Model Order Reduction for Nonlinear IC Models

Arie Verhoeven[1], Jan ter Maten[2], Michael Striebel[3], and Robert Mattheij[4]

[1] Eindhoven University of Technology (CASA), Den Dolech 2,
5600 MB Eindhoven, The Netherlands
Arie.Verhoeven@na-net.ornl.gov
[2] NXP Semiconductors
jan.ter.maten@nxp.com
[3] Technische Universität Chemnitz
michael.striebel@mathematik.tu-chemnitz.de
[4] Dept of Mathematics and Computer Science, TU Eindhoven, PO Box 513,
5600 MB Eindhoven The Netherlands
r.m.m.mattheij@tue.nl

Abstract. Model order reduction is a mathematical technique to transform nonlinear dynamical models into smaller ones, that are easier to analyze. In this paper we demonstrate how model order reduction can be applied to nonlinear electronic circuits. First we give an introduction to this important topic. For linear time-invariant systems there exist already some well-known techniques, like Truncated Balanced Realization. Afterwards we deal with some typical problems for model order reduction of electronic circuits. Because electronic circuits are highly nonlinear, it is impossible to use the methods for linear systems directly. Three reduction methods, which are suitable for nonlinear differential algebraic equation systems are summarized, the Trajectory piecewise Linear approach, Empirical Balanced Truncation, and the Proper Orthogonal Decomposition. The last two methods have the Galerkin projection in common. Because Galerkin projection does not decrease the evaluation costs of a reduced model, some interpolation techniques are discussed (Missing Point Estimation, and Adapted POD). Finally we show an application of model order reduction to a nonlinear academic model of a diode chain.

1 Introduction

The dynamics of electrical circuits at time t can be generally described by the nonlinear, first order, differential-algebraic equation (DAE) system of the form:

$$\begin{cases} \frac{d}{dt}[\mathbf{q}(\mathbf{x})] + \mathbf{j}(\mathbf{x}) + \mathbf{B}\mathbf{u} = \mathbf{0}, \quad \mathbf{x}(0) = \mathbf{x}_0, \\ \mathbf{y} = \mathbf{h}(\mathbf{x}), \end{cases} \tag{1}$$

where $\mathbf{x} : \mathbb{R} \to \mathbb{R}^d$ represents the unknown vector of circuit variables in time t, the vector-valued functions $\mathbf{q}, \mathbf{j} : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$ represent the contributions of, respectively, reactive elements (such as capacitors and inductors) and of nonreactive elements (such as resistors) and $\mathbf{B} \in \mathbb{R}^{d \times m}$ is the distribution matrix for the excitation vector $\mathbf{u} : \mathbb{R} \to \mathbb{R}^m$ that controls the output response $\mathbf{y} : \mathbb{R} \to \mathbb{R}^p$. We assume that $d >> m, p$. There are

A. Korytowski et al. (Eds.): System Modeling and Optimization, IFIP AICT 312, pp. 476–491, 2009.
© IFIP International Federation for Information Processing 2009

several established methods, such as sparse-tableau, modified nodal analysis, etc. which generate the system (1) from the netlist description of electrical circuit. The dimension d of the unknown vector $\mathbf{x}$ is of the order of the number of elements in the circuit, which means that it can be extremely large, as today's VLSI (Very Large Scale Integrated) circuits have hundreds of millions of elements.

Mathematical model order reduction (MOR) aims to replace the original model (1) by a system of much smaller dimension, which can be solved by suitable DAE solvers within acceptable time. Because we are only interested in the relationship between $\mathbf{u}$ and $\mathbf{y}$ in the time-domain, the model can be replaced by a low-order model for $\mathbf{z} : \mathbb{R} \to \mathbb{R}^r$, like

$$\begin{cases} \frac{d}{dt}\left[\tilde{\mathbf{q}}(\mathbf{z})\right] + \tilde{\mathbf{j}}(\mathbf{z}) + \tilde{\mathbf{B}}\mathbf{u} = \mathbf{0}, \quad \mathbf{z}(0) = \mathbf{z}_0, \\ \mathbf{y} = \tilde{\mathbf{h}}(\mathbf{z}). \end{cases} \tag{2}$$

At present, however, only linear MOR techniques are well-enough developed and properly understood to be employed [1]. To that end, we either linearize the system (1) or decouple it into nonlinear and linear subcircuits (interconnect macromodeling of parasitic subcircuits [9]). For dynamical systems the observability and controllability functions [1] are defined by

$$L_c(\mathbf{x}_0) = \min\{\tfrac{1}{2}\textstyle\int_{-\infty}^{0} \|\mathbf{u}(t)\|^2 dt : \mathbf{u} \in L_2(-\infty, 0),\ \mathbf{x}(-\infty) = 0,\ \mathbf{x}(0) = \mathbf{x}_0\}, \tag{3}$$

$$L_o(\mathbf{x}_0) = \tfrac{1}{2}\textstyle\int_{0}^{\infty} \|\mathbf{y}(t)\|^2 dt,\ \ \forall_{\tau\in[0,\infty)}\mathbf{u}(\tau) = \mathbf{0},\ \ \mathbf{x}(0) = \mathbf{x}_0. \tag{4}$$

They represent the minimum amount of input energy to reach state $\mathbf{x}_0$ and the output energy that comes free when starting at state $\mathbf{x}_0$ (compare kinetic and potential energy in mechanical systems). The system is in balanced form at basis $\mathbf{V}$ if the (energy) ratio $\frac{L_o(\mathbf{V}\mathbf{z})}{L_c(\mathbf{V}\mathbf{z})}$ is balanced. For linear time-invariant (LTI) systems as

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{x}(0) = \mathbf{x}_0, \\ \mathbf{y} = \mathbf{C}\mathbf{x}, \end{cases} \tag{5}$$

we have $L_c(\mathbf{x}_0) = \frac{1}{2}\mathbf{x}_0^T\mathbf{W}^{-1}\mathbf{x}_0$ and $L_o(\mathbf{x}_0) = \frac{1}{2}\mathbf{x}_0^T\mathbf{M}\mathbf{x}_0$, where $\mathbf{W}, \mathbf{M} \in \mathbb{R}^{d\times d}$ are the controllability and observability Gramians, which are symmetric positive definite matrices. They satisfy the well-known Lyapunov equations

$$\mathbf{A}\mathbf{W} + \mathbf{W}\mathbf{A}^T = -\mathbf{B}\mathbf{B}^T, \tag{6}$$

$$\mathbf{A}^T\mathbf{M} + \mathbf{M}\mathbf{A} = -\mathbf{C}^T\mathbf{C}. \tag{7}$$

An LTI system is balanced w.r.t. basis $\mathbf{V}$ if $\mathbf{W} = \mathbf{V}\Sigma\mathbf{V}^T$ and $\mathbf{M} = \mathbf{V}^{-T}\Sigma\mathbf{V}^{-1}$ are simultaneously diagonalized, such that

$$\frac{L_o(\mathbf{x})}{L_c(\mathbf{x})} = \frac{\mathbf{x}^T\mathbf{M}\mathbf{x}}{\mathbf{x}^T\mathbf{W}^{-1}\mathbf{x}} = \frac{\mathbf{x}^T\mathbf{V}^{-T}\Sigma\mathbf{V}^{-1}\mathbf{x}}{\mathbf{x}^T\mathbf{V}^{-1}\Sigma^{-1}\mathbf{V}^{-T}\mathbf{x}} = \frac{\mathbf{z}^T\Sigma^2\mathbf{z}}{\mathbf{z}^T\mathbf{z}}. \tag{8}$$

For redundant systems the singular values of Σ converge rapidly to zero. This allows to obtain an accurate reduced model by Truncated Balanced Realization (TBR). There also exist many other much cheaper MOR techniques for LTI systems, like PRIMA, PVL, PMTBR, SPRIM, etc. For the special case $\mathbf{A} = \mathbf{A}^T, \mathbf{B} = \mathbf{C}^T$ it follows from (6),

(7) that $\mathbf{W} = \mathbf{M}$. Then it is possible to find an orthogonal $\mathbf{V}$ such that $\mathbf{W} = \mathbf{M} = \mathbf{V}\Sigma\mathbf{V}^T$ are balanced.

