TECHNISCHE UNIVERSITÄT CHEMNITZ

Michael Striebel Joost Rommes

Model order reduction of nonlinear systems: status, open issues, and applications

CSC/08-07

Chemnitz Scientific Computing Preprints

Impressum:

Chemnitz Scientific Computing Preprints — ISSN 1864-0087
(1995–2005: Preprintreihe des Chemnitzer SFB393)

Herausgeber:
Professuren für
Numerische und Angewandte Mathematik
an der Fakultät für Mathematik
der Technischen Universität Chemnitz

Postanschrift:
TU Chemnitz, Fakultät für Mathematik
09107 Chemnitz
Sitz:
Reichenhainer Str. 41, 09126 Chemnitz

http://www.tu-chemnitz.de/mathematik/csc/

TECHNISCHE UNIVERSITÄT CHEMNITZ

Chemnitz Scientific Computing Preprints

Michael Striebel Joost Rommes

Model order reduction of nonlinear systems: status, open issues, and applications

CSC/08-07

Abstract

In this document we review the status of existing techniques for nonlinear model order reduction by investigating how well these techniques perform for typical industrial needs. In particular the TPWL-method (Trajectory Piecewise Linear-method) and the POD-approach (Proper Orthogonal Decomposion) is taken under consideration. We address several questions that are (closely) related to both the theory and application of nonlinear model order reduction techniques. The goal of this document is to provide an overview of available methods together with a classification of nonlinear problems that in principle could be handled by these methods.

CSC/08-07 ISSN 1864-0087 November 2008

The work presented is financed by the Marie Curie Host Fellowships for Transfer of Knowledge project O-MOORE-NICE! (Call identifier: FP6-2005-Mobility-3).

MOORE
NICE!

http://www.tu-chemnitz.de/mathematik/industrie_technik/projekte/omoorenice/

Author's addresses:

Michael Striebel
Professur Mathematik in Industrie und Technik
Fakultät für Mathematik
TU Chemnitz
D-09107 Chemnitz
michael.striebel@mathematik.tu-chemnitz.de

Joost Rommes
NXP Semiconductors
Corporate I&T / Design Technology & Flows
High Tech Campus 37, PostBox WY4-01
NL-5656 AE Eindhoven
The Netherlands
joost.rommes@nxp.com

Contents

1 Introduction

In this document we review the status of existing techniques for nonlinear model order reduction by investigating how well these techniques perform for typical industrial needs. We address several questions that are (closely) related to both the theory and application of nonlinear model order reduction techniques. The goal of this document is to provide an overview of available methods together with a classification of nonlinear problems that in principle could be handled by these methods.

Two well-known techniques for nonlinear model order reduction, Proper Orthogonal Decomposition [36, 6, 18, 24, 44, 30, 2, 19] or Karhunen-Loève expansion [25] and trajectory piecewise-linear (TPWL) methods [31, 32], have been studied extensively in the literature, but to less extent in the context of industrial applications [43, 39]. In this document we will address the latter point, i.e., application to industrial examples. In particular, we will investigate the nonlinear model order reduction problem and existing techniques by using the following questions as guidelines:

- What are the typical applications and problems of interest, i.e., what are the candidates for application of nonlinear model order reduction techniques?
- Where is reduction of nonlinear systems needed? Is the main need in speed up of simulations and/or is there a big need for reusable models of nonlinear systems?
- What type of reduction is needed? Is it, like for linear systems, mainly a reduction in the number of states and elements, or is a reduction/simplification of the nonlinearity of the system required?
- How well do existing techniques perform and which improvements are needed?
- Can multirate time integration techniques [39, 34, 37] be of use for nonlinear model order reduction?
- A reduced order model that is only accurate for a few inputs is of little practical value. How can reduced order models of nonlinear systems be made reusable? How do the training inputs need to be chosen in order to produce reusable models?

Most of the questions are discussed in detail in sections 4 – 7. For a summary of the answers to these questions, the reader is referred to section 9. The rest of this document is devoted to describing the theoretical and practical results that are needed to answer these questions.

This document is organized as follows. The problem of reduction of nonlinear system is formulated in section 2. In section 3 we give an overview of existing methods. Section 4 discusses the current status of these methods by identifying advantages, disadvantages, and incapabilities. Open issues are addressed in section 5. In section 6 we provide a classification of nonlinear problems and discuss which and how nonlinear model order reduction techniques can be applied. Numerical experiments are reported in section 7. Section 8 comments on implementation of nonlinear MOR in a circuit simulator and design flow. Section 9 concludes.

2 Reduction of nonlinear systems

The problem of reducing a nonlinear system is described as follows: Given a, possibly large-scale, nonlinear time-invariant dynamical system $\Sigma = (\mathbf{g}, \mathbf{f}, \mathbf{h}, \mathbf{x}, \mathbf{u}, \mathbf{y}, t)$

$$\Sigma = \begin{cases} \frac{d\mathbf{g}(\mathbf{x}(t))}{dt} & = & \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) & = & \mathbf{h}(\mathbf{x}, \mathbf{u}) \end{cases}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$, $\mathbf{y}(t) \in \mathbb{R}^p$, $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \mathbf{g}(\mathbf{x}(t)) \in \mathbb{R}^n$, $\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) \in \mathbb{R}^p$, find a reduced model $\widetilde{\Sigma} = (\tilde{\mathbf{g}}, \tilde{\mathbf{f}}, \tilde{\mathbf{h}}, \tilde{\mathbf{x}}, \mathbf{u}, \tilde{\mathbf{y}}, t)$

$$\widetilde{\Sigma} = \begin{cases} \frac{d\tilde{\mathbf{g}}(\tilde{\mathbf{x}}(t))}{dt} & = & \tilde{\mathbf{f}}(\tilde{\mathbf{x}}(t), \mathbf{u}(t)) \\ \tilde{\mathbf{y}}(t) & = & \tilde{\mathbf{h}}(\tilde{\mathbf{x}}, \mathbf{u}) \end{cases}$$

where $\tilde{\mathbf{x}}(t) \in \mathbb{R}^k$, $\mathbf{u}(t) \in \mathbb{R}^m$, $\tilde{\mathbf{y}}(t) \in \mathbb{R}^p$, $\tilde{\mathbf{f}}(\tilde{\mathbf{x}}(t), \mathbf{u}(t)), \tilde{\mathbf{g}}(\tilde{\mathbf{x}}(t)) \in \mathbb{R}^k$, $\tilde{\mathbf{h}}(\tilde{\mathbf{x}}(t), \mathbf{u}(t)) \in \mathbb{R}^p$, such that $\tilde{\mathbf{y}}(t)$ can be computed in much less time than $\mathbf{y}(t)$ and the approximation error $\mathbf{y}(t) - \tilde{\mathbf{y}}(t)$ is small.

Note that unlike for reduction of linear dynamical systems, there is here no explicit requirement $k \ll n$, since it is not clear whether such a requirement will help in achieving reduced simulation times.

In the context of circuit simulation the dynamical systems we are dealing with circuit blocks or subcircuits. Connection to and communication with a block's environment is done via its terminals, i.e. external nodes. Therefore, we can assume that the currents or voltages are always injected linearly into the circuit under consideration. A similar reasoning applies for the determination of the output signal $\mathbf{y}(t)$, which is also assumed to be not explicitly dependent on the input $\mathbf{u}(t)$. Hence, in the remainder of this document, we assume the dynamical systems to be of the form

$$\Sigma = \begin{cases} \frac{d\mathbf{g}(\mathbf{x}(t))}{dt} & = & \mathbf{f}(\mathbf{x}(t)) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) & = & \mathbf{C}^T\mathbf{x} \end{cases}$$

where $\mathbf{B} \in \mathbb{R}^{n\times m}$ and $\mathbf{C} \in \mathbb{R}^{n\times p}$.

3 Methods for nonlinear model order reduction

The two best-known methods for reduction of nonlinear systems are Proper Orthogonal Decomposition (POD), also known as Karhunen-Loève expansion [25], and trajectory piecewise-linear techniques (TPWL) [31, 32], which are discussed in section 3.1 and section 3.2, respectively. In section 3.3 an overview of variants and other methods is given.

3.1 Proper Orthogonal Decomposition (POD)

Proper Orthogonal Decomposition extends the Petrov-Galerkin projection based methods that are used for linear systems to nonlinear systems. By choosing a suitable $\mathbf{V} \in \mathbb{R}^{n\times k}$ and a test matrix $\mathbf{W} \in \mathbb{R}^{n\times k}$, where $\mathbf{W}$ and $\mathbf{V}$ are biorthonormal, i.e., $\mathbf{W}^T\mathbf{V} = \mathbf{I_{n\times n}}$, the reduced system is given by [1]

$$\left\{\begin{array}{rcl} \mathbf{W}^T\frac{d\mathbf{g}(\mathbf{V}\tilde{\mathbf{x}}(t))}{dt} & = & \mathbf{W}^T\mathbf{f}(\mathbf{V}\tilde{\mathbf{x}}(t)) + (\mathbf{W}^T\mathbf{B})\mathbf{u}(t) \\ \tilde{\mathbf{y}}(t) & = & (\mathbf{C}^T\mathbf{V})\tilde{\mathbf{x}} \end{array}\right.$$

Similar to linear model order reduction, the idea is that $\mathbf{V}$ captures the dominant dynamics, i.e., the states of the original system are approximated well by $\mathbf{V}\tilde{\mathbf{x}} \approx \mathbf{x}$. The test matrix $\mathbf{W}$ is chosen such that the Petrov-Galerkin condition $\mathbf{r} = \frac{d\mathbf{g}(\mathbf{V}\tilde{\mathbf{x}}(t))}{dt} - \mathbf{f}(\tilde{\mathbf{x}}(t)) - \mathbf{B}\mathbf{u}(t) \perp \mathbf{W}$ is met.

POD constructs the matrix $\mathbf{V}$ as follows. A time domain simulation of the complete system is done and snapshots of the states at suitably chosen times t_i are collected in the state matrix $\mathbf{X}$

$$\mathbf{X} = [\mathbf{x}(t_0),\ \mathbf{x}(t_1),\ \mathbf{x}(t_2),\ \cdots \mathbf{x}(t_{N-1})] \in \mathbb{R}^{n\times N},$$

where N is the number of time points t_i. To extract the subspace that represents that dominant dynamics, the singular value decomposition [16] of X is computed:

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{T},$$

where $\mathbf{U} \in \mathbb{R}^{n\times n}$, $\Sigma = [\mathrm{diag}(\sigma_1, \ldots, \sigma_n)\ 0_{n\times(N-n)}] \in \mathbb{R}^{n\times N}$ (if $N > n$), and $\mathbf{T} \in \mathbb{R}^{N\times N}$. Let the singular values $\sigma_1 \geq \sigma_2 \cdots \sigma_k \gg \sigma_{k+1} > \cdots > \sigma_n \geq 0$ be ordered in decreasing magnitude. POD chooses the matrix $\mathbf{V}$ to have as its columns the left singular vectors corresponding to the $k \ll n$ largest singular values:

$$\mathbf{V} = [\mathbf{u}_1,\ \mathbf{u}_2, \cdots, \mathbf{u}_k] \in \mathbb{R}^{n\times k}.$$

The number k of vectors to choose can depend on a tolerance based criterion like $\sigma_{k+1} < \epsilon$, or on the relative difference between σ_k and σ_{k+1}. The test matrix $\mathbf{W}$ is taken as $\mathbf{W} = \mathbf{V}$, i.e., the residual is orthogonal to the reduced state space.

The choice of the number of snapshots N depends on many factors. On the one hand one desires $N \ll n$ to limit the computational costs involved with the SVD, on the other hand all the dominant states should be captured. Also the choice of the snapshot times t_i can be a challenging task [29].

We stress that the reduction obtained from POD and similar projection based methods is solely in the number of states: k for the reduced systems vs. n for the original system and $k \ll n$. As also clearly argued in [31, section 2.3], the costs for evaluating nonlinear terms such as $\mathbf{W}^T\tilde{\mathbf{f}}(\mathbf{V}\tilde{\mathbf{x}}(t))$ will be larger than for the original system because at each timepoint

1. the reduced state $\tilde{\mathbf{x}}$ has to be projected back to the full state space, necessitating a matrix-vector product,

2. the complete function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ has to be evaluated,

3. the value obtained has to be projected back to the test space, and

4. the computation of the reduced system's Jacobian necessitates to compute the Jacobian of the full system.

