

# Parameterized Model Order Reduction of Nonlinear Dynamical Systems

Brad Bond

Research Laboratory in Electronics,  
Massachusetts Institute of Technology,  
77 Massachusetts Ave, Cambridge, MA, 02139  
Email: bnbond@mit.edu

Luca Daniel

Research Laboratory in Electronics,  
Massachusetts Institute of Technology,  
77 Massachusetts Ave, Cambridge, MA, 02139  
Email: luca@mit.edu

**Abstract**—In this paper we present a parameterized reduction technique for non-linear systems. Our approach combines an existing non-parameterized trajectory piecewise linear method for non-linear systems, with an existing moment matching parameterized technique for linear systems. Results and comparisons are presented for two examples: an analog non-linear circuit, and a MEM switch.

## I. INTRODUCTION

The presence of several non-linear analog circuits and Micro-Electro-Mechanical (MEM) components in modern mixed signal System-on-Chips (SoC) makes the fully automatic synthesis and optimization of such systems an extremely challenging task. The availability of techniques for generating Parameterized Reduced Order Models (PROM) of non-linear dynamical systems could serve as a first step toward the automatic and accurate characterization of geometrically complex components and subcircuits, eventually enabling their synthesis and optimization.

Several Parameterized Model Order Reduction techniques have been introduced in literature in the past few years. Some are based on statistical performance evaluation [1], [2], [3], [4], others are based on moment matching techniques [5], [6], [7], [8], [9], [10], or on Truncated Balance Realization (TBR) techniques [11], and on quasi-convex optimization techniques [12]. However, to our knowledge, almost all existing PMOR techniques apply only to linear systems. Very few, such as [4], also apply to non-linear systems.

Several non-parameterized Model Order Reduction (MOR) approaches are available for non-linear systems. For instance, the reduction of weakly non-linear systems has been shown using Volterra series and moment matching techniques [13], [14], [15], [16], [17], [18]. The reduction of strongly non-linear systems has been shown using Trajectory Piece-Wise Linear (TPWL) with moment matching techniques [19], [20], [21], [22], TPWL with TBR techniques [23], and trajectory PieceWise Polynomial (PWP) with moment matching techniques [24], [25], [26]. However, to our knowledge, all existing model order reduction techniques for non-linear systems apply only to non-parameterized systems.

The key contribution of this paper is the combination of an existing non-parameterized trajectory piecewise linear method for non-linear systems, with an existing moment matching

parameterized technique for linear systems, hence introducing a parameterized reduction technique for non-linear systems.

The rest of the paper is organized as follows: In Section II we briefly review non-parameterized and parameterized reduction for linear systems using moment matching, and the trajectory piecewise linear reduction for non-linear non-parameterized systems. In Section III we introduce our method combining the two approaches. Three alternatives are presented. In Section IV we demonstrate and compare our techniques modeling an analog circuit and a MEM switch.

## II. BACKGROUND

### A. Moment Matching MOR for Linear Systems

Consider a linear system

$$E \frac{dx}{dt} = Ax(t) + bu(t), \quad y(t) = c^T x(t), \quad (1)$$

where the state  $x$  has a very large order  $N$ . One approach to reduce the order of the system is to use an orthonormal projection matrix  $V$  such that  $x \approx V\hat{x}$ , where  $\hat{x}$  is size  $q \ll N$ , obtaining

$$\hat{E} \frac{d\hat{x}}{dt} = \hat{A}\hat{x}(t) + \hat{b}u(t), \quad \hat{y}(t) = \hat{c}^T \hat{x}(t), \quad (2)$$

where  $\hat{E} = V^T E V$ ,  $\hat{A} = V^T A V$ ,  $\hat{b} = V^T b$ , and  $\hat{c}^T = c^T V$  [27]. The projection matrix is carefully constructed to preserve the input/output relationship (e.g. transfer function) of the system. For instance a projection matrix chosen such that

$$\text{colspan}(V) \supseteq \text{span}\{b_M, Mb_M, M^2 b_M, \dots, M^{q-1} b_M\}, \quad (3)$$

would produce a reduced system matching the first  $q$  moments of the Taylor Series expansion in the Laplace variable  $s$  of the large system transfer function [28],

$$x = [sI - M]^{-1} b_M u = \sum_{m=0}^{\infty} s^m M^m b_M u, \quad (4)$$

where  $M = A^{-1}E$  and  $b_M = -A^{-1}b$ .

### B. Moment Matching PMOR for Linear Systems

Consider a linear system whose dynamical descriptor matrices in Laplace domain are functions of the Laplace frequency variable  $s$ , and of some other geometrical parameters,  $s_1, \dots, s_\mu$ ,

$$E(s, s_1, \dots, s_\mu)x = bu, \quad y = c^T x. \quad (5)$$

Using for instance a polynomial fitting technique,

$$E(s_1, \dots, s_\mu) = E_0 + \sum_i s_i E_i + \sum_{h,k} s_h s_k E_{h,k} + \dots$$

and introducing additional parameters  $\tilde{s}$  as shown in [8], [9],

$$\tilde{E}_i = \begin{cases} E_i & i = 0, \dots, \mu \\ E_{h,k} & h = 1, \dots, \mu; k = 1, \dots, \mu \\ \dots & \dots \end{cases}$$

$$\tilde{s}_i = \begin{cases} s_i & i = 1, \dots, \mu \\ s_h s_k & h = 1, \dots, \mu; k = 1, \dots, \mu \\ \dots & \dots \end{cases}$$

one can approximate the parameterized system as

$$[\tilde{E}_0 + \tilde{s}_1 \tilde{E}_1 + \dots + \tilde{s}_p \tilde{E}_p]x = bu \quad (6)$$