For nonlinear systems as (1) it is no longer possible to apply these linear MOR techniques. Then we try to exploit the (piecewise) linear structure as well as possible. The reduced model can be constructed for a benchmark simulation, such that it is accurate if the solution is in the neighborhood of the benchmark solution. In this paper we present the application of some promising nonlinear reduction methods on some electronic circuit models. These are the Trajectory Piecewise Linear approach (TPWL) [13] and the Proper Orthogonal Decomposition (POD) [2] supported by the Missing Point Estimation (MPE) technique [5]. This paper does not include an error analysis but interested readers could look at [7,10,13].

2 Model Order Reduction for Subcircuits

A continuously increasing number of functions is combined in each single integrated circuit. Therefore, complex devices are designed in a modular manner. Functional units like e.g., decoders, mixers, and operational amplifiers, are developed by different experts and stored in device libraries. Other circuit designers then choose these models according to their requirements and instantiate them in higher level circuits. To enable the combination of different blocks, each model is equipped with its own number of junctions, or *pins*, by which a communication with the outside world is possible.

In the first instance, numerical simulations are run to verify a design. Hence, it is desirable to have, besides the exact circuit schematic, a suitable description of the individual model that enables fast simulations, i.e., a library of reduced subcircuit models.

In circuits that are developed to act as a subcircuit in higher hierarchies a subset of its nodes are terminals. To these nodes both known inner elements as well as elements whose nature may change with different instantiations of the model are connected. Due to Kirchhoff's current law, the sum of all currents flowing into each single node is zero at each timepoint. In terms of the network equations (1) the contribution of currents from inner elements at the terminals is covered by $\frac{d}{dt}\mathbf{q}(\mathbf{x})$ and $\mathbf{j}(\mathbf{x})$, respectively. As the nature, i.e. reactive or nonreactive, of the adjacent elements in the final circuit is not known, when the cell is designed, additional unknowns $\mathbf{j_{pin}}$, i.e. *pin currents*, are introduced on the subcircuit level. We assume that the cell under consideration has d_e nodes and $d_{\text{pin}} < d_e$ of them are terminals. Then we can extend (1) to

$$\frac{d}{dt}\left[\mathbf{q}(\mathbf{x})\right] + \mathbf{j}(\mathbf{x}) + \mathbf{B}\mathbf{u}(t) + \mathbf{A_{pin}}\mathbf{j_{pin}} = \mathbf{0}, \tag{9}$$

with $\mathbf{j_{pin}} \in \mathbb{R}^{d_{\text{pin}}}$ and where $\mathbf{A_{pin}} \in \{0,1\}^{d \times d_{\text{pin}}}$ with d_{pin} columns containing exactly one non-zero element is an incidence matrix describing the topological distribution of the pin.

The pin currents $\mathbf{j_{pin}}$ can be determined when there is an external circuitry available, completing the network equations. During the process of developing the single cell a suitable test bench that emulates the typical environment the subcircuit will operate in later has to be defined by the designer.

Communication amongst electrical devices is done in terms of time varying voltages and currents. Regarding the cell (9) we can either inject the currents $\mathbf{j_{pin}}$ and get the voltage response at the terminals or supply the voltages at the pins and receive the according currents. The state $\mathbf{x}$ comprises the node voltages and the currents through inductors and voltage sources. With the pin currents' incidence matrix $\mathbf{A_{pin}}$ we can access the node voltages at the terminals by

$$\mathbf{x_{pin}} = \mathbf{A}_{\mathbf{pin}}^T \cdot \mathbf{x}. \tag{10}$$

Now, *current injection* means regarding $\mathbf{j_{pin}}$ as inputs returning $\mathbf{x_{pin}}$ as the output. Therefore, we can write

$$\mathbf{0} = \tfrac{d}{dt}\left[\mathbf{q}(\mathbf{x})\right] + \mathbf{j}(\mathbf{x}) + \left(\mathbf{B}\mathbf{A_{pin}}\right)\begin{pmatrix}\mathbf{u}(t)\\ \mathbf{j_{pin}}(t)\end{pmatrix}, \tag{11}$$

$$\mathbf{y} = \mathbf{A}_{\mathbf{pin}}^T\mathbf{x}. \tag{12}$$

Voltage injection on the other hand implies that the node voltages at the terminals are prescribed and corresponding pin currents are additional unknowns, i.e., they are added to the state vector:

$$\mathbf{0} = \tfrac{d}{dt}\begin{pmatrix}\mathbf{q}(\mathbf{x})\\ \mathbf{0}\end{pmatrix} + \begin{pmatrix}\mathbf{j}(\mathbf{x}) + \mathbf{A_{pin}}\mathbf{j_{pin}}\\ -\mathbf{A}_{\mathbf{pin}}^T\mathbf{x}\end{pmatrix} + \begin{pmatrix}\mathbf{B}\\ \mathbf{I}\end{pmatrix}\begin{pmatrix}\mathbf{u}(t)\\ \mathbf{x_{pin}}(t)\end{pmatrix} \tag{13}$$

$$\mathbf{y} = \begin{pmatrix}\mathbf{0} & \mathbf{I}\end{pmatrix}\begin{pmatrix}\mathbf{x}\\ \mathbf{j_{pin}}\end{pmatrix} \tag{14}$$

Finally, the common structure of both approaches is

$$\mathbf{0} = \tfrac{d}{dt}\left[\bar{\mathbf{q}}_\lambda(\bar{\mathbf{x}}_\lambda)\right] + \bar{\mathbf{j}}_\lambda(\bar{\mathbf{x}}_\lambda) + \begin{pmatrix}\bar{\mathbf{B}}_\lambda & \mathbf{C}_\lambda\end{pmatrix}\begin{pmatrix}\mathbf{u}_\lambda(t)\\ \mathbf{u}_{\mathbf{pin},\lambda}\end{pmatrix}, \tag{15a}$$

$$\mathbf{y}_\lambda = \mathbf{C}_\lambda^T\bar{\mathbf{x}}_\lambda \ \in \mathbb{R}^{d_{\mathrm{pin},\lambda}}, \tag{15b}$$

where we introduce λ as an identifier for the cell, taken from some set $\mathscr{I}$ of indices. Viewed from the outside, the cell (15) appears just in terms of its input-output behavior, i.e., given $\mathbf{u}_{\mathbf{pin},\lambda} \in \mathbb{R}^{d_{\mathrm{pin},\lambda}}$ it returns $\mathbf{y}_\lambda$.

Now, we turn our attention to the circuitry a cell might be embedded in. We assume that the state space of this circuit level has dimension d, i.e., it is described by the states $x \in \mathbb{R}^d$. Furthermore, we let this level consist of $r \in \mathbb{N}$ instantiations of cells, i.e., $\mathscr{I} = \{1,2,\ldots,r\}$, only. After due consideration we see that this electrical system is described by

$$\mathbf{0} = \sum_{\lambda\in\mathscr{I}} \mathbf{A}_\lambda^T\mathbf{y}_\lambda, \text{ with } \mathbf{A}_\lambda \in \{0,1\}^{d_{\mathrm{pin},\lambda}\times d} \tag{16}$$

where $\mathbf{y}_\lambda$ is determined by (15) with $\mathbf{u}_{\mathbf{pin},\lambda} = \mathbf{A}_\lambda\mathbf{x}$ for all $\lambda \in \mathscr{I}$.

As all the instantiated cells appear just in terms of their input-output behavior, we are free to reduce the order of single models (15) and use them again on level (16). Furthermore, as subcircuits are regarded as special elements, we can also include other elements on this level. Hence we can write in general form again

$$\frac{d}{dt}\left[\mathbf{q}(\hat{\mathbf{x}})\right] + \mathbf{j}(\hat{\mathbf{x}}) + \hat{\mathbf{B}}\hat{\mathbf{u}} = \mathbf{0}, \text{ with } \hat{\mathbf{x}} = \left(\mathbf{x}^T, \mathbf{y}_\mathbf{1}^T, \ldots, \mathbf{y}_\mathbf{r}^T\right)^T, \tag{17}$$

which can be seen as a subcircuit on another level again. In this way, a hierarchical model order reduction would be possible.

