones (small-signal distortions). On average, the macromodels deliver speedups of $6-9\times$ over the original circuits in our MATLAB implementations.

The remainder of this paper is organized as follows. In Section II, we briefly review the mathematical underpinnings of the existing macromodeling techniques. The PWP technique is presented in Section III, followed by the MPI method in Section IV. Validation of PWP and its generated macromodels is presented in Section V.

II. Previous Work and Background

In this section, we develop necessary background and mathematical notations, further reviewing the linear and weakly nonlinear MOR, TPWL, POD, and concepts from all which are incorporated into the PWP method.

A. LTI System MOR

The basic idea of MOR for an LTI system is to project high-dimension state space into a subspace which spans the solution space effectively. The prevalent algorithms are the Krylov-based techniques (e.g., [3]–[6], [30]–[42]).

Consider a size n LTI system described by ordinary differential equation

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

where $x(t) \in \mathbb{R}^n$ is the internal state and $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$ are the m-input and the p-output waveforms. The matrices are the following: $A \in \mathbb{R}^{n\times n}$, $E \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{p\times n}$.

This LTI system can be reduced to size q by a projection matrix (basis) $V \in \mathbb{R}^{n\times q}$ through the operation[1] $x = Vz$, $z \in \mathbb{R}^q$ such that

$$\hat{E} = V^{\mathrm{T}}EV \quad \hat{A} = V^{\mathrm{T}}AV \quad \hat{B} = V^{\mathrm{T}}B \quad \hat{C} = CV$$

leading to the reduced model

$$\hat{E}\frac{dz}{dt} = \hat{A}z(t) + \hat{B}u(t), \quad y = \hat{C}z(t). \tag{2}$$

The transfer functions of (1) and (2), i.e., $H(s) = C(sE - A)^{-1}B$ and $\hat{H}(s) = \hat{C}(s\hat{E} - \hat{A})^{-1}\hat{B}$, can be expanded with

[1]In this paper, we mainly consider Arnoldi-based projection. There are also Lanzcos-based methods, e.g., [4], [5], [30], and [31].

Taylor series at $s = 0$

$$H(s) = -C\left[A^{-1} + s(A^{-1}E)A^{-1} + s^2(A^{-1}E)^2A^{-1} + \cdots\right]B \tag{3}$$

$$\hat{H}(s) = -\hat{C}\left[\hat{A}^{-1} + s(\hat{A}^{-1}\hat{E})\hat{A}^{-1} + s^2(\hat{A}^{-1}\hat{E})^2\hat{A}^{-1} + \cdots\right]\hat{B} \tag{4}$$

where the coefficients of s are called moments, and B is usually called the starting vectors.

The projection basis V is defined by qth-order Krylov subspace

$$K_q(M, R) = \text{span}\{R, MR, \ldots, M^{q-1}R\} \tag{5}$$

where $M = A^{-1}E$, and $R = A^{-1}B$ for LTI system of (1).

It has been proved (e.g., [4] and [5]) that by choosing the qth-order Krylov subspace (5) as the projection matrix V, the first q moments of transfer functions of (3) and (4) will match to each other exactly. Typically, V can be calculated via, e.g., the Lanzcos or Arnoldi methods [3], [4], [42].

B. Proper Orthogonal Decomposition

POD is an alternative technique to generate a projection subspace, and it has been widely applied to many different MOR problems (e.g., [43]–[49]). The wide applications of POD are attributed to its nice properties of optimally representing system data with a small number of POD-basis vectors in the sense of least square approximation.

To apply the POD, a "snapshot" of the system solution is first obtained either by experimental measurements or by numerical simulation. These state-space vectors are then used to extract orthogonal set of POD-basis vectors which span the projection subspace. There are actually three different ways of implementing POD [50]: 1) Karlhunen–Loéve decomposition; 2) principal component analysis; and 3) singular value decomposition (SVD). In this paper, we adopt the SVD due to its simplicity and wide availability.

For example, let $x \in \mathbb{R}^n$ be the solution vector of the system being macromodeled. By running a transient simulation, one can assemble the system data as $\bar{X} = [x(t_1), x(t_2), \ldots, x(t_s)]$. The POD method seeks to find a basis V to maximize the representation of data points, which is equivalent to finding a projection basis V to minimize the overall projection error [45]

$$\|\bar{X} - VV^{\mathrm{T}}\bar{X}\|.$$

Obviously, the solution to this optimization problem is to perform the SVD on the data collection $\bar{X}$, i.e.,

$$V = \text{svd}(\bar{X})$$

where V is given by the dominant singular vectors. Note that the SVD of full-rank matrix is expensive $(O(n^3))$. However, the "snapshot" $\bar{X}$ typically has much less number of columns than rows (i.e., the system size); thus, the "economy-size" SVD of $\bar{X}$ can be performed more efficiently.

The advantage of POD is also its limitation. Because the POD basis is generated from system response with a specific input, the reduced model is only guaranteed to be close to the original system when the input is close to the training input. As a result, one needs to properly design the excitation signals to reveal most of the system dynamics.

C. Weakly Nonlinear System Macromodeling

A nonlinear circuit or system can be generally described by differential-algebraic equation as

$$\dot{q}\left(x(t)\right) = f\left(x(t)\right) + b(t). \tag{6}$$

As usual, all variables (except time t) are vector-valued. Without loss of generality (e.g., [10]), (6) can be expressed as

$$E\dot{x} = f(x) + Bu(t), \quad y = Cx \tag{7}$$

where $x \in \mathbb{R}^n$ is the unknown state vector, and $f(x)$ is a nonlinear vector function. $u(t) \in \mathbb{R}^{n\times m}$ is an m-input to the system. B and C are the same as the definition in (1). We shall use (7) all through this paper to describe the nonlinear system.

When the input signal $u(t)$ is small enough, the nonlinear term $f(x)$ can be adequately represented by polynomial expansions around dc operating point such that

$$E\dot{x} = A_1x + A_2x \otimes x + A_3x \otimes x \otimes x + \cdots + Bu(t) \tag{8}$$

where A_i is the ith order derivative, and the symbol $\otimes$ represents the Kronecker tensor product.

By Volterra theory [51], the solution of (8) is the summation of different orders of responses such that

$$x(t) = \sum_{i=1}^{\infty} x_i(t)$$

where $x_i(t)$ is the ith-order response given by the ith Volterra kernel $h_i(\tau_1, \ldots, \tau_i)$

$$x_i(t) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_i(\tau_1, \ldots, \tau_i), \ldots, u(t - \tau_i)d\tau_1, \ldots, d\tau_i.$$

More precisely, $x_i(t)$ can be recursively obtained by solving the same linear system under different inputs. As shown in Fig. 1, the first- through the third-order responses are the solutions of the following linear systems, respectively

$$E\dot{x}_1 = A_1x_1 + Bu(t) \tag{9}$$

$$E\dot{x}_2 = A_1x_2 + A_2x_1 \otimes x_1 \tag{10}$$

$$E\dot{x}_3 = A_1x_3 + A_2(x_1 \otimes x_2 + x_2 \otimes x_1) + A_3x_1 \otimes x_1 \otimes x_1. \tag{11}$$

Now, the weakly polynomial MOR problem has been recasted as the reduction of a series of LTI systems, where each is n-dimensional, that produce different order responses

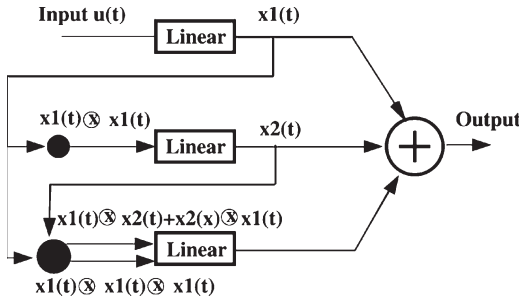


Fig. 1. Block diagram of the solution to weakly nonlinear system based on the Volterra series expansion.

$x_i(t)$. Based on this formulation, several approaches have been proposed.

1) Separate Projection: The first approach proposed in [12] is to reduce each LTI system with separated Krylov subspace. For example, the first-order LTI system of (9) can be reduced by the projection basis $V_1 \in \mathbb{R}^{n\times q_1}$ generated from the Krylov subspace $K_q(A_1^{-1}E, A_1^{-1}B)$. Then, the response $x_1(t)$ was approximated as $x_1(t) \approx V_1 z_1(t)$ and plugged into the second-order LTI system of (10), which can now be written as

$$E\dot{x}_2 = A_1 x_2 + B_2 u_2(t) \tag{12}$$

where $B_2 = A_2(V_1 \otimes V_1) \in \mathbb{R}^{n\times q_1^2}$, and $u_2(t) = z_1(t) \otimes z_1(t) \in \mathbb{R}^{q_1^2}$. This is actually a similar LTI system as (9) except that it has a q_1^2 input. As a result, the projection basis $V_2 \in \mathbb{R}^{n\times q_2}$ can be generated similarly from the Krylov subspace $K_{q_2}(A_1^{-1}E, A_1^{-1}B_2)$ with multiple starting vectors B_2. The projection-basis generation for the third-order system follows analogously.

The main difficulty associated with this method is the rapidly increasing dimension of the projection basis, resulting in inefficiently large reduced models.

2) Uniform Projection: To generate a compact model, it was proposed in [13] that the separated basis $V_1, V_2, \ldots$, can be merged via SVD to construct a single uniform basis V, i.e., $V = \text{svd}([V_1, V_2, \ldots])$. The goal is to obtain a more compact basis by "deflating" the subspace while retaining the similar properties of moment matching.

However, the improvement is not so attractive because the dimension of the Krylov subspaces for the second- and third-order systems increases exponentially due to the tenser product, which leads to the large dimension for merged subspace V even after the "deflating" process.

3) Nonlinear Model Order Reduction Method (NORM)—Momentwise Projection: To alleviate this obstruction, the relationship between moments of different order transfer functions and the Krylov subspace with corresponding starting vectors has been studied in depth in NORM [15]. It is shown that there is some redundancy among the Krylov subspaces of each LTI system, which can be removed by carefully choosing proper starting vectors when generating the Krylov subspaces. With the NORM, one can obtain a compact projection basis that is tailored precisely according to the moments to be matched, resulting in a very compact reduced model without loss of accuracy.

D. Trajectory Piecewise-Linear Method

For general nonlinear model reduction, a TPWL approach was first proposed in [16] and then extended in several ways [17], [18], [21]–[24], [26]. The idea is to represent a nonlinear system as a collage of linear models in adjoining polytopes, which is centered around the expansion points in the state space. The essence of the method is outlined as follows.