Next, from the moments of the multivariable Taylor series expansion

$$x = \sum_{m=0}^{\infty} (\tilde{s}_1 M_1 + \dots + \tilde{s}_p M_p)^m b_M u, \quad (7)$$

a projection matrix,  $V$ , can be constructed such that

$$\text{colspan}(V) \supseteq \text{span} \left\{ \bigcup_{m=0}^{m_q} \bigcup_{k_2=0}^{m-(k_p+\dots+k_3)} \dots \bigcup_{k_p=0}^m F_{k_2, \dots, k_p}^m b_M \right\} \quad (8)$$

where  $b_M = \tilde{E}_0^{-1} b$ ,  $M_i = -\tilde{E}_0^{-1} \tilde{E}_i$ , and

$$F_{k_2, \dots, k_p}^m = \begin{cases} 0 & \text{if } k_i \notin \{0, 1, \dots, m\} \quad i = 2, \dots, p \\ 0 & \text{if } k_2 + \dots + k_p \notin \{0, 1, \dots, m\} \\ I & \text{if } m = 0 \\ M_1 F_{k_2, \dots, k_p}^{m-1} + \dots + M_p F_{k_2, \dots, k_p}^{m-1} & \text{otherwise} \end{cases}$$

In [8] it is shown that the reduced system

$$[V^T \tilde{E}_0 V + \tilde{s}_1 V^T \tilde{E}_1 V + \dots + \tilde{s}_p V^T \tilde{E}_p V] \hat{x} = V^T b u \quad (9)$$

$$y = c^T V \hat{x},$$

assembled using such a projection matrix with  $q$  total columns, matches the first  $q$  moments (corresponding to the first  $m_q$  derivatives in each parameter) of the large system (6). In the same article it is also noted that, for a large number of parameters  $p$ , and a modest number of derivatives  $m_q$  matched for each parameter, this method generates systems of order  $q = O(p^{m_q})$ .

### C. TPWL for MOR of non-linear systems

Consider a non-linear system in the form

$$E \frac{dx}{dt} = A(x(t)) + bu(t), \quad y = c^T x(t). \quad (10)$$

The Trajectory Piece-Wise Linear (TPWL) method [21] constructs a collection of local linearizations,  $A_i x + K_i$ , of the non-linear function  $A(x(t))$ , around each of  $k$  linearization states  $x_i$ . The non-linear system is then approximated by weighted combinations of the linear models,

$$E \frac{dx}{dt} = \sum_{i=1}^k w_i(x, X) [A_i x + K_i] + bu(t) \quad (11)$$

where  $w_i(x, X)$  are some weighting functions, which depend on the state  $x$ , and on the  $k$  linearization points  $X = [x_1, x_2, \dots, x_k]$ . Since covering the entire state space with linear approximations would be extremely expensive, in the TPWL method the non-linear system is simulated with some ‘‘typical’’ training inputs, and only the trajectories of the state excited by those inputs are populated by linearized models ( $A_i, K_i$ ). One major drawback of this method is that the accuracy of the reduced model can be highly dependent on the ‘‘richness’’ of the inputs chosen for training. Furthermore, exact training trajectories can be in general expensive to generate since they involve simulating the full non-linear system. In [20], [21] it was noted that an alternative less expensive option for generating approximate training trajectories is to simulate only the partially created reduced model, and to generate and add a new linearized and reduced model  $\hat{A}_i$  whenever the reduced state  $\hat{x}(t)$  strays more than some distance  $\delta$  from all the reduced centers  $\hat{x}_i$  of the previous linearizations, i.e. when

$$\min_i \|\hat{x}(t) - \hat{x}_i\| > \delta. \quad (12)$$

In all our experiments, so far, we have verified that approximate training trajectory based approaches generate models as accurate as exact training trajectory ones, hence we will adopt this technique hereafter in this paper.

In the TPWL method, an orthonormal projection matrix  $V$  is created by assembling all the column vectors produced by applying (3) on each of the linearized models  $A_i$ . The final reduced system is then

$$\hat{E} \frac{d\hat{x}}{dt} = \sum_{i=1}^k \hat{w}_i(\hat{x}, \hat{X}) [\hat{A}_i \hat{x}(t) + \hat{K}_i] + \hat{b} u(t) \quad (13)$$

where  $\hat{A}_i = V^T A_i V$ ,  $\hat{K}_i = V^T K_i$ ,  $\hat{b} = V^T b$ ,  $\hat{E} = V^T E V$ ,  $\hat{x} = V^T x$ ,  $\hat{X} = [\hat{x}_1, \dots, \hat{x}_k]$ , and  $\hat{x}_i = V^T x_i$ . The relative weights,  $\hat{w}_i(\hat{x}, \hat{X})$ , of each linear model vary dynamically as the state evolves. One example of possible weighting functions is the one used in [21],

$$\hat{w}_i(\hat{x}, \hat{X}) = \frac{\exp\left(\frac{-\beta \|\hat{x} - \hat{x}_i\|}{m}\right)}{\sum_{i=1}^k \exp\left(\frac{-\beta \|\hat{x} - \hat{x}_i\|}{m}\right)}, \quad (14)$$

where  $\beta$  is some constant (typically we used 25), and  $m = \min_i \|\hat{x} - \hat{x}_i\|$ .

### III. TPWL-PMOR

A system possessing a non-linear dependence on both the state and some parameters may be of the form

$$E \frac{dx}{dt} = A(x(t), s_1, s_2, \dots, s_\mu) + bu(t) \quad y(t) = c^T x(t). \quad (15)$$

Using a collection of local linearizations around different states  $x_i$ , as in II-C and [21], one obtains

$$E \frac{dx}{dt} = \sum_{i=1}^k w_i(x, X) [A_i(s_1, \dots, s_\mu) x(t) + K_i(s_1, \dots, s_\mu)] + bu(t). \quad (16)$$

Using then, for instance, polynomial fitting, and introducing additional parameters  $\tilde{s}_j$  as in II-B and [8], [9] one obtains

$$E \frac{dx}{dt} = \sum_{i=1}^k \sum_{j=1}^p w_i(x, X) \tilde{s}_j [\tilde{A}_{ij} x(t) + \tilde{K}_{ij}] + bu(t) \quad (17)$$