3 Trajectory Piecewise Linear Model Order Reduction

The idea behind the Trajectory Piecewise Linear (TPWL) method is to linearize (1) several times along a given trajectory $\tilde{\mathbf{x}}(t)$ (corresponding to some typical input $\tilde{\mathbf{u}}(t)$) that satisfies

$$\frac{d}{dt}[\mathbf{q}(\tilde{\mathbf{x}})] + \mathbf{j}(\tilde{\mathbf{x}}) + \mathbf{B}\tilde{\mathbf{u}} = \mathbf{0}. \tag{18}$$

Note that in [16] an alternative version of TPWL is described where the nonlinear functions $\mathbf{q}(t,\mathbf{x}), \mathbf{j}(t,\mathbf{x})$ are linearized around the Linearization Tuples $(t_i, \tilde{\mathbf{x}}(t_i))$. Below the nonlinear system itself is linearized around the complete trajectory $\tilde{\mathbf{x}}(t)$. Furthermore we can use just Linearization Points (LPs) $\tilde{\mathbf{x}}(t_i)$ instead of Linearization Tuples because the system in (1) does not depend explicitly on t and behaves linearly with respect to $\mathbf{u}$. Define $\mathbf{y}(t) = \mathbf{x}(t) - \tilde{\mathbf{x}}(t)$ and $\bar{\mathbf{u}}(t) = \mathbf{u}(t) - \tilde{\mathbf{u}}(t)$. Linearizing the nonlinear equation (1) gives us

$$\frac{d}{dt}[\mathbf{q}(\tilde{\mathbf{x}})] + \mathbf{j}(\tilde{\mathbf{x}}) + \mathbf{B}\tilde{\mathbf{u}} + \frac{d}{dt}[\mathbf{C}(\tilde{\mathbf{x}})\mathbf{y}] + \mathbf{G}(\tilde{\mathbf{x}})\mathbf{y} + \mathbf{B}\bar{\mathbf{u}} = \mathbf{0}. \tag{19}$$

Because the trajectory $\tilde{\mathbf{x}}(t)$ satisfies (18) we obtain the following time-varying linear system for $\mathbf{y}$

$$\frac{d}{dt}[\mathbf{C}(\tilde{\mathbf{x}}(t))\mathbf{y}(t)] + \mathbf{G}(\tilde{\mathbf{x}}(t))\mathbf{y}(t) + \mathbf{B}\bar{\mathbf{u}}(t) = \mathbf{0}. \tag{20}$$

The main idea of TPWL is to approximate the time-varying Jacobian matrices $\mathbf{C}(\tilde{\mathbf{x}}(t)), \mathbf{G}(\tilde{\mathbf{x}}(t))$ by a weighted combination of piecewise constant matrices. Then a (finite) sequence of linearized local systems is used to create a globally reduced subspace. The final TPWL model is constructed as a weighted sum of all locally linearized reduced systems. The disadvantage of standard linearization methods is that they only deliver good results in the neighborhood of the chosen linearization point (LP) $\mathbf{x}(t_i)$ (see Fig. 1). To overcome this, several linearized models are created in TPWL. The LPs can be computed simultaneously with the numerical time-integration of (18) for the trajectory $\tilde{\mathbf{x}}(t)$. This procedure can be described by the following steps:

1. Set an absolute accuracy factor $\varepsilon > 0$, set $i = 1$.
2. Linearize the system around $\tilde{\mathbf{x}}_i = \tilde{\mathbf{x}}(t_i)$. This implies:

$$\mathbf{C}_i\dot{\mathbf{y}} + \mathbf{G}_i\mathbf{y} + \mathbf{B}\bar{\mathbf{u}}(t) = \mathbf{0}, \tag{21}$$

   with $\mathbf{C}_i = \frac{\partial}{\partial \dot{\mathbf{x}}}\mathbf{q}(\tilde{\mathbf{x}})\Big|_{\tilde{\mathbf{x}}_i}$ and $\mathbf{G}_i = \frac{\partial}{\partial \mathbf{x}}\mathbf{j}(\tilde{\mathbf{x}})\Big|_{\tilde{\mathbf{x}}_i}$, where $\tilde{\mathbf{x}}_i$ stays for $\tilde{\mathbf{x}}(t_i)$. Save $\mathbf{C}_i$, and $\mathbf{G}_i$.
3. Reduce the linearized system to dimension $r \ll d$ by an appropriate linear MOR method, like "Poor Man's TBR" [12] or by Krylov-subspace methods [11]. This implies

$$\mathbf{C}_i^r\dot{\mathbf{z}} + \mathbf{G}_i^r\mathbf{z} + \mathbf{B}_i^r\bar{\mathbf{u}}(t) = \mathbf{0}, \tag{22}$$

   where $\mathbf{C}_i^r = \mathbf{V}_i^T\mathbf{C}_i\mathbf{V}$, $\mathbf{G}_i^r = \mathbf{V}_i^T\mathbf{G}_i\mathbf{V}_i$, $\mathbf{B}_i^r = \mathbf{V}_i^T\mathbf{B}$ with $\mathbf{V}_i \in \mathbb{R}^{d\times r_i}$, $\mathbf{z} \in \mathbb{R}^{r_i}$ and $\mathbf{y} \approx \mathbf{V}_i\mathbf{z}$. Save the local projection matrix $\mathbf{V}_i$. To preserve sparsity it could be preferable to diagonalize the reduced systems afterwards, although this destroys the orthogonality.

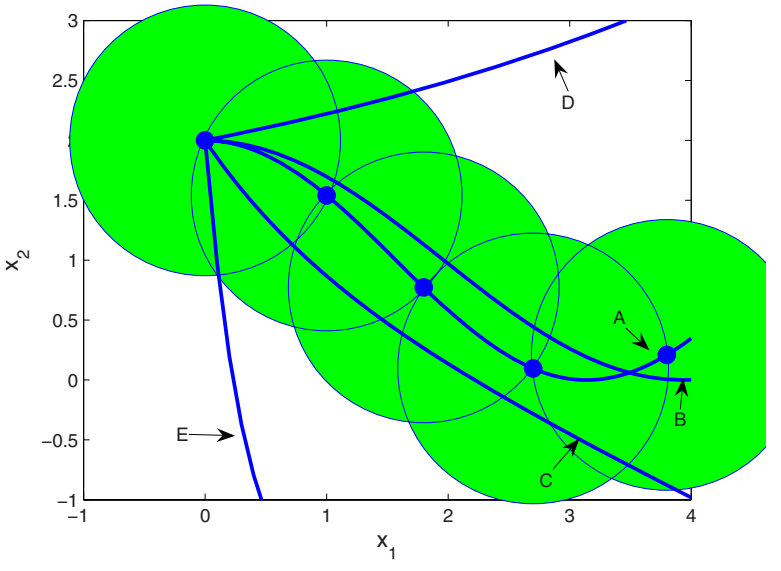

Fig. 1. The Linearization Points of this TPWL model are derived from the trajectory A. Because solutions B and C are in the neighborhood of the surrounding balls, they can be efficiently simulated using a TPWL model. But this is not the case for the solutions D and E.

4. Integrate the original system (1) until $\frac{||\tilde{\mathbf{x}}(t_k)-\tilde{\mathbf{x}}_i||}{||\tilde{\mathbf{x}}_i||} > \varepsilon$. Then we choose $\tilde{\mathbf{x}}(t_k)$ as $(i+1)$-th LP. Set $i = i+1$ and go to step 2.