1) Given a nonlinear system of (7), linearize it at various expansion points x_i, $i = 1, 2, \ldots, s$

$$E\dot{x} = f(x_i) + A_i(x - x_i) + Bu(t), \quad y = Cx.$$

2) Generate a projection basis V_i for each LTI model and calculate a common subspace V of the union $V_{\text{union}} = [V_1 V_2, \ldots, V_s]$ via $V = \text{svd}(V_{\text{union}})$. The size of V is usually larger than each V_i but smaller than the size of the original system.
3) Perform the linear model reduction using V, such as

$$\hat{E}\dot{z} = \hat{f}(x_i) + \hat{A}_i(z - z_i) + \hat{B}u(t), \quad y = \hat{C}z$$

where the reduced matrices $\hat{E}$, $\hat{A}$, $\hat{B}$, and $\hat{C}$ are the same as in (2), and $\hat{f}(x_i) = V^{\mathrm{T}} f(x_i)$.
4) The final reduced model is the weighted combination of all the reduced models

$$\hat{E}\dot{z} = \sum_{i=1}^{s} w_i(z)\left(f(z_i) + \hat{A}_i(z - z_i) + \hat{B}u(t)\right), \quad y = \hat{C}z$$

where $w_i(z)$ is the weight function.

The TPWL has excellent global approximations because of the piecewise nature but has limited local accuracy for small signal analysis. Intuitively, when the excitation is small enough to keep the states stay within one region, the system reduces to a pure LTI model, and no distortions could be captured. Nonlinearities induced exclusively by the nonlinear weight function $w_i(z)$ are generated only when states cross boundaries. Recently, some works [23], [24], [26] have greatly extended the original TPWL method, making it more scalable and practical. However, there is still less evidence in literatures to show the usage of the generated macromodel in other analysis, such as dc, ac, HB, etc. Moreover, this will be addressed in this PWP work.

III. PWP Approach

In this section, we first present the essential procedure of the PWP macromodeling algorithm and then discuss the implementation detail later in this section. To make it more concise, it is assumed that the projection basis V_i for each polynomial model has been obtained. We shall delay the discussion of generating such a basis using the MPI method until Section IV.

A. PWP Representations

Suppose that we have chosen s expansion points $\{x_1, x_2, \ldots, x_s\}$ from the state space of (7), each of which has

a quadratic expansion

$$E\dot{x} = f(x_i) + A_i^{(1)}x^{(1)} + A_i^{(2)}x^{(2)} + Bu(t), \quad y = Cx.$$

Here, $x^{(1)} = x - x_i$, $x^{(2)} = (x - x_i) \otimes (x - x_i)$, $A_i^{(1)}$, and $A_i^{(2)}$ are the first- and second-order derivatives. To simplify our discussion, we only present the system using quadratic model (extension to higher order terms is straightforward).

The projection basis V_i for each polynomial model can be constructed from the coefficient matrices using any weakly nonlinear MOR techniques. Similarly as TPWL [16], a uniform projection base V is then generated via SVD on the collection of all basis. If $V \in \mathbb{R}^{n\times q}$, a size-$q$-reduced model is given by

$$\hat{E}\dot{z} = \hat{f}(x_i) + \hat{A}_i^{(1)}z^{(1)} + \hat{A}_i^{(2)}z^{(2)} + \hat{B}_i u(t)$$

where $z_i = V^{\mathrm{T}}x_i$, $z^{(1)} = z - z_i$, $z^{(2)} = (z - z_i) \otimes (z - z_i)$, and $\hat{f}(x_i) = V^{\mathrm{T}}f(x_i)$. Similarly, the reduced matrices are $\hat{E} = V^{\mathrm{T}}EV$, $\hat{A}_i^{(1)} = VA_i^{(1)}V$, and $\hat{A}_i^{(2)} = V^{\mathrm{T}}A_i^{(2)}V \otimes V$.

The final reduced-order PWP model is obtained by a weighted combination of these regions such that

$$\begin{aligned}\hat{E}\dot{z} &= \sum_{i=1}^{m} w_i(z)\left(\hat{f}(x_i) + \hat{A}_i^{(1)}z^{(1)} + \hat{A}_i^{(2)}z^{(2)} + \hat{B}_i u(t)\right)\\ y &= C\left[\sum_{i=1}^{m} w_i(z)(x_i + V(z - z_i))\right] \end{aligned} \tag{13}$$

where $w_i(z)$ is a smooth weight function, as elaborated in Section III-E.

Although the general procedure of PWP looks simple at first glance, a practical implementation of the PWP involves considerable details that are critical to the model's accuracy, stability, and speedup. These implementation details will be discussed in the rest of the section.

B. Choose Expansion Points

To be useful in practice, a PWP-generated macromodel needs to cover certain range of state space with limited expansion regions. To start, one has to choose application-specific inputs to "train" the algorithm. For example, in this paper, we use sinusoidal signals with various amplitudes and frequencies as training inputs to an op-amp example.

An adaptive heuristic strategy to choose expansion points from one trajectory is summarized as follows.

1) Simulate the full system with a training input.
2) Start from an initial state x_0 (usually, the dc state), and construct an LTI model such that

$$f_{\mathrm{linear}} = f(x_0) + A_0(x - x_0)$$

where A_0 is the Jacobian matrix of $f(x)$ evaluated at x_0.

3) Traverse the trajectory, and ensure that the relative error $\mathrm{err} = (|f(x) - f_{\mathrm{linear}}(x)|/|f_{\mathrm{linear}}(x)|) < \alpha$, where α is the predefined error tolerance.
4) If $\mathrm{err} > \alpha$, add the current state x into the expansion point set. Start from this state, and construct a new LTI model. Repeat steps 2)–4) until the end of the trajectory.

Here, one can explore tradeoffs between the accuracy and the speedup by tuning α. A small α could lead to an accurate model with small errors but less speedup due to large number of regions. A typical value used in this paper is about 10^{-3}–10^{-6}.

It is noticed that this heuristic approach cannot guarantee capturing all necessary information. Sometimes, it might miss some key points that would cause significant runtime error. Certainly reducing α could bring in more points, but it will also increase the number of regions. A better way is to rerun the generated macromodel with the same training input and, if necessary, add expansion points manually to make the waveform match the original. Having this capability of manually adding the expansion points is particularly useful for some digital applications [21], [22] to obtain multiple dc solutions correctly.

C. Merge Multiple Trajectories

The key to generating a widely applicable PWP model is to maximize the state-space coverage with limited pieces of regions. This is done by merging regions from different trajectories. To avoid large number of regions, redundancy can be removed by examining the similarities among the regions using the following steps.

1) Choose a base set of expansion point, and ensure the model accuracy for that particular training input.
2) For new points on the new trajectory, check the L2-norm distances $d_1 = |x^i_{\mathrm{new}} - x^j_{\mathrm{base}}|_2$ between the new point set and the base set. This can be done efficiently using vectorized operation in MATLAB.
3) Select the points with L2 distance less than some predefined tolerance δ_1. Then, check the L2 distances of the Jacobian matrices between these selected points and the base set, i.e., $d_2 = \|A^i_{\mathrm{select}} - A^j_{\mathrm{base}}\|_2$.
4) Remove the points with both $d_1 \le \delta_1$ and $d_2 \le \delta_2$. Append the rest of the points into the base set.
5) Repeat steps 2)–4) for all trajectories. The typical value of δ_1 and δ_2 may vary from 10^{-2} to 10^{-6}, depending on different applications.

D. Uniform Projection Basis

For each region, a unique projection basis $V_i \in \mathbb{R}^{n\times q_i}$ $(q_i \ll n)$ is generated by certain weakly polynomial MOR technique. The projection operation $x = V_i z$ $(x \in \mathbb{R}^n, z \in \mathbb{R}^{q_i})$ implies that z is the local coordinate of x in the subspace spanned by column vectors of V_i. Thus, the reduced models are actually defined in different local coordinate systems. When simulating the macromodel in the reduced space, it is important to do it in one common subspace (coordinate system), which is possibly larger but contains all the underlying (smaller) subspaces. Otherwise, one cannot ensure

a smooth transition (among different coordinate systems) by only using the weight function. A straightforward way of finding such a common subspace is to collect dominant information from $V_{\text{union}} = [V_1, V_2, \ldots, V_s]$ via SVD, i.e., $V = \text{svd}(V_{\text{union}})$, and to keep only q $(q < n)$ dominant singular vectors.

It is possible that the dimension of the common subspace may also increase when including more and more regions from the combined trajectories. One may argue that finally, the dimension could be close to the system size, making the MOR meaningless. If so, it simply means that the solution space can hardly be reduced, for which none of these MOR techniques would work. Fortunately, circuits are highly connected system, and the number of truly independent variables (or order of freedom) is usually small compared with the system size.

Another key observation when performing the SVD is that the singular value always has a deep cut at certain position, indicating the existence of such a common subspace. However, the dimension of the common subspace is usually larger than each individual projection base, which puts a limit on the size of the final reduced model.[2]

E. Choice of Weight Functions

Weight functions play a key role in all trajectory-based approaches. Basically, each state calculated by the macromodel is the result of interpolation from all nearby linear/polynomial models. It is the weight function that amplifies the contributions from right neighbors and suppresses the "noise" from the others. Therefore, the value of the weight function $w_i(z)$ should be close to one when the state vector z approaches the center point z_i and should rapidly attenuate to zero as z leaves z_i. Additionally, weight functions should be continuous and differentiable, which is necessary to ensure the convergence of the transient simulation.

Although there is a considerable choice in functions satisfying this requirement, it is not trivial to make up a good weight function. In the original TPWL [16], the weight function $w_i(z)$ of current state z is calculated as the following procedure.

1) For $i = 1, \ldots, s$, compute $d_i = |z - z_i|_2$.
2) Take $m = \min(d_i)$ item. For $i = 1, \ldots, s$, compute $\hat{w}_i(z) = e^{-\beta d_i/m}$, where β is a constant, e.g, $\beta = 25$. item Normalize $\hat{w}_i(z)$ such that $S(z) = \sum \hat{w}_i(z)$ and $w_i(z) = \hat{w}_i(z)/S(z)$.

