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

Michał Rewieński, Jacob White

Abstract— **In this paper we present an approach to the nonlinear model reduction based on representing the 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 a nonlinear circuit and a micromachined fixed-fixed beam example. These examples demonstrate that the macromodels obtained with the proposed reduction algorithm are significantly more accurate than models obtained with linear or the recently developed quadratic reduction techniques. Finally, it is shown that the proposed technique is computationally inexpensive, and that the models can be constructed 'on-the-fly', to accelerate simulation of the system response.**

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 lead 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.

In this paper we present an approach to the nonlinear model reduction based on representing the 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 a circuit and a micromachined device example, 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 strategy and an approach for accelerating the needed simulation. Examples are examined in Section V, and in Section VI we present our conclusions.

M. Rewieński and J. White are with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA, E-mail: mrewiens@mit.edu, white@rle-vlsi.mit.edu

II. Examples of nonlinear dynamic systems

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

$$\begin{cases} \frac{dx(t)}{dt} = f(x(t)) + Bu(t) \\ y(t) = C^T x(t) \end{cases} \tag{1}$$

where $x(t) \in R^N$ is a vector of states, $f : R^N \to R^N$ is a nonlinear vector-valued function, B is an $N \times M$ input matrix, $u : R \to R^M$ is an input signal, C is an $N \times K$ output matrix and $y : R \to R^K$ is the output signal.

In this paper we will focus on two distinct examples of nonlinear systems which may be described by equations (1) and, due to their highly nonlinear dynamic behavior, illustrate well the challenges associated with nonlinear model order reduction.

The first example, considered by Chen et al. [1], is a nonlinear circuit shown in Figure 1. The circuit consists of resistors, capacitors and diodes with a constitutive 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 other example is a micromachined fixed-fixed beam structure shown in Figure 2. Following Huang et al. [8], the dynamic behavior of this coupled electro-mechanical-fluid system can be modeled with 1D Euler's beam equation and 2D Reynolds' squeeze film damping equation given below:

$$EI\frac{\partial^4 u}{\partial x^4} - S\frac{\partial^2 u}{\partial x^2} = F_{elec} + \int_0^w (p - p_a)dy - \rho\frac{\partial^2 u}{\partial t^2} \tag{2}$$

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

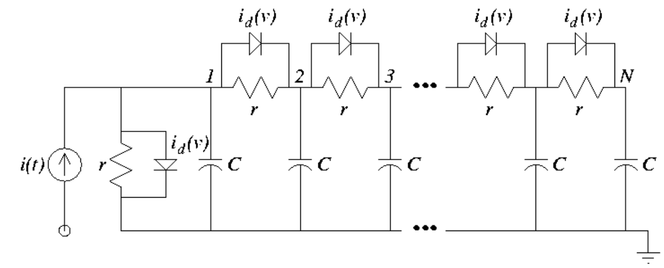

Fig. 1. The nonlinear circuit example.

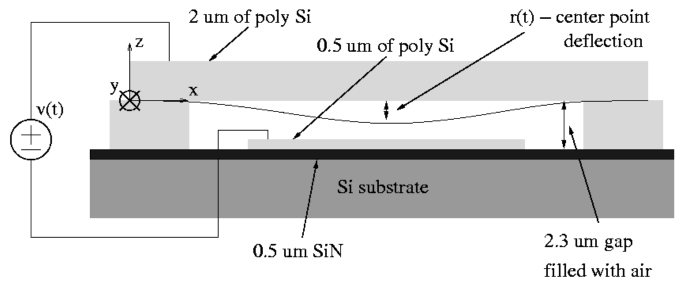

Fig. 2. Micromachined fixed-fixed beam (following Huang et al. [8]).

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

where x, y and z are as shown in Figure 2, 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 $p(x,y,t)$ is the pressure distribution in the fluid below the beam. The electrostatic force is approximated assuming nearly parallel plates and is given by $F_{elec} = -\frac{\epsilon_0 w v^2}{2u^2}$, where v is the applied voltage.

Spatial discretization of equations (2) and (3) using a standard finite-difference scheme (cf. [17]) leads to a large nonlinear dynamic system in form (1). In this case the state vector x consists of heights of the beam above the substrate (u) computed at the discrete grid points, their time derivatives, and the values of pressure below the beam. In this case 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. Figure 2).