Steps 2 to 4 are repeated until the end of the given trajectory. In this way, a finite number of locally reduced subspaces with bases $\mathbf{V}_1, ..., \mathbf{V}_s$ are created corresponding to the LPs $\{\tilde{\mathbf{x}}(t_1), \ldots, \tilde{\mathbf{x}}(t_s)\}$. All locally reduced subspaces are merged into a globally reduced subspace and each locally linearized system (21) is now projected onto this global subspace. The procedure can be described by the following steps:

1. Define $\tilde{\mathbf{V}} = [\mathbf{V}_1, \ldots, \mathbf{V}_s] \in \mathbb{R}^{d\times(r_1+\ldots+r_s)}$.
2. Calculate the SVD of $\tilde{\mathbf{V}}$: $\tilde{\mathbf{V}} = \mathbf{U}\Sigma\mathbf{W}^T$ with $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_d] \in \mathbb{R}^{d\times d}$, $\Sigma \in \mathbb{R}^{d\times \bar{r}s}$ and $\mathbf{W} \in \mathbb{R}^{\bar{r}s\times\bar{r}s}$, where $\bar{r} = (r_1 + \ldots + r_s)/s$.
3. Define the new global projection matrix $\mathbf{V} \in \mathbb{R}^{d\times r}$ as $[\mathbf{u}_1, \ldots, \mathbf{u}_r]$.
4. Project each local linearized system (21) onto $\mathbf{V}$.

Because of the construction of the global projection matrix $\mathbf{V}$ it is approximately true that $\mathscr{R}(\mathbf{V}_i) \subset \mathscr{R}(\mathbf{V})$ for $i = 1, \ldots, s$. All locally reduced linearized reduced systems are combined in a weighted sum to build the global TPWL model. Note that the TPWL model in [16] directly approximates $\mathbf{x}$ instead of $\mathbf{y} = \mathbf{x} - \tilde{\mathbf{x}}$ by $\mathbf{Vz}$. Then it is necessary to add the defect of the trajectory $\tilde{\mathbf{x}}$ to the new input vector. But if the original state $\mathbf{x} = \tilde{\mathbf{x}} + \mathbf{y}$ is approximated by $\tilde{\mathbf{x}} + \mathbf{Vz}$ the reduced state $\mathbf{z} \in \mathbb{R}^r$ satisfies

$$\sum_{i=1}^{s} w_i(\mathbf{z}) \left[\mathbf{V}^T\mathbf{C}_i\mathbf{V}\dot{\mathbf{z}} + \mathbf{V}^T\mathbf{G}_i\mathbf{V}\mathbf{z} + \mathbf{V}^T\mathbf{B}\bar{\mathbf{u}}(t)\right] = \mathbf{0}. \tag{23}$$

In (23) we need weighting functions $w_i(\mathbf{z})$ that satisfy

$$\sum_{i=1}^{s} w_i(\mathbf{z}) = 1,\ w_i(\mathbf{z}) \in [0,1]. \tag{24}$$

The weighting function $w_i(\mathbf{z})$ determines the influence of the i-th local system on the global system. Therefore it equals zero if $\mathbf{z}$ is far from the i-th projected Linearization Point $\mathbf{V}^T\mathbf{x}(t_i)$. A very simple weighting function is defined by

$$w_i(\mathbf{z}) = \begin{cases} 1 \text{ if } i = \min\{j \mid d_j(\mathbf{z}) = d_{\min}(\mathbf{z})\}, \\ 0 \text{ otherwise.} \end{cases} \tag{25}$$

Here $d_i(\mathbf{z})$ and $d_{\min}(\mathbf{z})$ are distance functions such that

$$d_i(\mathbf{z}) = \|\mathbf{z} - \mathbf{V}^T\mathbf{x}(t_i)\|,\ i = 1,\ldots,s, \tag{26}$$

$$d_{\min}(\mathbf{z}) = \min\{d_i(\mathbf{z}),\ i = 1,\ldots,s\}. \tag{27}$$

A more advanced alternative, with two free parameters $\alpha, \varepsilon > 0$, can be used like

$$\bar{w}_i(\mathbf{z}) = \begin{cases} \exp(-\frac{\alpha d_i(\mathbf{z})}{d_{\min}(\mathbf{z})}) \text{ if } \exp(-\frac{\alpha d_i(\mathbf{z})}{d_{\min}(\mathbf{z})}) > \varepsilon, \\ 0 \qquad\qquad\qquad \text{otherwise.} \end{cases} \tag{28}$$

$$w_i(\mathbf{z}) = \left[\textstyle\sum_{k=1}^{s} \bar{w}_k(\mathbf{z})\right]^{-1} \bar{w}_i(\mathbf{z}). \tag{29}$$

The TPWL method delivers reduced models that are cheap to simulate because the reduced model (23) does not need any evaluations of the original functions $\mathbf{q}, \mathbf{j}$ and Jacobian matrices $\mathbf{C}, \mathbf{G}$, because all matrices $\mathbf{V}^T\mathbf{C}_i\mathbf{V}, \mathbf{V}^T\mathbf{G}_i\mathbf{V}$ and $\mathbf{V}^T\mathbf{B}$ can be computed before the simulation. The reduction error of a TPWL method consists of a linearization and a truncation part. This error can be controlled by use of the Linearization Points [16,17]. Clearly the accuracy becomes higher for a large number of them. For strongly nonlinear systems the price is that a large number of Linearization Points is required to keep the linearization error sufficiently small. If the weighting functions $w_i(\mathbf{z})$ are not updated within the Newton method this will imply additional stepsize restrictions.

In the next three sections we will show how nonlinear systems can be reduced without linearization. Then the reduced models are obtained by Galerkin projection of the original model.

4 Empirical Balanced Truncation

For LTI systems the controllability and observability Gramians also satisfy

$$\mathbf{W} = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T t} dt,\ \mathbf{M} = \int_0^\infty e^{\mathbf{A}^T t}\mathbf{C}^T\mathbf{C}e^{\mathbf{A}t} dt. \tag{30}$$

Consider $\mathbf{X}(t) = [\mathbf{x}_1,\ldots,\mathbf{x}_m] = e^{\mathbf{A}t}\mathbf{B}$ and $\mathbf{Y}(t) = [\mathbf{y}_1,\ldots,\mathbf{y}_n] = \mathbf{C}e^{\mathbf{A}t}$. Let $\delta(t)$ be Dirac's delta function, then $\mathbf{x}_i$ and $\mathbf{y}_j$ satisfy

$$\tfrac{d}{dt}[\mathbf{q}(\mathbf{x}_i)] + \mathbf{j}(\mathbf{x}_i) = \mathbf{b}_i\delta(t),\ \mathbf{x}_i(0) = \mathbf{0}, \quad i = 1,\ldots,m, \tag{31}$$

$$\begin{cases} \tfrac{d}{dt}[\mathbf{q}(\mathbf{x}_j)] + \mathbf{j}(\mathbf{x}_j) = \mathbf{0},\ \mathbf{x}_j(0) = \mathbf{e}_j, \\ \mathbf{y}_j = \mathbf{h}(\mathbf{x}_j), \end{cases} \quad j = 1,\ldots,n. \tag{32}$$

Then it follows for LTI systems that the Gramians can be expressed in terms of the correlations of the states and outputs

$$\mathbf{W} = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T t} dt = \int_0^\infty \mathbf{X}(t)\mathbf{X}(t)^T dt = \sum_{i=1}^{m} \int_0^\infty \mathbf{x}_i(t)\mathbf{x}_i(t)^T dt, \tag{33}$$

and

$$\mathbf{M} = \int_0^\infty e^{\mathbf{A}^T t}\mathbf{C}^T\mathbf{C}e^{\mathbf{A}t} dt = \int_0^\infty \mathbf{Y}(t)^T\mathbf{Y}(t) dt = \sum_{i=1}^{n} \int_0^\infty \mathbf{y}_i(t)^T\mathbf{y}_i(t) dt. \tag{34}$$

If the states $[\mathbf{x}_1, \ldots, \mathbf{x}_m]$ and $[\mathbf{y}_1, \ldots, \mathbf{y}_n]$ are available, these Gramians $\mathbf{W}, \mathbf{M}$ can be numerically integrated as follows

$$\mathbf{W} \approx \hat{\mathbf{W}} = \sum_{i=1}^{m} \frac{1}{N} \sum_{k=1}^{N} \mathbf{x}_i(t_k)\mathbf{x}_i(t_k)^T, \quad \mathbf{M} \approx \hat{\mathbf{M}} = \sum_{i=1}^{n} \frac{1}{N} \sum_{k=1}^{N} \mathbf{y}_i(t_k)^T\mathbf{y}_i(t_k). \tag{35}$$

For LTI systems we have that $\hat{\mathbf{W}} \to \mathbf{W}$, $\hat{\mathbf{M}} \to \mathbf{M}$ if $N \to \infty$. Empirical balanced truncation (EBT) applies these formulae for $\hat{\mathbf{W}}, \hat{\mathbf{M}}$ to nonlinear systems with a larger set of inputs and initial values to include also the nonlinear properties. It is a powerful method because it really approximates the relationship between the input and output and neglects all other phenomena but also needs a lot of experiments. Then TBR or another linear MOR technique is used to balance $\hat{\mathbf{W}}, \hat{\mathbf{M}}$ by solving a system of Lyapunov equations. Thus a basis $\mathbf{V}$ can be constructed by truncation. The reduced model for $\mathbf{z} \in \mathbb{R}^r$ is constructed by Galerkin projection.