The resulting system is now described as a collection of  $k$  linear systems linearly dependent on  $p$  parameters. Hence one can use standard linear model order reduction techniques. For instance, using an orthonormal projection matrix  $V$ , the reduced system is

$$\begin{aligned} \hat{E} \frac{d\hat{x}}{dt} &= \sum_{i=1}^k \sum_{j=1}^p \hat{w}_i(\hat{x}, \hat{X}) \tilde{s}_j [\hat{A}_{ij} \hat{x}(t) + \hat{K}_{ij}] + \hat{b}u(t) \\ y &= \hat{c}^T \hat{x}(t) \end{aligned} \quad (18)$$

where  $\hat{E} = V^T E V$ ,  $\hat{A}_{ij} = V^T \tilde{A}_{ij} V$ ,  $\hat{K}_{ij} = V^T \tilde{K}_{ij}$ ,  $\hat{b} = V^T b$ ,  $\hat{c}^T = c^T V$ ,  $\hat{x}(t) = V^T x(t)$ ,  $\hat{X} = [\hat{x}_1, \dots, \hat{x}_k]$ , and  $\hat{x}_i = V^T x_i$ .

In order to complete the procedure, two algorithms remain to be specified: how to choose the linearization points  $x_i$ , and how to construct the projection matrix  $V$ . Alternatives for each of such two tasks will now be discussed in details in Sections III-A and III-B. Combining different alternatives will then produce in Section III-C our three proposed Non-Linear Parameterized Model Order Reduction (NLP MOR) algorithms.

#### A. Choosing Linearization Points

In the standard TPWL, the state linearization points are chosen along the state trajectories generated by applying typical training inputs. We can use here the same idea, and we will refer to it as “training at a single point in the parameter space”. However, one may notice that different trajectories may be generated not only by varying the training inputs, but also by training at different points in the parameter space. We will consider this option as well, and denote it as “training at multiple points in the parameter space”.

The additional training trajectories introduced by training at multiple points in the parameter space will increase the total number  $k$  of linearized models, however it will not affect the order  $q$  of the reduced system. Since the weighting functions in equation (18) are typically non-zero for just a few models at any particular time, a larger set of linearized models is not expected to significantly affect simulation time when using the reduced model.

#### B. Constructing Projection Matrix $V$

When constructing the orthonormal projection Matrix  $V$ , one could simply ignore the dependency on geometrical parameters, and assemble the columns of  $V$  as in the original TPWL method, using the MOR vectors produced by applying (3) on each of linearized models  $A_i$ . This procedure would produce reduced models that match one single moment with respect to each geometrical parameter. Hence we will denote this procedure as a simple “MOR moment matching in  $V$ ”.

However, as an alternative, one could also choose to use a multivariable Taylor series expansion about the Laplace variable  $s$  and each of the geometrical parameters  $\tilde{s}_j$ , and assemble the columns of  $V$  using the PMOR expression in (8). This procedure would produce reduced models that match more than one moment with respect to each set of geometrical parameter values. Hence we will denote this procedure as a “PMOR moment matching in  $V$ ”.

Matching multiple moments about multiple expansion points in the parameter space, for every linearized model may quickly increase the number of columns  $q$  of the projection matrix  $V$ , hence possibly affecting the reduced model simulation cost. In order to contain the final order  $q$  of the reduced model, one can use a Singular Value Decomposition (SVD) on  $V$ , and select its first  $q$  most relevant singular vectors, identified by its largest  $q$  singular values. Such SVD is relatively inexpensive since, although  $V$  can be very tall, it will be in most cases not very wide.

#### C. Proposed TPWL-PMOR Algorithms

Combining the parameterization options in sections III-A and III-B gives rise to four different algorithms, shown in Table I, for Trajectory PieceWise Linear Parameterized Model Order Reduction (TPWL-PMOR) of a non-linear system.

TABLE I  
FOUR OPTIONS FOR PARAMETERIZED REDUCTION OF NON-LINEAR SYSTEMS.

	Training at Single Point in Parameter Space	Training at Multiple Points in Parameter Space
MOR		
Moment Matching in $V$	TPWL	Algorithm 2
PMOR		
Moment Matching in $V$	Algorithm 1	Algorithm 3

**TPWL.** One can note that using a simple non-parameterized “MOR moment matching in  $V$ ”, while “training at a single point in parameter space”, is equivalent to applying the already available non-parameterized TPWL algorithm [21] on the parameterized system (17). Hence we will denote this algorithm as “TPWL”, since it is just a very trivial extension of that well known algorithm.

---

**Algorithm 1** Trained at Single Parameter Point with PMOR Moment Matching Projection Matrix  $V$ 

---

```
1:  $i \leftarrow 1$ 
2: for all training inputs do
3:    $t \leftarrow t_{\text{initial}}$ 
4:   while  $t < t_{\text{final}}$  do
5:     if  $\min_i \|\hat{x}(t) - \hat{x}_i\| > \delta$  then
6:       Linearize at current state  $\hat{x}_i \leftarrow \hat{x}(t)$ 
7:       Generate PMOR vectors (8)
8:       Orthogonalize and add them to  $V$ 
9:       Update reduced model using (18)
10:       $i \leftarrow i + 1$ 
11:     end if
12:     Simulate reduced model to next time step  $t$ 
13:   end while
14: end for
15: Perform SVD on projection matrix  $V$ 
16: Update reduced model using (18)
```

---

---

**Algorithm 2** Trained at Multiple Parameter Values with MOR Moment Matching Projection Matrix  $V$ 

---

```
1:  $i \leftarrow 1$ 
2: for all training points in the parameter space do
3:   for all training inputs do
4:      $t \leftarrow t_{\text{initial}}$ 
5:     while  $t < t_{\text{final}}$  do
6:       if  $\min_i \|\hat{x}(t) - \hat{x}_i\| > \delta$  then
7:         Linearize at current state  $\hat{x}_i \leftarrow \hat{x}(t)$ 
8:         Generate MOR vectors (3)
9:         Orthogonalize and add them to  $V$ 
10:        Update reduced model using (18)
11:        $i \leftarrow i + 1$ 
12:       end if
13:       Simulate reduced model to next time step  $t$ 
14:     end while
15:   end for
16: end for
17: Perform SVD on projection matrix  $V$ 
18: Update reduced model using (18)
```