III. Model Order Reduction for nonlinear systems

Suppose the initial dynamic system (1) is of order N, i.e. is described by N states. The main goal of model order reduction 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. Virtually all the numerical model order reduction strategies are based on the concept of projecting the states of the initial system onto a suitably selected reduced order state space. This may also be viewed as performing a change of variables:

$$x = Vz \tag{4}$$

where z is a q-th order projection of the state x (of order N) in the reduced order space and V is an $N \times q$ orthonormal matrix 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 order state space.

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

$$\left\{ \begin{array}{l} \frac{dz(t)}{dt} = V^T f(Vz(t)) + V^T Bu(t) \\ y(t) = C^T V z(t). \end{array} \right. \tag{5}$$

There are two key issues concerning representation (5) of the initial dynamic system (1). The first one is selecting a reduced basis V, such that system (5) provides good approximation of the initial system (1). For the linear case (i.e. if $f(\cdot)$ is a linear transformation), there are a number of methods for determining V. They include: selecting vectors from orthogonalized time-series data [8], computing singular vectors of the underlying differential equation Hankel operator [6] or examining Krylov subspaces [1], [2], [4], [7], [10], [11], [12], [15], [17]. 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 function $f(\cdot)$ [1], [2], [11], [17].

The other key issue in applying formulation (5) for reduced order modeling is finding a representation of $V^T f(V\cdot)$ which allows 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 $O(100,000)$ operations and is too costly. The simplest approximation for $f(\cdot)$, which allows $O(q)$ (not $O(N)$) storage and evaluation of $V^T f(V\cdot)$ is based on Taylor's expansion around the initial state (equilibrium point) 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. This approach leads to the following reduced order models proposed in [1], [2], [11] and [17]. For the linear case, the reduced order model (5) becomes:

$$\left\{ \begin{array}{l} \frac{dz(t)}{dt} = V^T f(x_0) + A_{0r} z + V^T Bu(t) \\ y(t) = C^T V z(t) \end{array} \right. \tag{6}$$

where $A_{0r} = V^T A_0 V$ is a $q \times q$ matrix. The quadratic reduced order model is given by [11][2]:

$$\left\{ \begin{array}{l} \frac{dz(t)}{dt} = V^T f(x_0) + A_{0r} z + \frac{1}{2} W_{0r} (z \otimes z) + V^T Bu(t) \\ y(t) = C^T V z(t) \end{array} \right. \tag{7}$$

[2] An alternative formulation of the quadratic reduced order model is presented in [1]. Both formulations give almost identical results.

where $W_{0r} = V^T W_0(V \otimes V)$ is a $q \times q^2$ matrix. In the above formulations, due to the fact that the reduced matrices are typically dense and must be represented explicitly, the cost of computing $V^T f(Vz)$ term and the cost of storing the reduced matrices 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 of nonlinearities [11], 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, and the computational cost is $O(q^5)$. In most cases it becomes inefficient to use so computationally expensive reduced order models.

IV. Piecewise-linear model order reduction

As described in the previous section, reduced order models based on Taylor series expansion become prohibitively expensive when the order of included nonlinearity becomes large. 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 a very limited range of inputs [17]. This leads us to proposing an approach towards model order reduction based on quasi-piecewise-linear approximations of nonlinear systems. 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 initial nonlinear system). The key issue in this approach is that we will be considering multiple linearizations around suitably selected states of the system, instead on relying on a single expansion around the initial state.

A. Piecewise-linear representation

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

$$\frac{dx}{dt} = f(x_i) + A_i(x - x_i) + Bu$$

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

$$\frac{dx}{dt} = \sum_{i=0}^{s-1} \tilde{w}_i(x) f(x_i) + \sum_{i=0}^{s-1} \tilde{w}_i(x) A_i(x - x_i) + Bu \qquad (8)$$

where $\tilde{w}_i(x)$ are weights depending on state x. (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):

$$\begin{cases} \frac{dz}{dt} = (A_r \cdot w(z)^T)z + \gamma \cdot w(z)^T + B_r u \\ y = C_r z \end{cases} \qquad (9)$$

where $B_r = V^T B$, $C_r = C^T V$, $A_r = [A_{0r} A_{1r} \ldots A_{(s-1)r}]$ and $A_{ir} = V^T A_i V$, $\gamma = [\gamma_0 \ldots \gamma_{s-1}] =$

$$\left[V^T(f(x_0) - A_0 x_0), \ldots, V^T(f(x_{s-1}) - A_{s-1} x_{s-1})\right]$$

and $[z_0, z_1, \ldots, z_{s-1}]$ are representations of linearization points $x_0, \ldots, x_{s-1}$ in the reduced basis:

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

Finally, $w(z) = [w_0(z) \ldots w_{s-1}(z)]$ is a vector of weights (norm $||w(z)|| = 1$ for all z). At this point we need to find a procedure for computing the weights w_i, given current state z and the linearization points z_i. We assume that weights w_i for the reduced models A_{ri} are computed based on the information about the distances $||z - z_i||$ of the linearization points from the current state z. We require that the 'dominant' model A_{jr} is the one corresponding to the linearization point z_j which is the closest to the current state of the system.