The initial experiment of using this weight function shows some problems during the transient simulation. Sometimes, the error is large when the current state is away from most of the expansion points, i.e., on the border of the space that is covered by the expansion points (Fig. 2). By experiments, we use the following weight function that seems to be more effective for PWP:

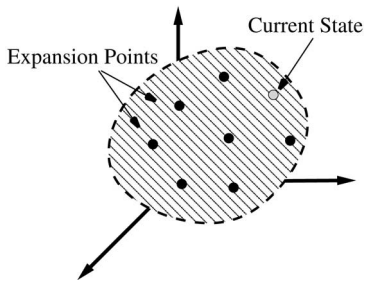


Fig. 2. Current state on the border of the space covered by the expansion points.

$$w_i(z) = \left[\frac{d_{\min}}{d_i(z)} e^{-\frac{d_i(z) - d_{\min}}{D_{\min}}} \right]^p \tag{14}$$

where $d_i(z) = |z - z_i|_2^2$, $d_{\min} = \min(d_i(z))$ for $i = 1, \ldots, s$, and $D_{\min}$ is the minimum distance among those center points $\{z_1, \ldots, z_s\}$. Parameter p (typically, $p = 1-2$) is used to make the transaction smoother or shaper when switching from one region to another. The whole weight function is finally normalized to satisfy $\sum_{i=1}^{s} w_i(z) = 1$.

The difference of these two weight functions can be illustrated using the following trivial test. Let the two center points be $z_1 = 0.99$ and $z_2 = 1.01$ and have z swept from zero to two. Ideally, $w_1(z)$ should be dominant when $z < 1$ and so does $w_2(z)$ when $z > 1$. We plot $w_1(z)$ and $w_2(z)$ for both of the weight functions [$p = 1$ for (14)] in Fig. 3.

One of the problems, as shown in Fig. 3(a), is that the weights do not attenuate to zero, as expected, when z is away from the center points.[3] This means that when the current state is on the margin of the space, as shown in Fig. 2, the weight function is averaging the results from all the regions. This might be reasonable if it is a mild nonlinear system and if all linearized models have some similarities. However, experiments show that it often leads to unpredictable behavior when the system exhibits strong nonlinear dynamics. In such case, a weight function, as shown in Fig. 3(b), that picks the best candidate model while suppressing the "noise" well from the others is more appreciated to get a stable transient simulation.

F. PWP Versus PWL

In this section, we demonstrate the advantage of PWP over PWL using an illustrative example. Fig. 4 shows a cascade NMOS amplifier, each stage of which is biased around $V_{\text{gs}} = 3$ with a gain that is slightly larger than one. Therefore, all stages will remain active when excited by the input signals. The whole circuit has a size of 50.

The training trajectory is obtained by a transient simulation with a square pulse signal around dc $= 3$. PWP is applied

[2]It is interesting to note that, recently, a grouping strategy has been proposed in [23] and [26], where the projection basis is generated from a "group" of local regions instead of calculating a common subspace from all regions via SVD. Therefore, it can deliver more compact macromodels with better speedups. However, the success of applying "group" seems to rely on the dense samplings in the state space, which is typically 10^3-10^4 points versus 30–40 points in PWP, to ensure smooth transitions from one local subspace to another.

[3]Another problem is that $\hat{w}_i(z) = e^{-\beta d_i/m}$ is not well defined at z_i as $m \to 0$. One has to arbitrarily force $w_i(z_i) = 1$ to avoid being "divided by zero." Moreover, the definition of distance function $d_i(z) = |z - z_i|_2$ is not differentiable at z_i, and it is replaced by $d_i(z) = (|z - z_i|_2)^2$.

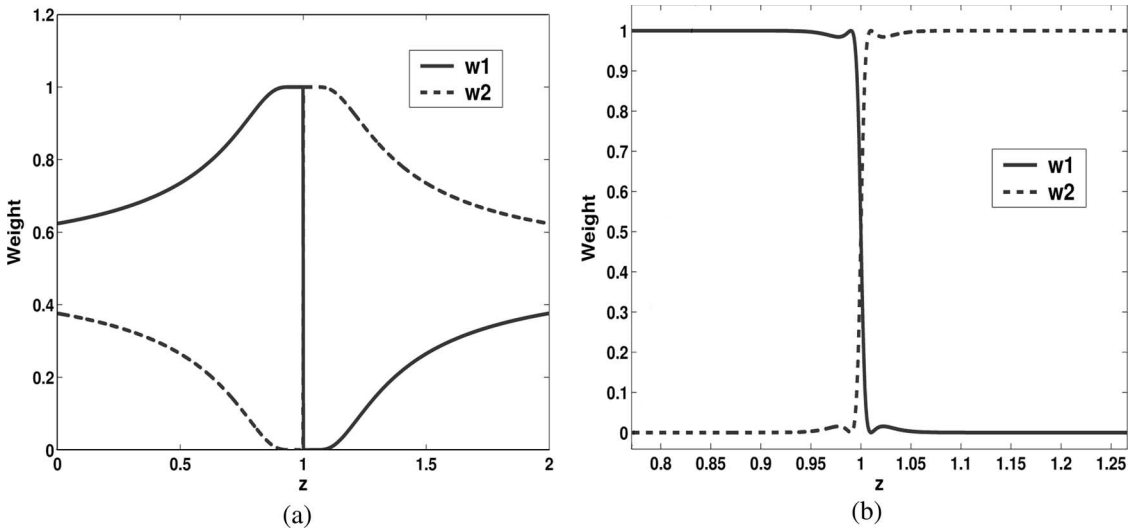


Fig. 3. Simple test result of two weight functions. (a) TPWL weight function. (b) PWP weight function.

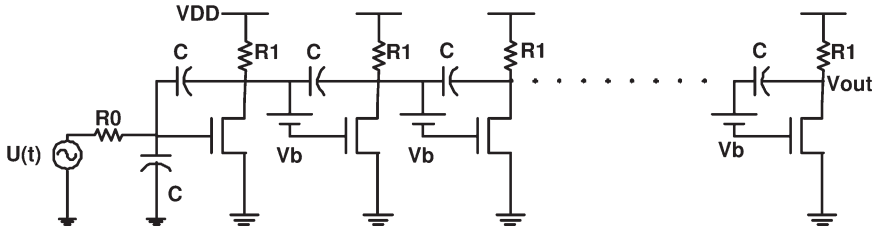


Fig. 4. Cascade NMOS amplifier.

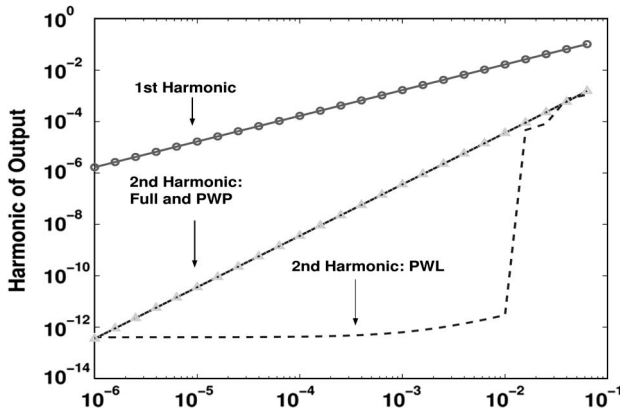


Fig. 5. Harmonic analysis of the cascade NMOS amplifier.

to generate a macromodel with 17 regions, each of which is reduced to a quadratic model with a size of 20. This macromodel is then used in HB analyses with an input $u(t) = 3 + A\sin(2\pi 100t)$, where A is swept from 10^{-8} to 10^{-1}. By skipping the quadratic term in the PWP model, we simulate the PWL model again and compare their first two harmonics with the full model. The results are shown in Fig. 5.

It is clearly shown that the first two harmonics of the PWP model are virtually identical to that of the full system. However, the PWL reduces to a pure LTI model when the input magnitude A is small, and thus, no second harmonic is captured. Only when the input becomes large will the second harmonic approach the full system due to the weight function.

It is out of the question that the PWP is superior to the PWL in capturing higher order nonlinearities within single region. However, the PWP relies on higher order derivatives and requires more CPU time and memory for evaluations. On the other hand, the PWL model demands less resources because of its simplicity but fails to capture critical nonlinearities. To improve its accuracy, it needs more expansion regions that may eventually compromise the efficiency. One has to explore these tradeoffs to generate proper "on-demand" macromodels.

Finally, it is worth mentioning that the PWP can adopt any existing weakly nonlinear MOR technique for each piecewise region. Good candidates include NORM [15] and the techniques in [12] and [14]. Alternatively, we propose another simple yet effective method, i.e., the MPI approach, as elaborated in the next section.

To conclude this section, we summarize the procedure of the PWP algorithm as follows.

Input: system equations of (7), derivative matrices $A_1 = (\partial f/\partial x)$, and $A_2 = (\partial^2 f/\partial^2 x)$.

Output: reduced PWP model of (13).

1) Choose a set of expansion points $X_s = \{x_1, x_2, \ldots, x_s\}$ by merging the trajectories from multiple application-specified training, e.g., transient, dc sweeps, etc.
2) Use any weakly polynomial MOR method (e.g., MPI (Section IV) or NORM [15]) to get a set of projection basis $\{V_1, V_2, \ldots, V_s\}$. Form a uniform basis via SVD, i.e, $V = \text{svd}([V_1, V_2, \ldots, V_s])$.

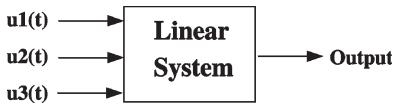


Fig. 6. Equivalent linear system with multiple inputs.

3) Perform a normal projection-based model reduction to get a set of reduced polynomial models as (13).
4) Apply the weight function of (14) to construct a final reduced PWP model as (13).

IV. Polynomial MOR With MPI

In practice, PWP relies on the weakly nonlinear MOR techniques to generate a projection basis for each region. Any existing approaches (e.g., [12], [13], [15], etc.) can be easily embedded in the PWP framework, and NORM [15] is known to be the best approach so far to generate a compact basis. In this section, we present an alternative to NORM, namely, the MPI method.

A. Volterra Series Expansion in Time Domain

Our MPI approach was originally inspired by rephrasing the Volterra series expansion in time domain. As shown in Fig. 1, a nonlinear system of (8) can be solved by solving the same linear system recursively with different inputs. By adding (9) to (11), we have

$$E\dot{x} = A_1 x(t) + A_2 u_2(t) + A_3 u_3(t) + Bu(t)$$

or in a companion form

$$E\dot{x} = A_1 x(t) + \underbrace{[\,B \quad A_2 \quad A_3\,]}_{B_{\rm eq}} \underbrace{\begin{bmatrix} u(t) \\ u_2(t) \\ u_3(t) \end{bmatrix}}_{u_{\rm eq}(t)} \tag{15}$$

where $x = x_1 + x_2 + x_3$, $u_2(t) = x_1(t) \otimes x_1(t) + x_1(t) \otimes x_2(t) + x_2(t) \otimes x_1(t)$, and $u_3(t) = x_1(t) \otimes x_1(t) \otimes x_1(t)$. As shown in Fig. 6, this is an equivalent linear system with multiple inputs $u_{\rm eq}$ and matrix $B_{\rm eq}$.