Hence with respect to simulation times no reduction will be obtained (unless additional measures are taken, see also section 4).

3.2 Trajectory piecewise-linear techniques (TPWL)

Trajectory piecewise linear techniques [31, 32, 43] linearize the nonlinear system around suitably selected states and approximate the nonlinear system by a piecewise linearization that is obtained from combining the (reduced) linearized systems via a weighting procedure.

Having s states $\mathbf{x}_0, \ldots, \mathbf{x}_{s-1}$, obtained from simulation of the original system on some finite time interval $[t_{\text{start}}, t_{\text{end}}]$, we linearize the original system around these states:

$$\frac{d}{dt}(\mathbf{g}(\mathbf{x}_i) + \mathbf{G_i}(\mathbf{x}(t) - \mathbf{x}_i)) = \mathbf{f}(\mathbf{x}_i) + \mathbf{F_i}(\mathbf{x}(t) - \mathbf{x}_i) + \mathbf{B}\mathbf{u}(t) \tag{1}$$

where $\mathbf{x}_0$ is the initial state of the system and $\mathbf{G_i}$ and $\mathbf{F_i}$ are the Jacobians of $\mathbf{g}$ and $\mathbf{f}$ evaluated at $\mathbf{x}_i$. Since each of the linearizations approximates the nonlinear system in the neighborhood of the expansion point $\mathbf{x}_i$, a model including all these linearizations could approximate the original system over a larger time interval and larger part of the state space. In [31] a weighting procedure is described to combine the models. The piecewise linear model becomes[1]

$$\frac{d}{dt}\left[\sum_{i=0}^{s-1} w_i(\mathbf{x})\mathbf{G_i}\mathbf{x}\right] = \sum_{i=0}^{s-1} w_i(\mathbf{x})\Big(\mathbf{f}(\mathbf{x}_i) + \mathbf{F_i}(\mathbf{x}-\mathbf{x}_i)\Big) + \mathbf{B}\mathbf{u}(t), \tag{2}$$

where $w_i(\mathbf{x})$ are state-dependent weights. A typical choice is to let $w_i(\mathbf{x})$ be large for $\mathbf{x} = \mathbf{x}(t)$ close to $\mathbf{x}_i$, and small otherwise, but other and more advanced schemes are also available [31]. Including time t in the process of calculating distances to linearisation tuples (x_i, t_i) $(i = 0, \cdots, s-1)$ can have distatrous effects. Resimulation with the same but time shifted input signal might already break down.

Simulation of a piecewise linearized system may already be faster than simulation of the original nonlinear system. However, the linearized system can be reduced by using model order techniques for linear systems.

The main difference between linear MOR and the nonlinear MOR-approach TPWL is that the latter introduces in addition to the application of a linear MOR technique the selection of linearization points (to get a linear problem) and the weighting of the linear submodels (to recover the global nonlinear behavior).

Selection of linearization points. The model extraction basically needs the solution of the full nonlinear system. In [31] a fast extraction method is proposed, but we will not give details here.

The TPWL-scheme is based on deciding when to add a linear substitute for the nonlinear problem automatically during simulation of the latter. Again there are several alternatives. Rewieński [31] proposes to check at each accepted timepoint t during simulation for the relative distance of the current state $\mathbf{x}$ of the nonlinear problem to all yet existing i linearization states $\mathbf{x}_0, \ldots, \mathbf{x}_{i-1}$. If the minimum is equal to or greater than some parameter $\delta > 0$, i.e.

$$\min_{0\le j\le i-1}\left(\frac{\|\mathbf{x}-\mathbf{x}_j\|_\infty}{\|\mathbf{x}_j\|_\infty}\right) \ge \delta, \tag{3}$$

$\mathbf{x}$ becomes the $(i+1)$st linearization point. Accordingly, a new linear model, arising from linearizing around $\mathbf{x}$ is added to the collection. To our experience as we will show later, the choice of δ already marks a critical point in the process. Here, Rewieński suggests to first calculate the steady state $\mathbf{x}_T$ of the linear system that arises from linearizing the nonlinear system at the DC-solution $\mathbf{x}_0$ and then setting $\delta = \frac{d}{10}$ where

$$d = \frac{\|\mathbf{x}_T - \mathbf{x}_0\|_\infty}{\|\mathbf{x}_0\|_\infty} \quad (d = \|\mathbf{x}_T\|_\infty \text{ if } \mathbf{x}_0 = 0). \tag{4}$$

[1] Note that $\frac{d}{dt}(\mathbf{g}(\mathbf{x_i}) - \mathbf{G_i}\mathbf{x_i}) \equiv \mathbf{0}$.

Choosing linearization points according to the criterion (3) essentially bases the decision on the slope of the trajectory the training is done on and not on the quality of the linear substitute model w.r.t. the nonlinear system. In Voß [43] the mismatch of nonlinear and linear system motivates the creation of a new linearization point and an additional linear model: as at each timepoint during training both the nonlinear and a currently active system are available, the latter one is computed in parallel to the former one. If the difference of the two approximations to the true solution at a timepoint t_n produced by the different models becomes too large, a new linear model is created from linearizing the nonlinear system around the state the system was in at the previous timepoint t_{n-1}.

Note that in both strategies a linear model around a linearization state $\mathbf{x}(t_{n-1})$ is constructed when the linear model arising from linearization around a former state does not extend to $\mathbf{x}(t_{n-1})$. However, it is not guaranteed that this new model extends backward, i.e., is able to reproduce the situation encountered before t_{n-1}. This circumstance could have a negative effect in resimulation where linear combinations of linear substitute systems are used to replace the full nonlinear system. That means during model extraction one deals with just one linear system in each situation but with combinations during resimulation.

Determination of the weights. During the training the nonlinear functions have been fragmented into linear ones, each part reflecting certain aspects of the “parent function”. When using the substitute collection for simulation, one will naturally aim at having to deal with a combination of just a small number of linear submodels. Hence, the weighting function has to have steep gradients to determine a few (in the ideal case just one) dominant linear models. As in Rewieński’s work we implemented a scheme that is depending on the absolute distance of a state to the linearization points. The importance of each single model is defined by

$$w_i(\mathbf{x}) = e^{-\frac{\beta}{m}\cdot\|\mathbf{x}-\mathbf{x}_i\|_2}, \quad \text{with } m = \min_i \|\mathbf{x}-\mathbf{x}_i\|_2. \tag{5}$$

With the constant β we can steer how abrupt the change of models is. In Rewieński [31] $\beta = 25$ is chosen. To guarantee a convex combination, the weights are normalized such that $\sum_i w_i(\mathbf{x}) = 1$.

Reduction of linear submodels. Basically, any MOR-technique for linear problems can be applied to the linear submodels. In [31] Rewieński proposes the usage of Krylov-based reduction using the Arnoldi-method, Vasilyev, Rewieński and White [38] introduce balanced truncation to TPWL and Voß [43] uses Poor Man’s TBR as linear MOR kernel. For comparison of different linear MOR strategies when applied to problems in circuit simulation we refer to [41, 22, 42].

The common basis of all these methods is that one restricts to a dominant subspace of the state-space. Suppose that the columns of $\mathbf{V} \in \mathbb{R}^{n\times k}$ span this dominant subspace of the state-spaces of the linearized systems, and that $\mathbf{W} \in \mathbb{R}^{n\times k}$ is the corresponding test matrix. Then a reduced order model for the piecewise-linearized system can be obtained as

$$\frac{d}{dt}\left[\sum_{i=0}^{s-1} w_i(\mathbf{V}\tilde{\mathbf{x}})\Big(\mathbf{W}^T\mathbf{g}(x_i) + \mathbf{W}^T G_i \mathbf{V}(\mathbf{V}\tilde{\mathbf{x}} - \mathbf{x}_i)\Big)\right]$$
$$= \sum_{i=0}^{s-1} w_i(\mathbf{V}\tilde{\mathbf{x}})\Big(\mathbf{W}^T\mathbf{f}(\mathbf{x}_i) + \mathbf{W}^T F_i \mathbf{V}(\mathbf{V}\tilde{\mathbf{x}} - \mathbf{x}_i)\Big) + \mathbf{W}^T\mathbf{B}\mathbf{u}. \quad (6)$$

Here, all linear submodels are reduced by projection with the overall matrices $\mathbf{V}$ and $\mathbf{W}$. Besides the choice of the reduction scheme, as mentioned above, the construction of these matrices is a further degree of freedom. In [31] two different strategies are presented. A simple approach is to regard only the linear model that arises from linearizing around the starting value $\mathbf{x}_0$, construct a reduced basis $\mathbf{V}_0$ and an according test space $\mathbf{W}_0$ for this linear model and set $\mathbf{V} = \mathbf{V}_0$ and $\mathbf{W} = \mathbf{W}_0$. Hereby one assumes that the characterization dominant vs. not dominant does not change dynamically. In a more sophisticated manner, one derives reduced bases and test spaces $\mathbf{V}_i$, $\mathbf{W}_i$, respectively, for each linear submodel $i = 0, \ldots, s-1$ and constructs $\mathbf{V}$ and $\mathbf{W}$ from $\{\mathbf{V}_0, \ldots, \mathbf{V}_{s-1}\}$ and $\{\mathbf{W}_1, \ldots, \mathbf{W}_{s-1}\}$. For more details we refer to [31, 43].

Construction of reduced order basis. For constructing a reduced order basis we have to take into account the linear submodels.

The simplest approach bases the reduction on the linear model that arises from linearization around the DC solution $\mathbf{x}_0$ only. From the corresponding system's matrices $\mathbf{G_0}, \mathbf{F_0}, \mathbf{B}$ (cf. (1)) a basis $\{\mathbf{v_1}, \ldots, \mathbf{v_l}\}$ for the lth order Krylov subspace using the Arnoldi algorithm might be constructed. Then the matrix $\mathbf{V} = [\mathbf{v_1}, \ldots, \mathbf{v_l}, \tilde{\mathbf{v}}_\mathbf{0}] \in \mathbb{R}^{N\times(l+1)}$, where $\tilde{\mathbf{v}}_\mathbf{0}$ arises from orthonormalization of $\mathbf{x}_0$ versus $\mathbf{v_1}, \ldots, \mathbf{v_l}$ can be taken as the projection matrix for Galerkin projection of the linear combination of the linear submodels.

A second, extended, approach might take into account all linear submodels. Here in a first step reduced order models for each single linear subsystem are constructed, which yields local reduced subspaces spanned by the columns of $\mathbf{V}_0, \ldots, \mathbf{V}_{s-1}$. In a second step an SVD is done on the aggregated matrix $\mathbf{V}_{\text{agg}} = [\mathbf{V}_0, \mathbf{x}_0; \ldots; \mathbf{V}_{s-1}, \mathbf{x}_{s-1}]$. The final reduced subspace is then spanned by the dominating left singular vectors. Note that by this truncation it can not be guaranteed that the linearization points are in the reduced space.

Determination of the weights for reduced linearized models. Note, that in (5) the absolute distance $\|\mathbf{x} - \mathbf{x}_i\|_2$ is taken in the full space $\mathbb{R}^n$. When using the reduced order model (6) for simulation one has to prolongate $\tilde{\mathbf{x}} \in \mathbb{R}^k$ to the full space. Computational costs could be reduced if this reprojection was not necessary, i.e. if we could measure the distances in the reduced space already. If the linearization points $\mathbf{x}_0, \dots, \mathbf{x}_{s-1}$ are in the space spanned by the columns of $\mathbf{V}$, it suffices to project them once to the reduced space and take the distances there, i.e. calculate $\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_i\|$ instead (cf. [31]). In the cited reference, it is not stated if extra steps are taken to guarantee that the linearization states are contained in the reduced space. Adding the linearization states to $\mathbf{V}$ after orthogonalizing them against the columns of $\mathbf{V}$ could be an appropriate activity, probably increasing the dimension of the reduced space.