The following procedure of computing w_i ensures that the above requirement is satisfied:

1. For $i = 0, \ldots, (s-1)$ compute: $d_i = ||z - z_i||_2$. (Alternatively we may take $d_i = ||C_r(z - z_i)||_2$.)
2. Compute $m = \min\{d_i : i = 0, \ldots, (s-1)\}$.
3. For $i = 0, \ldots, (s-1)$ compute $w_i = (\exp(d_i)/m))^{-25}$.
4. Normalize w_i.

One may note that, in the above procedure, the distribution of weights changes rather 'sharply' 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 1. This provides a rationale for referring to model (9) as a *piecewise-linear* reduced order model of nonlinear system (1). Clearly, the procedure presented above provides only an example. Nevertheless, as shown in the following sections, it may be effectively used in practice.

B. Generation of the piecewise-linear model

So far it has not been discussed how to generate the weighted model given by (8) or, more specifically, how to select linearization points x_i. We may assume that linearization of a nonlinear system, generated at state x_i is valid or accurate for a given state x if this state is 'close enough' to the linearization point x_i, i.e. $||x - x_i|| < \epsilon$, which means that x lies within a ball (in an N-dimensional space) of radius ϵ and centered at x_i. Suppose we would like to cover an N-dimensional state space with such balls. (Therefore assuring that for any state we will find a valid linearized model.) Then, assuming e.g. that the state space is an N-dimensional hypercube: $[0;1] \times \ldots [0;1] \in R^N$, $N = 1000$ and $\epsilon = 0.1$, the total number of models to be generated would equal roughly 10^{1000}. This is clearly a totally infeasible approach, due to enormous memory and computational costs.

Instead of finding linearized models covering the entire N-dimensional state space we propose to generate a collection of models along a single, fixed trajectory of the sys-

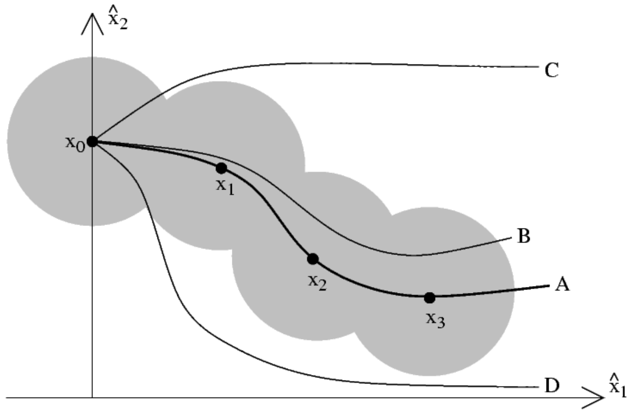

Fig. 3. Generation of the linearized models along a trajectory of a nonlinear system in a 2D state space.

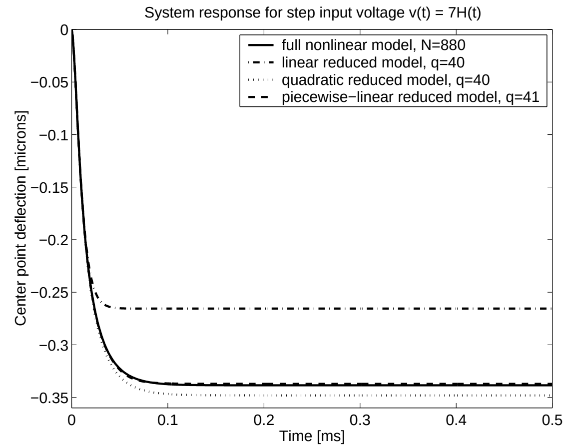

Fig. 4. Comparison of system response (micromachined beam example) computed with linear, quadratic and piecewise-linear reduced order models ($q = 40$ and $q = 41$) to the step input voltage $u(t) = 7H(t)$ ($H(t) \equiv 7$ for $t > 0$ and $H(t) \equiv 0$ for $t < 0$). The piecewise-linear model was generated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

