

A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices

Michał Rewieński and Jacob White, *Associate Member, IEEE*

Abstract—In this paper, we present an approach to nonlinear model reduction based on representing a nonlinear system with a piecewise-linear system and then reducing each of the pieces with a Krylov projection. However, rather than approximating the individual components as piecewise linear and then composing hundreds of components to make a system with exponentially many different linear regions, we instead generate a small set of linearizations about the state trajectory which is the response to a “training input.” Computational results and performance data are presented for an example of a micromachined switch and selected nonlinear circuits. These examples demonstrate that the macromodels obtained with the proposed reduction algorithm are significantly more accurate than models obtained with linear or recently developed quadratic reduction techniques. Also, we propose a procedure for *a posteriori* estimation of the simulation error, which may be used to determine the accuracy of the extracted trajectory piecewise-linear reduced-order models. Finally, it is shown that the proposed model order reduction technique is computationally inexpensive, and that the models can be constructed “on the fly,” to accelerate simulation of the system response.

Index Terms—Microelectromechanical systems (MEMS), model order reduction, nonlinear analog circuits, nonlinear dynamical systems, piecewise-linear models.

I. INTRODUCTION

INTEGRATED circuit fabrication facilities are now offering digital system designers the ability to integrate analog circuitry and micromachined devices, but such mixed-technology microsystems are extremely difficult to design because of the limited verification and optimization tools available. In particular, there are no generally effective techniques for automatically generating reduced-order system-level models from detailed simulation of the analog and micromachined blocks. Research over the past decade on automatic model reduction has led to enormous progress in strategies for linear problems, such as the electrical problems associated with interconnect and packaging, but these techniques have been difficult to extend to the nonlinear problems associated with analog circuits and micromachined devices.

Manuscript received May 31, 2002; revised September 3, 2002. This work was supported in part by the Defense Advanced Research Projects Agency Neocad program, the National Science Foundation, and by grants from the Singapore-MIT Alliance. This paper was recommended by Guest Editor G. Gielen.

The authors are with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: mrewiens@mit.edu; white@mit.edu).

Digital Object Identifier 10.1109/TCAD.2002.806601

In this paper, we present an approach to nonlinear model reduction based on representing a nonlinear system with a piecewise-linear system and then reducing each of the pieces with Krylov subspace projection methods. However, rather than approximating the individual components as piecewise-linear and then composing hundreds of components to make a system with exponentially many different linear regions, we instead generate a small set of linearizations about the state trajectory which is the response to a “training input.” At first glance, such an approach would seem to work only when all the inputs are very close to the training input, but as examples will show, this is not the case. In fact, the method easily outperforms recently developed techniques based on quadratic reduction.

We start in the next section by describing examples of nonlinear circuits and a micromachined switch, to make clear the nonlinear model reduction problem, and then in Section III, we describe the existing nonlinear reduction techniques in a more abstract setting. In Section IV, we present the trajectory-based piecewise-linear model order reduction (MOR) strategy, an approach for accelerating the needed simulation, and a procedure for *a posteriori* error estimation. Section V discusses a fast simulation technique emerging from the proposed MOR strategy. Computational results are examined in Section VI, and in Section VII, we present our conclusions.

II. EXAMPLES OF NONLINEAR DYNAMICAL SYSTEMS

A large class of nonlinear dynamical systems may be described using the following state-space approach:

$$\begin{cases} \frac{dg(x(t))}{dt} = f(x(t)) + B(x(t))u(t) \\ y(t) = C^T x(t) \end{cases} \quad (1)$$

where $x(t) \in R^N$ is a vector of states at time t , $f: R^N \rightarrow R^N$ and $g: R^N \rightarrow R^N$ are nonlinear vector-valued functions, B is a state-dependent $N \times M$ input matrix, $u: R \rightarrow R^M$ is an input signal, C is an $N \times K$ output matrix and $y: R \rightarrow R^K$ is the output signal.

In this paper, we will focus on three distinct examples of nonlinear systems which may be described by (1) and, due to their highly nonlinear dynamical behavior, illustrate well the challenges associated with nonlinear MOR.

The first example, considered by Chen *et al.* [1], is a nonlinear transmission line circuit model shown in Fig. 1. The circuit consists of resistors, capacitors, and diodes with a constitutive

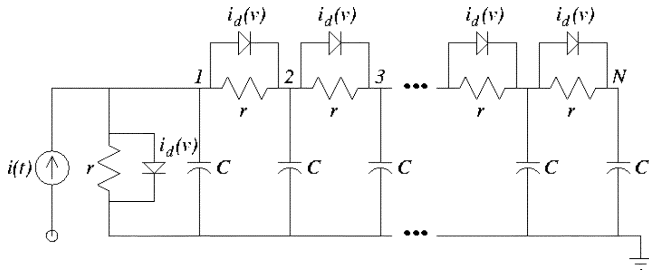


Fig. 1. Example of a nonlinear transmission line circuit model.

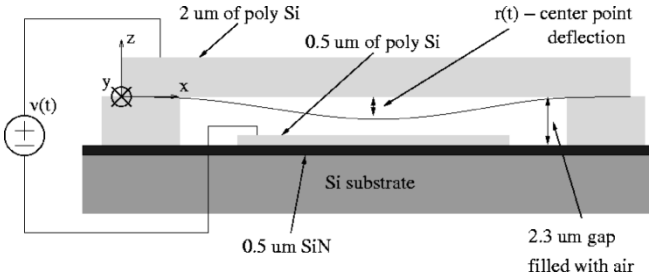


Fig. 2. Micromachined switch (following Hung *et al.* [10]).

equation $i_d(v) = \exp(40v) - 1$.¹ For simplicity, we assume that all the resistors and capacitors have unit resistance and capacitance, respectively, ($r = 1$, $C = 1$). In this case, the input is the current source entering node 1: $u(t) = i(t)$ and the (single) output is chosen to be the voltage at node 1: $y(t) = v_1(t)$.

The second example is a micromachined switch (fixed-fixed beam) shown in Fig. 2. Following Hung *et al.* [10], the dynamical behavior of this coupled electromechanical-fluid system can be modeled with the one-dimensional Euler's beam equation and two-dimensional Reynolds' squeeze film damping equation given below:

$$\hat{E}I \frac{\partial^4 u}{\partial x^4} - S \frac{\partial^2 u}{\partial x^2} = F_{\text{elec}} + \int_0^w (p - p_a) dy - \rho \frac{\partial^2 u}{\partial t^2} \quad (2)$$

$$\nabla \cdot ((1 + 6K)u^3 p \nabla p) = 12\mu \frac{\partial(pu)}{\partial t} \quad (3)$$

where x , y , and z are as shown in Fig. 2, \hat{E} is Young's modulus, I is the moment of inertia of the beam, S is the stress coefficient, ρ is the density, p_a is the ambient pressure, μ is the air viscosity, K is the Knudsen number, w is the width of the beam in y direction, $u = u(x, t)$ is the height of the beam above the substrate, and $p = p(x, y, t)$ is the pressure distribution in the air below the beam. The electrostatic force is approximated assuming nearly parallel plates and is given by $F_{\text{elec}} = -(\epsilon_0 w v^2 / 2u^2)$, where v is the applied voltage.

Spatial discretization of (2) and (3) using a standard finite-difference scheme (cf. [23]) leads to a large nonlinear dynamical system in form (1). For this system, the state vector (x) consists of heights of the beam above the substrate (u) computed at the discrete grid points, values of $\partial(u^3)/\partial t$, and the values of pressure below the beam. This vector of states is clearly only one of the possible choices. Still, it has an advantage that it allows

¹In the linear model, considered later on, we assume that $i_d(v) = 40v$ and in the quadratic model: $i_d(v) = 40v + 800v^2$.

one to obtain a system in form (1) with state-independent input matrix B . For the considered example, we select our output $y(t)$ as the deflection of the center of the beam from the equilibrium point ($y(t) \equiv r(t)$)—cf. Fig. 2).

The last of the examples we consider in this paper is an operational amplifier with differential input and output, and consisting of 70 MOSFETs, 13 resistors, and nine linear capacitors connected to 51 circuit nodes. Nodal analysis yields a nonlinear model of the device in form (1), with voltages at the circuit nodes defining a state vector. In order to simulate the amplifier, existing circuit simulators use separate nonlinear models for every transistor in the circuit, leading to complicated schemes of solving the resulting system of nonlinear equations. As we will show in the following sections, the proposed trajectory piecewise-linear (TPWL) approach allows one to model this complicated nonlinear dynamical system with a compact, easy-to-use macromodel, consisting of a small number of linearized models.

III. MODEL ORDER REDUCTION FOR NONLINEAR SYSTEMS

Suppose the original dynamical system (1) is of order N , i.e., is described by N states. The main goal of MOR techniques is to generate a model of this system with q states (where $q \ll N$), while preserving accurately the input/output behavior of the original system. Consequently, many MOR strategies are based on the concept of projecting the states of the original system onto a suitably selected reduced-order state space. This may also be viewed as performing a change of variables

$$x = Vz \quad (4)$$

where z is a q th order projection of state x (of order N) in the reduced-order space and V is an $N \times q$ orthonormal matrix ($V^T V = I$) representing a transformation from the original to the reduced-state space. In other words, columns of V define an orthonormal basis which spans the reduced state space.

Substituting (4) in (1) and multiplying the first of the resulting equations by V^T yields

$$\begin{cases} \frac{d}{dt} [V^T g(Vz(t))] = V^T f(Vz(t)) + V^T B(Vz(t))u(t) \\ y(t) = C^T Vz(t). \end{cases} \quad (5)$$

There are two key issues concerning representation (5) of the original dynamical system (1). The first one is selecting a reduced basis V , such that system (5) provides good approximation of the original system (1). For the linear case [i.e., if $f(\cdot)$ and $g(\cdot)$ are linear transformations, and $B(\cdot)$ is state independent], there are a number of methods for determining V . They include: selecting vectors from orthogonalized time-series data [10], [25], computing singular vectors of the underlying differential equation Hankel operator [6], or examining Krylov subspaces [1], [2], [4], [7], [14]–[16], [21], [23]. The approach based on using time-series data extends directly to the nonlinear cases, and the Hankel operator and Krylov subspace based strategies can be extended to the nonlinear case using linearization (Taylor's expansions) of the nonlinear system functions $f(\cdot)$ and $g(\cdot)$ [1], [2], [15], [23].

The other key issue in applying formulation (5) for reduced-order modeling is finding representations of $V^T f(V \cdot)$ and $V^T g(V \cdot)$ which allow low-cost storage and fast evaluation. Suppose, $N = 100\,000$ and $q = 10$. If no approximations are made to the nonlinear function $f(\cdot)$, then computing $V^T f(Vz)$ requires typically at least $O(100\,000)$ operations and is too costly. The simplest approximation for $f(\cdot)$, which allows less than $O(N)$ storage and evaluation of $V^T f(V \cdot)$ is based on Taylor's expansion about some (initial, equilibrium) state x_0

$$f(x) \simeq f(x_0) + A_0(x - x_0) + \frac{1}{2} W_0(x - x_0) \otimes (x - x_0)$$

where \otimes is the Kronecker product, and A_0 and W_0 are, respectively, the Jacobian and the Hessian of $f(\cdot)$ evaluated at the initial state x_0 . Analogously, we may take, e.g., a linearization of $g(x)$ about x_0 . Then,

$$\frac{dg(x)}{dt} \simeq \frac{d}{dt} (g(x_0) + G_0(x - x_0)) = G_0 \frac{dx}{dt}$$

where G_0 is the Jacobian of g at x_0 . Consequently, approximate evaluation of $d(V^T g(V \cdot))/dt$ becomes inexpensive. This approach leads to the following reduced-order models proposed in [1], [2], [15], and [23]. For the linear case, the reduced-order model (5) becomes

$$\begin{cases} G_{0r} \frac{d\bar{z}}{dt} = V^T f(x_0) + A_{0r} \bar{z} + B_{0r} u \\ y = C_r^T \bar{z} \end{cases} \quad (6)$$

where $A_{0r} = V^T A_0 V$ and $G_{0r} = V^T G_0 V$ are $q \times q$ matrices, $B_{0r} = V^T B(x_0)$, $C_r = V^T C$, and $V \bar{z} = x - x_0$. The quadratic reduced-order model is given by [15]²

$$\begin{cases} G_{0r} \frac{d\bar{z}}{dt} = V^T f(x_0) + A_{0r} \bar{z} + \frac{1}{2} W_{0r} (\bar{z} \otimes \bar{z}) + B_{0r} u \\ y = C_r^T \bar{z} \end{cases} \quad (7)$$

where $W_{0r} = V^T W_0 (V \otimes V)$ is a $q \times q^2$ matrix. One should note that the above quadratic model uses a linear approximation of $g(x)$. One could also consider a quadratic expansion for $g(x)$ which would lead to a more complicated reduced-order model.