In order to apply MOR to the equivalent linear system of (15), we can generate the Krylov projection bases using $B_{\rm eq}$ as the starting vectors. However, the kth-order derivative A_k would have n^k columns, prohibiting the direct usage of A_2 and A_3 even if they are very sparse in general. This can be remedied by exploiting the intrinsic correlations in $u_2(t)$ and $u_3(t)$ in time or frequency domain. For example, if $u_2(t)$ can be expressed as a linear combination of small number of vectors, such as

$$u_2(t) = U_2 \tilde{u}_2(t)$$

where $u_2(t) \in \mathbb{R}^{n^2}$, $U_2 \in \mathbb{R}^{n^2 \times q_2}$, $\tilde{u}_2(t) \in \mathbb{R}^{q_2}$, and $q_2 \ll n^2$, then $A_2 u_2 = A_2 U_2 \tilde{u}_2 = B_2 \tilde{u}_2$, where B_2 would only have q_2 columns. Therefore, the equivalent linear system of (15) becomes

$$E\dot{x} = A_1 x + \underbrace{[\,B \quad B_2 \quad B_3\,]}_{B_{\rm eq}} \underbrace{\begin{bmatrix} u(t) \\ \tilde{u}_2(t) \\ \tilde{u}_3(t) \end{bmatrix}}_{u_{\rm eq}(t)} \tag{16}$$

where $u_3 = U_3 \tilde{u}_3$, and $B_3 = A_3 U_3 \in \mathbb{R}^{n \times q_3}$. The total number of columns in $B_{\rm eq}$ would have been reduced from $m + n^2 + n^3$ to $m + q_2 + q_3$, where m is the number of inputs to the original system.

For simplicity, consider only expanding the system to the quadratic model such that $u_2(t) = x_1(t) \otimes x_1(t)$, where $x_1(t)$ is the response of the first-order LTI system (9). This motivates us to represent $x_1(t)$ with a compact basis, i.e., $x_1(t) = V_1 z_1(t)$, $V_1 \in \mathbb{R}^{q_1}$, such that $u_2(t) = V_1 \otimes V_1 z_1(t) \otimes z_1(t)$. It follows that $U_2 = V_1 \otimes V_1$, $\tilde{u}_2(t) = z_1(t) \otimes z_1(t)$, and $B_2 = A_2(V_1 \otimes V_1) \in \mathbb{R}^{n \times q_1^2}$. This is the essential idea behind the weakly polynomial MOR techniques that are discussed in Section II-C, where V_1 is obtained from the Krylov subspace using B as the starting vectors.

Alternatively, V_1 can also be calculated using the POD approach, as discussed in the next section.

B. MPI Approach With POD Basis

To simplify the discussion, we illustrate the MPI method using the quadratic expansion of (8), i.e.,

$$E\dot{x} = A_1 x + \underbrace{[\,B \quad A_2\,]}_{B_{\rm eq}} \underbrace{\begin{bmatrix} u(t) \\ u_2(t) \end{bmatrix}}_{u_{\rm eq}(t)} \tag{17}$$

where $A_2 \in \mathbb{R}^{n \times n^2}$ is the second derivative of $f(x)$, and $u_2(t) = x_1(t) \otimes x_1(t)$.

The POD basis can be calculated either from the time or frequency domain. In the time domain, we first solve $x_1(t)$ by running several steps of transient simulation for the LTI system of (9), collecting the samplings in $\bar{X} = [x_1(t_1), x_1(t_2), \ldots, x_1(t_i)]$. The POD basis is then given by SVD, i.e., $V_1 = \text{svd}(\bar{X}) \in \mathbb{R}^{n \times q_1}$ and $x_1(t) \approx V_1 z_1(t)$. The equivalent LTI system of (17) now becomes

$$E\dot{x} = A_1 x + \underbrace{[\,B \quad B_2\,]}_{B_{\rm eq}} \underbrace{\begin{bmatrix} u(t) \\ \tilde{u}_2(t) \end{bmatrix}}_{u_{\rm eq}(t)}$$

where $B_2 = A_2 V_1 \otimes V_1$, and $\tilde{u}_2 = z_1 \otimes z_1$. This LTI system can be further reduced through the Krylov-subspace projection with multiple starting vectors $B_{\rm eq} = [B, B_2]$.

This method is easily extended to higher order systems, for which the POD basis can be calculated more easily in the frequency domain. For example, to generate V_2 for the second-order LTI system of (12), one can get "snapshot" $\bar{X}_2$ in the frequency domain by calculating $H_2(s_i) = (s_i E - A_1)^{-1} B_2$ for selected frequency points s_i, where $B_2 = A_2(V_1 \otimes V_1)$. Once V_1 and V_2 are available, one can replace $x_1(t) = V_1 z_1(t)$

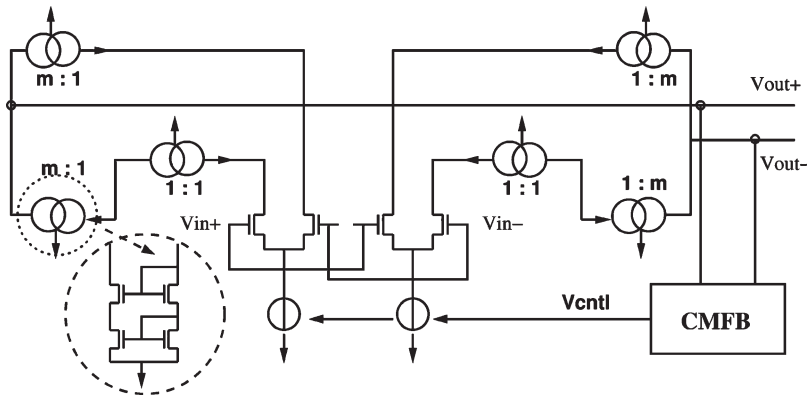


Fig. 7. Current-mirror op-amp with 50 MOSFETs and 39 nodes.

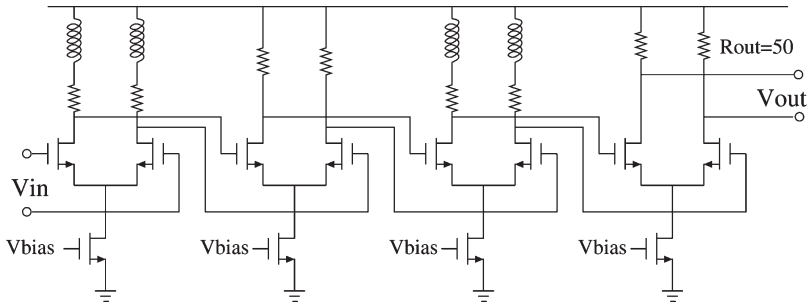


Fig. 8. Tapered CML buffer.

and $x_2(t) = V_2 z_2(t)$ in (15), formulate the equivalent LTI system in companion form as (16), and reduce it using the Krylov-subspace projection via Lanzcos or Arnoldi process with multiple starting vectors (e.g., [5]).

The benefit of using the POD basis is twofold. When applying the POD to the time-domain data, one can choose the input to the LTI system of (9) the same as the training input to the nonlinear system. Due to the "near optimal" property of the POD basis, it can effectively capture the LTI system dynamics under such an excitation. When generating the POD basis from the frequency-domain data, it boils down to the Poor Man's TBR (PMTBR) method [8], where it is shown that the POD basis converges quickly to the dominant eigenvectors of the controllability Gramians of the underlying LTI system. It may also be interpreted as a multipoint moment-matching method, which will match one moment at each frequency [52]. In fact, moment-matching is only one of the desirable properties to be preserved in the macromodels, and the macromodels generated with the Krylov subspace is not necessary to be optimal. In many cases, the POD basis (or PMTBR) can lead to a better reduced model in terms of accuracy and compactness [8]. Either way, it has the potential of using less dimension of V_1, without sacrificing too much accuracy, to achieve a compact macromodel for the weakly nonlinear system.

To conclude this section, we summarize the MPI method as follows.

Input: system equations of (8).
Output: projection basis V.

1) Get the data ensemble $\bar{X}_1$ either from the time or frequency domain.
2) Generate a POD basis by $V_1 = \text{svd}(\bar{X}_1)$. Keep the dominant singular vectors.
3) Let $B_2 = A_2(V_1 \otimes V_1)$, and form the equivalent starting vectors $B_{\text{eq}} = [B, B_2]$.
4) Generate a qth-order Krylov subspace using B_{eq}. colspan$\{V\} = K_q(A_1^{-1}E, A_1^{-1}B_{\text{eq}})$.

V. Validation of PWP

In this section, we conduct in-depth evaluations of the PWP method using three examples. For each circuit, a PWP macromodel is first generated, followed by variant macromodel-based simulations. The results are compared against the full simulations for validation purposes. The PWP-generated macromodel is further embedded in a larger system to demonstrate its capability of accelerating system-level simulations. Details of model generation and speedup numbers are provided at the end of this section.

A. Examples

1) Op-Amp: The first example is a current-mirror op-amp (Fig. 7) with 50 MOSFETs and 39 nodes, including a common-mode feedback block. It was designed to provide about 70 dB of dc gain, with a slew rate of 20 V/μs and an open-loop 3-dB bandwidth of $f_0 \approx 10$ kHz.

2) Current-Mode-Logic (CML) Buffer: The second example in Fig. 8 is a tapered CML buffer chain that is designed to drive a 50-Ω transmission line in high-speed digital communications [53]. Inductive peaking is employed in the first and third stages to increase the bandwidth. The sizing of each stage and the

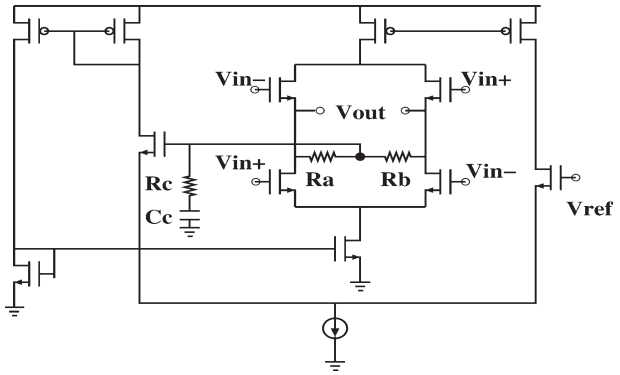


Fig. 9. LVDS buffer with a common-mode feedback loop.