---

**Algorithm 1.** A more interesting case from Table I, is obtained by combining “PMOR moment matching in  $V$ ”, while still “training at a single point in parameter space”. The resulting complete procedure is summarized in Algorithm 1. Compared to the previous TPWL case, Algorithm 1 is potentially more expensive since many more vectors are generated (each requiring a system solve) in order to match several moments with respect to each geometrical parameter. More precisely,  $O(kp^{m_q})$  vectors need to be generated to match  $m_q$  moments with respect to each of the  $p$  parameters for  $k$  linearized models  $A_i$ . An SVD is then performed on  $V$  to keep the final order  $q$  of the reduced system small.

---

**Algorithm 3** Trained at Multiple Parameter Values with PMOR Moment Matching Projection Matrix  $V$ 

---

```
1:  $i \leftarrow 1$ 
2: for all training points in the parameter space do
3:   for all training inputs do
4:      $t \leftarrow t_{\text{initial}}$ 
5:     while  $t < t_{\text{final}}$  do
6:       if  $\min_i \|\hat{x}(t) - \hat{x}_i\| > \delta$  then
7:         Linearize at current state  $\hat{x}_i \leftarrow \hat{x}(t)$ 
8:         Generate PMOR vectors (8)
9:         Orthogonalize and add them to  $V$ 
10:        Update reduced model using (18)
11:        $i \leftarrow i + 1$ 
12:       end if
13:       Simulate reduced model to next time step  $t$ 
14:     end while
15:   end for
16: end for
17: Perform SVD on projection matrix  $V$ 
18: Update reduced model using (18)
```

---

**Algorithm 2.** Another case of interest in Table I, trains the system at multiple points in the parameter space. Reduction is then performed with a non-parameterized MOR moment matching projection matrix  $V$ . The complete procedure is outlined in Algorithm 2. The final system is now a collection of linear models corresponding to different sets of parameter values. Compared to TPWL, the additional cost in Algorithm 2 lies in creating additional training trajectories. More specifically, training the system for  $p$  different parameters at  $r$  different values requires generating  $r^p$  additional trajectories. Constructing the projection matrix  $V$  now requires generating  $O(kp)$  vectors (i.e.  $O(kp)$  system solves), although the number  $k$  of linearized models  $A_i$  may be  $O(r^p)$  times larger than in TPWL or in Algorithm 1.

**Algorithm 3.** In the forth and final case described in Table I, the system is trained at multiple points in the parameter space, and the models are reduced with a PMOR moment matching projection matrix  $V$ . The complete procedure is summarized in Algorithm 3. Compared to all the previous Algorithms, a model created by Algorithm 3 can potentially cover the largest region of the parameter space, however it is the most expensive to build. Specifically, as in Algorithm 2, the number of additional training trajectories is  $O(r^p)$ . The projection matrix requires generating  $O(p^{m_q}r^p)$  vectors (i.e. system solves), where  $p^{m_q}$  is the number of vectors created by matching  $m_q$  moments with respect to each parameter at each of the  $r^p$  sets of parameter values (i.e. training trajectories).

#### IV. EXAMPLES

Two examples were chosen to test the methods presented in Sections III-C. Both examples are highly non-linear systems and depend geometrically on several parameters.

##### A. Non-Linear Analog Circuit Example Description

The first example considered is a non-linear analog circuit containing a chain of strongly non-linear diodes, together with resistors and capacitors as shown in Fig. 1. This example was first used by the authors in [21], [23] to illustrate the non-parameterized TPWL-MOR method. By using the same example here we will try to compare the advantages of our parameterized algorithms.

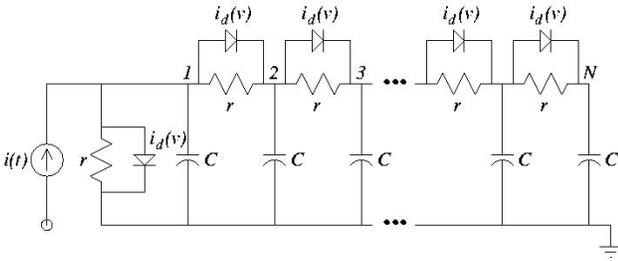


Fig. 1. A non-linear analog circuit example [21], [23]

The state of the system contains the nodal voltages,  $x(t) = [v_1 \ v_2 \ \dots \ v_N]^T$ , and the input is an ideal current source  $u(t) = i(t)$ . The system equations are derived using Kirchoff's current law and nodal analysis. An equation for interior node  $j$  would be of the form,

$$C \frac{dv_j}{dt} = \frac{1}{r} (v_{j-1} - v_j) - \frac{1}{r} (v_j - v_{j+1}) + I_d \left[ e^{\alpha(v_{j-1} - v_j)} - e^{\alpha(v_j - v_{j+1})} \right]$$

leading to a state space system of the form

$$E \frac{dx}{dt} = Gx + D(x) + bu(t). \quad (19)$$

Here  $G$  is the conductance matrix,  $E$  is the capacitance matrix,  $D(x)$  is a vector valued non-linear function containing the constitutive relations for the diodes, and  $b = [1 \ 0 \ \dots \ 0]^T$  is the input vector. All resistors have value  $r = 1\Omega$ , and all capacitors are  $C = 10\text{pF}$ . The diodes have a constitutive relation

$$i_d(v) = I_d(e^{\alpha v} - 1), \quad (20)$$

where  $\alpha = 1/v_T$ , and  $v_T$  is the threshold voltage. Nominal values for this device were  $I_d = 0.1\text{nA}$ , and  $\alpha = 40$  (corresponding to  $v_T = 25\text{mV}$ ).