In the above formulations, due to the fact that the reduced matrices are typically dense and must be represented explicitly, the cost of approximately computing $V^T f(Vz)$ and $d(V^T g(Vz))/dt$ terms and the cost of storing the reduced matrices G_{0r} , A_{0r} (A_{0r} and W_{0r} in the quadratic case) are $O(q^2)$ (in the linear case) and $O(q^3)$ (in the quadratic case). Therefore, although the method based on Taylor's expansions may be extended to higher orders [15], this approach is limited in practice to cubic expansions, due to exponentially growing memory and computational costs. For instance, if we consider quartic expansion of order $q = 10$, then the memory storage requirement exceeds $q^5 = 100\,000$ elements. In addition, the computational cost of evaluating the quartic approximation to $V^T f(V \cdot)$ is $O(q^5)$.

²An alternative formulation of the quadratic reduced-order model is presented in [1]. According to our experience, both formulations give almost identical results.

IV. PIECEWISE-LINEAR MOR

As described in the previous section, reduced-order models based on Taylor's series expansion become prohibitively expensive for high series order. On the other hand, a simple linearized reduced-order model (6), although computationally inexpensive, may be applied only to weakly nonlinear systems and is usually valid for only a very limited range of inputs [23]. This leads us to proposing an approach for MOR based on quasipiecewise-linear approximations of nonlinear systems [18]. The idea is to represent a system as a combination of linear models, generated at different linearization points in the state space (i.e., different states of the original nonlinear system). The key issue in this approach is that we will be considering multiple linearizations about suitably selected states of the system, instead of relying on a single expansion about the initial state.

A. Piecewise-Linear Representation

Let us assume we have generated s linearized models of the nonlinear system (1), with expansions about states x_0, \dots, x_{s-1}

$$\frac{d}{dt} (g(x_i) + G_i(x - x_i)) = f(x_i) + A_i(x - x_i) + B_i u$$

where x_0 is the initial state of the system, A_i (G_i) are the Jacobians of $f(\cdot)$ ($g(\cdot)$) evaluated at states x_i , and $B_i = B(x_i)$. We now consider a weighted combination of the above models

$$\begin{aligned} \frac{d}{dt} \left[\sum_{i=0}^{s-1} \tilde{w}_i(x) (g(x_i) + G_i(x - x_i)) \right] \\ = \sum_{i=0}^{s-1} \tilde{w}_i(x) (f(x_i) + A_i(x - x_i) + B_i u) \end{aligned} \quad (8)$$

where $\tilde{w}_i(x)$ s are state-dependent weights. (We assume that, for all x , $\sum_{i=0}^{s-1} \tilde{w}_i(x) = 1$.) The choice of weights is discussed later on in this section. Assuming we have already generated a q th order basis V [cf. (4)], we may consider the following reduced-order representation of system (8):

$$\begin{cases} \frac{d}{dt} \left(\left(\sum_{i=0}^{s-1} w_i(z) G_{ir} \right) z + \delta \cdot w(z) \right) \\ = \left(\sum_{i=0}^{s-1} w_i(z) A_{ir} \right) z + \gamma \cdot w(z) + \left(\sum_{i=0}^{s-1} w_i(z) B_{ir} \right) u \\ y = C_r z \end{cases} \quad (9)$$

where

$$G_{ir} = V^T G_i V$$

$$A_{ir} = V^T A_i V$$

$$B_{ir} = V^T B_i$$

$$C_r = C^T V$$

$$\gamma = [V^T (f(x_0) - A_0 x_0), \dots, V^T (f(x_{s-1}) - A_{s-1} x_{s-1})]$$

$$\delta = [V^T (g(x_0) - G_0 x_0), \dots, V^T (g(x_{s-1}) - G_{s-1} x_{s-1})]$$

and $w(z) = [w_0(z) \cdots w_{s-1}(z)]^T$ is a vector of weights [$\sum_{i=0}^{s-1} w_i(z) = 1$ for all z].

Let us assume that $[z_0, z_1, \dots, z_{s-1}]$ are projections of the linearization points x_0, \dots, x_{s-1} in the reduced basis, i.e.

$$[z_0, z_1, \dots, z_{s-1}] = [V^T x_0, V^T x_1, \dots, V^T x_{s-1}].$$

Then, weights $w_i(z)$ corresponding to the reduced-models associated with $[A_{ir}, G_{ir}, B_{ir}]$ can be computed using the information about the distances $\|z - z_i\|$ of the (projected) linearization points from the current state z . We require that the “dominant” model $[A_{jr}, G_{jr}, B_{jr}]$ is the one corresponding to the linearization point z_j which is the closest to the current state of the system. Also, we need to ensure that solutions of (9) remain smooth (e.g., have no discontinuities) as the “dominant” mode changes from one to another.

The following procedure of computing w_i was found to satisfy the above requirements.

- 1) For $i = 0, \dots, (s - 1)$ compute $d_i = \|z - z_i\|_2$.
- 2) Take $m = \min_{i=0, \dots, (s-1)} d_i$.
- 3) For $i = 0, \dots, (s - 1)$ compute $\hat{w}_i = e^{-\beta d_i/m}$.
- 4) Normalize \hat{w}_i at the evaluation point.
 - a) Compute $S(z) = \sum_{j=0}^{s-1} \hat{w}_j(z)$.
 - b) For $i = 0, \dots, (s - 1)$ set $w_i(z) = \hat{w}_i(z)/S(z)$.

In the above, β is a positive constant. In our implementation of the algorithm, we took $\beta = 25$. In this way, we obtain a weighting procedure such that the distribution of weights changes rather rapidly as the current state z evolves in the state space, i.e., once, e.g., z_j becomes the point closest to z , then weight w_j almost immediately becomes one. This provides a rationale for referring to model (9) as to a *piecewise-linear* reduced-order model of nonlinear system (1).

One of the reasons for using a procedure with rapidly changing weights is that “most of the time” the TPWL model actually reduces to a certain linear model, which may allow one to predict or control more easily some properties of that model. For example, if we know that all the Jacobians, A_{ir} , are stable (Hurwitz) matrices, then in the regions where only a single weight is nonzero, the “A” matrix for the TPWL system is clearly Hurwitz. On the other hand, in the regions with multiple nonzero weights, the associated “A” matrix may not be stable, since a convex combination of stable matrices may not be a stable matrix. Nevertheless, note that the discussed procedure still does not guarantee stability of the resulting TPWL model.

The weighting algorithm presented above is a simple heuristic with limited justification. Further investigation is needed in order to find out whether some extra knowledge on the system may be used to generate weighting procedures which would improve accuracy or preserve stability (or passivity) of the original system.

B. Generation of the Piecewise-Linear Model

One may assume that the linearized model is accurate for a given state x if this state is “close enough” to a linearization point x_i , i.e., $\|x - x_i\| < \epsilon$, or x lies within an N -dimensional ball of radius ϵ and centered at x_i . Then, it is obviously desirable to cover the entire N -dimensional state space with such balls, thereby assuring that any state is within ϵ of a linearization point, but the number of balls will grow exponentially with N . For

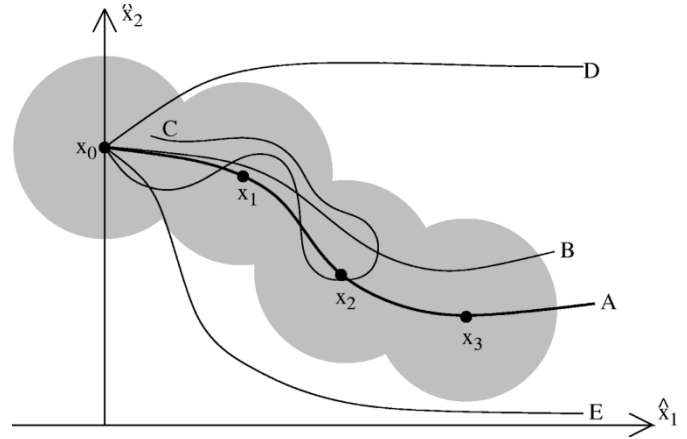


Fig. 3. Generation of the linearized models along a trajectory of a nonlinear system in a two-dimensional state space.

example, the number of radius 0.1 balls required to fill a 1000-dimensional unit hypercube is roughly 10^{1000} .

Since it is infeasible to cover the entire N -dimensional state space, we propose to generate a collection of models along a single, fixed trajectory of the system.³ We generate this trajectory by performing a single simulation of the nonlinear system for a fixed “training” input. (Instead of an exact simulation of the original nonlinear system we may perform an approximate, fast simulation—cf. Section IV-C2.) Given a training input signal $u(t)$, initial state x_0 , and a finite number of models to be generated s ($s > 1$), the proposed algorithm for generating the linearized models may be summarized in the following steps.

- 1) Generate a linearized model about the initial state x_0 , $i = 0$.
- 2) Simulate the nonlinear system while $\min_{0 \leq j \leq i} \|x - x_j\| < \delta$ ($\delta > 0$), i.e., while the current state x is close enough to any of the previous linearization points.
- 3) Generate a new linearized model about $x_{i+1} = x$, $i := i + 1$.
- 4) If $i < (s - 1)$ return to step 2).

This procedure is also depicted in Fig. 3. It should be stressed at this point that the piecewise-linear approach resulting from this procedure is different from methods presented, e.g., in [3] or [11], where piecewise-linear approximations of individual elements of the circuit (e.g., diodes or transistors) are considered and a very large collection of linear models is used. In our algorithm, piecewise-linear approximation applies to a trajectory of the *entire* nonlinear system, and, therefore, the number of linearized models may be kept small.

As illustrated in Fig. 3, the procedure proposed above allows one to “cover with models” only the part of the state-space located along the “training” trajectory (curve A). Let us assume that the reduced-order model (5) is composed of linear models generated along this trajectory. If a certain system’s trajectory, corresponding to a given input signal u , lies within the region of the state space covered by these models, we expect that the constructed piecewise-linear model (5) will suitably approximate the input/output behavior of the original nonlinear

³The idea of using a collection of linearized models along, e.g., an equilibrium manifold or a given trajectory is also used in design of gain-scheduled controllers for nonlinear systems—cf. [20], [22].