However, taking no extra steps and just projecting the linearisation points to the reduced space to take the distances there can be very dangerous as we present in Section 7.2.

Therefore, we strongly recommend to project the reduced space back to the full space for measuring the distance to the linearisation points.

3.3 Other techniques

Other techniques for reduction of nonlinear systems are Nonlinear Balanced Truncation [35, 21], Empirical Balanced Truncation [24] and higher-order polynomial expansion schemes [28].

Nonlinear Balancing. Nonlinear balancing extends Balanced Trunctation [27] for linear systems to the nonlinear case. The main terms of linear balanced truncation are the reachability and observability Gramians P and Q, respectively. These can be computed from Lyapunov equations, involving the system matrices $A, \mathbf{B}, \mathbf{C}, \mathbf{D}$ of the linear system. Knowing the Gramians, the energies connected to reach and observe a state can be determined by means of algebraic calculations. The full linear system is transformed to a *balanced representation*, i.e., to a form where states that are hard to observe are also hard to reach. Truncation is then done by eliminating these states. For this P and Q are simultaneously diagonalised:

$$P = Q = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n, \end{pmatrix}$$

where the so called Hankel singular values $\sigma_1, \dots, \sigma_n$ are the square roots of the eigenvalues of the product PQ. From the basis that arises from the transformation only those basis vectors that correspond to large Hankel singular values are kept. The main advantage of this approach is that there exists an a priori computable error bound for the truncated system.

In the nonlinear case there exist no constant Gramians that can be consulted for measuring the degree of reachability and observability. The reachability and observability functions $L_c(x)$ and $L_o(x)$, respectively, result from two differential equations, namely a Hamiltonian-Jacobi equation and a nonlinear Lyapunov-type equation.

A local coordinate transform $x = \Psi(z)$ (with $0 = \Psi(0)$) can be found such that the system is in *input normal representation*:

$$L_c(\Psi(z)) = \frac{1}{2} z^T z \quad \text{and} \quad L_o(\Psi(z)) = \frac{1}{2} z^T \begin{pmatrix} \tau_1(z) & & \\ & \ddots & \\ & & \tau_n(z) \end{pmatrix} z,$$

where $\tau_1, \ldots, \tau_n$ are smooth functions such that $\tau_1(z) \geq \cdots \geq \tau_n(z)$. These functions are called singular value functions.

Then another coordinate transformation $\bar{z} = \eta(z)$ is necessary to arrive at a balanced representation of the nonlinear system. This transformation is specified by the singular value functions in a special way. For further details we refer to [35].

The advantages of this approach are its stability and passivity preserving properties (cf. [21]). However, it is still not clear how this approach can be applied by means of numerical computations. The small examples that are available are treated analytically as the crucial points are the nonlinear basis transformations and the calculation of the singular value functions. From a mathematical point of view, nonlinear balanced truncation is interesting. However, there is no way at the moment to apply it to problems from circuit simulation. Furthermore, as with POD, one can not expect reduction in complexity as the full nonlinear function still has to be evaluated to extract the information for the reduced one from it.

Empirical Balanced Truncation. Empirical Balanced Truncation, like POD, collects snapshots to construct empirical Gramians. These gramians are used in a truncated balanced reduction method to construct projectors. The reduced order model is computed using these projectors in a similar way as in POD, and hence the disadvantage of EBT is that in terms of evaluation costs there is no reduction.

Volterra series. Higher-order polynomial expansion schemes (or Volterra series) aim at more accurate models by including also higher-order terms in the Taylor expansion. The main drawback here is that the memory and computational costs of these models grow exponentially with the number of nonlinear terms included. Consequently, this method is not applicable for large or even moderately sized nonlinear systems.

Inertial Manifolds and Nonlinear Galerkin Projections Motivated by the study of the long-time dynamics or asymptotic behavior of dynamical systems $\frac{d}{dt}\mathbf{x} = \mathbf{g}(\mathbf{x})$ inertial manifolds can be utilized for model order reduction, leading to nonlinear Galerking projections. We briefly describe the main ideas. Details about intertial manifolds can be found in [9] and their application in model order reduction are described in e.g., [5, 26].

We consider a dynamical system of the form

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{f}(\mathbf{x}) \tag{7}$$

with symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, as it might arise from linearization of $\frac{d}{dt}\mathbf{x} = \mathbf{g}(\mathbf{x})$ at $\mathbf{x} = \mathbf{0}$.

A set $\mathcal{M} \subset \mathbb{R}^n$ is called *inertial manifold* if it is invariant and attracts all solutions of (7) exponentially in time.

Inertial manifolds are closely connected to a separation of the state $\mathbf{x}$ in "fast" and "slow" modes. This identification in turn is based on a spectral decomposition of $\mathbf{A}$:

$$\mathbf{A}\,[\mathbf{Y}\ \mathbf{Z}] = [\mathbf{Y}\ \mathbf{Z}] \cdot \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}, \tag{8}$$

where $\mathbf{Y} \in \mathbb{R}^{n\times r}$ and $\mathbf{Z} \in \mathbb{R}^{n\times(n-r)}$ contain the eigenvectors corresponding to the r smallest eigenvalues $\lambda_1, \ldots, \lambda_r$ and the eigenvectors corresponding to the remaining eigenvalues $\lambda_{r+1}, \ldots, \lambda_n$, respectively. Accordingly we seperate the state

$$\mathbf{x} = \mathbf{Y}\mathbf{x_S} + \mathbf{Z}\mathbf{x_F}, \quad \text{with } \mathbf{Y}^T\mathbf{Z} = \mathbf{0}, \quad \mathbf{x_S} \in \mathbb{R}^r,\ \mathbf{x_F} \in \mathbb{R}^{n-r} \tag{9}$$

into a slow part $\mathbf{Y}\mathbf{x_S}$ and a fast part $\mathbf{Z}\mathbf{x_F}$. Projection of (7) onto the complementary spaces spanned by the columns of $\mathbf{Y}$ and $\mathbf{Z}$, respectively, leads to coupled dynamical systems

$$\frac{d}{dt}\mathbf{x_S} = \mathbf{Y}^T\mathbf{A}\mathbf{Y}\mathbf{x_S} + \mathbf{Y}^T\mathbf{f}(\mathbf{Y}\mathbf{x_S} + \mathbf{Z}\mathbf{x_F}), \tag{10a}$$

$$\frac{d}{dt}\mathbf{x_F} = \mathbf{Z}^T\mathbf{A}\mathbf{Z}\mathbf{x_F} + \mathbf{Z}^T\mathbf{f}(\mathbf{Y}\mathbf{x_S} + \mathbf{Z}\mathbf{x_F}). \tag{10b}$$

In the standard (linear) Galerkin method, we would neglect the fast part by setting $\mathbf{x_F} \equiv \mathbf{0}$ in (10a). In this way the problem (7) of dimension n is reduced to a problem

$$\frac{d}{dt}\mathbf{x_S} = \mathbf{Y}^T\mathbf{A}\mathbf{Y}\mathbf{x_S} + \mathbf{Y}^T\mathbf{f}(\mathbf{Y}\mathbf{x_S}) \tag{11}$$

of dimension $r < n$ from whose solution the full space solution is reconstructed as $\mathbf{x} \approx \mathbf{Y}\mathbf{x_S}$.

Nonlinear Galerkin projection does not neglect the fast part but its dynamic in comparison to the slowly changing parts, i.e., we set $\frac{d}{dt}\mathbf{x_F} \equiv \mathbf{0}$ and get

$$\frac{d}{dt}\mathbf{x_S} = \mathbf{Y}^T\mathbf{A}\mathbf{Y}\mathbf{x_S} + \mathbf{Y}^T\mathbf{f}(\mathbf{Y}\mathbf{x_S} + \mathbf{Z}\mathbf{x_F}), \tag{12a}$$

$$\mathbf{0} = \mathbf{Z}^T\mathbf{A}\mathbf{Z}\mathbf{x_F} + \mathbf{Z}^T\mathbf{f}(\mathbf{Y}\mathbf{x_S} + \mathbf{Z}\mathbf{x_F}). \tag{12b}$$

By this a reduction is achieved in the sense that the full n-dimensional problem is replaced by a dynamical system (12a) of dimension r on the inertial manifold, defined by $\mathbf{x_F} = \mathbf{\Phi}(\mathbf{x_S})$ where $(\mathbf{x_S}, \mathbf{x_F}) = (\mathbf{x_S}, \mathbf{\Phi}(\mathbf{x_S}))$ solves the algebraic equation 12b. As the full space solution can be reconstructed by $\mathbf{x} \approx Y\mathbf{x_S} + \mathbf{Z}\mathbf{\Phi}(\mathbf{x_s})$, more accurate results than the classical Galerkin approach yields are expected. In [5] it is pointed out that linear Galerkin projection applied to nonlinear problems can change basic properties like stability of the system.

Usually the manifold is not known exactly. Therefore, one searches for an approximate inertial manifold (AIM), i.e.,

$$\mathbf{x_F} \approx \tilde{\mathbf{x_F}} = \mathbf{\Phi}_{\text{app}}(\mathbf{x_S}), \tag{13}$$

which can be computed e.g., by discretising (10b) using implicit Euler, followed by a simple fixed point iteration to solve the nonlinear equation. For details we refer to [33].

Note that the AIM approach turns transforms the ODE (7) into a differential-algebraic problem (12).

In [23] and [26] we find applications of this approach to control problems and in analysis of wind power plants, respectively. To our knowledge it has not been applied yet to problems in circuit simulations.

4 Current status of nonlinear model order reduction methods

In this section we describe the status of POD and TPWL. We address the points in this section and section 5 by numerical experiments in section 7 as well.

4.1 Proper Orthogonal Decomposition (POD)

Although POD has been successfully used in many applications, in particular in computational fluid dynamics [36, 6, 18, 24], it has some drawbacks that could make it less applicable in circuit simulation [40, 39].

First, as also discussed in Section 3.1 and [31, Section 2.3], POD provides a reduction in the number of states, but may even increase the evaluation and simulation costs due to the way the reduced order model is constructed (via projection). A possible way to decrease these evaluation costs is Missing Point Estimation [2, 4, 3, 40].

Second, the choice of the snapshots, and the number of snapshots, is a challenging task and may be hard to automate in a robust way [29]. A related problem is that if the number of snapshots becomes too large, the computation of the SVD can become infeasible (although iterative eigenvalue methods can be of help here).

Third, the influence of the training input may be very big, making the reduced order models unsuitable for reuse in practice.

We discuss these points in more detail in Sec. 7. There we will present some results computed with the Missing Point Analysis/Adapted POD approach described in [40]. For details on the foundation of this approach we refer to the cited paper. However, we reflect the basic idea with the case of a simple ODE

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}), \tag{14}$$

of dimension n with nonlinear right hand side $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$. Like described in Sec. 3.1 a singular value decomposition $X = U\Sigma V^T$ of a matrix $X \in \mathbb{R}^{n\times N}$ of $N \geq n$ snapshots is computed, giving n singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. The orthogonal matrix $\mathbf{L} = \mathbf{U} \cdot \mathrm{diag}(\sigma_1, \ldots, \sigma_n) \in \mathbb{R}^{n\times n}$ is introduced, with its columns $\mathbf{l_1}, \ldots, \mathbf{l_n}$ spanning the complete space $\mathbb{R}^n$. Hence, one can change to the new basis, i.e., $\mathbf{x} = \mathbf{Ly}$ and apply a Galerkin-like projection to the system:

$$\mathbf{L}^T \frac{d}{dt}(\mathbf{Ly}) = \mathbf{L}^T\mathbf{f}(\mathbf{Ly}). \tag{15}$$

Strictly speaking we do not apply Galerkin projection as the columns of L are orthogonal, but not orthonormal.