tem.[3] This means we generate a trajectory by performing a single simulation of the nonlinear system for a fixed 'training' input. (Details of a fast simulation algorithm are presented in the following section.) This procedure is depicted in Figure 3. Given a training input signal $u(t)$ and initial state x_0 we proceed as follows: 1) We generate a linearized model around the initial state x_0 ($i = 0$); 2) We simulate the behavior of the nonlinear system while $||x - x_i|| < \delta$, i.e. while the current state x is close enough to the last linearization point; 3) We generate a new linearized model around $x_{i+1} = x$, take $i := i + 1$ and return to step 2). In this procedure we may fix the maximum number of models we want to generate. It should be stressed that this piecewise-linear approach is different from methods presented e.g. in [3] or [9], 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 Figure 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 initial nonlinear system (cf. curve B).[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. results in Section V). This case is also illustrated in Figure 4, which shows computational results for the example of a micromachined beam (cf. Section II). This figure presents the system response to a 7-volt step input voltage, computed with an 41-th order piecewise-linear reduced model of the device, generated for with an 8-volt step input training voltage. (The model was generated using the fast algorithm proposed in Section IV-C.) One 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 around 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 C and D in Figure 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 Figure 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 response of the nonlinear system. Now, if we generate the piecewise-linear model with a 9-volt training input (cf. Figure 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

[3] The 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. [14], [16].

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

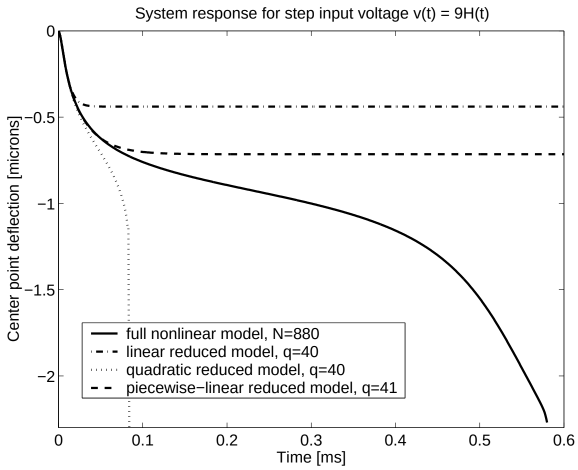

Fig. 5. Comparison of system response (micromachined beam example) computed with linear, quadratic and piecewise-linear reduced order models ($q = 40$ and $q = 41$) to the step input voltage $u(t) \equiv 9$ ($t > 0$). The piecewise-linear model was generated for the 7-volt step input voltage.

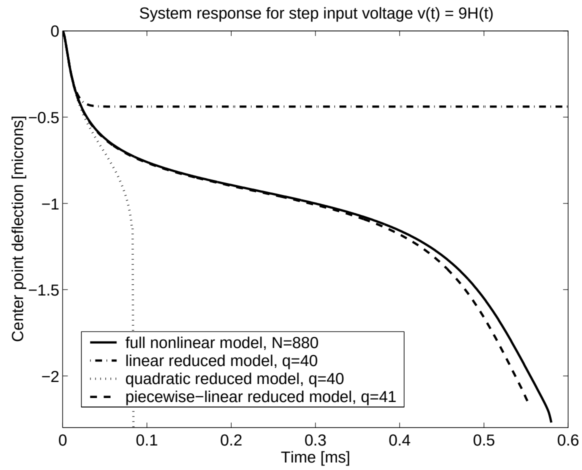

Fig. 6. Comparison of system response (micromachined beam example) computed with linear, quadratic and piecewise-linear reduced order models ($q = 40$ and $q = 41$) to the step input voltage $u(t) \equiv 9$ ($t > 0$). The piecewise-linear model was generated for the 9-volt step input voltage.

beam is pulled down to the substrate), which is of particular importance in applications [8]. 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.

One may note that the proposed method of generating the piecewise-linear model of a nonlinear dynamic system requires performing simulation of the initial nonlinear system (1) which may be very costly, due to the initial 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. This means we may perform a much faster approximate simulation (performed e.g. in the reduced order state space) and leads us to the following algorithm.

C. Fast generation of piecewise-linear models

This section presents an approach towards efficient generation of piecewise-linear reduced order model described above. The proposed numerical algorithm proceeds in the two stages: 1) Generation of the reduced basis, used to represent approximately the state space vectors (x); 2) Approximate simulation of the nonlinear system response to the training input, using the reduced basis and piecewise-linear approximation of nonlinear function $f(x)$ along a trajectory of the nonlinear dynamic system (1). This approach shares features with reduced basis methods for solving parabolic problems [5]. Below these two stages are described in more detail.