5 Proper Orthogonal Decomposition

The Proper Orthogonal Decomposition (POD), also known as the Principal Component Analysis (PCA) and the Karhunen-Loéve expansion, is a special case of Empirical Balanced Truncation. It approximates the controllability Gramian $\hat{\mathbf{W}}$ by using only one trajectory

$$\hat{\mathbf{W}} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{x}_1(t_k)\mathbf{x}_1(t_k)^T = \mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^T. \tag{36}$$

Because the two Gramians are assumed to be equal, the POD basis can be found from the singular value decomposition

$$\hat{\mathbf{W}} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^T, \tag{37}$$

where $\mathbf{V} \in \mathbb{R}^{d \times d}$ is an orthogonal matrix and $\boldsymbol{\Sigma}$ a positive real diagonal matrix.

Thus the POD basis $\mathbf{V}_r = \begin{pmatrix} \mathbf{v}_1 & \ldots & \mathbf{v}_r \end{pmatrix}$ is an orthonormal basis and derived from the collected state evolutions (snapshots)

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}(t_1) & \ldots & \mathbf{x}(t_N) \end{pmatrix}. \tag{38}$$

The POD method is particularly popular for systems governed by nonlinear partial differential equations describing computational fluid dynamics. Analytical solutions do not exist for such systems and the collected data may serve as the only adequate description of the system dynamics. The POD basis is found by minimizing the time-averaged approximation error given by

$$\mathrm{av}\left(\| \mathbf{x}(t_k) - \mathbf{x}_n(t_k) \|_2\right). \tag{39}$$

The averaging operator $\mathrm{av}(\cdot)$ is defined as:

$$\mathrm{av}(f) := \frac{1}{N}\sum_{k=1}^{N} f(t_k). \tag{40}$$

Solving the minimization problem of (39) is equivalent to computing the eigenvalue decomposition of $\frac{1}{N}\mathbf{X}\mathbf{X}^T$. Because $\frac{1}{N}\mathbf{X}\mathbf{X}^T$ is a symmetric positive definite matrix there exists an orthogonal matrix $\mathbf{V}_r \in \mathbb{R}^{d\times r}$ and a positive real diagonal matrix $\Lambda_r \in \mathbb{R}^{r\times r}$ such that

$$\frac{1}{N}\mathbf{X}\mathbf{X}^T\mathbf{V}_r = \mathbf{V}_r\Lambda_r. \tag{41}$$

The term $\frac{1}{N}\mathbf{X}\mathbf{X}^T$ equals the state covariance matrix. The POD basis is a subset of the eigenvectors of this covariance matrix and is stored by the matrix $\mathbf{V}_r$. The most important POD basis function is the eigenvector corresponding to the first eigenvalue. The truncation degree is determined from the eigenvalue distribution in Λ_r=diag$(\lambda_1,\ldots,\lambda_r)$. Based on the commonly adopted ad-hoc criterion, the truncation degree r should at least capture 99% of the total energy. The POD basis minimizes, in Least Squares sense, (39) over all possible bases. Error estimates for the solutions obtained from the reduced model are available in [10].

6 Galerkin Projection

For each t let the state $\mathbf{x}(t) \in \mathbb{R}^d$ belong to a separable Hilbert space $\mathscr{X}$, equipped with the Euclidian inner product. Then for all t the state $\mathbf{x}$ can be expanded in a basis $\mathbf{V} = \begin{pmatrix}\mathbf{v}_1 & \ldots & \mathbf{v}_d\end{pmatrix}$

$$\mathbf{x}(t) = \sum_{i=1}^{d} z_i(t)\mathbf{v}_i. \tag{42}$$

The basis is derived from various criteria based on the approximation quality of the original state $\mathbf{x}$ by its truncated expansion $\mathbf{x}_r$ as defined in (43)

$$\mathbf{x}(t) \approx \mathbf{x}_r(t) = \sum_{i=1}^{r} z_i(t)\mathbf{v}_i. \tag{43}$$

The order r of the truncated expansion is lower than the order d of the original expansion. Different reduction methods yield different bases.

The reduced order model is the model that describes the dynamics of the basis coefficients or the reduced state $\mathbf{z} = \{z_1,\ldots,z_r\}$. In many methods the reduced order model

is derived by replacing the original state $\mathbf{x}$ by its truncated expansion $\mathbf{x}_r$ and projecting the original equations onto a truncated basis

$$\mathbf{W}_r = \left(\mathbf{w}_1 \ \ldots \ \mathbf{w}_r\right). \tag{44}$$

Galerkin projection of (1) onto $\mathbf{V}_r$ along $\mathbf{W}_r$ results in the reduced DAE model

$$\begin{cases} \frac{d}{dt}\left[\mathbf{W}_r^T\mathbf{q}(\mathbf{V}_r\mathbf{z})\right] + \mathbf{W}_r^T\mathbf{j}(\mathbf{V}_r\mathbf{z}) + \mathbf{W}_r^T\mathbf{B}\mathbf{u} = \mathbf{0}, \quad \mathbf{z}(0) = \mathbf{z}_0, \\ \mathbf{y} = \mathbf{h}(\mathbf{V}_r\mathbf{z}). \end{cases} \tag{45}$$

The original d-dimensional DAE model is reduced to an r-dimensional DAE reduced order model by means of the Galerkin projection. Unfortunately, the resulting reduced order model (45) for $\mathbf{z} \in \mathbb{R}^r$ is not always solvable for any arbitrary truncation degree r. Furthermore, in contrast to TPWL this reduced model still needs evaluations of the original model, because the functions $\mathbf{V}_r^T\mathbf{q}(t, \mathbf{V}_r\mathbf{z})$ and $\mathbf{V}_r^T\mathbf{j}(t, \mathbf{V}_r\mathbf{z})$ cannot be expanded before the simulation.

For circuit models the snapshots can be collected from a transient simulation with fixed parameters and sources. The reduced model can also be used to approximate the model for different parameters or sources as long as the solution still approximately lies in the projected space. For circuit models with a lot of redundancy the reduced model can have a much smaller dimension. Unfortunately, direct application of POD to circuit models does not work well in practice. Firstly, for Differential Algebraic Equations the Galerkin projection scheme may yield an unsolvable reduced order model. This problem has been studied in more detail in [5,14]. Secondly, the computational effort required to solve the reduced order model and the original model is about the same in nonlinear cases. This is due to the fact that the evaluation costs of the reduced model (45) are not reduced at all because $\mathbf{V}_r$ will be a dense matrix in general.