Classical POD reduction acts on $\mathbf{x} = L\mathbf{y}$ in the sense that the expansion of $\mathbf{x}$ in the basis $\mathbf{l}_1, \ldots, \mathbf{l}_n$ where $(\mathbf{l}_1, \ldots, \mathbf{l}_n) = \mathbf{L} = (\sigma_1{\cdot}\mathbf{v}_1, \ldots, \sigma_n{\cdot}\mathbf{v}_n)$ with $(\mathbf{v}_1, \ldots, \mathbf{v}_n) =$

$\mathbf{U}$ is truncated with respect to the magnitude of the singular values $\sigma_1, \dots, \sigma_n$:

$$\begin{aligned}
\mathbf{x} = \mathbf{L}\mathbf{y} &= (\sigma_1 \mathbf{v}_1) \cdot y_1 + \cdots + (\sigma_r \mathbf{v}_r) \cdot y_r + (\sigma_{r+1} \mathbf{v}_{r+1}) \cdot y_{r+1} + \cdots + (\sigma_n \mathbf{v}_n) \cdot y_n \\
&\approx (\sigma_1 \mathbf{v}_1) \cdot y_1 + \cdots + (\sigma_r \mathbf{v}_r) \cdot y_r + 0 \cdot y_{r+1} + 0 \cdot y_n \\
&= (\mathbf{l}_1, \dots, \mathbf{l}_r, 0, \dots, 0) \cdot \mathbf{y} \\
&= (\mathbf{L}\mathbf{P_r}^T \mathbf{P}_r) \cdot \mathbf{y}, \quad \text{with } \mathbf{P_r} = \begin{pmatrix} \mathbf{I_{r\times r}} & \mathbf{0_{r\times(n-r)}} \end{pmatrix} \in \{0,1\}^{r\times n} \\
&= (\mathbf{L}\mathbf{P_r}^T) \cdot (\mathbf{P_r}\mathbf{y}) = (\mathbf{L}\mathbf{P_r}^T) \cdot \mathbf{z_r} \quad \text{with } \mathbf{z_r} = (y_1, \dots, y_r)^T \in \mathbb{R}^r
\end{aligned}$$

where r usually is chosen in such a way that $\sigma_{r+1} < \text{tol}$ or $\sigma_{r+1} \ll \sigma_r$.

This procedure can also be interpreted as keeping the r most "dominant" columns of L and zeroising the rest, where a column's norm is taken as a criterion. That means, $\mathbf{L}$ is approximated by

$$\mathbf{L} \approx \mathbf{L}\mathbf{P_r}^T \mathbf{P_r}, \quad \text{with } \mathbf{P_r} \in \{0,1\}^{r\times n}. \tag{16}$$

where $\mathbf{P_r} = \begin{pmatrix} \mathbf{I_{r\times r}} & \mathbf{0_{r\times(n-r)}} \end{pmatrix}$ selects these columns. By construction of $\mathbf{L} = \mathbf{U} \cdot \text{diag}(\sigma_1, \dots, \sigma_n)$, where $\mathbf{U}^T\mathbf{U} = \mathbf{I_{n\times n}}$, we have $\|\mathbf{v}_i\|_2 = \sigma_i$ for $i = 1, \dots, n$. In this respect the r most dominant columns are therefore $\mathbf{l}_1, \dots, \mathbf{l}_r$.

In the adapted POD presented in [40] this perception is carried over to the transposed $\mathbf{L}^T$. That means, one selects, again based on the norms, the $g \in \mathbb{N}$ most dominant columns $\{\tilde{\mathbf{l}}_{\mu_1}, \dots, \tilde{\mathbf{l}}_{\mu_g}\}$ of $\mathbf{L}^T = (\tilde{\mathbf{l}}_1, \dots, \tilde{\mathbf{l}}_n)$ and zeroizes the rest:

$$\mathbf{L}^T \approx \mathbf{L}^T \mathbf{P_g}^T \mathbf{P_g}, \quad \text{with } \mathbf{P_g} \in \{0,1\}^{g\times n}. \tag{17}$$

First, these approximations to $\mathbf{L}$ and $\mathbf{L}^T$ from (16) and (17), respectively, are inserted into (15):

$$\mathbf{L}^T \mathbf{P_g}^T \mathbf{P_g} \frac{d}{dt}(\mathbf{L}\mathbf{P_r}^T \mathbf{P_r}\mathbf{y}) = \mathbf{L}^T \mathbf{P_g}^T \mathbf{P_g} \mathbf{f}(\mathbf{L}\mathbf{P_r}^T \mathbf{P_r}\mathbf{y}) \tag{18}$$

From (16) and (17) it also follows that

$$\mathbf{L}^T \approx \mathbf{P_r}^T \mathbf{P_r} \mathbf{L}^T \mathbf{P_g}^T \mathbf{P_g}, \tag{19}$$

and multiplying (18) with $\mathbf{P_r}$ (consider $\mathbf{P_r}\mathbf{P_r}^T = \mathbf{I_{r\times r}}$), the system (18) turns into

$$\mathbf{P_r}\mathbf{L}^T \mathbf{P_g}^T \mathbf{P_g} \frac{d}{dt}(\mathbf{L}\mathbf{P_r}^T \mathbf{P_r}\mathbf{y}) = \mathbf{P_r}\mathbf{L}^T \mathbf{P_g}^T \mathbf{P_g} \mathbf{f}(\mathbf{L}\mathbf{P_r}^T \mathbf{P_r}\mathbf{y}) \tag{20}$$

As $\mathbf{L}\mathbf{P_r}^T = (\sigma_1 \mathbf{v}_1, \dots, \sigma_r \mathbf{v}_r) = \mathbf{U_r}\Sigma_r$ (for $\mathbf{U_r} = (\mathbf{v}_1, \dots, \mathbf{v}_r), \Sigma_r = \text{diag}(\sigma_1, \dots, \sigma_r)$) we get

$$\Sigma_r \mathbf{U_r}^T \mathbf{P_g}^T \frac{d}{dt}[\mathbf{P_g}\mathbf{U_r}\Sigma_r \mathbf{P_r}\mathbf{y}] = \Sigma_r \mathbf{U_r}^T \mathbf{P_g}^T \mathbf{P_g} \mathbf{f}(\mathbf{U_r}\Sigma_r \mathbf{P_r}\mathbf{y}), \quad \mathbf{L}\mathbf{y} = \mathbf{x}.$$

The above equation states a system of dimension r for $\mathbf{y} \in \mathbb{R}^n$. Therefore, we introduce the reduced state vector $\mathbf{y}_r \in \mathbb{R}^r$:

$$\mathbf{y}_r = \Sigma_r \mathbf{P_r y},$$

from which we can approximately reconstruct the coefficients of the full state in the basis spanned by the columns of $\mathbf{L}$ by:

$$\mathbf{y} \approx \mathbf{P_r}^T \Sigma_r^{-1} \mathbf{y}_r,$$

This in turn lets us approximate the full state in the original basis:

$$\mathbf{x} \approx \mathbf{U_r y}_r,$$

because $\mathbf{x} = \mathbf{Ly} \approx \mathbf{LP_r}^T \Sigma_r^{-1} \mathbf{y}_r = \mathbf{U_r} \Sigma_r \Sigma_r^{-1} \mathbf{y}_r$. This part is consistent with the classical POD.

In addition to the reduction in the state space the adapted POD downsizes $\mathbf{f}(\cdot)$ by observing that the term $\mathbf{P_g f}(\cdot)$ corresponds to just the inclusion of g components $f_{\mu_1}(\cdot), \ldots, f_{\mu_g}(\cdot)$ of $\mathbf{f}(\cdot) = (f_1(\cdot), \ldots, f_r(\cdot))^T$. Hence, it suffices to evaluate the g-dimensional function

$$\bar{\mathbf{f}} : \mathbb{R}^n \to \mathbb{R}^g : \mathbf{x} \mapsto (f_{\mu_1}(\mathbf{x}), \ldots, f_{\mu_g}(\mathbf{x}))^T.$$

After scaling with Σ_r^{-1} the reduced system for the reduced state vector $\mathbf{y}_r \in \mathbb{R}^r$ becomes

$$\mathbf{U_r}^T \mathbf{P_g}^T \frac{d}{dt}[\mathbf{P_g U_r y}_r] = \mathbf{U_r}^T \mathbf{P_g}^T \bar{\mathbf{f}}(\mathbf{U_r y}_r), \quad \mathbf{x} = \mathbf{U_r y}_r \tag{21}$$

For the general case of having not an ODE (14) but a DAE

$$\frac{d}{dt}\mathbf{g}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) + \mathbf{Bv}$$

to deal with, one gets a reduced problem

$$\mathbf{U_r}^T \mathbf{P_g}^T \frac{d}{dt} \bar{\mathbf{g}}(\mathbf{U_r y_r}) = \mathbf{U_r}^T \mathbf{P_g}^T \bar{\mathbf{f}}(\mathbf{U_r y_r}) + \mathbf{U_r}^T \mathbf{Bv}. \tag{22}$$

with $\bar{\mathbf{g}} : \mathbb{R}^n \to \mathbb{R}^g : \mathbf{x} \mapsto (g_{\mu_1}(\mathbf{x}), \ldots, g_{\mu_g}(\mathbf{x}))^T$.

Remarks. Although results from applying the adapted POD to a chain of diodes as presented in [40] (see also Sec. 7) are promising, we also see severe drawbacks that have to be analyzed in more detail.

1. The downsizing of the nonlinear function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ to $\bar{\mathbf{f}} : \mathbb{R}^n \to \mathbb{R}^g$ with $g \ll n$ is based on the columns of $\mathbf{L}^T$. We recall that the matrix $\mathbf{L}$ arises from scaling the orthonormal basis vectors $\mathbf{v}_1, \dots, \mathbf{v}_n$ with the corresponding singular values. As there is no reasoning for this scaling identifiable it is somehow arbitrary.

   The decision about the importance of the items of $\mathbf{f}$ is based solely on $\mathbf{L}^T$. It is questionable whether the magnitude of $\mathbf{f}$ can be neglected in this way. Furthermore, also the differential operator $\frac{d}{dt}$ is reduced in the same manner – which becomes even clearer when looking at the reduced DAE-system (22). Hence, the same questions hold for the downsized differential part of the system.

2. Combining the approximations to $\mathbf{L}$ and $\mathbf{L}^T$ from (16) and (17) in (19) corresponds to zeroising all but the g columns $\mu_1, \dots, \mu_g$ and all but the first r rows of $\mathbf{L}^T$. It has to be studied carefully, which impact this has for the structure and especially the DAE-index of the system.

3. The role of the leading matrix product $\mathbf{W}_{r,g} = \mathbf{U_r}^T \mathbf{P_g}^T \in \mathbb{R}^{r\times g}$ in both (21) and (22) has to be analyzed more precisely. It is not completely clear if we can guarantee that $\mathbf{W}_{r,g}$ has full rank in general.

In [40] and in Sec. 7.2 the adapted POD is tested with a diode chain, showing nice behaviour. However, it is questionable if one can make it standard practice. In the following we present two small examples. The first shows that in general we cannot deduce a statement on the columns of $\mathbf{L}^T$ from decreasing singular values. In the second the problem of scaling is treated.

Example 1. Assume that the singular value decomposition $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$ of a snapshot matrix $X \in \mathbb{R}^{5\times 5}$ yields:

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{3}} & 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ 0 & \frac{1}{\sqrt{3}} & 0 & 0 & -\frac{2}{\sqrt{6}} \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} 10^5 & & & & \\ & 10^5 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}.$$

Then $\mathbf{L}^T$ calculates to:

$$\mathbf{L}^T = \begin{pmatrix} \frac{10^5}{\sqrt{2}} & 0 & \frac{10^5}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{10^5}{\sqrt{3}} & 0 & \frac{10^5}{\sqrt{3}} & \frac{10^5}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 & \frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} \end{pmatrix}$$

The euclidian norm of the columns of $\mathbf{L}^T$ are approximately

$$\frac{10^5}{\sqrt{2}}, \quad \frac{10^5}{\sqrt{3}}, \quad \frac{10^5}{\sqrt{2}}, \quad \frac{10^5}{\sqrt{3}}, \quad \frac{10^5}{\sqrt{3}}.$$

In this example the singular values, i.e., the elements on the diagonal of Σ, decrease rapidly. We clearly see that this does not allow to make a statement on the columns of $\mathbf{L}^T$.