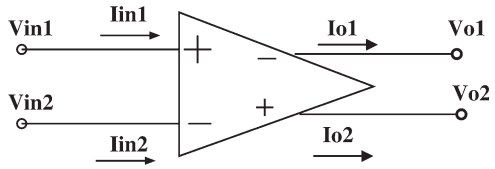


Fig. 10. PWP-model generation for the op-amp.

parameters are optimized to minimize the buffer delay [54]. The circuit size of this example is 28, and $V_{\rm dd} = 1.8$ V.

3) Low-Voltage Differential-Signaling (LVDS) Buffer: The third example in Fig. 9 is an LVDS output buffer ($V_{\rm dd} = 3.3$ V) with a common-mode feedback loop [55], which is also commonly used in digital communications. The common-mode voltage of inputs is enforced by $V_{\rm ref}$ to be around 1.25 V. It is designed to drive a 50-Ω with about 0.5-V voltage swing. The size of the circuit is 18.

For all the aforementioned examples, the MOS devices were modeled using a smooth bulk-referred version of the Schichman–Hodges (MOS Level 1) equations, plus considering the channel-length-modulation effect. It should be noted that the PWP-generated macromodels automatically abstract relevant features of all underlying device models in the original circuit, no matter how simple or complex they are. Finally, all circuit simulations and verifications represent apple-to-apple comparisons in a MATLAB prototyping environment running on a 1.8-GHz Pentium-4 Linux box.

B. PWP-Model Generation

1) Op-Amp: The PWP model of the op-amp was generated with four inputs and four outputs, as shown in Fig. 10. Besides the original two inputs ($V_{\rm in1}$ and $V_{\rm in2}$) and two outputs ($V_{\rm o1}$ and $V_{\rm o2}$), another two inputs ($I_{\rm o1}$ and $I_{\rm o2}$) and two outputs ($I_{\rm in1}$ and $I_{\rm in2}$) were added to capture bidirectional loading effects, such that the generated macromodel can be used as drop-in replacement and simulated with peripheral circuits.

As mentioned in Section III, the expansion points were chosen along the trajectory with certain training input. The choice of training input was dictated by a desire to exercise the circuit through all its important nonlinear and dynamical behaviors. In this test, we obtain multiple trajectories using transient simulation with step function and several sinusoidal inputs (amplitude varying from 10^{-6} to 10^{-1} and frequency varying from 10^2 to 10^5) as well as some dc sweeps of the full circuit.

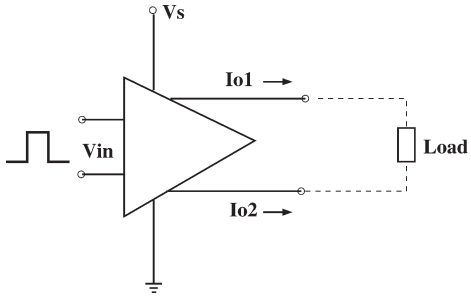


Fig. 11. PWP-model generation for the IO buffer circuits.

TABLE I
MACROMODEL SIZE AND GENERATION TIME FOR THE OP-AMP AND BUFFER CIRCUITS

examples	original size	reduced size	# of regions	gen. time [s]
op-amp	39	24	47	411
CML	28	15	32	610
LVDS	18	11	21	400

Each individual polynomial is reduced to size 12 with the MPI method. These projection bases are then combined, and a common subspace with a size of 24 is obtained via SVD. Eventually, the PWP-generated macromodel has 47 piecewise regions, each of which is approximated by a polynomial model with a state space of size 24.

2) Buffers: For buffer circuits in digital applications, we are primarily interested in their switching activities of the buffers with large signal inputs, which are dominated by the coverage of piecewise regions and the smoothing function. For such cases, weak nonlinearities captured by the polynomials inside each region are not as important as in other applications (e.g., op-amps and mixers). Through experimentation, we have found that using the linear-only models within each region is adequate for meeting the accuracy requirements.[4] The weight function (14) and the merging of multiple training trajectories, as described in Section III-B, are both very important for developing macromodels that work well in large-signal transient analysis.

Fig. 11 shows the block diagram for the macromodel generation of buffer circuits in Figs. 8 and 9. The buffer is modeled with five inputs and two outputs: Two differential inputs track different input patterns, two loading currents tackle loading variations, and power grid noise is captured via port V_s. Two differential outputs are connected to the load. Several transient simulations of the full buffer circuit with an input pattern of "010" and different loads (e.g., 50-Ω resistor and 1-pF capacitor) are used to generate the training trajectories, along which the piecewise regions are selected and merged.

Finally, the size and the macromodel generation time of three examples are summarized in Table I.

[4]Being able to leave out the polynomial terms significantly improves the macromodel's efficiency.

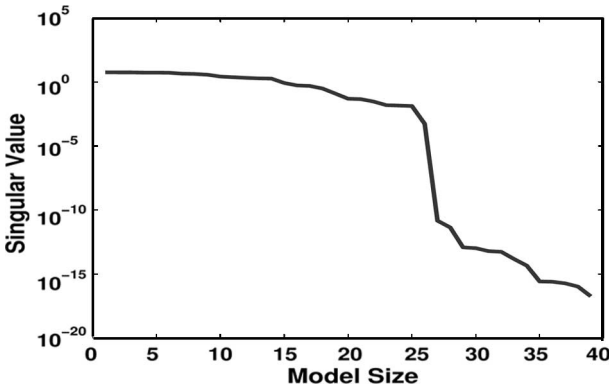


Fig. 12. Singular value of common subspace.

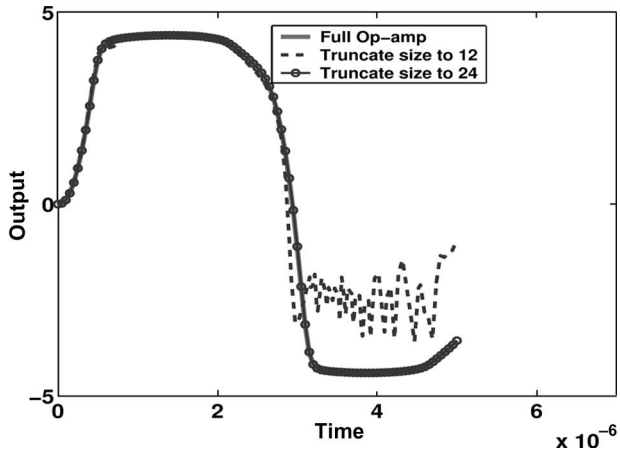


Fig. 13. Transient result of the PWP-generated macromodels with different model sizes.

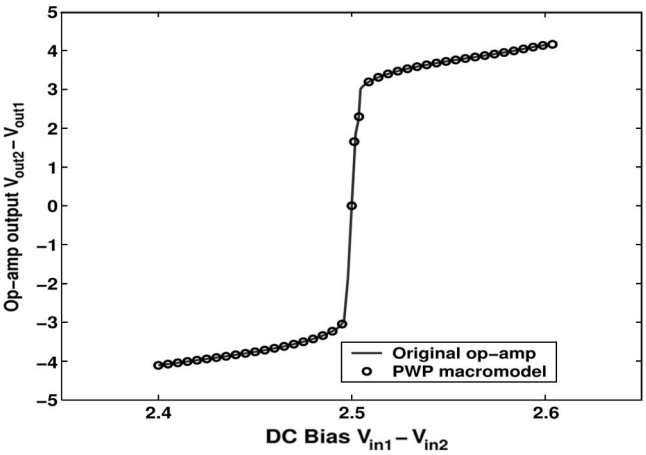


Fig. 14. DC sweep of the op-amp.

C. Importance of Accurate Common Subspace

It is important to choose the dimension of a common subspace according to the singular value drop. We validate this using the op-amp example.

We generated 32 models along a trajectory with a sinusoidal training input. For each region, the model size was reduced from 39 to 8 by the MPI method, i.e., $V_i \in \mathbb{R}^{39\times 8}$ for $i = 1, \ldots, 32$. Fig. 12 shows the singular value of the collections of $V = [V_1, \ldots, V_{32}]$. It is seen that there is a cut at size 24 (or 25). The insight is that even if the original system could have a large number of unknowns, they are partially correlated to each other, and the intrinsic freedom is limited. Therefore, it is possible to project the system into a subspace that effectively spans the solution space.

It is important to identify the correct size of the common subspace. To see the problem, we run a comparison test on the PWP-generated macromodels with sizes 12 (overcut) and 24, as shown in Fig. 13. It is seen that the size-24 model matches the original model very well. However, the size-12 macromodel with an overcutting subspace has large errors, mainly in the second half of the period. This is because the overcutting subspace excludes the critical information of those regions presented in the latter half of the training trajectory. Meanwhile, since the model is not accurate, it has converged problem during the simulation that makes it much slower than the size-24 model. In practice, one can detect the dramatic change of the singular value at runtime to determine the proper size of the common subspace.

D. Op-Amp: DC and AC Analyses

We first perform the dc-sweep analysis to the open-loop configuration of the op-amp, and part of the dc operating points are used to generate the final PWP macromodel. We then compare the results of the full op-amp with that of the PWP-generated macromodel. As shown in Fig. 14, two models are precisely matched.

Next, we compare Bode plots, which are obtained by the ac analysis, of the PWP-generated macromodel against those of the full op-amp. Two ac sweeps, which are obtained at different dc biases, are shown in Fig. 15. The PWP also provides excellent matches around each bias point.

E. Op-Amp: Distortion Via HB Simulations

When the op-amp is used as a linear amplifier with small inputs, distortion and intermodulation are important performance metrics. One of the strengths of the PWP-generated macromodels is that weak nonlinearities, which are responsible for distortion and intermodulation, are captured well. Such weakly nonlinear effects are best simulated using the frequency-domain HB analysis, for which we choose the one-tone sinusoidal input $V_{\text{in1}} - V_{\text{in2}} = A\sin(2\pi \times 100t)$ and $C_{\text{load}} = 10$ pF. The input magnitude A is swept over several decades to verify the valid range of macromodel, and the first two harmonics are shown in Fig. 16.

It can be seen that for the entire input range, there is an excellent match of the distortion component from the macromodel versus that of the full circuit (at very small input magnitudes, the distortion component of both is dominated by numerical noise). Note that the same macromodel is used for this HB simulation as for all the other analyses presented. The CPU time and the speedups are shown later in Table II.