7 Missing Point Estimation (MPE)

As mentioned before, many MOR techniques for nonlinear systems as (1) use Galerkin projection to obtain a reduced model of the following type

$$\frac{d}{dt}\left[\mathbf{W}^T\mathbf{q}(\mathbf{V}\mathbf{z})\right] + \mathbf{W}^T\mathbf{j}(\mathbf{V}\mathbf{z}) + \mathbf{W}^T\mathbf{B}\mathbf{u} = \mathbf{0}. \tag{46}$$

The original state can be obtained by $\mathbf{x} = \mathbf{V}\mathbf{z}$. Thus indeed it is assumed that $\mathbf{x} \in \mathscr{R}(\mathbf{V})$. If $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{V} \in \mathbb{R}^{d\times r}$ where $r \ll d$ it is clear that the reduced model (46) is of much smaller size than the original model (1). For LTI systems with $\mathbf{q}(t,\mathbf{x}) = \mathbf{C}\mathbf{x}$ and $\mathbf{j}(t,\mathbf{x}) = \mathbf{G}\mathbf{x} - \mathbf{s}(t)$ it is really possible to reduce the simulation time for small r. In particular if the reduced model is diagonalized, we certainly get a model that is very cheap to solve. For the general case it is much worse because then the evaluation costs are not reduced at all. But if the linear algebra part is dominant, we still can expect a speed-up. Despite the resulting low dimensional model, the computational effort required to solve the reduced order model and the original model is relatively the same in nonlinear cases. It may even occur that the original model is cheaper to evaluate than the reduced order model. The low dimensionality is obtained by means of projection, either by the Galerkin projection

method or the least square method. In the projection schemes, the original numerical model must be projected onto the projection space. It implies that the original model must be re-evaluated when the original numerical model is time-varying, which is the general case for nonlinear systems. A consequence is that the evaluation costs for the reduced model are not reduced at all.

Missing Point Estimation (MPE) is a well-known technique that modifies the matrix $\mathbf{V}$ such that only a part of the equations of the original model have to be evaluated. This makes POD applicable for model order reduction of nonlinear DAEs. The Missing Point Estimation (MPE) was proposed in [2] as a method to reduce the computational cost of reduced order, nonlinear, time-varying models. The method is inspired by the Gappy-POD approach that was introduced by Everson and Sirovich in [8]. More details can be found in [5,15].

7.1 Adapted POD Method

Assume that we have a benchmark solution $\tilde{\mathbf{x}}(t)$ of the DAE (1). Consider the snapshot matrix $\mathbf{X} \in \mathbb{R}^{d\times N}$. Consider the singular value decomposition of $\mathbf{X}$:

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T, \tag{47}$$

where $\mathbf{U} \in \mathbb{R}^{d\times d}, \mathbf{V} \in \mathbb{R}^{N\times N}$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{d\times N}$. Thus the correlation matrix satisfies $\mathbf{W} = \frac{1}{N}\mathbf{X}\mathbf{X}^T = \frac{1}{N}\mathbf{U}\Sigma\Sigma^T\mathbf{U}^T$. Because $\Sigma\Sigma^T \in \mathbb{R}^{d\times d}$ is a positive real diagonal matrix we can write $\Sigma\Sigma^T = \Gamma^2$, where $\Gamma \in \mathbb{R}^{d\times d}$ is another positive real diagonal matrix.

In contrast to POD we introduce the matrix $\mathbf{L} = \mathbf{U}\Gamma \in \mathbb{R}^{d\times d}$, such that also $\mathbf{W} = \frac{1}{N}\mathbf{L}\mathbf{L}^T$. Note that the columns of $\mathbf{L}$ are still an orthogonal basis but not orthonormal. Then we transform the original system (1) by writing $\mathbf{x} = \mathbf{L}\mathbf{y}$ and using orthogonal Galerkin projection as follows

$$\frac{d}{dt}\left[\mathbf{L}^T\mathbf{q}(\mathbf{L}\mathbf{y})\right] + \mathbf{L}^T\mathbf{j}(\mathbf{L}\mathbf{y}) + \mathbf{L}^T\mathbf{B}\mathbf{u} = \mathbf{0}, \;\; \mathbf{x} = \mathbf{L}\mathbf{y}. \tag{48}$$

Note that we are able to compute the matrix $\mathbf{L}^T\mathbf{B}$ before the simulation in contrast to the nonlinear functions $\mathbf{L}^T\mathbf{q}(\mathbf{L}\mathbf{y})$ and $\mathbf{L}^T\mathbf{j}(\mathbf{L}\mathbf{y})$. Therefore we are going to approximate $\mathbf{L}^T$ and $\mathbf{L}$ such that $\mathbf{L}^T\mathbf{q}(\mathbf{L}\mathbf{y})$ and $\mathbf{L}^T\mathbf{j}(\mathbf{L}\mathbf{y})$ become cheaper to evaluate. Note that we will use different approximations for $\mathbf{L}$ and $\mathbf{L}^T$. Because $\mathbf{L} = \mathbf{U}\Gamma$ we can approximate it by $\mathbf{U}_r\Gamma_r\mathbf{P}_r = \mathbf{L}\mathbf{P}_r^T\mathbf{P}_r$ where $\mathbf{U}_r \in \mathbb{R}^{d\times r}$ and $\Gamma_r \in \mathbb{R}^{r\times r}$ consists of the r most dominant singular values of Γ and $\mathbf{P}_r \in \{0,1\}^{r\times d}$ is a selection matrix. The matrices $\mathbf{U}_r, \Gamma_r$ and $\mathbf{P}_r$ easily follow from the singular value decomposition. But if we use this approximation we still have the problem that for each function $\mathbf{f}$ the projected function $\mathbf{L}^T\mathbf{f} \approx \mathbf{P}_r^T\Gamma_r\mathbf{U}_r^T\mathbf{f}$ needs all elements of $\mathbf{f}$. Therefore we use here also another approximation

$$\mathbf{L}^T \approx \mathbf{T}_g\mathbf{P}_g = \mathbf{L}^T\mathbf{P}_g^T\mathbf{P}_g, \tag{49}$$

where $\mathbf{P}_g \in \{0,1\}^{g\times d}$ is another selection matrix and $\mathbf{T}_g \in \mathbb{R}^{d\times g}$ contains the g columns of $\mathbf{L}^T$ with largest norm. If the singular values of Γ decrease rapidly we often need just

a small number g of columns. This means that the aliasing error $\|\mathbf{T}_g\mathbf{P}_g - \mathbf{L}^T\|$ also converges rapidly to zero. Now we can approximate the transformed DAE (48) by

$$\frac{d}{dt}\left[\mathbf{L}^T\mathbf{P}_g^T\mathbf{P}_g\mathbf{q}(\mathbf{L}\mathbf{P}_r^T\mathbf{P}_r\mathbf{y})\right] + \mathbf{L}^T\mathbf{P}_g^T\mathbf{P}_g\mathbf{j}(\mathbf{L}\mathbf{P}_r^T\mathbf{P}_r\mathbf{y}) + \mathbf{L}^T\mathbf{B}\mathbf{u} = \mathbf{0},\ \ \mathbf{x} = \mathbf{L}\mathbf{y}. \tag{50}$$

Because $\mathbf{L} \approx \mathbf{L}\mathbf{P}_r^T\mathbf{P}_r$ and $\mathbf{L}^T \approx \mathbf{L}^T\mathbf{P}_g^T\mathbf{P}_g$ it also follows that

$$\mathbf{L}^T \approx \mathbf{P}_r^T\mathbf{P}_r\mathbf{L}^T\mathbf{P}_g^T\mathbf{P}_g. \tag{51}$$

Writing $\mathbf{a} = \mathbf{P}_r\mathbf{y} \in \mathbb{R}^r$ we get the following truncated system of r equations

$$\frac{d}{dt}\left[\mathbf{P}_r\mathbf{L}^T\mathbf{P}_g^T\mathbf{P}_g\mathbf{q}(\mathbf{L}\mathbf{P}_r^T\mathbf{a})\right] + \mathbf{P}_r\mathbf{L}^T\mathbf{P}_g^T\mathbf{P}_g\mathbf{j}(\mathbf{L}\mathbf{P}_r^T\mathbf{a}) + \mathbf{P}_r\mathbf{L}^T\mathbf{B}\mathbf{u} = \mathbf{0},\ \mathbf{x} = \mathbf{L}\mathbf{P}_r^T\mathbf{a}. \tag{52}$$