Obviously, it does not make sense here to approximate $\mathbf{L}^T$ by some other $\mathbf{L}^T\mathbf{P_g}^T\mathbf{P_g}$. However, if we determine $\mathbf{P_r}$ and $\mathbf{P_g}$ in a naive way by defining $r = g = 2$ and just keeping the corresponding most dominant columns of $\mathbf{L}$ and $\mathbf{L}^T$ this yields:

$$\mathbf{P_r} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{P_g} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix},$$

such that

$$\mathbf{U_r}^T\mathbf{P_g}^T = (\mathbf{P_g}U\mathbf{P_r}^T)^T = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 \end{pmatrix} \quad \text{which only has rank 1.}$$

Example 2. The reasoning for downsizing $\mathbf{f}$ is actually based on the Galerkin-like projection of the system (see (15)) where $\mathbf{L}^T\mathbf{L} = \Sigma^2 \neq \mathbf{I_{n\times n}}$ holds. There is no obvious reason why the special scaling should be used. This example shows what can happen, when we apply Petrov-Galerkin projection with $\mathbf{W} = \bar{\mathbf{L}} = U\Sigma^{-1}$ and $\mathbf{V} = \mathbf{L} = U\Sigma$ (cp. Sec. 3.1):

$$\bar{\mathbf{L}}^T \frac{d}{dt}\mathbf{L}y = \bar{\mathbf{L}}^T f(\mathbf{L}y).$$

With the same line of argument that led from the full nonlinear problem (14) to the reduced system (21) we can also downsize $\mathbf{f}$ to $\bar{P}_g\mathbf{f}$ where $\bar{P}_g \in \mathbb{R}^{g\times n}$ is determined such that $\bar{\mathbf{L}}^T \approx \bar{\mathbf{L}}^T\bar{P}_g^T\bar{P}_g$. In this way we get a leading matrix product $\mathbf{U_r}^T\bar{P}_g^T$, whose rank has to be determined.

Assuming that a singular value decomposition now gives

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ 0 & 0 & \frac{1}{\sqrt{3}} & 0 & -\frac{2}{\sqrt{6}} \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} 10^5 & & & & \\ & 10^5 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}.$$

And therefore

$$\bar{\mathbf{L}}^T = \begin{pmatrix} \frac{10^{-5}}{\sqrt{2}} & 0 & \frac{10^{-5}}{\sqrt{2}} & 0 & 0 \\ \frac{10^{-5}}{\sqrt{2}} & 0 & -\frac{10^{-5}}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 & \frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} \end{pmatrix} \approx \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 & \frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} \end{pmatrix}.$$

Hence, we might choose

$$\mathbf{P_r} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \bar{P}_g = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

such that

$$\mathbf{U_r}^T \bar{P}_g^T = (\bar{P}_g U \mathbf{P_r}^T)^T = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{which even has rank 0.}$$

Clearly both examples lead to problems of reduced rank. In [3] it is proposed to solve these kind of problems with the least squares technique.

4.2 Trajectory piecewise-linear techniques (TPWL)

Trajectory piecewise-linear techniques deal better with reducing evaluation costs of the reduced order model. Although construction of the model can be expensive (depending on the number of linearization points), evaluation of the reduced order models is usually (much) cheaper due to the fact that the models are linear and smaller.

The choice of linearization points is automatic and there exist alternatives (see Sec. 3.2 and [31, 43]). Nevertheless, the linearization points must be chosen with care, since missing an important dynamic can make the linearization (and reduced order model) much less accurate. Related to this is the determination of the weight during simulation for different inputs.

With respect to training inputs, TWPL has the advantage that linear models are valid for all inputs by default. However, for the nonlinear system, this holds only in the neighborhood of the linearization point: if the system due to a different input reaches a state that was not reached during the training phase (and not covered by one of the linearized systems), the linearized model becomes of practically no value. In [31, 6.1.2] it is reported that big changes in amplitude of the testing and training input can have a dramatic impact on the accuracy.

Another point that is not much considered in the literature is the influence of the observer function. Most of the focus is on the states, but it might be fruitful to take the observer function $\mathbf{y}(t) = \mathbf{C}^T\mathbf{x}$ into account as well, during both the determination of the linearization points and the weights.

5 Open issues in nonlinear model order methods

The most important open issues in nonlinear MOR methods that block application in practice are:

1. Determination of training inputs: how does the quality of the reduced model depend on the training input used during construction and for which inputs is the reduced model reusable? Can the range of inputs for which a model is valid be identified a priori? What types of inputs are relevant in industrial applications?

2. Identification and classification of nonlinear systems that are suitable for reduction by currently available reduction techniques. Is there a way to determine a priori if a system is suitable for reduction, or if other techniques such as multirate time integration are needed?

3. Properties of linearized systems: since the snapshots are usually not equilibrium points, except in the case of the DC state, the linearized systems have an additional forcing term. What can be said about the structure, index, stability and passivity of these systems?

4. Automatic construction of reusable reduced order models: practical (re)use requires automatic construction of reduced order models. Furthermore, the models must be available in such a way that they can be (re)used in other designs just like any other building block, without additional work. Special care has to be taken when choosing the correct time interval $[t_{\text{start}}, t_{\text{end}}]$ because only those nonlinear effects that occured during the training can be reconstructed by the model.

5.1 Issues related to the training input

The choice of training inputs might even lead to a paradox: on the one hand one wants to obtain a reduced order model with much less states and hence only a small part of the state space should be dominant in order to achieve that. On the other hand, one wants an accurate model for possibly many different inputs and consequently many states become dominant, preventing a big reduction. A way out of this would be to define classes of inputs and construct a reduced order model for every class. In practice this is feasible provided the number of different classes is limited for the application at hand.

One could also think of taking suitably selected (hierarchical) basis functions as training functions. In real life applications, however, the choice of the training inputs must be based on the properties of the underlying system and the expected input signals. Even then finding suitable training inputs, by hand or automatically, can be challenging.

5.2 Classification of nonlinear systems

Systems showing digital behavior, i.e., having parts that switch on and off during certain time intervals due to a clock signal passing through, may not be good candidates for reduction by nonlinear model reduction techniques. Especially if many parts switch at different time points, which will introduce many independent states, piecewise linearization would require many linearization points. Examples of such systems are inverter chains.

On the other hand, systems showing a more smooth nonlinear behavior, such as transmission lines, may be potential candidates for reduction. If the system states, for the input signals of interests, only span a small subspace of the complete state space, the number of linearization points will be limited and hence piecewise linearization, with or without additional reduction, can be effective.

5.3 Properties of the linearized systems

5.3.1 Stability and passivity

As also discussed in [31, Section 5.1.4], it is not possible to make statements about the stability of the individual linearized systems, let alone about convex combinations of these systems, without making strict assumptions on $\mathbf{f}$ and $\mathbf{g}$ of the original system. If the original nonlinear system has an exponentially stable equilibrium point $\mathbf{x}_0$, then $\mathbf{x}_0$ is also an exponentially stable equilibrium point of the linearized system around $\mathbf{x}_0$. Exponential stability can be hard to check in practice, and furthermore, it does not say anything about the stability of the linearizations around non-equilibrium points. Also, the (convex) combination of stable linearizations is not guaranteed to be stable. Another artifact of piecewise-linearization is the possible generation of additional equilibria that are not equilibria of the original system, which is clearly an undesired feature. Since stability is hard to guarantee, this by definition is also the case for passivity.

In [31, Chapter 5] some techniques for computing stabilizing weights are described. We refer to [8] for recent developments on stabilizing schemes for piecewise-linearized models. Considerations on passivity and stability inforcing algorithms can be found in [10].

5.3.2 Structure and index

Linearization around (arbitrary) states does not change the underlying physical structure and index of the problem, since both properties are defined for all states $\mathbf{x}(t)$ [17]. If the index is known beforehand, this observation may help in choosing the model order reduction scheme for the linearized systems.

There is one structural property of the system that does change due to linearization around non-equilibrium points: an additional nonzero forcing term is introduced, cf. the term $\mathbf{f}(\mathbf{x}_i)$ in (1). We can regard this forcing term as an additional constant input to the linearized system and hence we have to take this into account when simulating the piecewise-linearized system and when computing reduced order models.

5.4 Automatic construction of reusable models

In principle, piecewise-linearized models can be constructed in an automatic way, provided that values for parameters such as β are given (see section 8 for some comments on implementation in a circuit simulator). The numerical experiments in section 7, however, show that the TPWL procedure in its current form is not robust enough to allow automatic construction of accurate models. Especially the strategy to choose linearization points and the determination of the weights both need to be improved.

Concerning the reuse of linearized models there is, apart from determining proper training inputs (a prerequisite for producing a reusable model), an additional difficulty. Realization or synthesis of a single linear reduced order model as a circuit is possible and allows for natural reuse in the design phase and simulations, without making any changes to the simulation software. However, piecewise-linear models, and especially the weighting procedure associated with resimulating such models, cannot be (re)used without making changes to the simulation software or adding additional functionality to the netlist language (to define, for instance, piecewise-linear models). An option is to define a new circuit block for piecewise-linear models, and to adapt the circuit simulator to deal with such blocks, see also section 8.

6 Applications of nonlinear model order reduction

In this section we describe applications in industrial practice that may take advantage of nonlinear model order reduction techniques. As far as possible, we identify how much current nonlinear model order techniques are applicable and where improvements or new developments are needed. Numerical results for a selection of the examples will be discussed in section 7.

- Transmission lines: like linear RLC transmission lines are well known examples for linear MOR techniques, good candidates for nonlinear MOR techniques are transmission lines with nonlinear elements (diodes).
- Inverter chains: due to the repetitive structure of the inverter chains, they may be reducible. Their typical switching behavior, however, might complicate reduction and could be handled better by multirate time integration techniques (see section 7).
- Phase noise analysis of oscillators: simple oscillator models can be handled with techniques based on Perturbation Projection Vectors [12]. Application to full schematics of oscillator circuits may require reduction techniques. Starting points for study can be [31, section 6.2],[15] and [17].
- Complete functional blocks on chip: large components such as radios, GPS, bluetooth, are (re)used in different application. Availability of reduced order models could help in the design and verification phase. There is no present knowledge on the actual need for this.
- RF Building Blocks: design of RF systems requires optimization of several performance parameters. This optimization process may involve thousands of circuit simulations. Although the building blocks are usually relatively small circuits, there is need for reduced or behavioral models that map inputs (design parameters) to outputs (performance parameters) in a much faster way. It is unclear whether nonlinear MOR techniques are of use here. Probably, response surface modeling techniques are of more use.

7 Numerical experiments

In this section we present results of computations done with matlab implementations. Concerning TPWL we have two matlab-codes: one uses the strategy presented by Rewieński [31], the second one the strategy used by Voß [43] to select the linearizations points.

As mentioned in section 3.2 the TPWL method is based on linearizations around properly chosen states the system reaches when a chosen time varying signal is applied at the input ports. As we are interested especially in the applicability of model order reduction techniques in an industrial environment, we concentrate on the observations presented in the following on the robustness of the models w.r.t. varying input signals and the degree of automatism, offered by the corresponding scheme.

7.1 Testcases

Testcases for the considerations are the nonlinear transmission line model from Fig. 1 with $N = 100$ nodes, a problem of dimension $n = 100$, the chain of $N = 300$ inverters from Fig. 2, a problem of dimension $n = 302$ and the diode chain given in Fig. 3, a problem of dimension $n = 301$.

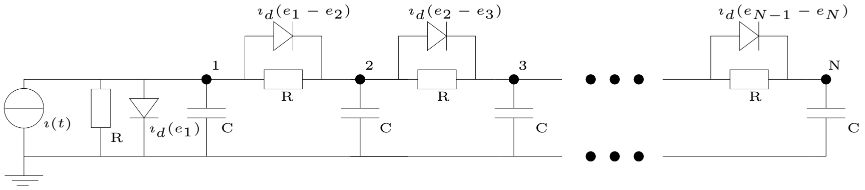

Figure 1: Nonlinear transmission line [31, 32].