The analog circuit was parameterized in both  $\alpha$  and  $I_d$ . We approximated the non-linear dependency on such parameters using

$$E \frac{dx}{dt} = Gx + D(x, \alpha_0, I_{d0}) + \frac{\partial D(x, \alpha_0, I_{d0})}{\partial \alpha} (\alpha - \alpha_0) + \frac{\partial D(x, \alpha_0, I_{d0})}{\partial I_d} (I_d - I_{d0}) + bu(t)$$

The model then produced at each linearization state  $x_i$ , is linear both in the parameters and in the state, and has the same frequency domain form as in (6):

$$\begin{aligned} sEx &= Gx + D_i + J_{D_i} \cdot (x - x_i) + \\ &+ D_{\alpha i} \cdot (\alpha - \alpha_0) + J_{D_{\alpha}} \cdot (\alpha - \alpha_0) \cdot (x - x_i) + \\ &+ D_{I_d i} \cdot (I_d - I_{d0}) + J_{D_{I_d}} \cdot (I_d - I_{d0}) \cdot (x - x_i) + bu(t), \end{aligned}$$

where  $s$  is the Laplace variable,  $D_i = D(x_i, \alpha_0, I_{d0})$ ,  $J_{D_i}$  is the Jacobian of  $D(x, \alpha_0, I_{d0})$  with respect to  $x$ ,  $D_{\alpha i} = \frac{\partial D}{\partial \alpha}(x_i, \alpha_0, I_{d0})$ ,  $J_{D_{\alpha}}$  is the Jacobian of  $\frac{\partial D}{\partial \alpha}$  with respect to  $x$ ,  $D_{I_d i} = \frac{\partial D}{\partial I_d}(x_i, \alpha_0, I_{d0})$ , and  $J_{D_{I_d}}$  is the Jacobian of  $\frac{\partial D}{\partial I_d}$  with respect to  $x$ .

Training trajectories were created with a sinusoidal input,  $u(t) = [\cos(\omega t) + 1]/2$ , with  $\omega = 2\pi\text{GHz}$ . When matching moments with respect to frequency, both in MOR and in PMOR we used  $s_0 = j2\pi\text{GHz}$  as a Taylor series expansion point. When matching moments with a simple non-parameterized MOR projection matrix  $V$ , or when training at a single point in parameter space, we evaluated the system at the nominal values  $\alpha = 40$ , and  $I_d = 0.1\text{nA}$ . When matching moments with a PMOR procedure, or when training at multiple moments in parameter space, we used  $\alpha = 40, 50$ , and  $I_d = 0.1\text{nA}, 0.3\text{nA}$ .

Results from this example are shown in Fig. 4, 5, 6, and will be illustrated and discussed in details in the next Section.

##### B. MEM Switch Example

The second example tested is a micromachined switch [21], [23]. The switch consists of a polysilicon fixed-fixed beam suspended over a polysilicon pad on a silicon substrate as shown in Fig. 2. When a voltage is applied between the beam and the substrate, the electrostatic force generated pulls the beam down toward the pad. If the force is great enough, the beam will come into contact with the pad closing the circuit. The unknowns of interest in this system have been chosen to be the deflection of the beam,  $z$ , and the air pressure between the beam and substrate,  $p$ .

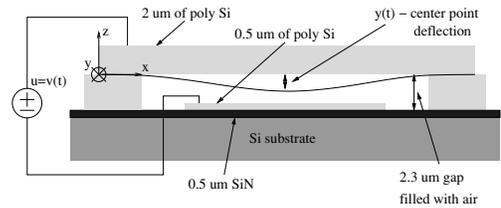


Fig. 2. The MEMS switch is a polysilicon beam fixed at both ends and suspended over a semiconducting pad and substrate [21], [23].

The system of equations was assembled by discretizing the coupled 1D Euler's Beam Equation (21), and the 2D Reynold's squeeze film damping equation (22), taken from [21]. A finite difference scheme was used for the discretization, and since the length of the beam is much greater than the width, the vertical deflection was assumed to be uniform across the width and only pressure was discretized in the width.

$$\hat{E}I_0h^3w\frac{\partial^4z}{\partial x^4} - S_0hw\frac{\partial^2z}{\partial x^2} = F_{elec} + \int_0^w (p - p_a)dy - \rho_0hw\frac{\partial^2z}{\partial t^2} \quad (21)$$

$$\nabla \cdot [(1 + 6K)z^3 p \nabla p] = 12\mu\frac{\partial(pz)}{\partial t} \quad (22)$$

Here,  $F_{elec} = -(\epsilon_0 w v^2)/(wz^2)$  is the electrostatic force across the plates resulting from the applied voltage  $v$ , while  $u = v^2$  is the input to the system. The system output is the height of the beam center point. The beam is  $610\mu\text{m}$  in length, and has a width of  $40\mu\text{m}$ . The other constants are permittivity of free space  $\epsilon_0 = 8.854 \cdot 10^{-6}\text{F/m}$ , permeability  $\mu = 4\pi \cdot 10^{-7}\text{H/m}$ , moment of inertia  $I_0 = 1/12$ , Young's modulus  $\hat{E} = 149\text{GPa}$ , Knudsen number  $K = 0.064/z_0$ , stress coefficient  $S_0 = -3.7$ , and density  $\rho_0 = 2300 \text{ kg/m}^3$ . By setting the state-space variables to  $x_1 = z$ ,  $x_2 = \frac{\partial z^2}{\partial t}$ , and  $x_3 = p$ , the following dynamical system results.

$$\begin{aligned} \frac{dx_1}{dt} &= \frac{x_2}{3x_1^2} \\ \frac{dx_2}{dt} &= \frac{2x_2^2}{3x_1^3} + \frac{3x_1^2}{\rho_0hw} \left[ \int_0^w (x_3 - p_a)dy + S_0hw\frac{\partial^2x_1}{\partial x^2} - EIh^3w\frac{\partial^4x_1}{\partial x^4} \right] - \frac{3\epsilon_0}{2\rho_0h}v^2 \\ \frac{dx_3}{dt} &= -\frac{x_2x_3}{3x_1^3} + \frac{1}{12\mu x_1} \nabla \cdot \left[ \left(1 + 6\frac{\lambda}{x_1}\right) x_1^3 x_3 \nabla x_3 \right] \end{aligned}$$

The beam is fixed at both ends and initially in equilibrium, so the applied boundary conditions are

$$z(x, 0) = z_0, \quad p(x, y, 0) = p_a, \quad z(0, t) = z(l, t) = z_0. \quad (23)$$

Other constraints enforced are,

$$\frac{\partial p(0, y, t)}{\partial x} = \frac{\partial p(l, y, t)}{\partial x} = 0, \quad p(x, 0, t) = p(x, w, t) = p_a \quad (24)$$

The initial height and pressure used were  $z_0 = 2.3\mu\text{m}$ , and  $p_a = 1.103 \cdot 10^5\text{Pa}$ .