C.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. We consider the linearization of the dynamic system (1) around the initial state x_0:

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

where A_0 is the Jacobian of $f(x)$, evaluated at $x = x_0$. We construct an orthogonal basis $\tilde{V} = \{v_1, \dots, v_{l \cdot M}\}$ in the l-th order Krylov subspace:

$$K_l(A_0^{-1}, A_0^{-1}B) = \text{span}\{A_0^{-1}B, \dots, A_0^{-l}B\},$$

using the Arnoldi algorithm [17] (or block Arnoldi algorithm [13] 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) [11].

2. We orthonormalize the initial state vector x_0 with respect to the columns of $\tilde{V}$ and obtain vector v_{lM+1}. To this end we may use e.g. the SVD algorithm.

3. We 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 the initial state of the system is zero, then steps 2 and 3 become unnecessary.) Exact representation of the initial state assures 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]

[5] We presented only the simplest (and the least computationally expensive) algorithm of 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.

C.2 Fast approximate simulation

The second stage of the proposed MOR algorithm may be summarized in the following steps:
1. Using basis V construct and save a reduced-order model of dynamic system (1), linearized around state x_i:

$$\begin{cases} \frac{dz}{dt} = V^T A_i V z + V^T f(x_i) - V^T A_i x_i + V^T B u \\ y = C^T V z \end{cases} \tag{11}$$

where z is a reduced order approximation of state vector x ($x \approx Vz$). Initially $x_i = x_0$, where x_0 is the initial state. This step requires computation of the Jacobian A_i of $f(x)$ (at $x = x_i$) in the non-reduced state space.
2. Simulate reduced order linear dynamic system (11), 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, i.e. when:

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

where α is an appropriately selected constant (cf. the comments below).
3. Change the linearization point from x_i to $x_{i+1} = Vz(t_j)$ ($i := i + 1$) and return to Step 1 of the algorithm.

There is an important issue concerning the piecewise-linear MOR algorithm proposed above. In order to be able to reproduce nonlinear effects in the behavior of the dynamic 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 the following. First, for a given input signal, we perform a reduced order simulation of the linearized dynamic system, with linearization around 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 dynamic 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 $\alpha = d/10$.

D. Fast piecewise-linear simulator

One should also note that the MOR algorithm presented in Section IV-C.2 may be used as a fast simulator for nonlinear dynamic systems. The simulator (as described in Section IV-C) has been implemented for the example of a nonlinear circuit given in Section II. Selected results of numerical tests are presented below.

Figure 7 shows the output voltage $v_1(t)$ for a step input current, computed using full order linear and quadratic models as well as the proposed piecewise-linear simulator. The reference result is computed with a simulator using a full nonlinear model. In the simulation the number of time-steps was 1000 ($T = 10$, $\Delta T = 0.01$), the piecewise-linear simulator used the reduced basis of order $q = 30$ (the original system size was $N = 1500$) and the linearization point changed 20 times (i.e. it used 21 different linear models during the simulation). The output voltage obtained with this method matches very well the reference result (the curves overlap almost perfectly). Table I shows the relative error between the voltage $v = [v(0), v(\Delta T), \ldots, v(T)]$ computed with linear, quadratic and piecewise-linear 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. Unlike the two other it is also able to accurately match the steady state of the system.

Table II compares performance of the full nonlinear simulator of the considered nonlinear circuit and the proposed piecewise-linear solver, using reduced basis computations for three different inputs. In order to assure appropriate accuracy, in the case of the step input the order of the reduced basis applied equaled $q = 30$ ($N = 1500$), and for the sinusoidal input – $q = 10$ ($N = 100$). Both algorithms were implemented in Matlab, therefore the execution times should be used for comparison only. High performance

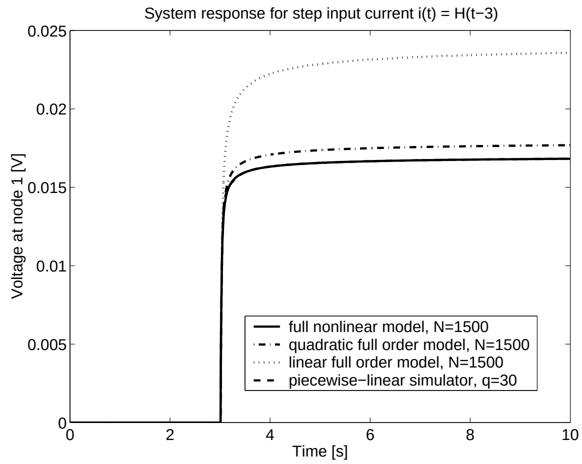

Fig. 7. Comparison of system response computed with linear, quadratic, full nonlinear and piecewise-linear models to the step input current $i(t) = H(t-3)$ ($H(t) \equiv 1$ for $t > 3$ and $H(t) \equiv 0$ for $t < 3$). $N = 1500$.