F. Op-Amp: Slewing/Clipping Via Transient Simulations

Another strength of PWP is that it can capture the effects of strong nonlinearities excited by large signal swings. To

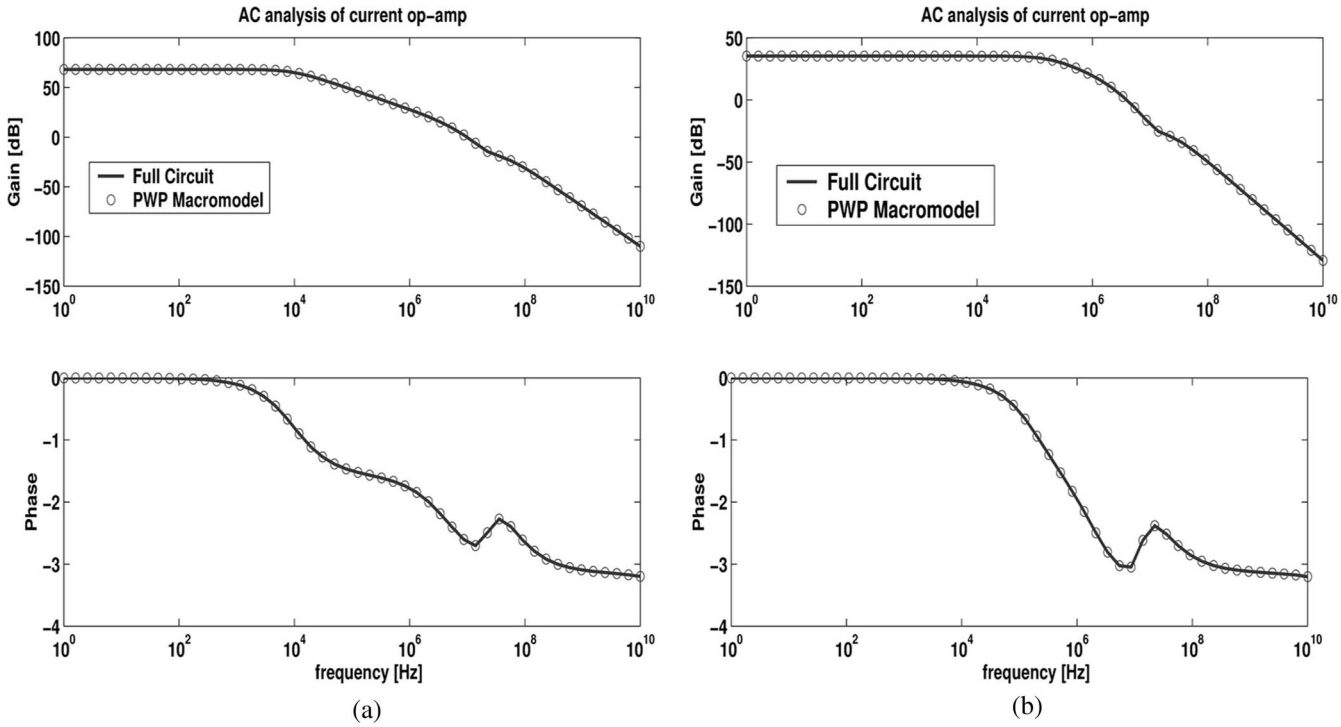


Fig. 15. AC analysis with different dc biases. (a) $V_{\rm in1} = V_{\rm in2} = 2.5$ V. (b) $V_{\rm in1} = V_{\rm in2} = 2.0$ V.

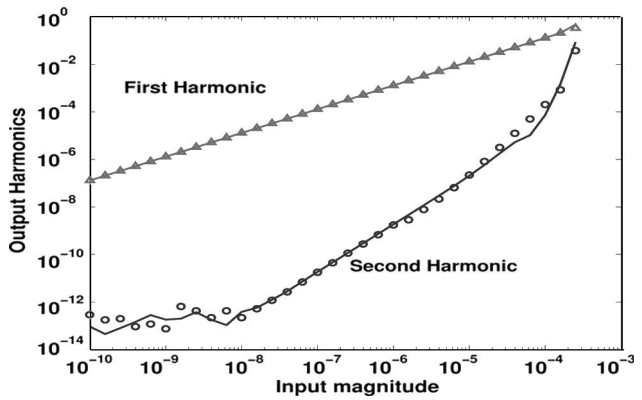


Fig. 16. Harmonic analysis of the current-mirror op-amp. Solid line—full op-amp; discrete point—PWP model.

TABLE II
MACROMODEL SIMULATION TIME AND SPEEDUPS

example	analyses	full [s]	PWP [s]	speedup
op-amp	Harmonic Balance	125.9	15.58	8.1
	transient with clipping	351.7	39.05	9
	transient with feedback	791	102	7.1
CML	transient with diff. loadings	885.82	112.3	8
	transient with crosstalks	1215.9	138.1	8.8
LVDS	transient with SSN	612.7	80.2	7.6

demonstrate this, a transient analysis was run with a large fast step input, and the comparisons of output are shown in Fig. 17.

The slope of the step input was chosen to excite slew-rate limiting, which is a dynamical phenomenon caused by strong nonlinearities (saturation of differential amplifier structures).

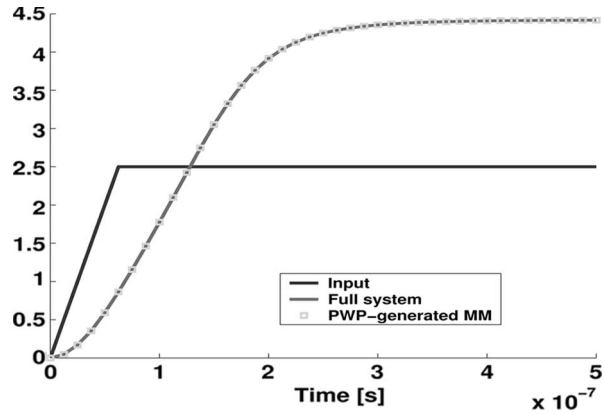


Fig. 17. Transient analysis of the current-mirror op-amp with fast step input.

To illustrate the clipping due to the power and ground rails, another transient simulation was run with large input

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

Comparisons of the macromodel versus the original are shown in Fig. 18. The CPU time and the speedup number are listed in Table II.

G. Op-Amp: Embedded in Negative Feedback Loop

The main purpose of generating macromodels is to use them as drop-in replacement to speed up simulation with other circuits. To illustrate this idea, we embedded the op-amp in a negative feedback loop, as shown in Fig. 19.

A transient-simulation result with large sinusoidal input $V_{\rm in1} - V_{\rm in2} = 4\sin(2\pi 10^6 t)$ is shown in Fig. 20. The

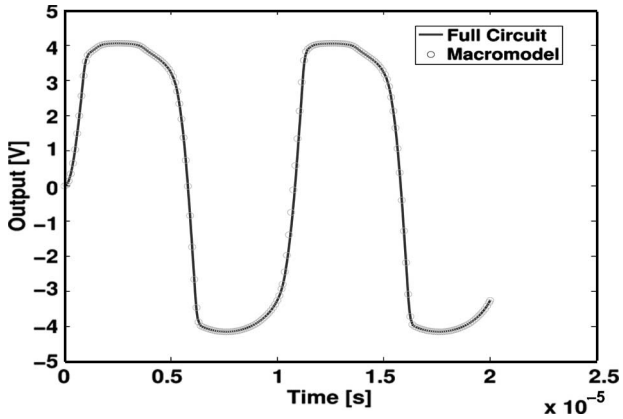


Fig. 18. Transient analysis of the current-mirror op-amp with large sinusoidal input.

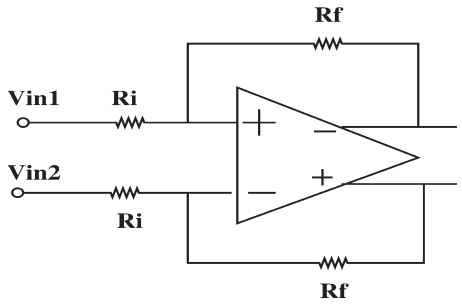


Fig. 19. Op-amp embedded in the negative feedback loop, $R_i/R_f = 10\,\mathrm{K}/1\,\mathrm{K}$.

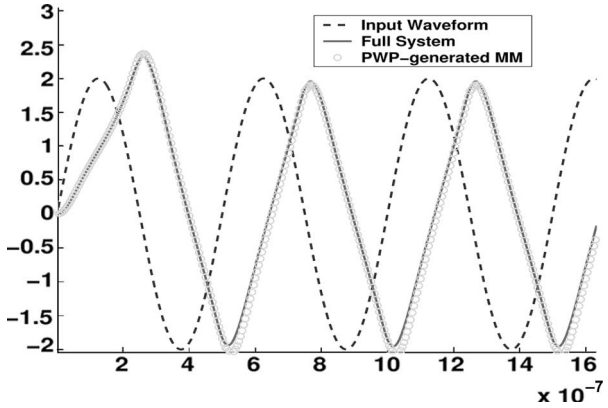


Fig. 20. Large sinusoidal transient simulation of the op-amp in a feedback loop, revealing slewing effects.

magnitude and the frequency of input signal are chosen such that the op-amp presents a slewing effect on its output. It was observed that the PWP-generated macromodel accurately captures this strong nonlinear effect. In this test, the original system takes 791 s in the transient analysis, whereas the macromodel-based simulation takes 102 s, which results in about 7.7× speedup.

H. CML Buffer: Different Loading Effects

We verify the capturing of different loading effects using the macromodel from the second example (CML buffer). Three transmission lines (modeled with lumped RLC network) are connected to the buffer in the test. The voltage waveforms across the load at the far end of the transmission line against the full circuit simulation are shown in Fig. 21. The three cases are the following:

1) lossless transmission line: $Z_\mathrm{c} = 75\ \Omega$, $T_\mathrm{d} = 0.4$ ns, $Z_\mathrm{load} = Z_\mathrm{c}$, and input pattern "0100101;"
2) lossy transmission line: $Z_\mathrm{c} = 100\,\Omega$, $T_\mathrm{d} = 0.5$ ns, $Z_\mathrm{dc} = 2\,\Omega$, $Z_\mathrm{load} = Z_\mathrm{c}$, and input pattern "0101100 transmission;"
3) lossy line: $Z_\mathrm{c} = 75\ \Omega$, $T_\mathrm{d} = 0.5$ ns, $Z_\mathrm{dc} = 2\ \Omega$, $Z_\mathrm{load} =$ 1 pF, and input pattern "0110010."