For this MEMS switch system, we are interested in producing a model as a function of the width  $w$ . The dependency on the parameter is easily made linear in this case by introducing variable  $\beta = 1/w$ , and parameterizing the system with respect to  $\beta$  instead of  $w$

$$\frac{dx}{dt} = A_0(x) + \beta A_1(x) + bu(t) \quad (25)$$

Training trajectories were created using a step input,  $u(t) = v^2$  for  $t > 0$ , with  $v = 7$ . When matching moments with a simple non-parameterized MOR projection matrix  $V$ , or when training at a single point in parameter space, we evaluated the system at the nominal values  $\beta = 1/40$ . When matching moments with a PMOR moment matching projection matrix  $V$ , or when training at multiple moments in parameter space, we used  $\beta = 1/40, 1/44.4$ .

Results from this example are shown in Fig. 3 and will be illustrated and discussed in details in the next Section.

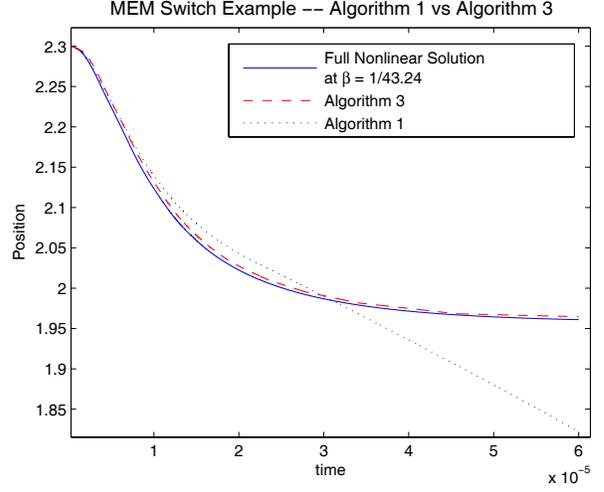


Fig. 3. Comparison of models created by Alg. 1 (trained at  $\beta = 1/40$ ) vs. Alg. 3 (trained at  $\beta = 1/40, 1/44.4$ ). Both used a PMOR projection  $V$  matching moments about  $\beta = 1/40$ . Both models were verified against the full nonlinear system at  $\beta = 1/43.24$ , and a  $v=7$  volts step input.

### C. Results and Discussion

The algorithms from Table I were tested on the non-linear analog circuit and on the MEM switch examples. Reduced order models were built, and then simulated for some set of parameter values which differ from the training values. Comparisons in this Section are made to determine the effects of the alternative options described in Sections III-A and III-B, available when choosing linearization points, and when constructing projection matrix  $V$ , respectively.

**Benefits of training at multiple parameter points.** In order to examine the benefits of training at multiple points in the parameter space as opposed to training at a single point (as described in Section III-A), reduced order models for the MEM switch example created by Algorithms 3 and 1 are compared in Figure 3. Both models used a PMOR expansion about  $\beta = 1/40$  for the construction of  $V$ . Algorithm 3 was trained at  $\beta = 1/40, 1/44.4$  creating 50 linearized models, while Algorithm 1 was trained only at  $\beta = 1/40$  creating 80 linearized models. Both models were then reduced from an original size  $N = 150$  to a final  $q = 40$ , and simulated at  $\beta = 1/43.24$ . The figure clearly shows that Algorithm 3 produced a much more accurate model for this value of  $\beta$  than Algorithm 1, even though it contains 30 fewer linearized models.

**Benefits of PMOR moment matching projection.** In order to examine the benefits of using a PMOR moment matching projection  $V$  as opposed to a simple non-parameterized MOR moment matching  $V$ , (as described in Section III-B), reduced order models for the analog circuit example created by TPWL and Algorithm 1 are compared in Fig. 4. The same benefits can also be evaluated in Fig. 5, which compares models created by Algorithm 2 and Algorithm 3.

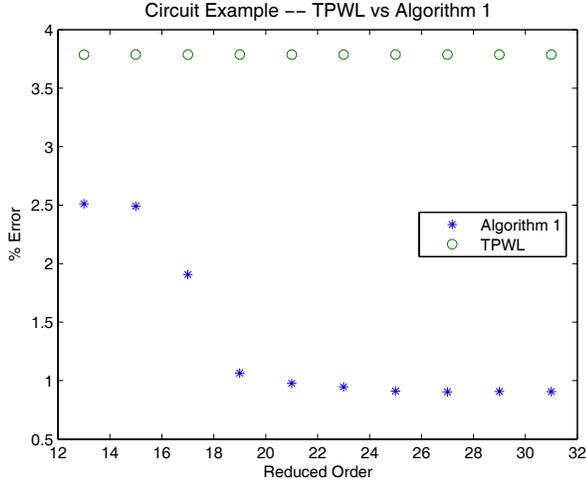


Fig. 4. Comparison of models created with TPWL (MOR projection at  $I_d = 0.1\text{nA}$ ) and Alg. 1 (PMOR projection at  $I_d = 0.1\text{nA}$ ). Errors are in percentage and are measured against the full non-linear system at  $I_d = 0.045\text{nA}$  for increasing orders of the reduced models.