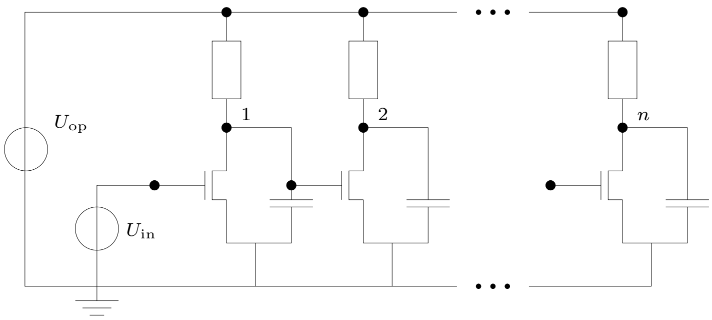

Figure 2: Inverter chain [43].

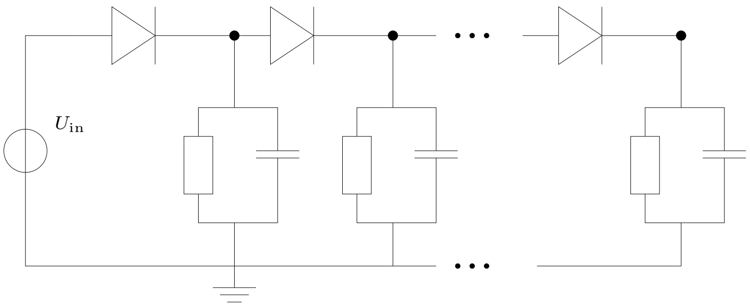

Figure 3: Diodechain [40].

We present observations we made when trying to reproduce the results presented in [31, 32] with the nonlinear transmission line and when changing the input signals with the inverter chain model.

The nonlinear transmission line (Fig. 1). The diodes introduce the designated nonlinearity to the circuit, as the current $\imath_d$ traversing a diode is modeled by $\imath_d(v) = \exp(40 \cdot v) - 1$ where v is the voltage drop between the diode's terminals. The resistors and capacitors contained in the model have unit resistance and capacitance ($R = C = 1$), respectively. The current source between node 1 and ground marks the input to the system $u(t) = \imath(t)$ and the output of the system is chosen to be the voltage at node 1: $y(t) = v_1(t)$.

Introducing the state vector $\mathbf{x} = (v_1, \ldots, v_N)$, where for $i = 1, \ldots, N$ v_i describes the node voltage at node i, modified nodal analysis leads to the network equations:

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= f(\mathbf{x}) + \mathbf{B} \cdot u, \\ y &= \mathbf{C}^T \cdot \mathbf{x}, \end{aligned}$$

where $\mathbf{B} = \mathbf{C} = (1, 0, \ldots, 0)^T$ and $f : \mathbb{R}^N \to \mathbb{R}^N$ with

$$\begin{aligned} f(\mathbf{x}) = &\begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{pmatrix} \cdot \mathbf{x} + \\ &+ \begin{pmatrix} 2 - \exp(40x_1) - \exp(40(x_1 - x_2)) \\ \exp(40(x_1 - x_2)) - \exp(40(x_2 - x_3)) \\ \vdots \\ \exp(40(x_{N-2} - x_{N-1})) - \exp(40(x_{N-1} - x_N)) \\ \exp(40(x_{N-1} - x_N)) - 1 \end{pmatrix} \end{aligned}$$

The inverter chain (Fig. 2). The nonlinearity is introduced by the MOSFET-transistors. Basically, in a MOSFET transistor the current from drain to source is controlled by the gate-drain and gate-source voltage drops. Hence, the easiest way to model this element is to regard it as a voltage controlled current source and assume the leakage currents from gate to drain and gate to source to be zero:

$$\imath_{\text{ds}} = k \cdot f(u_g, u_d, u_s),$$

with $f(u_g, u_d, u_s) = \max(u_g - u_s - U_{\text{thres}}, 0)^2 - \max(u_g - u_d - U_{\text{thres}}, 0)^2$, where the threshold voltage $U_{\text{thres}} = 1$ and the constant $k = 2 \cdot 10^{-4}$ is chosen. For a proper definition of the corresponding network equations we refer to [43].

The diode chain (Fig. 3). The diode chain depicted in Fig. 3 is somewhat similar to the nonlinear transmission line given in Fig. 1. Here we do not have a resistive path in parallel to each diode. Therefore, we expect, that signals applied to the diode chain via the voltage source U_{in} die out much faster than in the case of the transmission line, i.e., we expect a much higher redundancy.

Clearly, the diodes are the nonlinear elements, described by the correlation of terminal voltages and current traversing:

$$\imath_d(v) = \imath_d(v_a - v_b) = g(v_a, v_b) = \begin{cases} I_s\left(e^{\frac{v_a - v_b}{V_T}} - 1\right) & \text{if } v_a - v_b > 0.5V, \\ 0 & \text{otherwise} \end{cases}$$

Here, the treshold voltage $V_t = 0.0256V$ and the static current $I_s = 10^{-14}$ are chosen for all the diode models. The resistors and capacitors have uniform size $R = 10k\Omega$ and $C = 1pF$.

7.2 Test runs

The nonlinear transmission line. Rewieński [31] uses this model to show the behaviour of the automatic model extraction and the robustness of the TPLW-model w.r.t. input signals that differ from the input signals. There, the simple approach for constructing the overall reduced subspace, i.e. just taking the reduction coming from the linear model that arises from linearization around the starting value is used. Additional linearization points are chosen according to the relative distance (3) existing ones. No specification for δ therein is given. The Arnoldi-process to match $l = 10$ moments is used for reduction. Furthermore, in the weighting procedure (5) $\beta = 25$ is chosen.

With these settings, five models are extracted automatically and the resimulation shows very good match even for differing input signals. With the same training- and simulation-input we try to reproduce these results.

For the training the shifted Heaviside step function

$$\imath(t) = H(t-3) = \begin{cases} 0, & \text{if } t < 3 \\ 1, & \text{if } t \geq 3 \end{cases}$$

is used. Figure 4 shows both the input signal (left) and the state response, i.e. the node voltages (right). It can be seen, that with increasing number, the reaction of the nodes becomes weaker.

For resimulation a different input signal $\imath(t) = 0.5 \cdot (1 + \cos(2\pi t/10))$ was used. Plots presented in the cited papers [31, 32] show almost perfect match with the reference solutions.

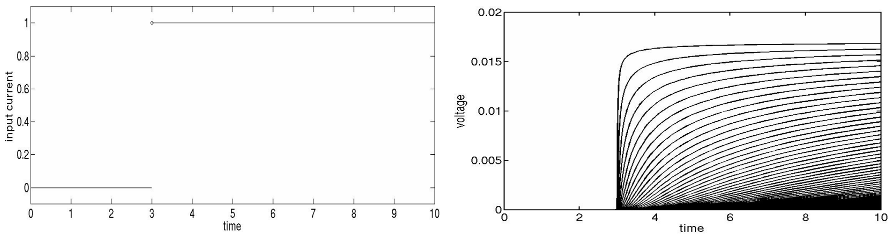

Figure 4: Transmission line: training input (left) and state response (right, all stages).

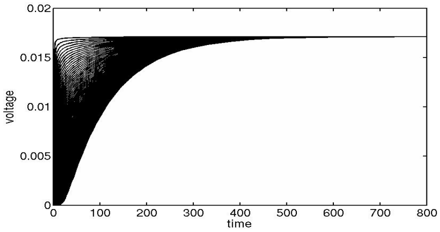

Figure 5: Transmission line: long term simulation with training signal.

We try to reproduce these plots but arrive at different results. In the following we give some details of the settings we used and the results we got.

- To get a guess for δ, which is used for the selection of the linearization points, we run a simulation of an order reduced linear model that corresponds to the linearization around $\mathbf{x}_0$ until $t_{\text{end}} = 300$ as at $t = 10$ steady state is not yet reached (cf. Fig. 5). As the starting value satisfies $\mathbf{x}_0 = 0$, d and δ where chosen to be (cf. (4))

$$d = \|\mathbf{x}_T\|_\infty \approx 0.0171 \quad \text{and} \quad \delta = 0.00171 \quad \text{with } T = 800.$$

- Both the simple and the extended approach for generating the reduced order basis were tested. For the latter attempt, the magnitude of the smallest singular value regarded meaningful was 1% of the magnitude of the largest one.

Figures 6 and 7 show the situation that arises when we apply the above mentioned settings, with the simple strategy for deriving the global reduced space, taking the distances in the full space. Except of a delay in the phase where the signal decreases, we see quite a good match in Fig. 6. But, automatically not 5 but 82 linear models are chosen. Fig. 7 shows the average activity of each of the extracted models in the sense that we multiply for each model the weight with the duration this weight is valid during integration, i.e, weight times timestep, build the sum of these products and divide it by the total length of the time interval. In Fig. 8 we see the impact of taking the distance in the weighting

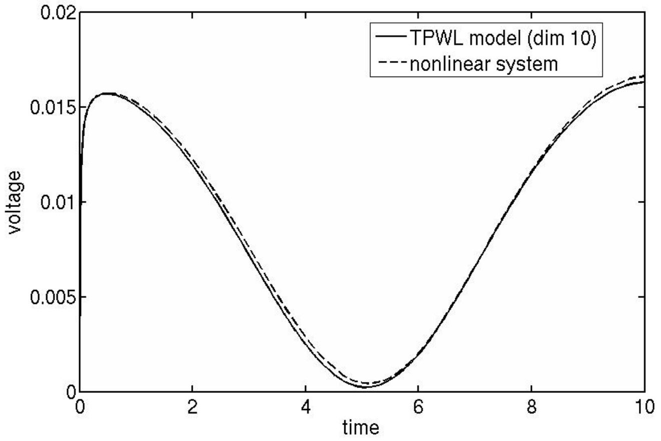

Figure 6: Transmission line: TPWL-resimulation, simple basis, $\delta = 0.00171$, 82 linear models, reduction to order 10, distance in full space.

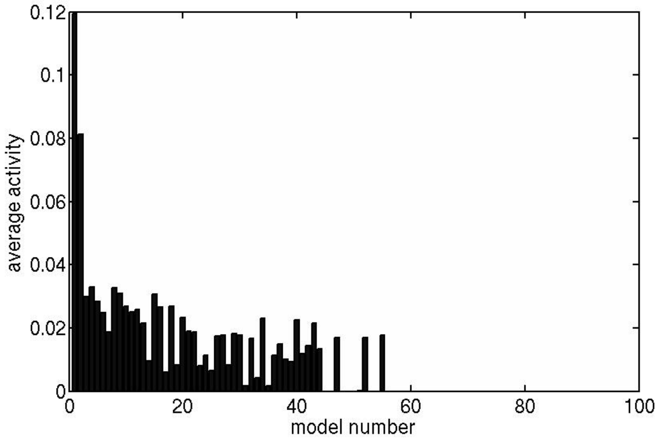

Figure 7: Transmission line: average activity of models, simple basis, $\delta = 0.00171$, 82 linear models, distance in full space.

process in the reduced space without taking care, if the linearization points are within the reduced space.

From Fig. 7 one can see, that some of the 82 linear submodels extracted with the Heaviside step function seem to be far more important for resimulating with the changed input signal than others. We chose five models and allowed only them to be involved in the resimulation. However, also this was not done automatically but by hand in several iterations. The models we used correspond to linearizations around the states the system was in at the timepoints

$$0 \quad 3.0021 \quad 3.0075 \quad 3.0219 \quad 3.1487$$

Indeed, we have a good match as can be seen from Fig. 9 from only five models. But, recall these were not chosen automatically.

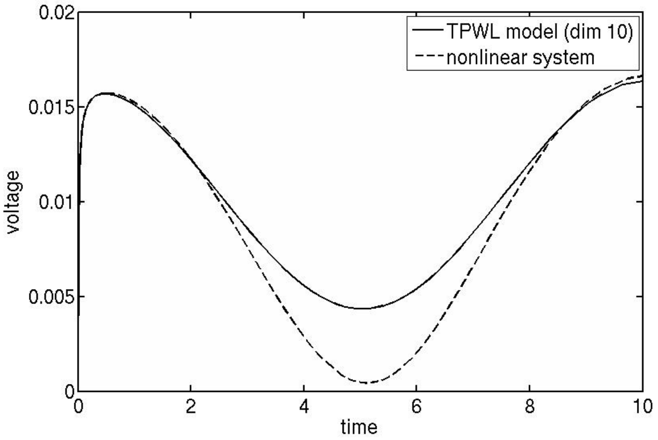

Figure 8: Transmission line: TPWL-resimulation, simple basis, $\delta = 0.00171$, 82 linear models, reduction to order $k = 10$, distance in reduced space.