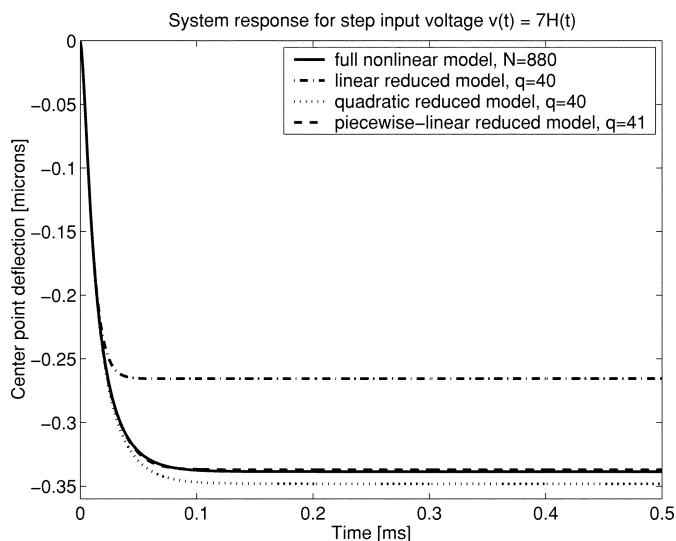


Fig. 4. Comparison of system response (micromachined switch example) computed with linear, quadratic, and piecewise-linear reduced-order models ($q = 40$ and $q = 41$) to the step input voltage $u(t) = 7H(t)$ [$H(t) \equiv 7$ for $t > 0$ and $H(t) \equiv 0$ for $t < 0$]. The piecewise-linear model was generated for the 8-V step input voltage.

system (cf. curves B and C).⁴ It should also be stressed at this point that, although the considered trajectory stays close to the “training” trajectory in the state space, the corresponding input signal can be dynamically very different from the “training” input. In other words, we may apply the piecewise-linear model for inputs which are significantly different from the “training” input, provided the corresponding trajectories stay in the region of the state space covered by the linearized models (cf. curve C and results in Section V). This case is also illustrated in Fig. 4, which shows computational results for the example of a micromachined switch (cf. Section II). This figure presents the system response to a 7-V step input voltage, computed with a 41st-order piecewise-linear reduced model of the device, generated for an 8-V step input training voltage. (The model was generated using the fast algorithm proposed in Section IV-C.) We should stress that, in fact, the input to the system is the squared input voltage $u(t) = v^2(t)$. One may note that the obtained output signal approximates very accurately the output signal computed with the full nonlinear model of the device (the curves on the graphs overlap almost perfectly). In this case, the piecewise-linear model provides significantly more accurate results than the linear or quadratic models based on single expansions about the initial state.

A different situation occurs when the input signal causes the trajectory to leave the region covered by the linearized models (cf. curves D and E in Fig. 3). Then, the piecewise-linear model (5) will most likely *not* provide significantly better approximation to the nonlinear system than a simple linear reduced model (6). This situation has been illustrated in Fig. 5. Due to a significant difference in scales (amplitudes) between the “training” input ($u(t) \equiv 7^2$) and the testing input ($u(t) \equiv 9^2$) the piecewise-linear model is no longer able to reproduce accurately the

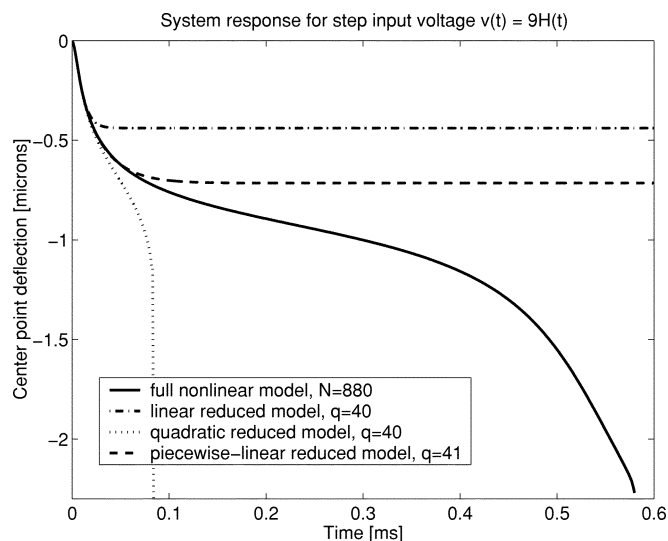


Fig. 5. Comparison of system response (micromachined switch example) computed with linear, quadratic, and piecewise-linear reduced-order models ($q = 40$ and $q = 41$) to the step input voltage $u(t) \equiv 9$ ($t > 0$). The piecewise-linear model was generated for the 7-V step input voltage.

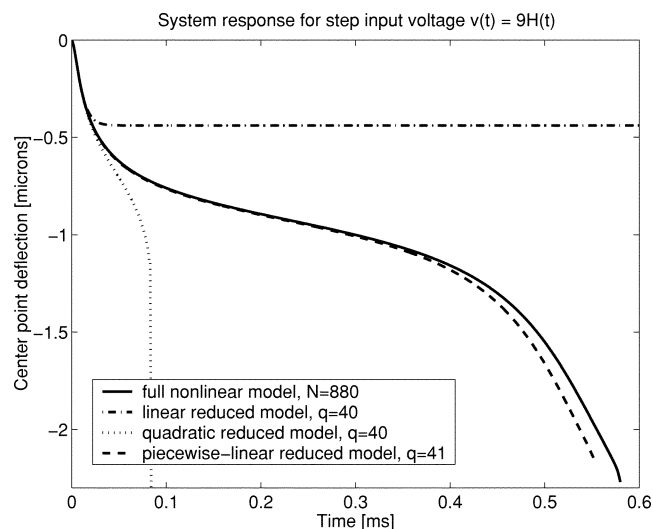


Fig. 6. Comparison of system response (micromachined switch example) computed with linear, quadratic, and piecewise-linear reduced-order models ($q = 40$ and $q = 41$) to the step input voltage $u(t) \equiv 9$ ($t > 0$). The piecewise-linear model was generated for the 9-V step input voltage.

response of the nonlinear system. Now, if we generate the piecewise-linear model with a 9-V training input (cf. Fig. 6), then this model is able to reproduce accurately the nonlinear response. One should note that in this case the piecewise-linear model is able to accurately model the dynamics of a highly-nonlinear pull-in effect (the beam is pulled down to the substrate), which is of particular importance in applications [10]. One may note from the graph that the linear model is not able to reproduce this phenomenon, while the quadratic model is unable to reproduce the correct dynamics. Still, this example shows that if the piecewise-linear model is to be used for inputs with very different scales one should consider more complicated schemes of generating the linearized models, based, e.g., on multiple training inputs.

⁴The additional rationale for this observation is that in typical situations the dimensions of observable and controllable spaces of a dynamical system are much lower than the dimension of its state space. (This is expected to be true for the examples of nonlinear dynamical systems presented in Section II.)

C. Fast Generation of TPWL Models

The above method for generating piecewise-linear models of nonlinear dynamical systems requires simulating the original nonlinear system (1). This simulation can be costly, due to the original size of the problem. In order to reduce the computational effort, we note that it is unnecessary to compute the *exact* trajectory for the “training” input in order to generate a collection of linearized models. In fact, it suffices to compute an *approximate* trajectory and obtain only approximate linearization points. In this section, we present an approach for efficient generation of piecewise-linear reduced-order models based on this idea. The proposed numerical algorithm proceeds in the two stages: 1) generation of the reduced basis, used to represent approximately the state space vectors (x) and 2) approximate simulation of the nonlinear system response to the training input, using the reduced basis and piecewise-linear approximation of the nonlinear functions $f(x)$ and $g(x)$ along a trajectory of the nonlinear dynamical system (1). This approach shares features with reduced-basis methods for solving parabolic problems [5]. Below, these two stages are described in more detail.

1) *Generation of the Reduced Basis:* The reduced-order basis $V = [v_1, \dots, v_q]$, where $v_i \in R^N$, is obtained in the following three steps.

- 1) Generate the linearization of the dynamical system (1) about the initial state x_0

$$\begin{cases} G_0 \frac{dx(t)}{dt} = f(x_0) + A_0(x - x_0) + B_0 u(t) \\ y(t) = C^T x(t) \end{cases} \quad (10)$$

where $B_0 = B(x_0)$, and A_0 (G_0) is the Jacobian of $f(x)$ ($g(x)$), evaluated at $x = x_0$, and construct an orthogonal basis $\tilde{V} = \{v_1, \dots, v_{lM}\}$ in the l th order Krylov subspace

$$\begin{aligned} K_l(A_0^{-1}G_0, A_0^{-1}B_0) \\ = \text{span}\{A_0^{-1}B_0, \dots, (A_0^{-1}G_0)^{l-1}A_0^{-1}B_0\}, \end{aligned} \quad (11)$$

using the Arnoldi algorithm [23] (or block Arnoldi algorithm [17] if the number of inputs $M > 1$). This choice of basis \tilde{V} ensures that l moments of the transfer function of the *reduced-order* linearized model match l moments of the transfer function for the original linearized model (10) [8], [15].

- 2) Orthonormalize the initial state vector x_0 with respect to the columns of \tilde{V} and obtain vector v_{lM+1} . (To this end, one may use e.g., the singular value decomposition (SVD) algorithm.)
- 3) Take V as a union of \tilde{V} and v_{lM+1} : $V = [\tilde{V}; v_{lM+1}]$.

So, the final size of the reduced basis equals $q = lM + 1$. The last two steps ensure that we will be able to represent exactly the initial state x_0 in the reduced basis V . [Note that if $x_0 \in \text{colspan}(\tilde{V})$, then steps 2) and 3) become unnecessary.] Exact representation of the initial state ensures that we may correctly

start the fast approximate simulation of the nonlinear system in the reduced-order space as described in the following section.⁵

2) *Fast Approximate Simulation:* The second stage of the proposed MOR algorithm may be summarized in the following steps.

- 1) Take $i = 0$, set x_0 to be the initial state.
- 2) While $i < s$ do
 - a) Using basis V , construct a reduced-order model of dynamical system (1), linearized about state x_i

$$\begin{cases} V^T G_i V \frac{dz}{dt} = V^T A_i V z + V^T (f(x_i) - A_i x_i) + V^T B_i u \\ y = C^T V z \end{cases} \quad (12)$$

where z is a reduced-order approximation of state vector x ($x \approx Vz$). This step requires computation of the Jacobians A_i and G_i (at $x = x_i$) in the nonreduced state space.

- b) Simulate reduced-order linear dynamical system (12), i.e., compute $z(t)$ for subsequent time steps $t = t_j$, while the state $Vz(t_j)$ is close enough to the initial state x_i ($x_i \neq 0$), i.e., when

$$\|Vz(t_j) - x_i\| / \|x_i\| < \alpha$$

where α is an appropriately selected constant (cf. the comments below).

- c) Take the next linearization point $x_{i+1} = Vz(t_j)$, $i := i + 1$.

There is an important issue concerning the TPWL MOR algorithm proposed above. In order to be able to reproduce nonlinear effects in the behavior of a dynamical system, the linearization points should be changed “frequently enough” during the proposed piecewise-linear simulation. This is determined by the constant parameter α in the algorithm presented above. The proper choice of α was found to depend significantly on the amplitude of the input signal $u(t)$.

A simple procedure for determining an appropriate value of α automatically is as follows. First, for a given input signal, we perform a reduced-order simulation of the linearized dynamical system, with linearization about the initial state, to find the final (steady-state) vector x_T . Although in most cases x_T will *not* be the correct steady state of our nonlinear dynamical system, it will give us information about the scale of changes between the initial and final state

$$d = \|x_T - x_0\| / \|x_0\|.$$

(If $x_0 = 0$, we may take $d = \|x_T\|$.) It is clear that in order to capture any nonlinear effects one has to select the value of α such that $\alpha < d$. In practical situations, it is usually enough to select $\alpha = d/5$ or $\alpha = d/10$.

3) *Generation of the Reduced Basis—An Extended Algorithm:* The simple algorithm for generating the reduced-order basis, which constructed a Krylov subspace only at the initial

⁵In this section, we presented only the simplest (and the least computationally expensive) algorithm for generating the reduced basis V . One may easily extend this scheme to construct a basis which includes e.g., states used as subsequent linearization points and Krylov subspaces corresponding to these states. Such an approach is presented in Section IV-C3.

state of the system, can be extended in order to include in the reduced-order basis Krylov subspaces corresponding to other linearization points located along the training trajectory [19]. This extension of the reduced basis is motivated by the following fact. As already mentioned in the previous section, if we use basis V spanning Krylov subspace (11) to construct a linear reduced-order model, then the first l moments of the transfer function for this reduced-order linear model match first l moments of the transfer function for the original linearized model (10). Consequently, important dynamical features of the nonreduced linearized model are preserved by the reduced-order linear model [8], [15]. Since in the TPWL model we use a collection of reduced-order linear models, taking a union of bases in Krylov subspaces corresponding to subsequent linearized models as a reduced-order basis V will ensure that, for every resulting reduced-order linear model, the first few moments of its transfer function will match the first few moments of the transfer function of the corresponding *nonreduced*-linearized model. Consequently, we may expect that important dynamical properties for each of the linearized models will be preserved after the projection process.