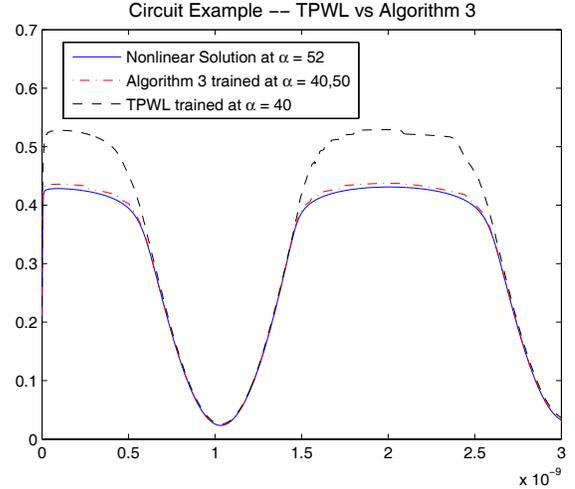


Fig. 6. Comparison of models created by TPWL (trained at  $\alpha = 40$ ) and Alg. 3 (trained and expanded about  $\alpha = 40,50$ ). Both models were verified against the full non-linear system at  $\alpha = 52$ . Our proposed Alg. 3 shows in this example a maximum error of 2.5% compared to the 25% error of TPWL.

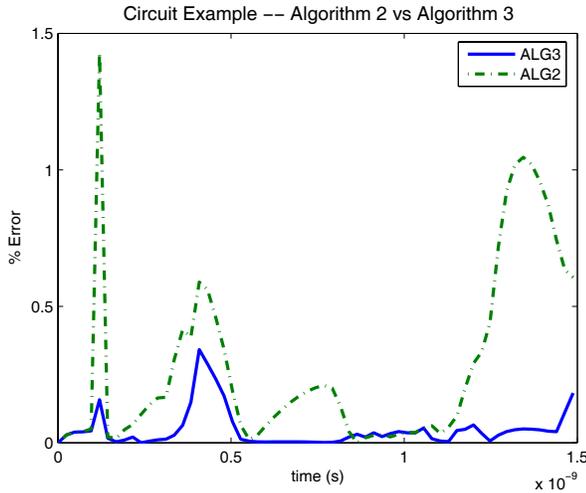


Fig. 5. Comparison of models created with Alg. 2 (MOR projection) vs. Alg. 3 (PMOR projection) Both models were trained at  $I_d = 0.1\text{nA}, 0.03\text{nA}$ . Both models were compared against the full non-linear system at  $I_d = 0.045\text{nA}$ .

Specifically, Figure 4 compares the error for increasing order of the models, created by TPWL and by Algorithm 1. Both models were trained at a single point  $I_d = 0.1\text{nA}$ ,  $\alpha = 40$ , and contain 107 linearized models. TPWL used a simple non-parameterized MOR moment matching projection  $V$  constructed at  $I_d = 0.1\text{nA}$ ,  $\alpha = 40$ . Algorithm 1 used instead a PMOR moment matching projection  $V$  parameterized in  $I_d$  with a Taylor series expanded about  $I_d = 0.1\text{nA}$ . The model errors were then measured against the full non-linear system at  $I_d = 0.045\text{nA}$ ,  $\alpha = 40$ . The figure clearly shows that increasing the order by adding vectors with a simple non-parameterized MOR projection (TPWL) does not appear to improve the error. Instead, adding vectors to the projection  $V$  with a PMOR moment matching procedure (Alg. 1) can produce more accurate models for larger orders.

Figure 5 compares the errors of two reduced order models created by Algorithm 2 and Algorithm 3 on the same circuit example. Both models were trained at multiple points in the parameter space ( $I_d = 0.1\text{nA}, 0.03\text{nA}$ ), resulting in 165 linear models, and were reduced from  $N = 100$  to  $q = 35$ . Algorithm 3 used a PMOR moment matching projection matrix  $V$  expanded about  $I_d = 0.1\text{nA}, 0.03\text{nA}$ , while Algorithm 2 used a simple non-parameterized MOR moment matching projection matrix constructed at  $I_d = 0.1\text{nA}$ . The reduced models were then simulated and compared to the full non-linear system at  $I_d = 0.045\text{nA}$ . Although both reduced order models approximate the original system well, Algorithm 3 clearly shows the additional advantage of using PMOR projection vs. the non-parameterized MOR projection in Algorithm 2.

**Benefits of combining PMOR projection with training at multiple parameters values.** Models of the analog circuit example created by TPWL and by Algorithm 3 are compared in Figure 6 in order to present the benefits of combining both ideas: PMOR moment matching projection matrix  $V$ , and training at multiple parameter values. In this case the circuit example was parameterized in  $\alpha$ , while  $I_d$  was kept constant at  $0.1\text{nA}$ . Algorithm 3 was trained at  $\alpha = 40,50$  and expanded about  $\alpha = 40,50$  for the PMOR projection matrix. In TPWL, the model was trained at  $\alpha = 40$  and the non-parameterized MOR projection matrix was constructed at the same parameter value  $\alpha = 40$ . Both reduced order models were then simulated and verified against the full non-linear model at a parameter value of  $\alpha = 52$ . Figure 6 shows a 25% maximum error of TPWL versus a 2.5% maximum error of our proposed Algorithm 3.

## V. CONCLUSION

In this paper we have shown that it is possible to create reduced order non-linear systems which can accurately approximate large non-linear systems over a practical range of geometrical parameter variations. Several approaches have been tested on two examples: an analog non-linear circuit and a MEM switch.

In our tests, we have observed that approximated training trajectories can produce models with the same accuracy of the models generated by much more expensive exact training trajectories.

We have also observed that using multiple approximated training trajectories corresponding to different values of the parameters, together with a simple moment matching MOR projection matrix (Algorithm 2) can produce models more accurate than using a single exact training trajectory corresponding to a nominal parameter value together with a moment matching PMOR projection matrix (Algorithm 1).