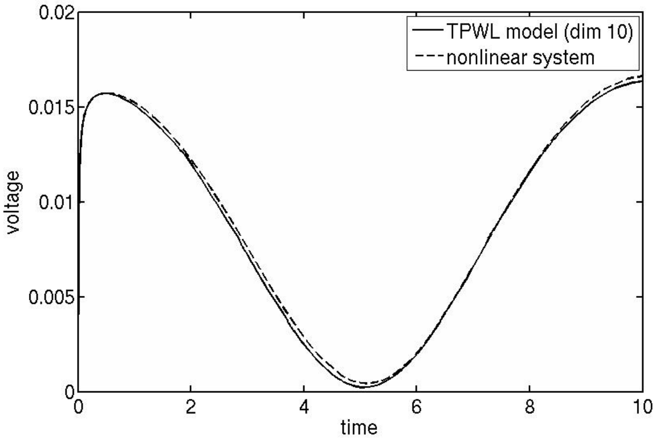

Figure 9: Transmission line: TPWL-resimulation, simple basis, $\delta = 0.00171$, 82 linear models, reduction to order $k = 10$, just 5 models allowed – chosen by hand.

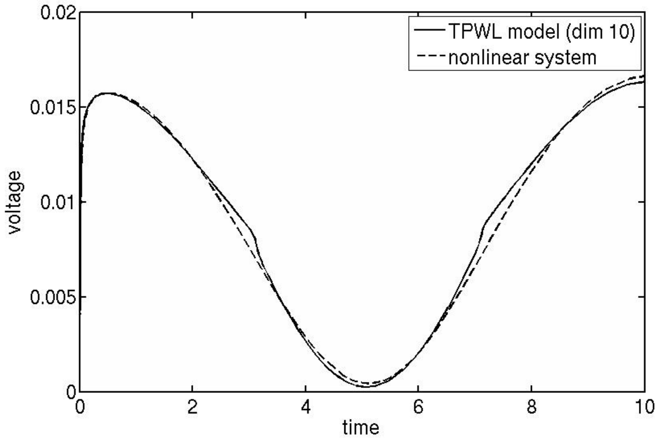

Figure 10: Transmission line: TPWL-resimulation, simple basis, $\delta = 0.01675$, 6 linear models, reduction to order $k = 10$, distance in full space.

In an automatic way just six nonlinear models where extracted only if we relaxed the tolerance in (3) to $\delta = 0.01675$. In this case however, the solution of the reduced system shows some kinks as we can see in Fig. 10. The timepoints at which the linearization points where chosen during simulation are

$$0 \quad 7.8691 \quad 8.3383 \quad 8.8075 \quad 9.2767 \quad 9.7459$$

We can see clearly that the model constructed in this way is not able to cope with the chosen new input signal. However, this is not a matter of the reduction but of the process choosing the linearization points. This observation can be made from having a look at Fig. 11, which shows the result from replacing the full nonlinear with a full linear model, i.e., using the TPWL-procedure without reducing the linear submodels.

The result of constructing the overall reduced space with the extended approach is shown in Fig. 12, with Fig. 13 showing the average activity of each model. Figure 14 shows the situation when allowing just a selection of models (in fact the same we refered to before) to be involved in the usage of the TPWL-model for simulation purposes.

Finally, Fig. 15 shows the result gained from extracting and resimulating with the implementation according to the TPWL procedure described in [43]. Here, 37 linear models where created, the system was reduced to dimension 19 and the resimulation again shows the edges we already seen before.

The inverter chain. The inverter chain constitutes a special class of circuit problems. Here a signal passes through the system, activating at each timeslot just a view elements and leaving the others untouched. However, as the signal

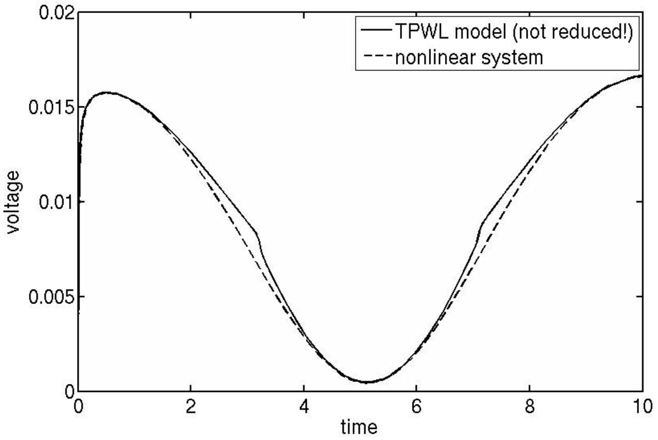

Figure 11: Transmission line: TPWL-resimulation, $\delta = 0.01675$, 6 linear models, no reduction, distance in full space.

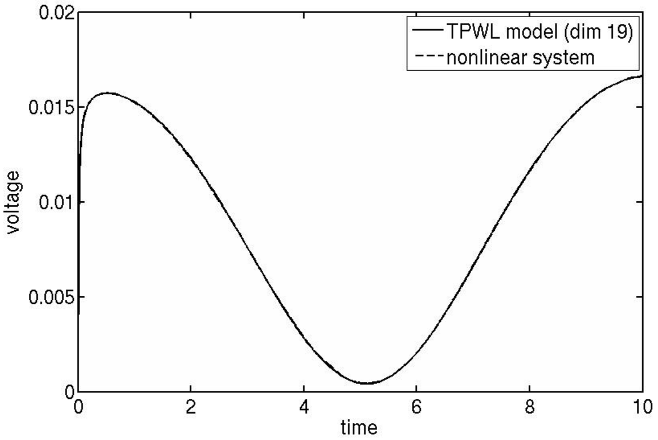

Figure 12: Transmission line: TPWL-resimulation, extended basis approach, $\delta = 0.00171$, 82 linear models, reduction to order $k = 19$, distance in full space.

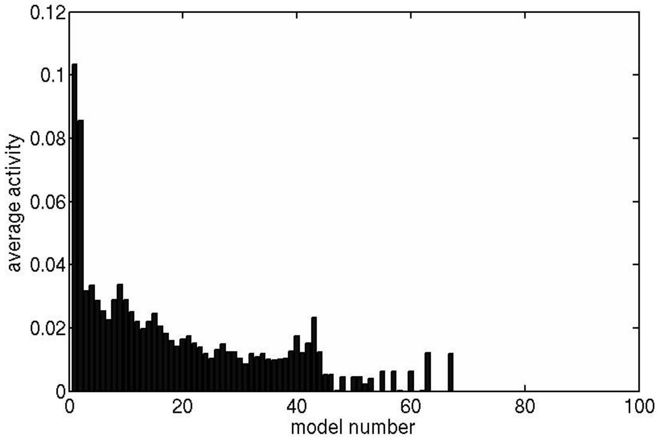

Figure 13: Transmission line: average activity of models, extended basis approach, $\delta = 0.00171$, 82 linear models, distance in full space full space.

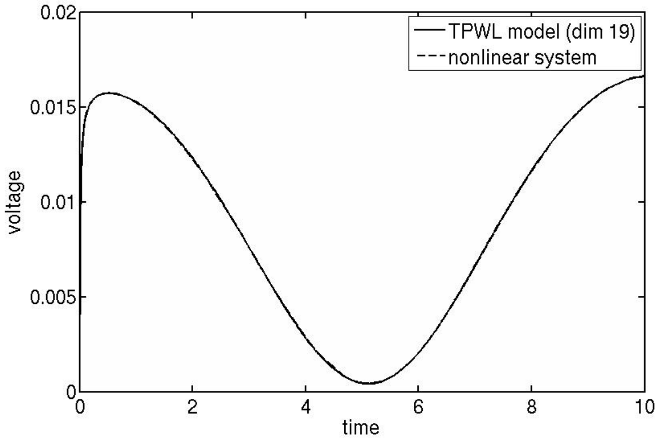

Figure 14: Transmission line: TPWL-resimulation, extended basis approach, $\delta = 0.00171$, 82 linear models, reduction to order $k = 19$, just 5 models allowed – chosen by hand, distance in full space.

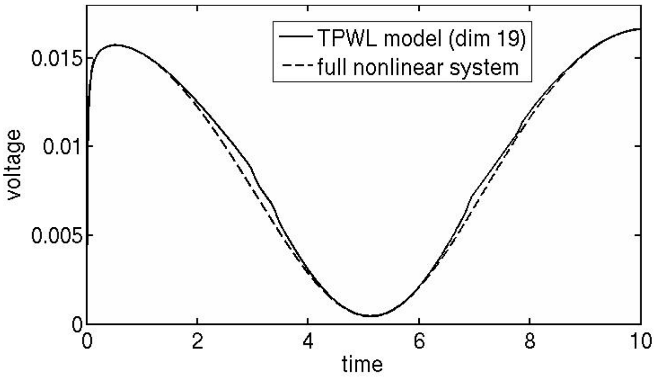

Figure 15: Transmission line: TPWL-resimulation, PMTBR, reduction to order $k = 19$, extended extraction of linear models and overall reduced subspace, distance in full space.

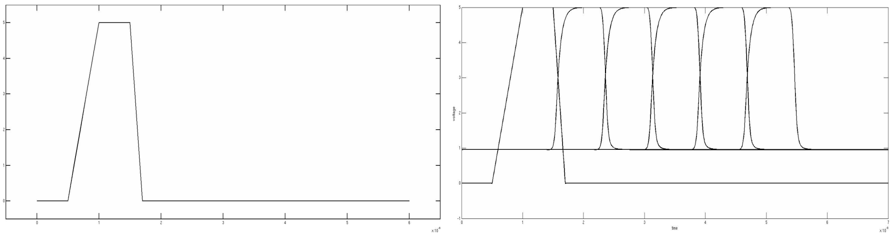

Figure 16: Inverter chain: training input (left) and state response (right, all stages).

passes through, each element is active at some time and sleeping at some others. As in [43], the training of the inverter chain during the TPWL model extraction was done with a single piecewise linear voltage at $U_{\text{in}} = u(t)$ (see also Fig. 16), defined by

$$u(0) = 0, u(5\text{ns}) = 0, u(10\text{ns}) = 5, u(15\text{ns}) = 5, u(17\text{ns}) = 0$$

In Figs. 17 and 18 we see again the danger of taking distances to linearization points not in the full space but in the reduced space. Both are showing the signal at inverter 24. In Fig. 18 the second impulse is just not recognized where this seems to be no problem in the first one. However, something else seems to be missing, even if we take the distance in the full space. In Figs. 19 and 20 the voltage at inverters 68 and 92 is given. In both cases, the signal cannot be recovered correctly. In the latter one it is even not recognized at all. At the moment we cannot state reasons for that. Obviously this is not caused by the reduction but by linearization or the weighting procedure as we get similar results when turning off the reduction step.

The impact of broadening the signal can be seen in Figs. 21, 22 and 23, which display the voltage at inverters 4, 18 and 68. In the first two figures the signals

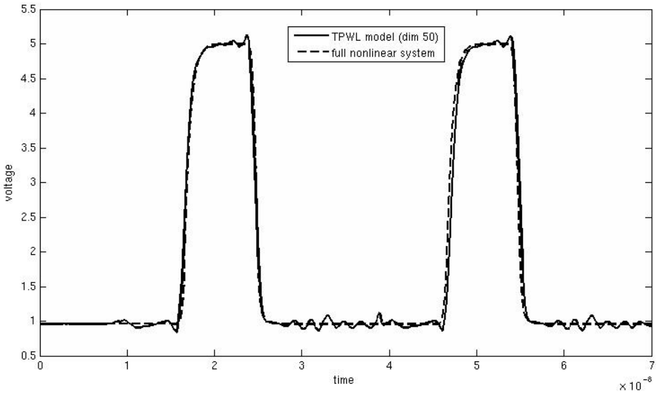

Figure 17: Inverter chain: TPWL-resimulation, reduction to order 50, repeated pulse, distance in full space, inverter 24.