Two important technical details arise during the construction of the union of bases at different linearization points. Firstly, if we linearize about a nonequilibrium point two input terms appear (instead of one, if we linearize about an equilibrium)—one associated with term $B_i u$, and the other—associated with term $(f(x_i) - A_i x_i)$, where x_i is a given linearization point [cf. (12)]. Consequently, bases corresponding to two different Krylov subspaces $K_l(A_i^{-1}G_i, A_i^{-1}B_i)$ and $K_l(A_i^{-1}G_i, A_i^{-1}(f(x_i) - A_i x_i))$ [cf. (11)] need to be constructed. Secondly, we need to eliminate redundant (e.g., almost parallel) vectors (which may appear after we take a union of a collection of bases) from the reduced basis. To this end, we may apply, e.g., SVD algorithm and discard vectors corresponding to the smallest singular values.

Using the above motivation, we developed an extended algorithm for generating the projection basis, which may be summarized in the following steps.

- 1) Let $V_{\text{agg}} = []$, $i = 0$.
- 2) Repeat until the “training” simulation is completed.
 - a) Consider linearization of dynamical system (1) about state x_i

$$\begin{cases} G_i \frac{dx(t)}{dt} = f(x_i) + A_i(x(t) - x_i) + B_i u(t) \\ y(t) = C^T x(t) \end{cases} \quad (13)$$

where A_i is the Jacobian of $f(x)$, evaluated at $x = x_i$, G_i is the Jacobian of $g(x)$ evaluated at $x = x_i$, and $B_i = B(x_i)$ and construct two orthogonal bases V_1 and V_2 in the following q_1 th order Krylov subspaces

$$\begin{aligned} \text{span}\{V_1\} &= K_{q_1}(A_i^{-1}G_i, A_i^{-1}B_i) \\ \text{span}\{V_2\} &= K_{q_1}(A_i^{-1}G_i, A_i^{-1}(f(x_i) - A_i x_i)) \end{aligned}$$

using the Arnoldi algorithm [23] (or block Arnoldi algorithm [17] if there are multiple inputs).

- b) Take \tilde{V} as $\tilde{V} = [V_1 \ V_2 \ x_i]$.
 - c) Orthogonalize the columns of \tilde{V} using the SVD algorithm and construct a new basis \hat{V} which contains orthogonalized columns of \tilde{V} corresponding to singular values larger than a given $\epsilon_1 > 0$.
 - d) Take $V_{\text{agg}} := [V_{\text{agg}} \ \hat{V}]$.
 - e) Using \hat{V} , construct a linearized reduced-order system at x_i and simulate this system until you reach the next linearization point x_{i+1} , set $i := i + 1$.
- 3) Orthogonalize the columns of the aggregate basis V_{agg} using the SVD algorithm and construct the final reduced-order basis V which includes orthogonalized columns of V_{agg} corresponding to singular values larger than some $\epsilon_2 > 0$.

Step 2c) from the above algorithm may be omitted if we simulate a full order nonlinear system (instead of a linearized reduced-order system) in order to find subsequent linearization points. Then, one should take $\hat{V} = \tilde{V}$.

One may note that the above method is more expensive than the simple algorithm presented earlier, since we need to generate two orthogonal bases at every linearization point. To this end, we need to perform LU factorization of Jacobians A_i at every linearization point x_i . In the simple approach, presented in Section IV-C1, this had to be done only once. The extended algorithm also requires additional SVD steps.

Nevertheless, since we generate a “richer” basis we expect that it will more adequately approximate the initial state space. One may argue though that the above method may generate models of significantly larger order than the simple algorithm. In fact, the situation is often the opposite. As shown in Section IV-C, the extended algorithm has potential to generate suitable, accurate reduced bases with a lower order than the simple algorithm using a single linearization about the initial state.

D. A Posteriori Error Estimation

In this section, we will present a method for *a posteriori* estimating the error of solving (1) with a TPWL model (9). The following derivation of the error estimator is based on the assumption that the original nonlinear function f [cf. (1)] is negative monotone, i.e., [24]:

$$\exists \lambda > 0 \forall x, y (x-y)^T (f(x) - f(y)) \leq -\lambda (x-y)^T (x-y). \quad (14)$$

The above assumption is satisfied by a number of nonlinear systems, including, e.g., a certain class of nonlinear analog circuits. Also, one may easily note that if f is negative monotone, then system (1) is L_p stable for any admissible p , provided $g(x) = Gx$, where G is a symmetric positive definite matrix.

For simplicity, we will also assume in this section that $g(x)$ from (1) is an identity transformation and that $B(x) \equiv B$. Still, the following derivation may be easily extended for the case of an arbitrary invertible transformation $g(x)$ (with appropriately modified assumptions) and a state-dependent input matrix $B(x)$.

We will look for an estimate of $\|\delta x(t)\|_2$, where $\delta x(t) = x(t) - \hat{x}(t) = x(t) - Vz(t)$, and $x(t)$ and $z(t)$ are solutions at time t of (1) and (9), respectively (for the same initial condition x_o ⁶ and the same input signal u). From (1) and (9) we have

$$\begin{aligned}\dot{x} &= f(x) + Bu \\ \dot{\hat{x}} &= \sum_{i=0}^{s-1} w_i(z) [VV^T f(x_i) + VV^T A_i V(z - z_i)] + VV^T Bu\end{aligned}$$

where $x = x(t)$, $z = z(t)$ and $u = u(t)$. From the above we obtain

$$\delta \dot{x} = f(x) - f(\hat{x}) + h(z) + (I - VV^T)Bu \quad (15)$$

where

$$h(z) = f(Vz) - \sum_{i=0}^{s-1} w_i(z) [VV^T f(x_i) + VV^T A_i V(z - z_i)].$$

Since $\sum_{i=0}^{s-1} w_i(z) = 1$, for every z , we may write

$$\begin{aligned}f(Vz) &= \sum_{i=0}^{s-1} w_i(z) \left\{ [f(x_i) + A_i(Vz - x_i)] \right. \\ &\quad \left. + \left[\int_0^1 (1-s)W(\theta_i(s)) ds \cdot (Vz - x_i) \otimes (Vz - x_i) \right] \right\}\end{aligned}$$

where $\theta_i(s) = x_i + s(Vz - x_i)$ and W is the Hessian of f . Then $h(z)$ becomes

$$\begin{aligned}h(z) &= \sum_{i=0}^{s-1} w_i(z) \left\{ [(I - VV^T)(A_i Vz + f(x_i)) \right. \\ &\quad \left. + VV^T A_i V z_i - A_i x_i] \right. \\ &\quad \left. + \left[\int_0^1 (1-s)W(\theta_i(s)) ds \cdot (Vz - x_i) \otimes (Vz - x_i) \right] \right\}.\end{aligned} \quad (16)$$

Left multiplying (15) by δx^T , and applying property (14) and Schwartz inequality gives

$$\frac{d}{dt} (\|\delta x\|_2^2) \leq -\lambda \|\delta x\|_2^2 + \|\delta x\|_2 \|h(z) + (I - VV^T)Bu\|_2. \quad (17)$$

Let us now consider time interval $[t_i, t_{i+1}]$. Suppose we know $\|\delta x(t_i)\|_2$. Then, applying Comparison Lemma [9] to differential inequality (17) yields

$$\begin{aligned}\|\delta x(t)\|_2 &\leq \frac{1}{\lambda} \sup_{\tau \in [t_i, t_{i+1}]} \|h(z(\tau)) + (I - VV^T)Bu(\tau)\|_2 \\ &\quad \times (1 - \exp(-\lambda(t - t_i))) + \|\delta x(t_i)\|_2 \\ &\quad \times \exp(-\lambda(t - t_i))\end{aligned} \quad (18)$$

for all $t \in [t_i, t_{i+1}]$. The above inequality leads us to proposing the following scheme of computing error estimates $e(t)$ of $\|\delta x(t)\|_2$ at time steps t_0, t_1, t_2, \dots

⁶If x_o cannot be represented exactly in basis V , the initial condition for the reduced system is taken as $z_o = V^T x_o$.

- 1) At t_0 we have $\|\delta x(t_0)\|_2 = \|(I - VV^T)x_o\|_2 \leq \|I - VV^T\|_2 \|x_o\|_2 = e(t_0)$, where x_o is a known initial condition.
- 2) For $i = 1, 2, \dots$ we iteratively compute

$$\begin{aligned}e(t_i) &= \frac{1}{\lambda} \sup_{\tau \in [t_{i-1}, t_i]} \|h(z(\tau)) + (I - VV^T)Bu(\tau)\|_2 \\ &\quad \times (1 - \exp(-\lambda(t_i - t_{i-1}))) + e(t_{i-1}) \\ &\quad \times \exp(-\lambda(t_i - t_{i-1})).\end{aligned} \quad (19)$$

Clearly, it follows from (18) that $e(t_i) \geq \|\delta x(t_i)\|_2$ for every t_i . In practice, we replace the supremum in the above formula by a maximum over a discrete set of time steps between t_{i-1} and t_i , corresponding to a certain numerical time integration scheme. (If t_i are the same as subsequent integration steps, then we take a maximum of the two values at the ends of the considered time interval.) Clearly, this method of evaluating the supremum implicitly assumes that neither $h(z(t))$ nor $u(t)$ behave pathologically between subsequent integration time steps.

The main challenges associated with using the above scheme are related to: 1) finding λ [compare (14)] which would be as precise as possible. (Quality of the error estimates heavily depends on this parameter, therefore, one could consider using different λ s in different regions of the state space, if at all possible and computationally feasible.) and 2) finding estimates of $\|h(z(t))\|_2$, given by (16), which typically requires estimating $\|W(x)\|_2$ —the norm of the Hessian of f .

If we apply the aggregate reduced-order basis V , described in Section IV-C3, then one may easily note that $Vz_i = x_i$, for every i . Furthermore, if we include $(f(x_i) - A_i x_i)$ ($\forall i$) in the reduced basis V , then $VV^T(f(x_i) - A_i x_i) = (f(x_i) - A_i x_i)$ [cf. (16)] and the following estimate of $\|h(z)\|_2$ may be given:

$$\begin{aligned}\|h(z)\|_2 &\leq \sum_{i=0}^{s-1} w_i(z) \left[\frac{1}{2} \sup_x \|W(x)\|_2 \|z - z_i\|_2^2 \right. \\ &\quad \left. + \|(I - VV^T)A_i V\|_2 \|z\|_2 \right].\end{aligned}$$

Then, we may replace (19) with

$$\begin{aligned}e(t_i) &= \frac{1}{\lambda} \sup_{\tau \in [t_{i-1}, t_i]} \sum_{i=0}^{s-1} w_i(z(\tau)) \\ &\quad \times \left[\frac{1}{2} \sup_x \|W(x)\|_2 \|z(\tau) - z_i\|_2^2 \right. \\ &\quad \left. + \|(I - VV^T)A_i V\|_2 \|z(\tau)\|_2 \right. \\ &\quad \left. + \|(I - VV^T)B\|_2 \|u(\tau)\|_2 \right] \\ &\quad \times (1 - \exp(-\lambda(t_i - t_{i-1}))) + e(t_{i-1}) \\ &\quad \times \exp(-\lambda(t_i - t_{i-1})).\end{aligned} \quad (20)$$

One should note that since the values of norms $\|(I - VV^T)A_i V\|_2$ (for every i) and $\|(I - VV^T)B\|_2$ can be computed during construction of the reduced model, the cost of evaluating (20) is $O(sq)$ only. This means that error estimation may be performed ‘‘on the fly,’’ along with the reduced-order simulation, without increasing the complexity of the fast solver.

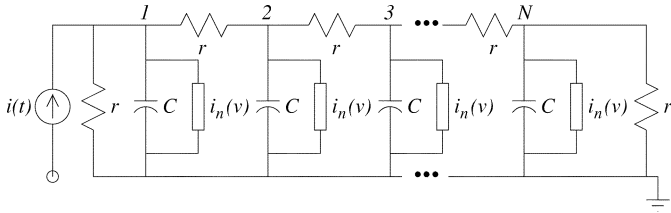


Fig. 7. Example of a circuit with quadratic nonlinearity.