Results could be in general application specific. However, combining both ideas by using multiple approximated training trajectories together with a moment matching PMOR projection matrix (Algorithm 3), can potentially provide enough trade off to accommodate the needs of a large range of applications.

## ACKNOWLEDGMENTS

This work was supported by the MARCO Gigascale Systems Research Center, by the Semiconductor Research Corporation, and by the National Science Foundation.

## REFERENCES

- [1] S. Pullela, N. Menezes, and L.T. Pileggi. Moment-sensitivity-based wire sizing for skew reduction in on-chip clock nets. *IEEE Trans. Computer-Aided Design*, 16(2):210–215, February 1997.
- [2] Y. Liu, Lawrence T. Pileggi, and Andrzej J. Strojwas. Model order reduction of RCL interconnect including variational analysis. In *Proc. of the ACM/IEEE Design Automation Conference*, pages 201–206, New Orleans, Louisiana, June 1999.
- [3] P. Heydari and M. Pedram. Model reduction of variable-geometry interconnects using variational spectrally-weighted balanced truncation. In *Proc. of IEEE/ACM International Conference on Computer Aided-Design*, San Jose, CA, November 2001.
- [4] H. Liu, A. Singhee, R. A. Rutenbar, and L. R. Carley. Remembrance of circuits past: macromodeling by data mining in large analog design spaces. In *Proc. of the ACM/IEEE Design Automation Conference*, pages 437–42, June 2002.
- [5] D. S. Weile, E. Michielssen, Eric Grimme, and K. Gallivan. A method for generating rational interpolant reduced order models of two-parameter linear systems. *Applied Mathematics Letters*, 12:93–102, 1999.
- [6] P. Gunupudi and M. Nakhla. Multi-dimensional model reduction of vlsi interconnects. In *Proc. of the Custom Integrated Circuits Conference*, pages 499–502, Orlando, FL, 2000.
- [7] C. Prud'homme, D. Rovas, K. Veroy, Y. Maday, A.T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bounds methods. *Journal of Fluids Engineering*, 2002.
- [8] L. Daniel, C. S. Ong, S. C. Low, K. H. Lee, and J. K. White. A multiparameter moment matching model reduction approach for generating geometrically parameterized interconnect performance models. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 23(5):678–93, May 2004.
- [9] L. Daniel and J. White. Automatic generation of geometrically parameterized reduced order models for integrated spiral rf-inductors. In *IEEE Intern. Workshop on Behavioral Modeling and Simulation*, San Jose, CA, September 2003.
- [10] P. Li, F. Liu, S. Nassif, and L. Pileggi. Modeling interconnect variability using efficient parametric model order reduction. In *Design, Automation and Test Conference in Europe*, March 2005.
- [11] J.R. Phillips. Variational interconnect analysis via PMTBR. In *Proc. of IEEE/ACM International Conference on Computer Aided-Design*, pages 872–9, November 2004.
- [12] K. C. Suo, A. Megretski, and L. Daniel. A quasi-convex optimization approach to parameterized model order reduction. In *Proc. of the IEEE/ACM Design Automation Conference*, CA, June 2005.
- [13] M. Celik, A. Atalar, and M. A. Tan. Transient analysis of nonlinear circuits by combining asymptotic waveform evaluation with volterra series. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 42(8):470–473, August 1995.
- [14] J. R. Phillips. Automated extraction of nonlinear circuit macromodels. In *Proceedings of the Custom Integrated Circuit Conference*, pages 451–454, Orlando, FL, May 2000.
- [15] J.R. Phillips. Projection frameworks for model reduction of weakly nonlinear systems. In *Proc. of the ACM/IEEE Design Automation Conference*, pages 184–9, June 2000.
- [16] J. Chen and S. M. Kang. An algorithm for automatic model-order reduction of nonlinear MEMS devices. In *Proceedings of ISCAS 2000*, pages 445–448, 2000.
- [17] J.R. Phillips. Projection-based approaches for model reduction of weakly nonlinear, time-varying systems. *IEEE Trans. Computer-Aided Design*, 22(2):171–87, 2003.
- [18] P. Li and L. T. Pileggi. Norm: Compact model order reduction of weakly nonlinear systems. In *Proc. of the ACM/IEEE Design Automation Conference*, June 2003.
- [19] Y. Chen. Model order reduction for nonlinear systems. M.S. Thesis, Massachusetts Institute of Technology, September 1999.
- [20] M. Rewienski and J. K. White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. In *Proc. of IEEE/ACM International Conference on Computer Aided-Design*, pages 252–7, San Jose, CA, USA, November 2001.
- [21] M. Rewienski and J. White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Trans. Computer-Aided Design*, 22(2):155–70, Feb 2003.
- [22] S. Tiwary and R. Rutenbar. Scalable trajectory methods for on-demand analog macromodel extraction. In *42<sup>nd</sup> ACM/IEEE Design Automation Conference*, June 2005.
- [23] D. Vasilyev, M. Rewienski, and J. White. A tbr-based trajectory piecewise-linear algorithm for generating accurate low-order models for nonlinear analog circuits and mems. In *Proc. of the ACM/IEEE Design Automation Conference*, pages 490–5, June 2003.
- [24] N. Dong and J. Roychowdhury. Piecewise polynomial nonlinear model reduction. In *Proc. of the ACM/IEEE Design Automation Conference*, June 2003.
- [25] N. Dong and J. Roychowdhury. Automated extraction of broadly applicable nonlinear analog macromodels from spice-level descriptions. In *Proc. of the IEEE Custom Integrated Circuits Conference*, October 2004.
- [26] Y. Wan and J. Roychowdhury. Operator-based model-order reduction of linear periodically time-varying systems. In *42<sup>nd</sup> ACM/IEEE Design Automation Conference*, June 2005.
- [27] A. Odabasioglu, M. Celik, and L. T. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. *IEEE Trans. Computer-Aided Design*, 17(8):645–654, August 1998.
- [28] Eric Grimme. *Krylov Projection Methods for Model Reduction*. PhD thesis, Coordinated-Science Laboratory, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, 1997.