Because $\mathbf{L} = \mathbf{U}\Gamma$ and $\mathbf{L}^T = \Gamma\mathbf{U}^T$ we can also write this system as

$$\frac{d}{dt}\left[\Gamma_r\mathbf{U}_r^T\mathbf{P}_g^T\mathbf{P}_g\mathbf{q}(\mathbf{U}_r\Gamma_r\mathbf{a})\right] + \Gamma_r\mathbf{U}_r^T\mathbf{P}_g^T\mathbf{P}_g\mathbf{j}(\mathbf{U}_r\Gamma_r\mathbf{a}) + \Gamma_r\mathbf{U}_r^T\mathbf{B}\mathbf{u} = \mathbf{0},\ \ \mathbf{x} = \mathbf{U}_r\Gamma_r\mathbf{a}. \tag{53}$$

This system is still badly scaled. Therefore we have to multiply all equations by Γ_r^{-1} and write $\mathbf{z} = \Gamma_r\mathbf{a}$, such that we get

$$\frac{d}{dt}\left[\mathbf{U}_r^T\mathbf{P}_g^T\mathbf{P}_g\mathbf{q}(\mathbf{U}_r\mathbf{z})\right] + \mathbf{U}_r^T\mathbf{P}_g^T\mathbf{P}_g\mathbf{j}(\mathbf{U}_r\mathbf{z}) + \mathbf{U}_r^T\mathbf{B}\mathbf{u} = \mathbf{0},\ \ \mathbf{x} = \mathbf{U}_r\mathbf{z}. \tag{54}$$

We need just g elements of the functions $\mathbf{q},\mathbf{j}$ in this case. Define $\bar{\mathbf{q}} = \mathbf{P}_g\mathbf{q}, \bar{\mathbf{j}} = \mathbf{P}_g\mathbf{j}$ and the matrices $\mathbf{W}_{r,g} = \mathbf{P}_r\mathbf{U}^T\mathbf{P}_g^T = \mathbf{U}_r^T\mathbf{P}_g^T \in \mathbb{R}^{r\times g}$, $\bar{\mathbf{B}}_r = \mathbf{U}_r^T\mathbf{B}$. Then we get indeed

$$\frac{d}{dt}\left[\mathbf{W}_{r,g}\bar{\mathbf{q}}(\mathbf{U}_r\mathbf{z})\right] + \mathbf{W}_{r,g}\bar{\mathbf{j}}(\mathbf{U}_r\mathbf{z}) + \bar{\mathbf{B}}_r\mathbf{u} = \mathbf{0}, \quad \mathbf{x} = \mathbf{U}_r\mathbf{z}. \tag{55}$$

Because the g selected elements $\bar{\mathbf{q}},\bar{\mathbf{j}}$ of $\mathbf{q},\mathbf{j}$ only need a small subset of the elements of $\mathbf{U}_r\mathbf{z}$, it is possible to replace the dense matrix $\mathbf{U}_r$ by a sparse matrix $\mathbf{P}_h^T\mathbf{P}_h\mathbf{U}_r$ such that all unused rows of $\mathbf{U}_r$ are replaced by zero rows. The selection matrix $\mathbf{P}_h$ can easily be found from the average absolute values of the Jacobian matrices $\mathbf{C},\mathbf{G}$ along the benchmark solution. For many applications, e.g. circuit models, the required number h of rows is just slightly larger than g. Thus the matrix $\bar{\mathbf{U}}_{r,h} = \mathbf{P}_h\mathbf{U}_r \in \mathbb{R}^{h\times r}$ is often of a relatively small size. In this manner we finally get the following reduced model for $\mathbf{z} \in \mathbb{R}^r$

$$\frac{d}{dt}\left[\mathbf{W}_{r,g}\bar{\mathbf{q}}(\mathbf{P}_h^T\bar{\mathbf{U}}_{r,h}\mathbf{z})\right] + \mathbf{W}_{r,g}\bar{\mathbf{j}}(\mathbf{P}_h^T\bar{\mathbf{U}}_{r,h}\mathbf{z}) + \bar{\mathbf{B}}_r\mathbf{u} = \mathbf{0},\ \ \mathbf{x} = \mathbf{U}_r\mathbf{z}. \tag{56}$$

This reduced model can be simulated very efficiently because it does not need expensive function evaluations.

8 Applications

We consider the academic diode chain model shown in Fig. 2, that is described by the following equations

$$
\begin{aligned}
V_1 - U_{\text{in}}(t) &= 0,\\
i_E - g(V_1, V_2) &= 0,\\
g(V_1, V_2) - g(V_2, V_3) - C\tfrac{d}{dt}V_2 - \tfrac{1}{R}V_2 &= 0,\\
&\vdots\\
g(V_{N-1}, V_N) - g(V_N, V_{N+1}) - C\tfrac{d}{dt}V_N - \tfrac{1}{R}V_N &= 0,\\
g(V_N, V_{N+1}) - C\tfrac{d}{dt}V_{N+1} - \tfrac{1}{R}V_{N+1} &= 0,
\end{aligned}
$$

$$
g(V_a, V_b) = \begin{cases} I_s(e^{\frac{V_a - V_b}{V_T}} - 1) & \text{if } V_a - V_b > 0.5\ \text{V},\\ 0 & \text{otherwise,} \end{cases}
$$

$$
U_{\text{in}}(t) = \begin{cases} 20 & \text{if } t \le 10\ \text{ns},\\ 170 - 15t & \text{if } 10\ \text{ns} < t \le 11\ \text{ns},\\ 5 & \text{if } t > 11\ \text{ns}. \end{cases}
$$

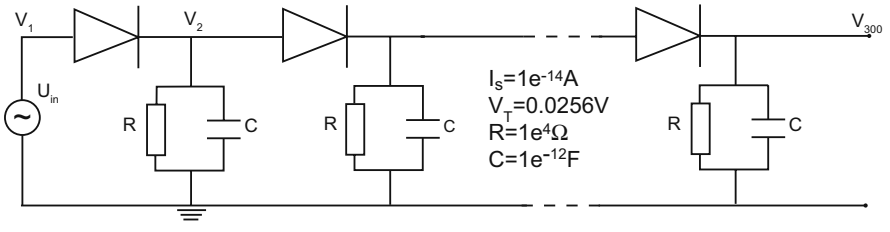

Fig. 2. Structure of the test circuit

Fig. 3 shows the numerical solution (nodal voltage in each node) of the original model at $[0, 70\ \text{ns}]$, computed in MATLAB by the Euler Backward method with fixed step sizes of 0.1 ns.

Fig. 4 indicates the redundancy of the model, as most of the eigenvalues of the correlation matrix $\frac{1}{N}\mathbf{X}\mathbf{X}^T$ can be neglected (left) and also the aliasing error of $\mathbf{L}^T$ rapidly converges to zero (right). Fig. 5 shows the relative errors over all nodes in the time interval $[0, 70\ \text{ns}]$, defined as $\frac{||\mathbf{V_z} - \mathbf{x}||}{||\mathbf{x}||}$, for the reduced models of different orders constructed by TPWL (left) and POD (right). For TPWL the relative error is most of the time lower then the chosen error bound $\varepsilon = 0.025$. Furthermore, for higher order reduced models, a smaller number of LPs has been used than for the reduced models with lower order, as the local systems with higher orders are more accurate. E.g. for a reduced model of order 100 we have used 42 LPs and for smaller reduced models 60 LPs. The POD models are, as expected, more accurate, but much slower to simulate than the TPWL models (see the corresponding extraction and simulation times in Table 1). However, a