In order to verify the proposed method of estimating the simulation error for the TPWL reduced-order model, we considered a simple test example of an RC ladder shown in Fig. 7. A quadratic nonlinearity is introduced to the circuit by adding nonlinear resistors to the ground at each node

$$i_n(v) = g \cdot \text{sgn}(v)v^2$$

where $\text{sgn}(v) = 1$ if $v \geq 0$ and $\text{sgn}(v) = -1$ if $v < 0$. If we take $C = r = g = 1$, then the nonlinear operator f [cf. (1)] takes the following form:

$$f(v) = Av - n(v)$$

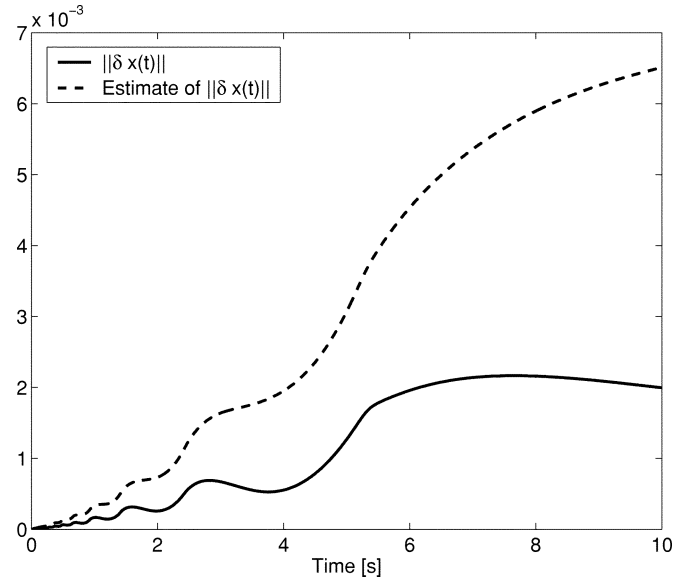
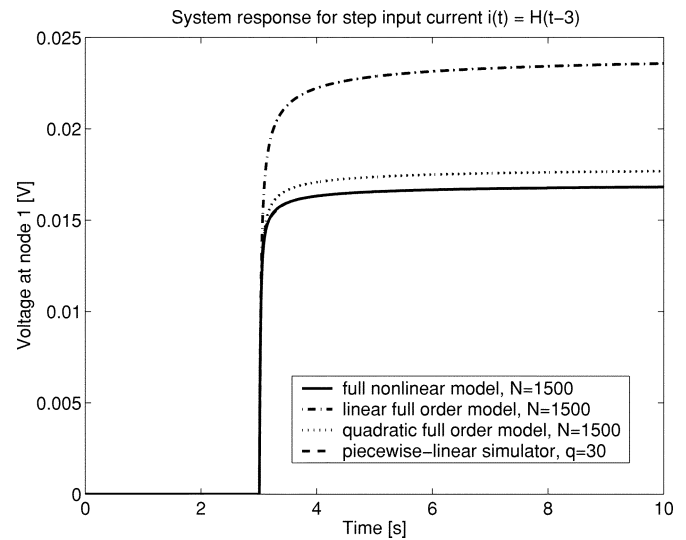
where

$$A = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & & 1 & -2 \end{bmatrix} \quad n(v) = \begin{bmatrix} \text{sgn}(v_1)v_1^2 \\ \text{sgn}(v_2)v_2^2 \\ \dots \\ \text{sgn}(v_N)v_N^2 \end{bmatrix}$$

and $v = [v_1, \dots, v_N]$ is the vector of states ($N = 100$ in our test). It may easily be proved that f is negative monotone, provided all v_i are nonnegative at all times (which is satisfied if the input current $i(t) \geq 0$ for all t). The value of λ [cf. (14)] may then be taken as $\lambda = -1 \cdot \max\{\lambda_i \in \sigma(A)\}$, where $\sigma(A)$ is the spectrum of matrix A . For $N = 100$, $\lambda = 9.67 \cdot 10^{-4}$. We also have that $\|W(x)\|_2 = 2g = 2$ for all x . Knowing λ and $\|W(x)\|_2$ we are ready to use formula (20) to compute error estimates.

In a numerical test, we generated a reduced-order TPWL model of order $q = 25$ (with $s = 16$ linearization points) and simulated both original nonlinear system and TPWL reduced-order model, with the input current $i(t)$ equal to unit step. (It should be stressed that q is relatively large, as compared to N for this example, and, therefore, it may be inefficient to use the extracted TPWL model in practice. Still, this reduced model provides useful insight while considering the problem of error estimation.) The actual error $\|\delta x\|_2$ and its estimate were computed at every time step. Fig. 8 shows a comparison of the actual error and its estimate for the considered case. One may note that formula (20) gives reasonable estimates of the error of approximating the original nonlinear system with a TPWL reduced-order model.

One should note that the error estimation procedure described above may be used not only to assess errors of simulation with an existing TPWL reduced-order model, but also to improve the algorithm for generating the TPWL models (or, more precisely, the algorithm for selecting subsequent linearization points). Currently, during the ‘‘training simulation,’’ the subsequent linearization points are selected using a simple geometric criterion: if the current state is ‘‘far enough’’ from all previous linearization points, then it becomes the next linearization


 Fig. 8. *A posteriori* error estimates for a TPWL reduced-order model of a nonlinear RC ladder.

 Fig. 9. Comparison of system response (nonlinear transmission line circuit model) computed with linear, quadratic, full nonlinear, and TPWL models to the step input current $i(t) = H(t - 3)$ [$H(t) \equiv 1$ for $t > 3$ and $H(t) \equiv 0$ for $t < 3$]. $N = 1500$.

point. Instead of this geometric criterion, perhaps one might use a measure based on error estimates (which use information on the nonlinear system at hand) to select a collection of linearization points. Clearly, the subject of where to place linearization points needs further investigation.

V. FAST PIECEWISE-LINEAR SIMULATOR

One should note that the MOR algorithm presented in Section IV-C may be used as a fast simulator for nonlinear dynamical systems. The simulator has been implemented for the example of a nonlinear transmission line circuit model described in Section II. Selected results of numerical tests are presented below.

Fig. 9 shows the output voltage $v_1(t)$ for a step input current, computed using full order linear and quadratic models as well as

TABLE I
QUALITY OF APPROXIMATION FOR LINEAR, QUADRATIC AND TPWL SIMULATORS FOR THE STEP INPUT CURRENT. $v = [v(0), v(\Delta T), \dots, v(T)]$ IS THE COMPUTED OUTPUT VOLTAGE AT NODE 1, v_{ref} IS THE REFERENCE OUTPUT VOLTAGE COMPUTED WITH FULL NONLINEAR MODEL

Model	$\ v - v_{\text{ref}}\ _2$
linear, $N = 1500$	0.384
quadratic, $N = 1500$	0.049
TPWL, $N = 1500, q = 30$	0.003

TABLE II
COMPARISON OF SIMULATION TIMES FOR THE FULL ORDER NONLINEAR SIMULATOR AND THE PROPOSED TPWL REDUCED ORDER SIMULATOR

Input, problem size	Simulation time [s], full nonlinear model	Simulation time [s], TPWL model
$i(t) = H(t - 3)$ ($N = 1500, q = 30$)	9573.3	80.8
$i(t) = \exp(-t)$ ($N = 1500, q = 30$)	11713.1	110.9
$i(t) = \sin(2\pi t/10)$ ($N = 100, q = 10$)	25.4	2.7

the proposed TPWL simulator. The reference result is computed with a simulator using a full order nonlinear model. In the simulation the number of time steps was 1000 ($T = 10, \Delta T = 0.01$), the TPWL simulator used the reduced basis of order $q = 30$ (the original problem size was $N = 1500$) and the linearization point changed 20 times (i.e., it used 21 different linear models during the simulation). As Fig. 9 clearly shows, the output voltage obtained by the TPWL method matches very well the reference result (the curves overlap almost perfectly). Table I shows the relative error between the voltage $v = [v(0), v(\Delta T), \dots, v(T)]$ computed with linear, quadratic, and TPWL simulators and the reference voltage v_{ref} obtained with the full nonlinear model of the circuit. It is apparent that the proposed piecewise-linear algorithm gives significantly more accurate results than the linear or quadratic simulations, producing results which accurately match the steady state of the system.

Table II compares performance of the full nonlinear simulator for the considered nonlinear transmission line circuit model and the proposed TPWL solver, which performs reduced-basis computations, for three different inputs. In order to assure appropriate accuracy, for the circuit with $N = 1500$ nodes, the order of the reduced basis equaled $q = 30$, and for the circuit with $N = 100, q = 10$. The simulators were implemented in Matlab, therefore, the presented execution times should be used for comparison only. High-performance implementations will most likely give significantly lower absolute execution times and may change the relative performance of the two algorithms. The tests were performed on a Linux workstation with a Pentium III Xeon processor. It is apparent that for either small or large original problem sizes, the piecewise-linear simulator is significantly faster than the full nonlinear solver. For $N = 1500$, a hundredfold acceleration in computation time was achieved.

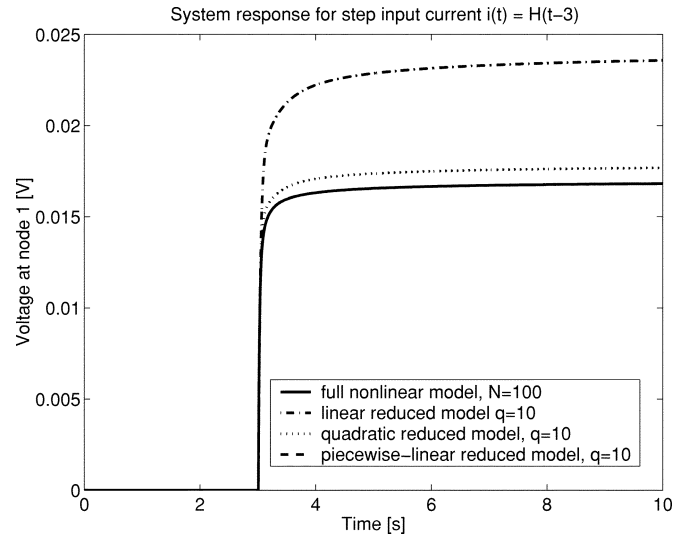


Fig. 10. Comparison of system response (nonlinear transmission line circuit model) computed with linear, quadratic, and TPWL reduced-order models (of order $q = 10$) for the step input current $i(t) = H(t - 3)$ [$H(t) \equiv 1$ for $t > 3$ and $H(t) \equiv 0$ for $t < 3$]. The TPWL model was generated using a unit step input current.

VI. COMPUTATIONAL RESULTS

A. Model Verification—Transient Simulations

This section presents results of computations using TPWL reduced-order models, obtained with the MOR technique proposed in Section IV-C. Our main goal is to find out whether this technique does really generate a model of our system. Let us recall that, in the proposed MOR algorithm, the model [which basically consists of a collection of reduced-order $q \times q$ matrices $A_{0r}, \dots, A_{1r}, \dots, A_{(s-1)r}$] is obtained by performing a fast simulation for a given training input signal. In order to show that we have indeed generated a model we should verify that it gives correct outputs not only for the input it was generated with, but also for other inputs.

This verification was done experimentally. We considered our nonlinear transmission line circuit model (cf. Section II) with $N = 100$ nodes and generated a reduced-order TPWL model of order $q = 10$ using a step input $i(t) = H(t - 3)$. For this example, the linearization point changed four times, therefore, our model consisted of five reduced-order matrices A_{0r}, \dots, A_{4r} . The reduced-order model was tested for different inputs, including the step input used to generate it. Fig. 10 shows the result for the step input (the same input we used for model extraction). Figs. 11 and 12 show the reduced-order simulation results for a cosinusoidal input and an exponentially decaying input, respectively. In all the cases, the output voltages obtained with the TPWL reduced-order model accurately approximate the reference voltages (the curves overlap almost perfectly). This indicates that our reduced-order system provides a sensible model for the original nonlinear circuit.