TABLE I

QUALITY OF APPROXIMATION FOR LINEAR, QUADRATIC AND PROPOSED PIECEWISE-LINEAR MODEL FOR THE STEP INPUT CURRENT. $v \equiv v(t)$ IS THE COMPUTED OUTPUT VOLTAGE AT NODE 1, v_{ref} IS THE REFERENCE OUTPUT VOLTAGE COMPUTED WITH FULL NONLINEAR MODEL.

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

TABLE II

Comparison of simulation times for the full order nonlinear simulator and the proposed piecewise-linear reduced order simulator.

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

implementations would most likely give significantly lower absolute execution times. The tests were performed in a Linux workstation with Pentium III Xeon processor. It is apparent that for either small or large initial problem sizes the piecewise-linear simulator is significantly faster than the full nonlinear solver. For $N = 1500$ a 100-fold acceleration in computation time was achieved.

V. Computational results

A. Model verification – transient simulations

This section presents results of computations using piecewise-linear 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}, A_{1r}, ..., $A_{(s-1)r}$) 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 for not only for the input it was generated with, but also for other inputs.

This verification was done experimentally. We considered our nonlinear circuit for $N = 100$ and generated a reduced order piecewise-linear model of order $q = 10$ using a step input $i(t) = H(t-3)$. For this example, the linearization point changed 4 times, therefore our model consisted of 5 reduced order matrices A_{0r}, ..., A_{4r}. The reduced order model was tested for different inputs, including the step input used to generate it. Three of the results are shown in Figures 8-10. Figure 8 shows the result for the step input (the same input as used for model extraction). Figures 9 and 10 show the reduced order simulation time for a cosinusoidal input and an exponential input, respectively. In both cases the output voltages obtained with the piecewise-linear 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 initial nonlinear circuit.

Figure 11 provides an analogous test for the example of a micromachined fixed-fixed beam described in Section

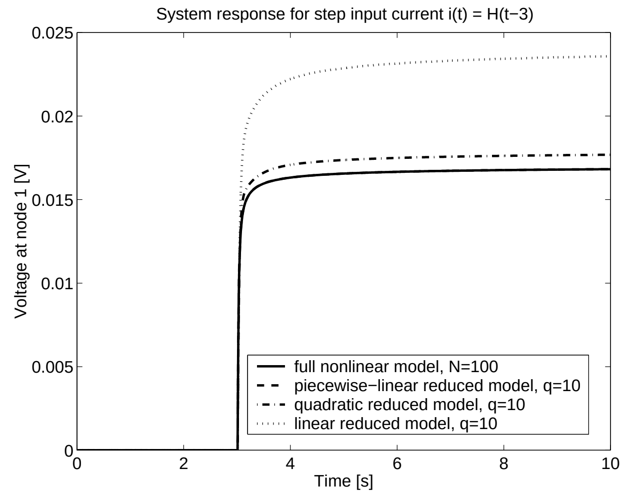

Fig. 8. Comparison of system response (nonlinear circuit example) computed with linear, quadratic and piecewise-linear reduced order models (of order $q = 10$) for the step input current $i(t) = H(t-3)$ ($H(t) \equiv 1$ for $t > 3$ and $H(t) \equiv 0$ for $t < 3$).

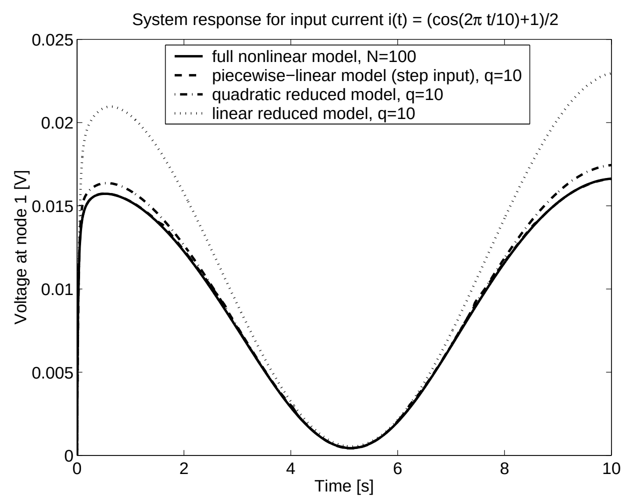

Fig. 9. Comparison of system response (nonlinear circuit example) computed with linear, quadratic and piecewise-linear reduced order models (of order $q = 10$) for the input current $i(t) = (\cos(2\pi t/10) + 1)/2$.