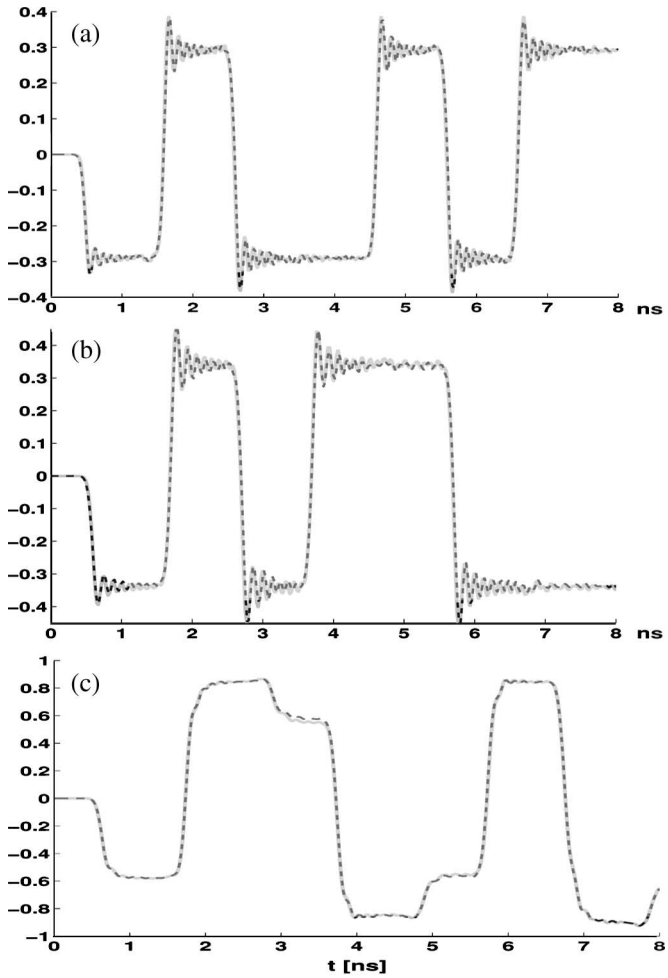


Fig. 21. Voltage waveform across the load. Solid line: Full circuit simulation; dashed line: Macromodel simulation.

It is seen that the macromodel is capable of capturing different loading effects, and its accuracy in matching the full circuit simulation is more than adequate. The relative error is less than 5% on average. The runtime comparison is shown later in Table II.

I. CML Buffer: Crosstalk

We further investigate the CML buffer macromodel for crosstalk simulation. As shown in Fig. 22, two coupled lossy transmission lines ($Z_\mathrm{c} = 75\ \Omega$, $T_\mathrm{d} = 0.5$ ns, and $Z_\mathrm{dc} = 2\ \Omega$) are driven by two buffers: One is active with an input pattern of "0101100," and the other remains quiet.

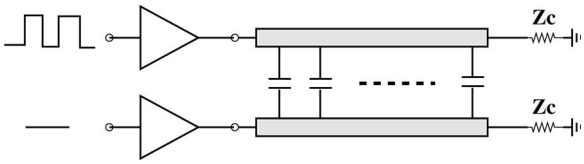


Fig. 22. Circuit for the crosstalk simulation.

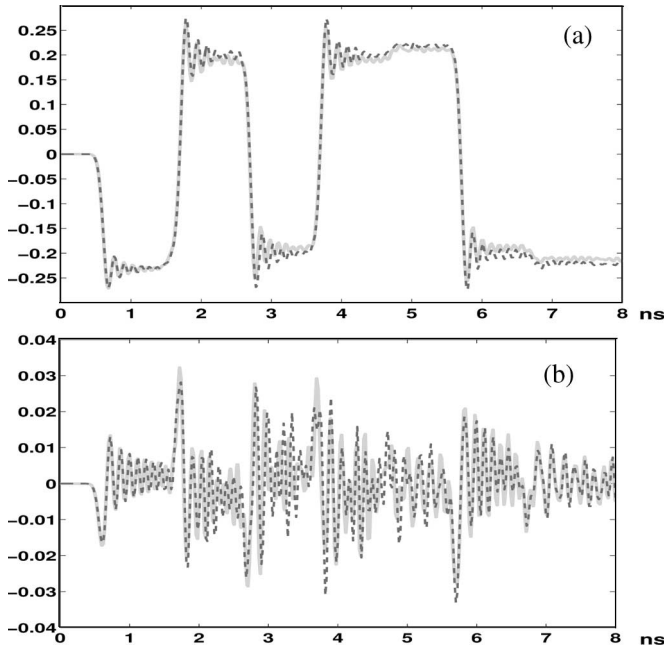


Fig. 23. Macromodel in the crosstalk simulation. Solid line: Full circuit; dashed line: Macromodel. (a) Voltage across the load on the active line. (b) Voltage across the load on the quiet line.

The voltage waveforms on the load impedance at the far end of both lines are shown in Fig. 23. It is seen that the macromodel reproduces the dynamic behaviors of the buffer and captures the crosstalk noise quite well.

The runtime comparison is shown later in Table II.

J. LVDS Buffer: Simultaneous Switching Noise (SSN)

The macromodel of the third example (LVDS buffer in Fig. 9) is used in this test. As shown in Fig. 24, M identical drivers are loaded with lossy transmission line ($Z_{\rm c} = 100, T_{\rm d} = 0.5$ ns, and $Z_{\rm dc} = 2\,\Omega$). An ideal power supply $V_{\rm dd}$ is connected to the power supply port V_s of drivers through L_s and R_s. In the simulation, $M = 7$, $L_s = 0.1$ nH, and $R_s = 1$ mΩ. All drivers have the same input stream "0100101."

The simulation results, as shown in Fig. 25, confirm that the macromodel accurately captures the sensitive SSN noise in both the voltage and current waveforms.

Finally, we summarize the speedup results for all test cases in Table II. It is evident from the aforementioned three examples that the PWP-generated macromodels can be profitably used as general-purpose drop-in replacements with various analysis, resulting in an order of speedups with little loss of accuracy. The speedups are mainly due to the two factors: the reduced system size and the simple model evaluations. Therefore, more attractive speedups can be expected for large circuits with complex transistor models. The generated macromodels are easily targeted to a variety of model-description languages, including MATLAB/Simulink blocks [19]–[22], Verilog-A, VHDL-AMS, and SPICE subcircuits [23], [26].

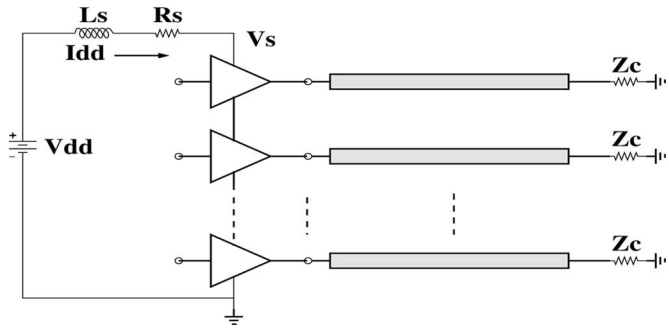


Fig. 24. SSN validation.

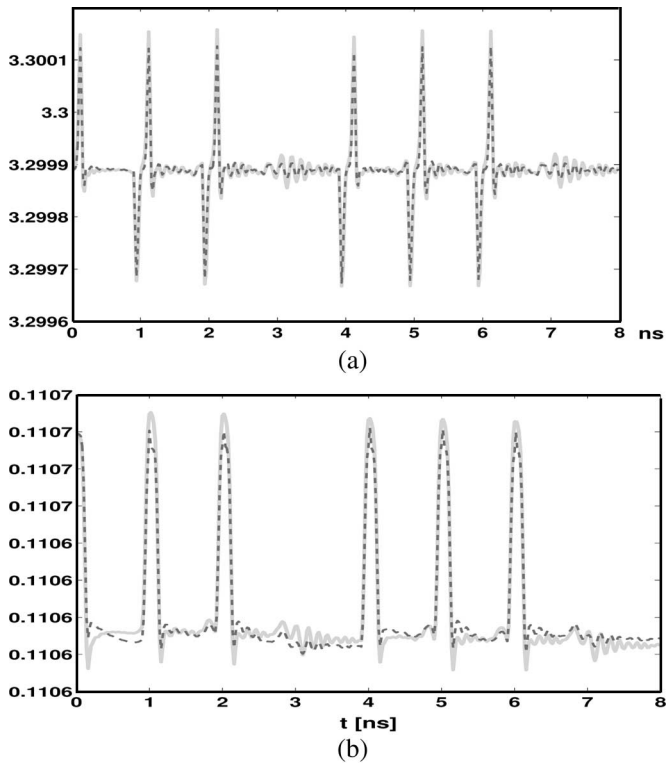


Fig. 25. SSN using the macromodel of the LVDS buffer. Solid line: Full circuit; dashed line: Macromodel. (a) Voltage waveform at node V_s. (b) Noisy supply current $I_d d$.

VI. Conclusion

We have presented a PWP approach for a general-purpose nonlinear model reduction. Our approach draws inspiration from and improves upon the previous work in [16], [19], and [20]. It combines good global and local accuracy properties, thereby making the reduced models suitable for both the large-signal transient analysis and the small-signal distortion analysis. Numerical results confirm these expectations quantitatively. We have also developed a reliable and easily implemented weakly polynomial model-reduction technique,

the MPI method, which combines the POD and the Krylov subspace to generate a proper projection basis. The PWP has a considerable potential as an accelerator for the system-level simulations with large individual blocks.

REFERENCES

[1] Accellera Verilog-AMS Group, [Online]. Available: http://www.eda.org/verilog-ams/.

[2] L. Pillage and R. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 9, no. 4, pp. 352–366, Apr. 1990.

[3] P. Feldmann and R. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 14, no. 5, pp. 639–649, May 1995.

[4] R. Freund, "Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation," Bell Laboratories, Murray Hill, NJ, Tech. Rep. 11273-980217-02TM, 1998.

[5] R. W. Freund, "Krylov-subspace methods for reduced-order modeling in circuit simulation," *J. Comput. Appl. Math.*, vol. 123, no. 1/2, pp. 395–421, Nov. 2000.

[6] A. Odabasioglu, M. Celik, and L. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 1997, pp. 58–65.

[7] J. R. Phillips, L. Daniel, and M. Silveira, "Guaranteed passive balancing transformations for model order reduction," in *Proc. IEEE Des. Autom. Conf.*, 2002, pp. 52–57.

[8] J. Phillips and L. M. Silveira, "Poor man's TBR: A simple model reduction scheme," in *Proc. Des., Autom. Test Eur. Conf. Exhib.*, 2004, pp. 938–943.