Fig. 13 provides an analogous test for the example of a micromachined beam described in Section II. In this case, the reduced-order model was generated for the step 8-V training input voltage ($q = 41$, the model used nine linearization points). Then it was tested for a cosinusoidal input with a 7-V amplitude. Once again, the transient obtained with the TPWL model matches

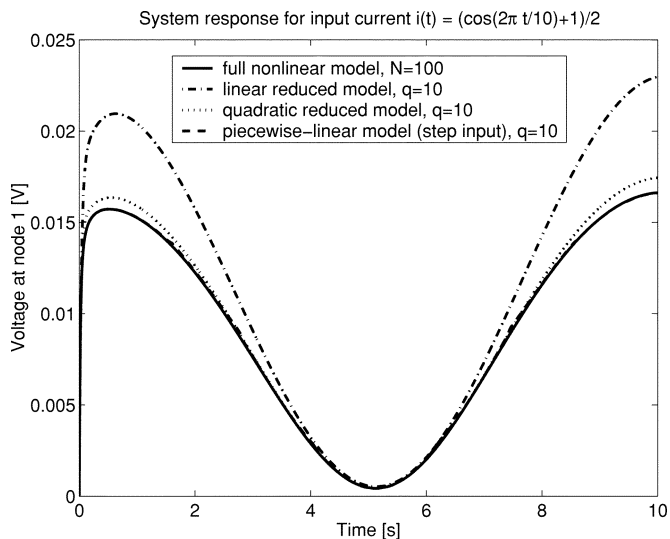


Fig. 11. Comparison of system response (nonlinear transmission line circuit model) computed with linear, quadratic, and TPWL reduced-order models (of order $q = 10$) for the input current $i(t) = (\cos(2\pi t/10) + 1)/2$. The TPWL model was generated using a unit step input current.

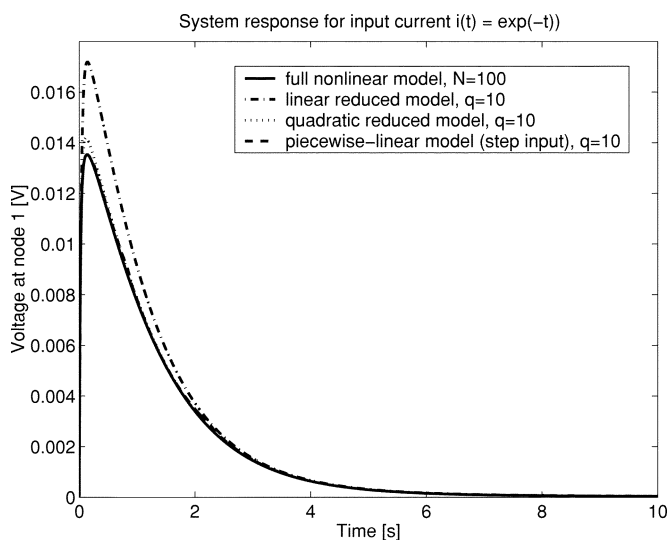


Fig. 12. Comparison of system response (nonlinear transmission line circuit model) computed with linear, quadratic, and TPWL reduced-order models (of order $q = 10$) for the input current $i(t) = \exp(-t)$. The TPWL model was generated using a unit step input current.

very accurately the reference result obtained with the full nonlinear model of order $N = 880$.

Figs. 10–13 also provide a comparison of the proposed TPWL reduced-order model with linear and quadratic reduced models, generated using methods described in [1], [15], and [23]. It is apparent from the graphs that the TPWL reduced-order model gives significantly more accurate results than the linear and quadratic reduced-order models using Taylor expansions about the initial state. It should be stressed at this point that all models (linear, quadratic, and TPWL) were of the same order and, moreover, applied the same basis V (obtained with the procedure described in Section IV-C1).

Next, we considered the operational amplifier example, described in Section II. The examined circuit had $N = 51$ nodes and eight inputs: 1) the differential input with input signals v_{in1}

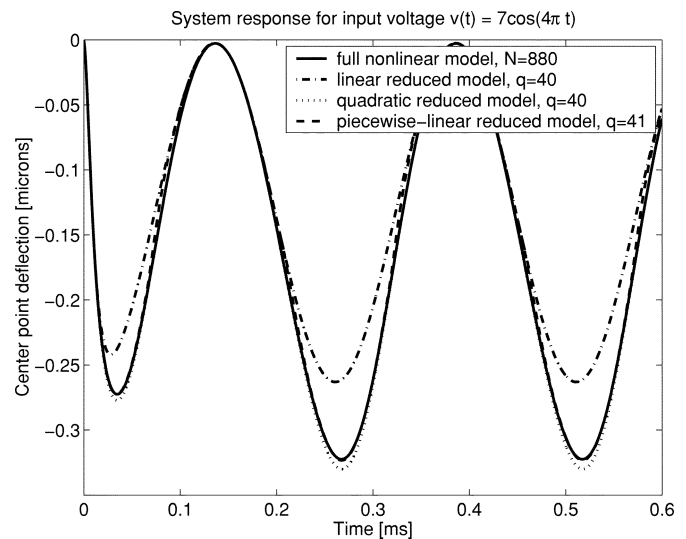


Fig. 13. Comparison of system response (micromachined switch example) computed with linear, quadratic, and piecewise-linear reduced-order models (of order $q = 40$ and $q = 41$) for the input voltage $u(t) = 7 \cos(4\pi t)$. The piecewise-linear model was generated for the 8-V step input voltage.

and v_{in2} , and 2) the auxiliary inputs v_{cmmrst} , v_{gnd} , v_{intn} , v_{intp} , v_{rst} , and v_{cmmin} used in common-mode rejection testing. We considered two full-order models of the op-amp: 1) simplified: employing linearized capacitance models for MOSFETs [then, $g(x) = Gx$ —cf. equation (1)] and 2) regular: employing nonlinear capacitance models (i.e., with a state-dependent capacitance matrix). In both cases, the full nonlinear simulations were performed using NITSWIT circuit simulator, described in [12] and [13]. In order to generate the reduced-order TPWL models, we applied the following set of training inputs:

$$v_{in1}(t) = \begin{cases} 0, & t < 290 \\ 12.5 \cdot 10^{-3}(t - 290)/10, & 290 \leq t < 300 \\ 12.5 \cdot 10^{-3}, & t \geq 300. \end{cases}$$

$v_{in2} = -v_{in1}$ (cf. Fig. 14) and auxiliary input signals shown in Fig. 15.

For the case with a linearized capacitance matrix, we obtained a TPWL model of order $q = 18$ (with 35 linearization points and eight inputs), and for the case with the nonlinear capacitance we obtained a model of order 34 (with 29 linearization points). As one may note, due to an increased complexity of the applied MOSFET device models, the order of the resulting reduced-order model is higher in the second case.

The obtained reduced-order TPWL models were then tested for the following input (cf. Fig. 14):

$$v_{in1}(t) = \begin{cases} 0, & t < 290 \\ 11.5 \cdot 10^{-3}(t - 290)/110, & 290 \leq t < 400 \\ 11.5 \cdot 10^{-3}, & t \geq 400. \end{cases}$$

($v_{in2} = -v_{in1}$). Figs. 16 and 17 show a comparison of the transients computed with NITSWIT and with the reduced-order TPWL models for one (of the two) output nodes of the amplifier. One may note excellent agreement of the output signals for both cases, which indicates that suitable reduced-order TPWL models of the original systems have been constructed. The results also indicate that the proposed MOR method may be suc-

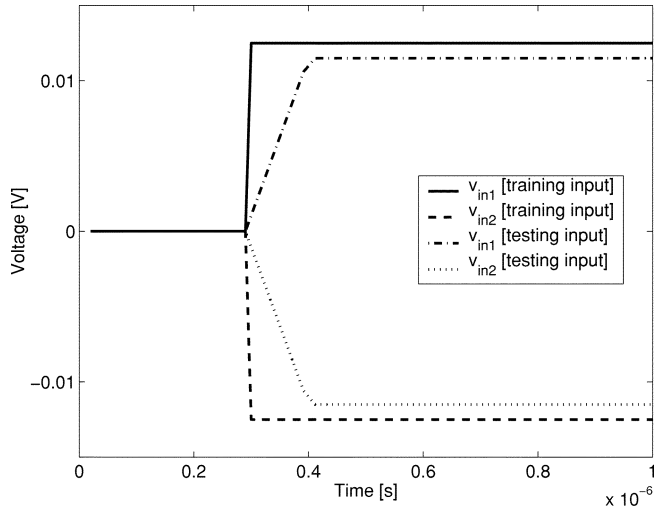
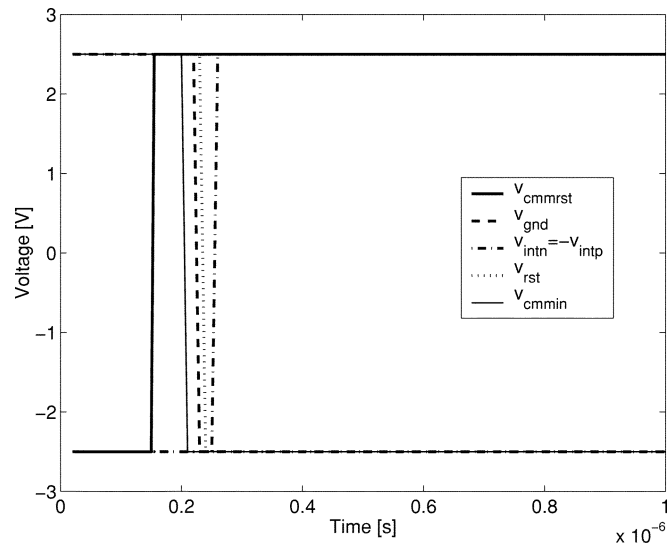
Fig. 14. Op-amp input signals v_{in1} and v_{in2} .

Fig. 15. Auxiliary inputs for the op-amp circuit.

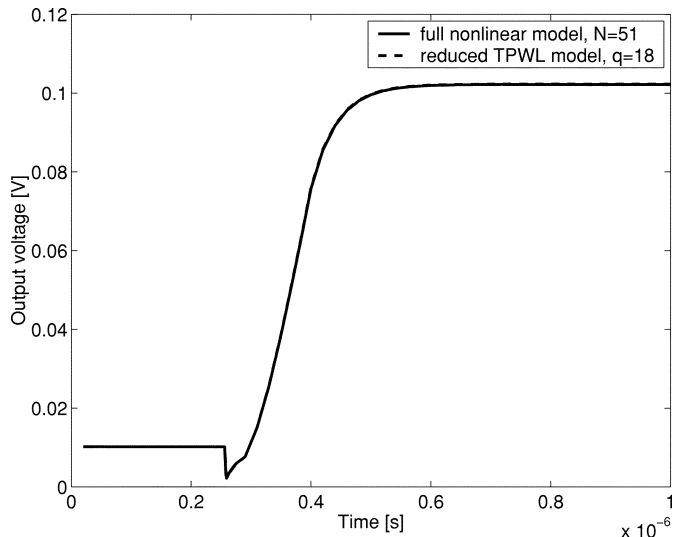


Fig. 16. Comparison of the output voltage (op-amp example, simplified linearized capacitance models), computed with a reduced-order TPWL model and NITSWIT circuit simulator.

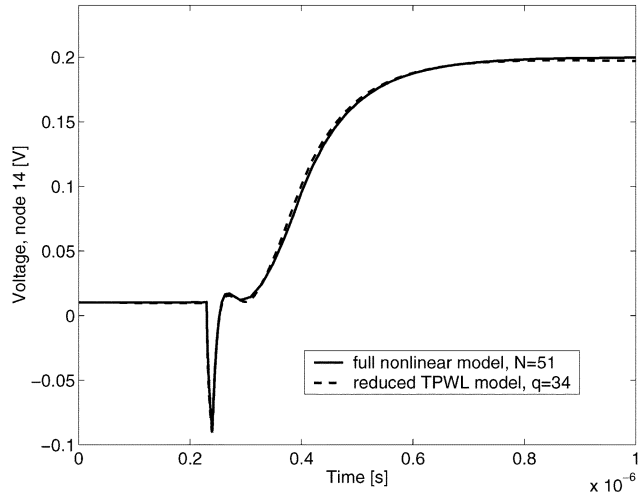


Fig. 17. Comparison of the output voltage (op-amp example, regular nonlinear capacitance models), computed with a reduced-order TPWL model and NITSWIT circuit simulator.