II. In this case the reduced order model ($q = 41$) was generated for the step 8-volt training input voltage. (The model used 9 linearization points.) Then it was tested for a cosinusoidal input with a 7-volt amplitude. Once again, the transient obtained with the proposed model matches very accurately the reference result obtained with the full nonlinear model of order $N = 880$.

Figures 8-11 also provide a comparison of the proposed piecewise-linear reduced order model with linear and quadratic reduced models, generated using methods described in [1], [11] and [17]. It is apparent from the graphs that the piecewise-linear reduced order model gives significantly more accurate results than the linear and quadratic reduced order models using Taylor expansions around the initial state. It should be stressed at this point that all models (linear, quadratic and piecewise-linear) were of the

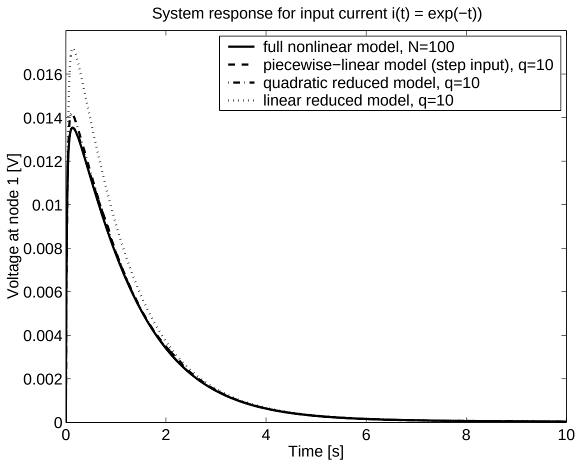

Fig. 10. Comparison of system response (nonlinear circuit example) computed with linear, quadratic and piecewise-linear reduced order models (of order $q = 10$) for the input current $i(t) = \exp(-t)$.

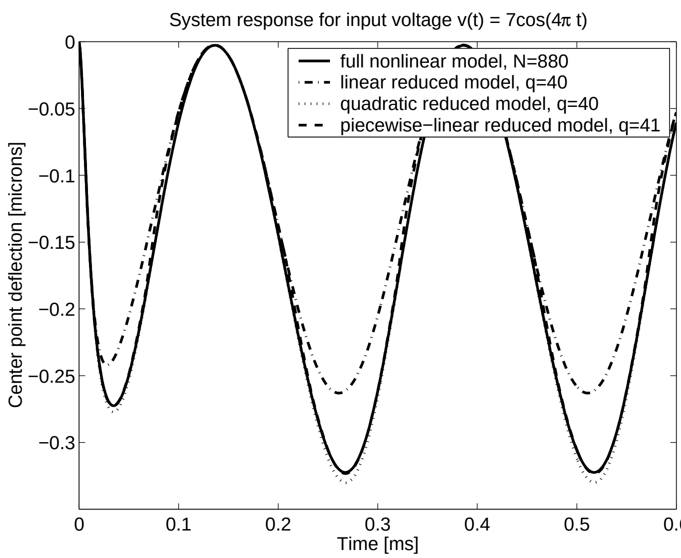

Fig. 11. Comparison of system response (micromachined beam example) computed with linear, quadratic and piecewise-linear reduced order models (of order $q = 40$ and $q = 41$) for the input voltage $u(t) = 7\cos(4\pi t)$. The piecewise-linear model was generated for the 8-volt step input voltage.

same order and, moreover, applied the same basis V (obtained with the procedure described in Section IV-C.1).

B. Performance and complexity of the MOR algorithm

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

TABLE III

Comparison of the times of generation of the reduced model and reduced order simulations for the quadratic and piecewise-linear MOR techniques. The initial problem had size $N = 1500$. The reduced model had size $q = 30$. The tests were run for the nonlinear circuit example.

MOR method	Model generation time [s]	Simulation time [s]
linear MOR	44.8	1.18
quadratic MOR	2756.5	31.5
piecewise-linear MOR	80.7	8.0

The memory complexity of the piecewise-linear 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 for the linear reduced order simulator (which is $O(q^2)$). The 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 our example (cf. Figures 8–11, $s = 5 = q/2$ or $s = 9 < q/4$), so in fact the memory used by the piecewise-linear algorithm equaled roughly only half (or a quarter) of the memory used by the quadratic solver.