[9] J.-R. Li, F. Wang, and J. White, "An efficient Lyapunov equation-based approach for generating reduced-order models of interconnect," in *Proc. IEEE Des. Autom. Conf.*, 1999, pp. 1–6.

[10] J. Roychowdhury, "Reduced-order modelling of time-varying systems," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 10, pp. 1273–1288, Nov. 1999.

[11] J. Phillips, "Model reduction of time-varying linear systems using approximate multipoint Krylov-subspace projectors," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 1998, pp. 96–102.

[12] J. Roychowdhury, "Reduced-order modelling of linear time-varying systems," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 1998, pp. 92–95.

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

[14] J. R. Phillips, "Automated extraction of nonlinear circuit macromodels," in *Proc. IEEE Custom Integr. Circuits Conf.*, 2000, pp. 451–454.

[15] P. Li and L. T. Pileggi, "NORM: Compact model order reduction of weakly nonlinear systems," in *Proc. IEEE Des. Autom. Conf.*, 2003, pp. 472–477.

[16] M. Rewienski and J. White, "A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 2001, pp. 252–257.

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

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

[19] N. Dong and J. Roychowdhury, "Piecewise polynomial nonlinear model reduction," in *Proc. IEEE Des. Autom. Conf.*, 2003, pp. 484–489.

[20] N. Dong and J. Roychowdhury, "Automated extraction of broadly applicable nonlinear analog macromodels from SPICE-level descriptions," in *Proc. IEEE Custom Integr. Circuits Conf.*, 2004, pp. 117–120.

[21] N. Dong and J. Roychowdhury, "Automated nonlinear macromodelling of output buffers for high-speed digital applications," in *Proc. IEEE Des. Autom. Conf.*, 2005, pp. 51–56.

[22] S. Dabas, N. Dong, and J. Roychowdhury, "Automated extraction of accurate delay/timing macromodels of digital gates and latches using trajectory piecewise methods," in *Proc. IEEE Asia South Pacific Des. Autom. Conf.*, 2007, pp. 361–366.

[23] S. K. Tiwary and R. A. Rutenbar, "Scalable trajectory methods for on-demand analog macromodel extraction," in *Proc. IEEE Des. Autom. Conf.*, 2005, pp. 403–408.

[24] S. Tiwary and R. A. Rutenbar, "On-the-fly fidelity assessment for trajectory-based circuit macromodels," in *Proc. IEEE Custom Integr. Circuits Conf.*, 2006, pp. 185–188.

[25] H. Liu, A. Singhee, R. Rutenbar, and L. Carley, "Remembrance of circuits past: Macromodeling by data mining in large analog design spaces," in *Proc. IEEE Des. Autom. Conf.*, 2002, pp. 437–442.

[26] S. K. Tiwary and R. A. Rutenbar, "Faster, parametric trajectory-based macromodels via localized linear reductions," in *Proc. Int. Conf. Comput.-Aided Des.*, 2006, pp. 876–883.

[27] X. Ren and T. J. Kazmierski, "Behavioral-level performance modeling of analog and mixed-signal systems using support vector machines," in *Proc. IEEE Int. Behavioral Model. Simul. Conf.*, 2006, pp. 28–33.

[28] M. Ding and R. Vemuri, "A combined feasibility and performance macromodel for analog circuits," in *Proc. IEEE Des. Autom. Conf.*, 2005, pp. 63–68.

[29] T. Kiely and G. Gielen, "Performance modeling of analog integrated circuits using least-squares support vector machines," in *Proc. Des., Autom. Test Eur. Conf. Exhib.*, 2004, pp. 448–453.

[30] Z. Bai, P. M. Dewilde, and R. W. Freund, "Reduced-order modeling," Bell Lab., Murray Hill, NJ, Tech. Rep., 02-4-13, Mar. 2002.

[31] Z. Bai and D. Skoogh, "Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems," *Appl. Numer. Math.*, vol. 43, no. 1/2, pp. 9–44, Oct. 2002.

[32] M. Kamon, F. Wang, and J. White, "Generating nearly optimally compact models from Krylov-subspace based reduced-order models," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 4, pp. 239–248, Apr. 2000.

[33] P. Feldmann and R. Freund, "Circuit noise evaluation by Pade approximation based model-reduction techniques," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 1997, pp. 132–138.

[34] I. Elfadel and D. Ling, "A block rational Arnoldi algorithm for multipoint passive model-order reduction of multiport RLC networks," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 1997, pp. 66–71.

[35] I. Jaimoukha, "A general minimal residual Krylov subspace method for large-scale model reduction," *IEEE Trans. Autom. Control*, vol. 42, no. 10, pp. 1422–1427, Oct. 1997.

[36] L. M. Silveira, M. Kamon, and J. White, "Efficient reduced-order modeling of frequency-dependent coupling inductances associated with 3-D interconnect structures," in *Proc. IEEE Des. Autom. Conf.*, Jun. 1995, pp. 376–380.

[37] L. Daniel, A. Sangiovanni-Vincentelli, and J. White, "Techniques for including dielectrics when extracting passive low-order models of high speed interconnect," in *Proc. Int. Conf. Comput.-Aided Des.*, 2001, pp. 240–244.

[38] L. Daniel, C. S. Ong, S. C. Low, K. H. Lee, and J. White, "Geometrically parameterized interconnect performance models for interconnect synthesis," in *Proc. Int. Symp. Phys. Des.*, 2002, pp. 202–207.

[39] L. Daniel, O. C. Siong, L. S. Chay, K. H. Lee, and J. White, "A multiparameter moment matching model reduction approach for generating geometrically parameterized interconnect performance models," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 5, pp. 678–693, May 2004.

[40] I. Balk, "On a passivity of the Arnoldi based model order reduction for full-wave electromagnetic modeling," *IEEE Trans. Adv. Packag.*, vol. 24, no. 3, pp. 304–308, Aug. 2001.

[41] L. Daniel and J. R. Phillips, "Model order reduction for strictly passive and causal distributed systems," in *Proc. IEEE Des. Autom. Conf.*, 2002, pp. 46–51.

[42] E. Grimme, "Krylov projection methods for model reduction," Ph.D. dissertation, EE Dept., Univ. Illinois, Urbana–Champaign, IL, 1997.

[43] H. Banks, R. del Rosario, and H. Tran, "Proper orthogonal decomposition-based control of transverse beam vibrations: Experimental implementation," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 5, pp. 717–726, Sep. 2000.

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

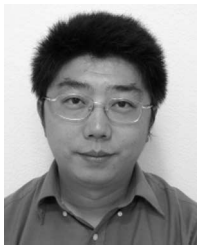
[45] M. Rathinam and L. R. Petzold, "A new look at proper orthogonal decomposition," *SIAM J. Numer. Anal.*, vol. 41, no. 5, pp. 1893–1925, 2003.

[46] A. Noor, "Recent advances in reduction methods for nonlinear problems," *Comput. Struct.*, vol. 13, no. 1–3, pp. 31–44, Jun. 1981.

[47] G. Kepler, H. Tran, and H. Banks, "Reduced order model compensator control of a species transport in a CVD reactor," *Optim. Control Appl. Methods*, vol. 21, no. 4, pp. 143–160, 2000.

[48] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 4, no. 3, pp. 519–524, Mar. 1987.

[49] K. Willcox, J. Peraire, and J. White, "An Arnoldi approach for generation of reduced-order models for turbomachinery," *Comput. Fluids*, vol. 31, no. 3, pp. 369–389, Mar. 2002.
[50] Y. Liang, H. Lee, S. Lim, W. Lin, and K. Lee, "Proper orthogonal decomposition and its applications—Part I: Theory," *J. Sound Vib.*, vol. 252, no. 3, pp. 527–544, May 2002.
[51] W. Rugh, *Nonlinear System Theory—The Volterra–Wiener Approach*. Baltimore, MD: Johns Hopkins Univ. Press, 1981.
[52] S. Mijalković, "Using frequency response coherent structures for model-order reduction in microwave applications," *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 9, pp. 2292–2297, Sep. 2004.
[53] J. Savoj and B. Razavi, "A 10-Gb/s CMOS clock and data recovery circuit with a half-rate linear phase detector," *IEEE J. Solid-State Circuits*, vol. 36, no. 5, pp. 761–768, May 2001.
[54] P. Heydari, "Design and analysis of low-voltage current-mode logic buffers," in *Proc. 4th Int. Symp. Quality Electron. Des.*, 2003, pp. 293–298.
[55] A. Boni, A. Pierazzi, and D. Vecchi, "LVDS I/O Interface for Gb/s-per-pin operation in 0.35-μm CMOS," *IEEE J. Solid-State Circuits*, vol. 36, no. 4, pp. 706–711, Apr. 2001.



Ning Dong (S'03) received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1997 and 2000, respectively, and the Ph.D. degree in electrical engineering from the University of Minnesota, Minneapolis, in 2006.

He is currently with Texas Instruments Incorporated, Dallas. His research interests include circuit- and system-level analyses, automated nonlinear macromodeling, and simulation of analog, RF, and mixed-signal systems.

Jaijeet Roychowdhury (S'85–M'87–SM'06) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1987, and the Ph.D. degree in electrical engineering and computer science from the University of California, Berkeley, in 1993.

From 1993 to 1995, he was with the Computer-Aided Design (CAD) Laboratory, AT&T Bell Laboratories, Allentown, PA. From 1995 to 2000, he was with the Communication Sciences Research Division, Bell Laboratories, Murray Hill, NJ. From 2000 to 2001, he was with CeLight Inc. (an optical networking startup), Silver Spring, MD. Since 2001, he has been with the Department of Electrical and Computer Engineering and the Digital Technology Center, University of Minnesota, Minneapolis. Over the years, he has authored or coauthored five best or distinguished papers at ASP-DAC, DAC, and ICCAD. He is the holder of ten patents. His professional interests include the design, analysis, and simulation of electronic, electrooptical, and mixed-domain systems, particularly for high-speed and high-frequency communication circuits.

Dr. Roychowdhury was cited for Extraordinary Achievement by Bell Laboratories in 1996. He was an IEEE Circuits and Systems Society Distinguished Lecturer from 2003 to 2005 and served as a Program Chair of IEEE's CANDE and BMAS workshops in 2005. Currently, he serves on the Technical Program Committees of DAC, DATE, ASP-DAC, ISQED, and BMAS, on the Executive Committee of ICCAD, on the Nominations and Appointments Committee of CEDA, and as a Treasurer of CANDE.