TABLE III
COMPARISON OF THE TIMES OF MODEL EXTRACTION AND REDUCED ORDER SIMULATION FOR THE LINEAR, QUADRATIC AND TPWL MOR TECHNIQUES. THE ORIGINAL PROBLEM HAD SIZE $N = 1500$. THE REDUCED MODEL HAD SIZE $q = 30$. THE TESTS WERE RUN FOR THE NONLINEAR TRANSMISSION LINE EXAMPLE

MOR method	Model extraction time [s]	Simulation time [s]
linear	44.8	1.18
quadratic	2756.5	31.5
TPWL	80.7	8.0

cessfully used for multiple input systems. It is important to point out that not only do the TPWL models have a lower order than the original system, but also they are much easier to use. Since a TPWL model consists of a weighted combination of linear models, the time stepping is very straightforward. In a simplified backward Euler time stepping scheme we compute the weights w_i [cf. (9)] e.g., using the previous state of the system or a predictor of the next state and then, assuming that these weights are fixed, we find the state at the next time step by performing only a single Newton update (i.e., solving a low-order linear system of equations). In a more sophisticated time stepping scheme, one can account for derivatives of $w_i(x)$, which is also straightforward, since the weights are simple scalar functions. In a regular simulator, if using backward Euler scheme, finding the next state requires computation of a number of Newton updates for the full order nonlinear system, which is considerably more complex.

B. Performance and Complexity of the MOR Algorithm

Table III shows a comparison of the performance of the discussed MOR techniques and the reduced-order solvers. All the algorithms were implemented in Matlab. The tests were performed on a Linux workstation with a Pentium III Xeon processor. One may note that performance for linear and TPWL models is comparable. The generation of the quadratic model is significantly more expensive, due to the costly reduction of

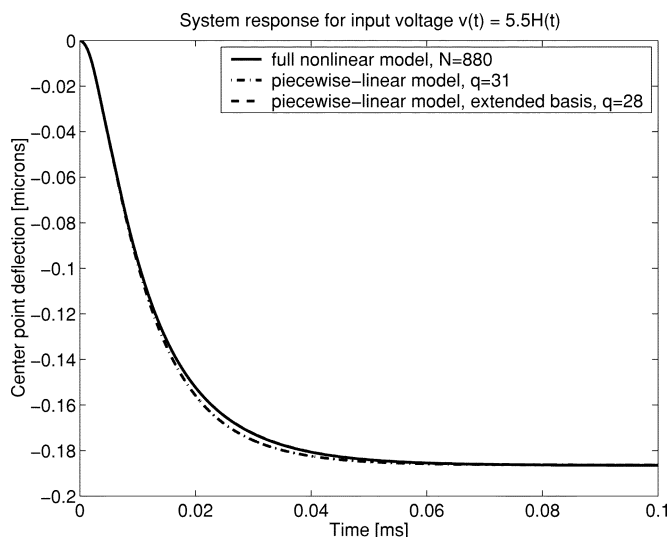


Fig. 18. Comparison of system response (micromachined switch example) computed using the TPWL models extracted with the simple and the extended algorithm for generating the reduced-order basis. In both cases, the models were generated for the 5.5-V step input voltage.

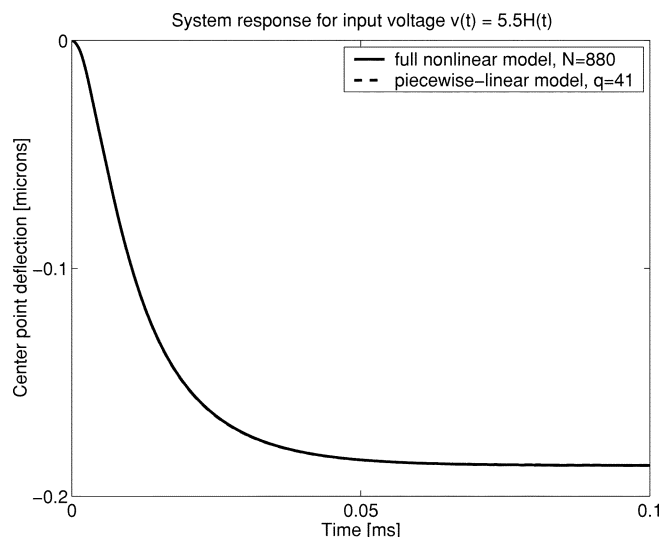


Fig. 19. Comparison of system response (micromachined switch example) computed using the full nonlinear simulator and the TPWL reduced-order model extracted with the simple algorithm for generating the reduced-order basis. The model of order $q = 41$ was generated for the 5.5-V step input voltage.

the Hessian matrix, which requires q^2 computations of the matrix-vector product $W(x \otimes x)$, where W is a full order $N \times N^2$ Hessian matrix (usually represented implicitly—cf. [1]).

The memory complexity of the TPWL reduced-order solver is $O(sq^2)$, where s is the number of linearization points. Consequently, the memory cost is roughly s times larger than the cost for the linear reduced-order simulator [which is $O(q^2)$]. The memory cost of the quadratic reduced-order solver is $O(q^3)$ (the reduced-order Hessian must be stored explicitly as a matrix), so if $s \approx q$, then the memory requirements for the piecewise-linear solver are approximately the same as for the quadratic solver. For the examples of the nonlinear transmission line and the micromachined switch (cf. Figs. 10–13, $s = 5 = q/2$ or $s = 9 < q/4$), so in those cases the memory used by the piecewise-linear algorithm equaled roughly only half (or a quarter) of the memory used by the quadratic solver. In the case of the operational amplifier $s \approx 2q$ (linearized capacitance case), which translates to doubled storage requirements as compared to the quadratic model.

C. Performance of the Extended Algorithm for Generating the Reduced-Order Basis

This section presents computational results comparing two algorithms for generating the reduced-order basis: a simple one, presented in Section IV-C1, and the extended one, introduced in Section IV-C3. Fig. 18 shows the deflection of the center of the micromachined fixed-fixed beam computed using the two considered methods. In both MOR methods, the 5.5-V step input voltage was used as a “training” input and the number of linearization points equaled six. For the simple algorithm, the order of the reduced-model equaled $q = 31$. In the extended algorithm, a basis of order seven [$= 2 \cdot q_1 + 1$, $q_1 = 3$ —cf. Step 2)a) in the algorithm from Section IV-C3] was generated at each of the linearization points. Then the size of the aggregate basis V_{agg} has been reduced from $42 = 6 \cdot 7$ to 28 using the SVD algorithm. One may note that the TPWL model of order $q = 28$,

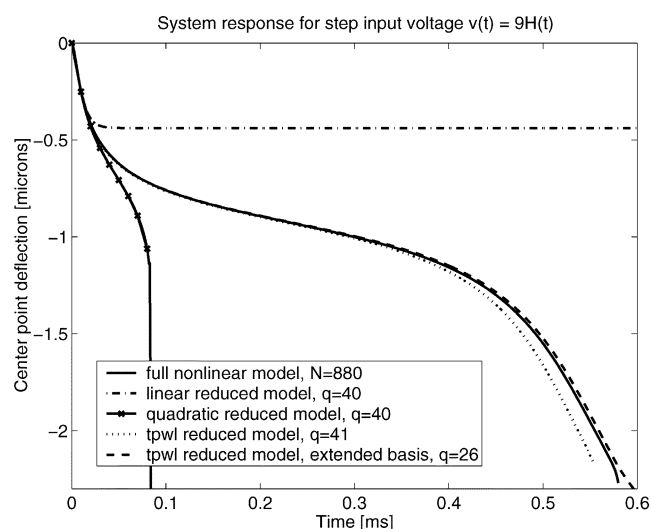


Fig. 20. Comparison of system response (micromachined switch example) computed with different MOR algorithms. TPWL models were generated for the 9-V step input voltage.

generated with the extended reduced-order basis, gives significantly more accurate results than the TPWL model generated with a simple basis. (On the graph, the dashed line overlaps perfectly with the solid line.) In order to obtain the desired accuracy with the model extracted with the simple basis generation algorithm, the order of the basis needs to be increased to $q = 41$ in the considered case (cf. Fig. 19).

Fig. 20 shows the simulated pull-in effect for the micromachined switch example. Again, in this case, the MOR method employing the extended algorithm to generate the reduced basis provides the best accuracy among the considered MOR techniques. It also generates a model with the lowest order.

One should note that in the extended algorithm, we generate a collection of very-low-order bases at different linearization points rather than a larger basis at a single linearization point, as in the initial approach. As shown by the presented results, this

TABLE IV

COMPARISON OF THE TIMES OF MODEL EXTRACTION AND REDUCED ORDER SIMULATION FOR THE TPWL MOR ALGORITHMS USING TWO DIFFERENT METHODS OF GENERATING THE REDUCED ORDER BASIS. THE TESTS WERE RUN FOR THE MICROMACHINED SWITCH EXAMPLE. THE ORIGINAL PROBLEM SIZE EQUALED $N = 880$

Reduced basis generation algorithm	Model extraction time [s]	Simulation time [s]
simple, $q=28$	26.5	3.1
simple, $q=31$	27.4	3.4
simple, $q=41$	30.9	6.5
extended, $q=28$	66.8	3.1

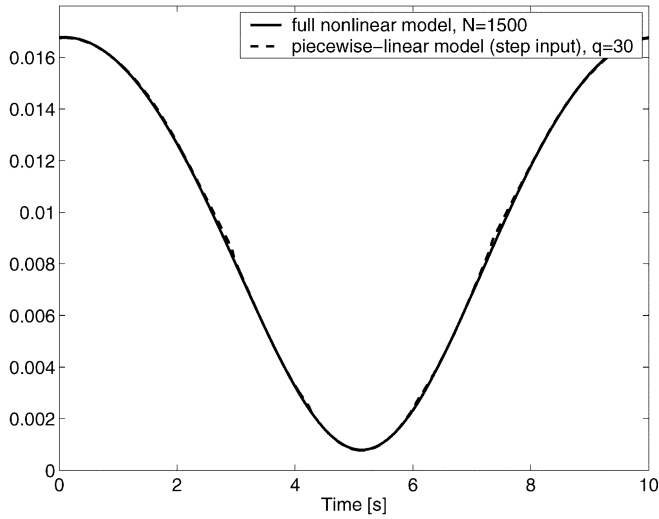


Fig. 21. Comparison of sinusoidal steady state (nonlinear transmission line circuit model) computed using full nonlinear and TPWL models. The input signal was $i(t) = (\cos(2\pi t) + 1)/2$.

may lead to a model with a lower order, which is faster to simulate. The tradeoff is that the extended basis generation algorithm is computationally more expensive. This has been illustrated in Table IV, which shows performance of the MOR techniques for the considered micromachined beam example.

D. Sinusoidal Steady-State Simulations

This section compares results of simulation of the sinusoidal steady state computed with the full-order nonlinear models and the reduced-order TPWL models. The tests were performed for the examples of a nonlinear transmission line circuit model (cf. Fig. 1) and the operational amplifier, described in Section II.

In the first series of tests, we computed the sinusoidal steady state for the nonlinear transmission line model with $N = 1500$ nodes, excited with the input current $i(t) = (\cos(2\pi t) + 1)/2$. The simulation was performed with a simple, fixed time step shooting method. Fig. 21 shows the computed sinusoidal steady-state output of the system in time domain. One may note that the result obtained with the reduced-order model of order $q = 30$ closely matches the reference result.