VI. Conclusions

In this paper we have proposed an efficient numerical approach towards automatic model order reduction and simulation of nonlinear systems. The results obtained for the examples of a nonlinear circuit and a micromachined beam indicate that this method provides good accuracy for different applications. 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, a number of its aspects require further investigation, including the procedure of merging (weighting) the linearized models or the method of selecting linearization points. There are also many possible extensions of the presented technique, which may include application of multiple reduced bases (instead of a single basis generated at the initial state) in the reduced order piecewise-linear simulators or developing schemes for automatic model generation with multiple 'training' inputs, which may allow one to extend the validity of the quasi-piecewise-linear reduced order model to inputs with different scales of amplitudes.

It should be stressed that application of the discussed piecewise-linear reduced order approach is not limited to the class of SISO or MIMO dynamic systems found in circuit or MEMS modeling. The proposed technique may be easily extended for use in macromodeling of second order systems arising in e.g. coupled domain problems involving micromachined electromechanical devices.

VII. Acknowledgements

This work was sponsored by the Singapore-MIT Alliance and the DARPA neoCAD program.

References

[1] Y. Chen, J. White, "A Quadratic Method for Nonlinear Model Order Reduction", in proceedings of the International Conference on Modeling and Simulation of Microsystems, pp. 477-480, 2000.

[2] J. Chen, S-M. Kang, "An algorithm for automatic model-order reduction of nonlinear MEMS devices", in proceedings of the IEEE International Symposium on Circuits and Systems, pp. 445-8, vol. 2, 2000.

[3] C. T. Dikmen, M. M. Alaybeyi, S. Topcu, A. Atalar, E. Sezer, M. A. Tan, R. A. Rohrer, "Piecewise Linear Asymptotic Waveform Evaluation for Transient Simulation of Electronic Circuits", in proceedings of the International Symposium on Circuits and Systems, pp. 854-857, Singapore, 1991.

[4] P. Feldmann, R. W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process", IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 14, pp. 639-649, 1995.

[5] E. Gallopoulos, Y. Saad, "Efficient solution of parabolic equations by Krylov approximation methods", SIAM Journal of Scientific and Statistical Computing, Vol. 13, No. 5, pp. 1236-1264, 1992.

[6] K. Glover, "All Optimal Hankel-norm Approximations of Linear Multivariable Systems and Their L^∞ error Bounds", International Journal of Control, Vol. 39, No. 6, pp. 1115-1193, 1984.

[7] E. J. Grimme, D. C. Sorensen, P. Van Dooren, "Model Reduction of State Space Systems via an Implicitly Restarted Lanczos Method", Numerical Algorithms, vol. 12, no. 1-2, pp. 1-31, 1996.

[8] E. Huang, Y. Yang, S. Senturia, "Low-Order Models For Fast Dynamical Simulation of MEMS Microstructures", in proceedings of the IEEE International Conference on Solid State Sensors and Actuators (Transducers '97), Vol. 2, pp. 1101-1104, 1997.

[9] R. Kao, M. Horowitz, "Timing analysis for Piecewise Linear Rsim", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 13, No. 12, pp. 1498-1512, 1994.

[10] A. Odabasioglu, M. Celik, L. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm", in proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp. 58-65, 1997.

[11] J. R. Phillips, "Automated extraction of nonlinear circuit macromodels", in proceedings of the Custom Integrated Circuit Conference, pp. 451-454, 2000.

[12] J. R. Phillips, "Model reduction of time-varying systems using approximate multipoint Krylov-subspace projectors", in proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp. 96-102, 1998.

[13] D. Ramaswamy, "Automatic Generation of Macromodels for MicroElectroMechanical Systems (MEMS)", Ph.D. thesis, Massachusetts Institute of Technology, 2001.

[14] J. S. Shamma, "Analysis and Design of Gain Scheduled Control Systems", Ph.D. thesis, Massachusetts Institute of Technology, 1988.

[15] L. M. Silveira, M. Kamon, J. White, "Efficient Reduced-Order Modeling of Frequency-Dependent Coupling Inductances associated with 3-D Interconnect Structures", in proceedings of the European Design and Test Conference, pp. 534-538, 1995.

[16] D. J. Stilwell, W. J. Rugh, "Interpolation Methods for Gain Scheduling", in proceedings of the 37th IEEE Conference on Decision and Control, pp. 3003-3008, 1998.

[17] F. Wang, J. White, "Automatic model order reduction of a microdevice using the Arnoldi approach", in proceedings of the International Mechanical Engineering Congress and Exposition, pp. 527-530, 1998.