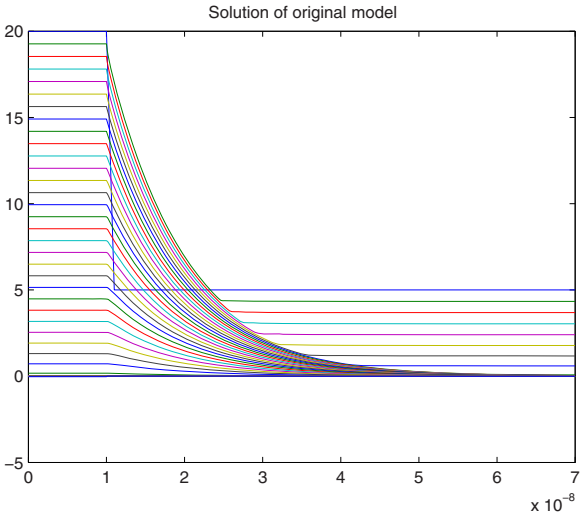

Fig. 3. Numerical solution of the full-scale nonlinear diode chain model

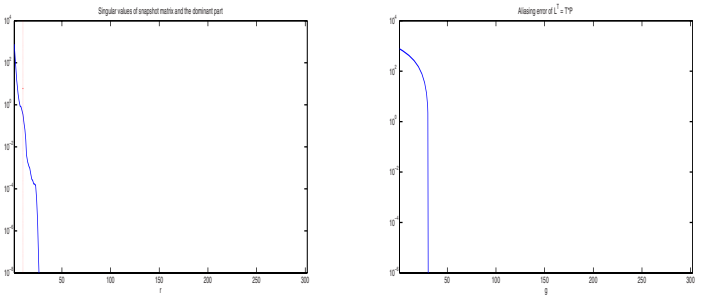

Fig. 4. The eigenvalues of the correlation matrix $\frac{1}{N}XX^T$ (left) and the aliasing error of $\mathbf{L}^T$ (right)

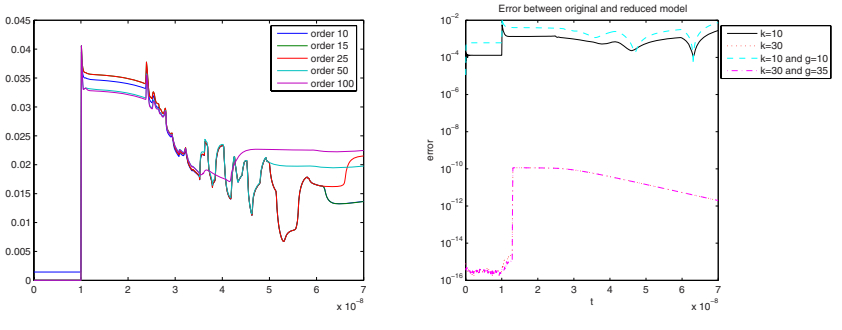

Fig. 5. Relative errors over all nodes for the reduced models created by TPWL (left) and by POD (right)

Table 1. Comparison of extraction and simulation times in seconds

Model	r	Extr. time	Sim. time	Model	r	g	Extr. time	Sim. time
Original	302	0	79	POD	10	302	107	72
TPWL	10	285	1.0	POD	30	302	107	74
TPWL	25	278	1.4	POD + MPE	10	10	107	3.9
TPWL	50	202	2.4	POD + MPE	30	35	107	10.2

significant speed up can be achieved by combining the POD with MPE. The MATLAB scripts can be optimized by using the command `pcode *.m`. Using also a modified Newton method it is even possible to simulate the smallest POD model ($k = g = 10$) in 2.4 seconds, which is even about 33 times faster! These results highly improve the numerical results in [14] for the same example.

9 Conclusion and Outlook

In this paper we studied how nonlinear IC models can be reduced by TPWL and POD. The first method has the advantage that it really approximates the system behavior of the linearized model. Well-developed linear model reduction techniques can be used to reduce the linearized models. However, to maintain sufficient accuracy a large number of LPs is required, which implies a large extraction time. The POD method delivers reduced models which are more accurate because there is no linearization error. Adapted versions are necessary to achieve a reduction of the simulation time at all because of the expensive function evaluations. TPWL and POD have in common that the reduced model is created around a benchmark solution that has to be found first. To make nonlinear MOR applicable in practice it is therefore essential that a proper benchmark solution can be calculated. This could be done by a cheap integration method at a coarse time-grid or in a hierarchical way from typical trajectories per subcircuit. Both the MOR methods TPWL and POD seems to be promising for reducing the simulation time for nonlinear DAE systems. They offer a good starting point for further research on MOR of non-linear dynamical systems.

Acknowledgements. We would like to thank Dr. B. Tasić for his help with the diode chain model and Dr. J. Rommes for his support with the tool Hstar.

References

1. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. Society for Industrial and Applied Mathematics, Philadelphia (2005)
2. Astrid, P.: Reduction of Process Simulation Models: a Proper Orthogonal Decomposition Approach: Ph.D. Thesis, Eindhoven University of Technology, Department of Electrical Engineering (2004)
3. Astrid, P., Weiland, S.: On the construction of POD models from partial observations. In: Proceedings of the 44th IEEE Conf. on Decision and Control, Spain, pp. 2272–2277 (2005)

4. Astrid, P., Weiland, S., Willcox, K., Backx, A.: Missing Point Estimation in models described by Proper Orthogonal Decomposition. In: Proceedings of the 43th IEEE Conf. on Decision and Control, Bahamas, vol. 2, pp. 1767–1772 (2004)
5. Astrid, P., Verhoeven, A.: Application of Least Squares MPE technique in the reduced order modeling of electrical circuits. In: Yamamoto, Y., Sugie, T., Ohta, Y. (eds.) Proceedings of 17th Int. Symposium on Mathematical Theory of Networks and Systems (CDROM), Japan, pp. 1980–1986 (2006)
6. Benner, P., Mehrmann, V., Sorensen, D.C.: Dimension Reduction of Large-Scale Systems. Springer, Heidelberg (2006)
7. d'Elia, M.: Reduced Basis Method and Model Order Reduction for Initial Value Problems of Differential Algebraic Equations: Master's Thesis, Politecnico di Milano, Facoltà di Ingegneria dei Sistemi, Milano Leonardo (2007)
8. Everson, R., Sirovich, L.: The Karhunen-Luève procedure for gappy data. Journal Opt. Soc. Am. 12, 1657–1664 (1995)
9. Freund, R.W.: Krylov-subspace methods for reduced order modeling in circuit simulation. Journal of Computational and Applied Mathematics 123, 395–421 (2000)
10. Homescu, C., Petzold, L.R., Serban, R.: R Serban: Error estimation for reduced-order models of dynamical systems. SIAM Journal on Numerical Analysis 43, 1693–1714 (2005)
11. Odabasioglu, A., Celik, M., Pileggi, L.T.: PRIMA: Passive reduced-order interconnect macromodeling algorithm. IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems 17(8), 645–654 (1998)
12. Phillips, J., Silvera, L.M.: Poor Man's TBR: A simple model reduction scheme. IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems 14(1) (2005)
13. Rewienski, M., White, J.: A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. In: Proc. of the Int. Conf. on CAD, pp. 252–257 (2001)
14. Verhoeven, A.: Redundancy Reduction of IC Models by Multirate Time-Integration and Model Order Reduction: Ph.D. thesis, Eindhoven University of Technology, Department of Mathematics and Computer Science, Eindhoven (2008)
15. Verhoeven, A., Voss, T., Astrid, P., ter Maten, E.J.W., Bechtold, T.: Model order reduction for nonlinear problems in circuit simulation. Presented at the ICIAM Conf., Zürich (2007)
16. Voss, T.: Model Reduction for Nonlinear Differential Algebraic Equations: Master's Thesis, University of Wuppertal (2005)
17. Voss, T., Pulch, R., ter Maten, E., El Guennouni, A.: Trajectory piecewise linear approach for nonlinear differential-algebraic equations in circuit simulation. In: Ciuprina, G., Ioan, D. (eds.) Scientific Computing in Electrical Engineering, Sinaia, Romania, pp. 167–173. Springer, Heidelberg (2007)
18. Voss, T., Verhoeven, A., Bechtold, T., ter Maten, J.: Model order reduction for nonlinear differential-algebraic equations in circuit simulation. In: Bonilla, L.L., Moscoso, M., Vega, J.M. (eds.) Progress in Industrial Mathematics at ECMI 2006, Madrid, Spain, pp. 518–523. Springer, Heidelberg (2006)
19. Willcox, K.: Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. Computers and Fluids 35, 208–226 (2006)