The frequency domain analysis has also been performed for the sinusoidal steady-state output signal $y(t)$ shown in Fig. 21. [$y(t) = v_1(t)$, where $v_1(t)$ is the voltage at node 1 of the circuit.] We computed a complex discrete Fourier transform (DFT)

TABLE V

COMPARISON OF THE SUBSEQUENT HARMONICS OF THE SINUSOIDAL STEADY STATE, COMPUTED USING A FULL ORDER NONLINEAR MODEL AND THE REDUCED ORDER TPWL MODEL

Harmonics	Full nonlinear model	Reduced order TPWL model	Error [%]
dc (c_0)	9.3736	9.4061	0.4
1st (c_1)	3.9684-0.2625i	3.9630-0.2641i	0.2
2nd (c_2)	-0.2803+0.0617i	-0.3100+0.0578i	10.5
3rd (c_3)	0.0223-0.0106i	0.0233-0.0138i	13.5

TABLE VI

COMPARISON OF THE MAIN INTERMODULATION HARMONICS OF THE SINUSOIDAL STEADY STATE, COMPUTED USING A FULL-ORDER NONLINEAR MODEL AND THE REDUCED ORDER TPWL MODEL ($q = 39$)

Harmonics	Full nonlinear model	Reduced order TPWL model	Error [%]
dc	-0.4967	-0.4915	-1.1
$f_2 - f_1$	0.0072 - 0.0223i	0.0077 - 0.0232i	4.3
$2f_1 - f_2$	0.0046 - 0.0037i	0.0044 - 0.0029i	-10.9
f_1	-0.0667 + 0.1116i	-0.0667 + 0.1116i	0.1
f_2	-0.0638 - 0.1003i	-0.0629 - 0.1008i	-0.1
$2f_2 - f_1$	0.0013 - 0.0050i	0.0015 - 0.0051i	1.9
$2f_1$	0.0104 + 0.0008i	0.0108 + 0.0010i	3.8

of the discrete output signal $\hat{y}(n) = y(n\Delta t)$. The first four Fourier coefficients c_k (not normalized) are shown in Table V. One may note that the sinusoidal steady state $y(t)$ obtained with the discussed reduced-order model matches closely up to the third harmonic of the reference sinusoidal steady state of the considered nonlinear transmission line model. This result suggests that the extracted reduced-order models may be used to analyze second-order effects like harmonic distortion.

In a different series of tests, we applied a TPWL reduced-order model to compute intermodulation distortion for the considered op-amp example (using nonlinear capacitance models for MOSFETs). We generated two TPWL models: the first one of order $q = 39$ (with 39 linearization points), and the second one of order $q = 35$ (with 36 linearization points) for a sinusoidal training input (which was the same in both cases)

$$v_{in1}(t) = -v_{in2}(t) = (2 \cdot 20 \cdot \sin(2\pi f_1 t) + 24) \cdot 10^{-3}$$

where $f_1 = 1$ MHz. Then, the models were tested for the input signal being a sum of two sinusoids with different frequencies:

$$v_{in1}(t) = -v_{in2}(t) = [20 \cdot (\sin(2\pi f_1 t) + \sin(2\pi f_2 t)) + 24] \cdot 10^{-3}$$

where $f_1 = 1$ MHz and $f_2 = 1.2$ MHz, and the spectrum of the computed sinusoidal steady state was extracted using DFT. Tables VI and VII show the complex amplitudes of the main intermodulation products and the driving harmonics obtained with the TPWL reduced-order models and the full nonlinear model of the considered op-amp. The error shown in the tables is the relative error of the computed amplitude. The results indicate that the TPWL reduced model is able to effectively reproduce the intermodulation distortion effects in the considered case. At the same time, comparison between the two tables shows that

TABLE VII
COMPARISON OF THE MAIN INTERMODULATION HARMONICS OF THE
SINUSOIDAL STEADY STATE, COMPUTED USING A FULL ORDER
NONLINEAR MODEL AND THE REDUCED ORDER
TPWL MODEL ($q = 35$)

Har- monics	Full nonlinear model	Reduced order TPWL model	Error [%]
dc	-0.4967	-0.4921	-1.0
$f_2 - f_1$	0.0072 - 0.0223i	0.0081 - 0.0226i	2.4
$2f_1 - f_2$	0.0046 - 0.0037i	0.0056 - 0.0042i	18.8
f_1	-0.0667 + 0.1116i	-0.0667 + 0.1108i	-0.6
f_2	-0.0638 - 0.1003i	-0.0634 - 0.1005i	-0.1
$2f_2 - f_1$	0.0013 - 0.0050i	0.0019 - 0.0048i	-0.3
$2f_1$	0.0104 + 0.0008i	0.0108 + 0.0004i	3.7

the computed amplitudes of intermodulation products vary significantly depending on the applied TPWL model which suggests that in this case we may not further reduce the order of the macromodel.

VII. CONCLUSION

In this paper, we have proposed an efficient numerical approach for MOR and simulation of nonlinear dynamical systems. The results obtained for the examples of nonlinear circuits and a micromachined switch indicate that the presented method provides very good accuracy for different applications (and both single- and multiple-input systems). The method also proves to be characterized by low computational and memory requirements, therefore, providing a cost-efficient alternative for the nonlinear MOR techniques based on linear and quadratic models.

Although the algorithm in its current state has proved to be very effective, its performance still depends on a few parameters, which need to be adjusted more or less arbitrarily for a given application example. The discussed parameters are related mainly to the weighting procedure, as well as the method of selecting subsequent linearization points. Consequently, further developments of the proposed MOR algorithm are necessary in order to achieve its true robustness. Topics for further investigation include e.g.: 1) developing more sophisticated weighting procedures which would exploit available information on the original system in order to obtain more accurate TPWL models and/or to preserve stability or passivity of the original system; 2) incorporating *a posteriori* error estimation procedures to the algorithm of selecting the collection of linearization points; and 3) controlling approximation errors in the proposed fast simulation algorithm. A separate problem is to define what is an optimal training input for a given nonlinear system.

There are also many possible extensions of the presented MOR technique, which may include applying different types of bases in the reduced-order TPWL simulators or developing schemes for automatic model generation with multiple "training" inputs, which may allow one to extend the validity of the quasipiecewise-linear reduced-order model to inputs with different scales of amplitudes. One should also note that appli-

cation of the discussed TPWL MOR approach is not limited to single-input single-output or multiple-input multiple-output dynamical systems given in form (1). For instance, if we extend the weighting procedure to take into account not only the state space of the system, but also the space of input signals, we may be able to construct TPWL macromodels for systems with fully nonlinear input operators.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for suggestions that substantially improved the paper. In particular, the reviewers recommended extending the TPWL algorithm to the case with nonlinear capacitors. They also suggested examining intermodulation distortion.

REFERENCES

- [1] Y. Chen and J. White, "A quadratic method for nonlinear model order reduction," in *Proc. Int. Conf. Modeling and Simulation of Microsystems*, 2000, pp. 477–480.
- [2] J. Chen and S.-M. Kang, "An algorithm for automatic model-order reduction of nonlinear MEMS devices," in *Proc. IEEE Int. Symp. Circuits and Syst.*, vol. 2, 2000, pp. 445–448.
- [3] C. T. Dikmen, M. M. Alaybeyi, S. T. A. Atalar, E. Sezer, M. A. Tan, and R. A. Rohrer, "Piecewise linear asymptotic waveform evaluation for transient simulation of electronic circuits," in *Proc. Int. Symp. Circuits and Syst.*, vol. 2, Singapore, 1991, pp. 854–857.
- [4] 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.
- [5] E. Gallopoulos and Y. Saad, "Efficient solution of parabolic equations by Krylov approximation methods," *SIAM J. Sci. Stat. Comput.*, vol. 13, no. 5, pp. 1236–1264, 1992.
- [6] K. Glover, "All optimal Hankel-norm approximations of linear multi-variable systems and their L^∞ error bounds," *Int. J. Control*, vol. 39, no. 6, pp. 1115–1193, 1984.
- [7] E. J. Grimme, D. C. Sorensen, and P. Van Dooren, "Model reduction of state space systems via an implicitly restarted Lanczos method," *Numer. Algorithms*, vol. 12, no. 1–2, pp. 1–31, 1996.
- [8] E. J. Grimme, "Krylov projection methods for model reduction," Ph.D. dissertation, Univ. Illinois, Urbana-Champaign, IL, 1997.
- [9] J. K. Hale, *Ordinary Differential Equations*. New York: Wiley, 1969.
- [10] E. Hung, Y. Yang, and S. Senturia, "Low-order models for fast dynamical simulation of MEMS microstructures," in *Proc. IEEE Int. Conf. Solid-State Sensors and Actuators (Transducers '97)*, vol. 2, 1997, pp. 1101–1104.
- [11] R. Kao and M. Horowitz, "Timing analysis for piecewise linear Rsim," *IEEE Trans. Computer-Aided Design*, vol. 13, pp. 1498–1512, Dec. 1994.
- [12] K. Kundert, J. White, and A. Sangiovanni-Vincentelli, "A mixed frequency-time approach for finding the steady-state solution of clocked analog circuits," in *Proc. IEEE Custom Integrated Circuits Conf.*, 1988, pp. 6.2/1–6.2/4.
- [13] —, "A mixed frequency-time approach for distortion analysis of switching filter circuits," *IEEE J. Solid-State Circuits*, vol. 24, pp. 443–451, Apr., 1989.
- [14] A. Odabasioglu, M. Celik, and L. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, 1997, pp. 58–65.
- [15] J. R. Phillips, "Automated extraction of nonlinear circuit macromodels," in *Proc. Custom Integrated Circuit Conf.*, 2000, pp. 451–454.
- [16] —, "Model reduction of time-varying systems using approximate multipoint Krylov-subspace projectors," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, 1998, pp. 96–102.
- [17] D. Ramaswamy, "Automatic generation of macromodels for MicroElectroMechanical Systems (MEMS)," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, 2001.

- [18] M. Rewieński 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. Computer-Aided Design*, vol. 1, 2001, pp. 252–257.
- [19] —, "Improving trajectory piecewise-linear approach to nonlinear model order reduction for micromachined devices using an aggregated projection basis," in *Proc. 5th Int. Conf. Modeling and Simulation of Microsystems*, vol. 1, 2002, pp. 128–131.
- [20] J. S. Shamma, "Analysis and design of gain scheduled control systems," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, 1988.
- [21] 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. European Design and Test Conf.*, 1995, pp. 534–538.
- [22] D. J. Stilwell and W. J. Rugh, "Interpolation methods for gain scheduling," in *Proc. 37th IEEE Conf. Decision and Control*, vol. 3, 1998, pp. 3003–3008.
- [23] F. Wang and J. White, "Automatic model order reduction of a microdevice using the Arnoldi approach," in *Proc. Int. Mechanical Engineering Congr. and Exposition*, 1998, pp. 527–530.
- [24] J. K. White and A. Sangiovanni-Vincentelli, *Relaxation Techniques for the Simulation of VLSI Circuits*. Norwell, MA: Kluwer, 1987.
- [25] K. E. Willcox, J. D. Paduano, J. Peraire, and K. C. Hall, "Low order aerodynamic models for aeroelastic control of turbomachines," in *Proc. Amer. Inst. Aeronautics Astronautics Structures, Structural Dynamics, Materials Conf.*, vol. 3, 1999, pp. 2204–2214.



Michał Rewieński received the M.S. degree in computer science and the Ph.D. degree in electrical engineering from the Technical University of Gdansk, Gdansk, Poland, in 1998 and 2000, respectively.

He is currently a Research Assistant in the Computational Prototyping Group, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge. His research interests include developing fast numerical algorithms for full-wave modeling and simulation of electromagnetic systems in high frequencies, MOR

techniques for nonlinear dynamical systems, and large-scale parallel scientific computing.



Jacob White (A'88) received the B.S. degree from the Massachusetts Institute of Technology, Cambridge, in 1980, and the S.M. and Ph.D. degrees from the University of California, Berkeley, in 1983 and 1985, respectively, all in electrical engineering and computer science.

He was with the IBM T. J. Watson Research Center from 1985 to 1987, was the Analog Devices Career Development Assistant Professor at the Massachusetts Institute of Technology from 1987 to 1989, and was a 1988 Presidential Young Investigator.

He is currently a Professor at the Massachusetts Institute of Technology and an Associate Director of the Research Laboratory of Electronics. His current research interests are in numerical algorithms for problems in circuit, interconnect and microelectromechanical system design.

Prof. White was an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, from 1992 until 1996, and was Technical Program Chair of the International Conference on Computer-Aided Design in 1998.