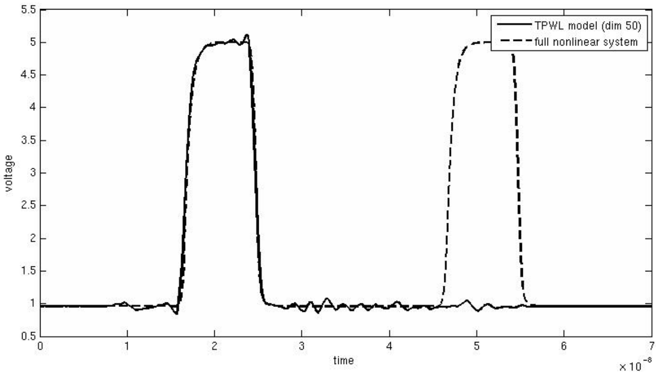

Figure 18: Inverter chain: TPWL-resimulation, reduction to order 50 repeated pulse, distance in reduced space, inverter 24.

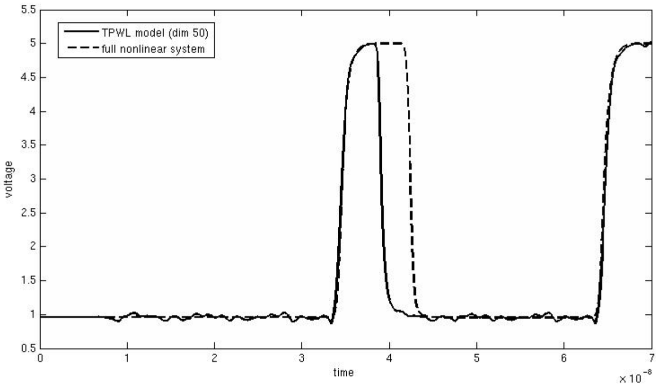

Figure 19: Inverter chain: TPWL-resimulation, reduction to order 50, repeated pulse, distance in full space, inverter 68.

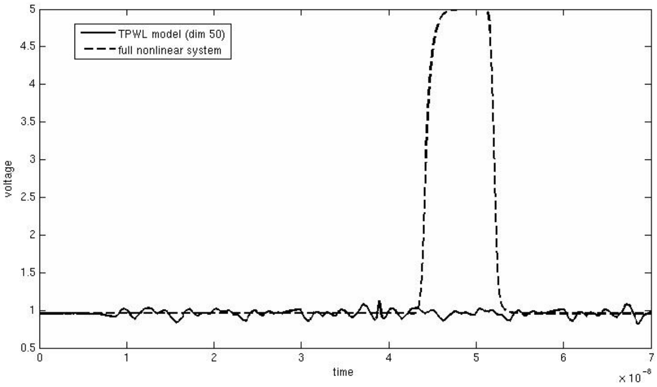

Figure 20: Inverter chain: TPWL-resimulation, reduction to order 50, repeated pulse, distance in full space,inverter 92.

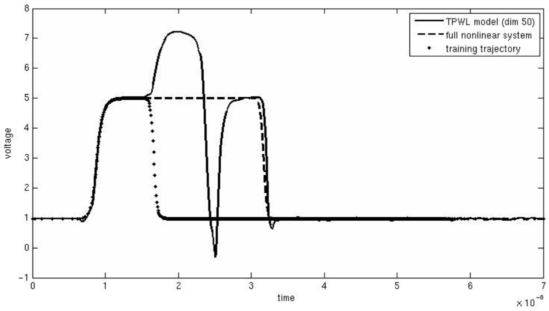

Figure 21: Inverter chain: TPWL-resimulation, reduction to order 50, wider pulse, distance in full space, inverter 4.

are far away from the expected behaviour. However, there seems to be a trend towards the situation that was encountered during the training. And indeed in Fig. 23, at inverter 68 we find a time shifted version of the training signal instead of the wide signal that has been applied.

Finally, in Figs. 24 and 25 the result of using the reduced model that arises from training with the single input of given width with a slightly tighter input signal is given for the inverters 6 and 68, respectively. In the former the characteristic is reflected quite well. However, in the latter the output signal seems to be just a time shifted version of the situation during the training.

For the time being also the ringing of the signal when the reduced model is simulated, as seen e.g., in Fig. 17, remains an open question. Having a closer look at how the inverter chain is modelled we see that the input voltage is applied at a floating node. This could give reasoning for the behavior encountered. However, also the backward and forward validity of the linear models (cp.Sec. 3.2) could be candidates.

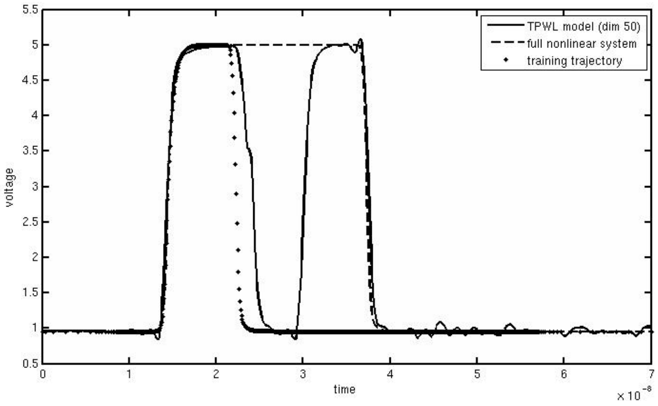

Figure 22: Inverter chain: TPWL-resimulation, reduction to order 50, wider pulse, distance in full space, inverter 18.

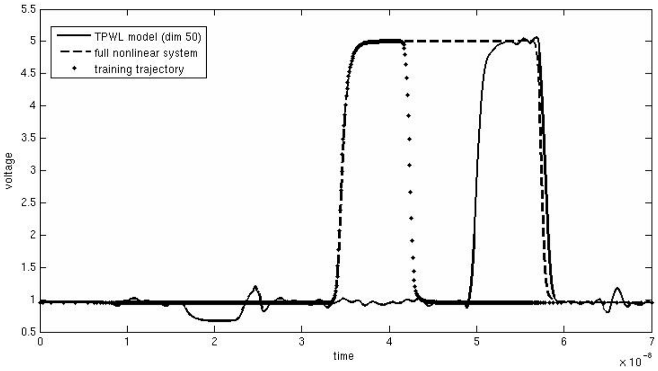

Figure 23: Inverter chain: TPWL-resimulation, reduction to order 50, wider pulse, distance in full space, inverter 68.

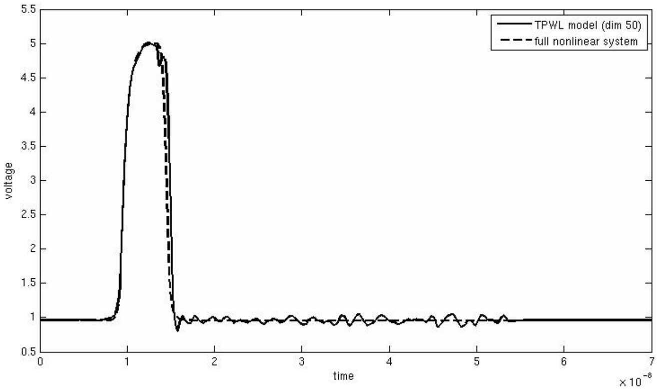

Figure 24: Inverter chain: TPWL-resimulation, reduction to order 50, tighter impulse, distance in full space, inverter 6.

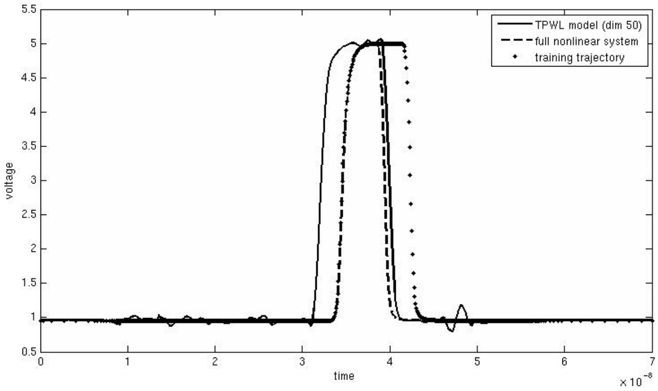

Figure 25: Inverter chain: TPWL-resimulation, reduction to order 50, tighter impulse, distance in full space, inverter 68.

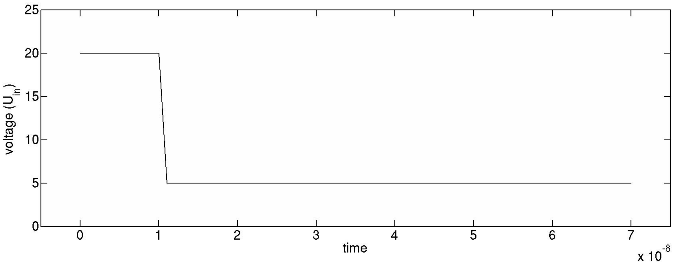

Figure 26: Diode chain: training input.

The diode chain. The diode chain is used as testcase in Verhoeven [40] to show the effectiveness of the adapted POD approach.

As training input the stepfunction $U_{\text{in}}(t)$, depicted in Fig. 26 that stays on $20\,V$ until $t = 10\,ns$ and then degrades linearly to $5\,V$ at $t = 11\,ns$.

In Fig. 27 the response at all nodes is given (left) and the singular values from the decomposition of the snapshot matrix. Like we expected, we see that the signal dies out very quickly and just the first 30 diodes operate. This reflects also in the singular values which drop very rapidly (see Fig. 27).

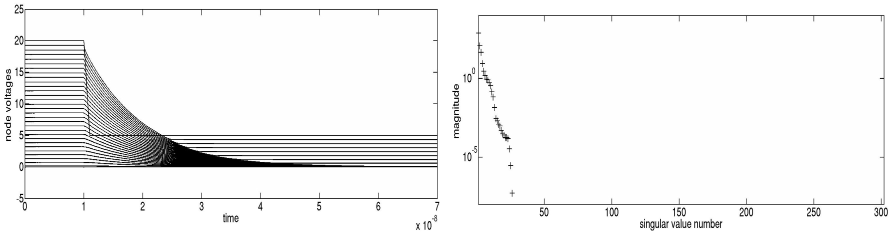

Figure 27: Diode chain: system's response and singular values.

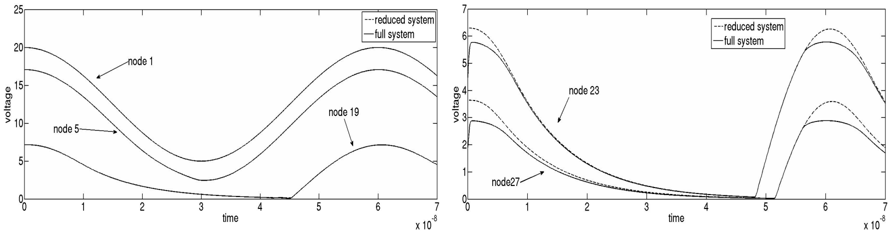

Figure 28: Diode chain: adapted POD-resimulation, reduction to dimesion 30 (g=35), differing input signals: $7.5 \cos \ldots$ (left), $9.5 \cos \ldots$ (right).

For extracting a reduced order model we start the algorithm with the parameters $r = 30$ and $g = 35$, i.e., the state space is reduced to dimension 30 and the nonlinear functions are downsized to dimension 35.

When the according reduced order model of the diode chain is used in a simulation with an input signal that is different from the one the model was trained with, but which still does not have larger signal, we get a very good match between behaviour of the full system and the reduced one. However, a larger amplitude of the input signal may produce inaccuracies. The two graphs in Fig. 28 show this behaviour. In both cases a reduced order model that arose from a training with the input signal U_{in} from Fig. 26 was used. A resimulation with the reduced substitude model was done for the input signals

$$7.5 \cos\left(\frac{2\pi t}{60 \cdot 10^{-9}}\right) + 12.5 \quad \text{and} \quad 9.5 \cos\left(\frac{2\pi t}{60 \cdot 10^{-9}}\right) + 12.5.$$

The maximum of the first signal does not exceed the amplitude of the training input signal. Hence, also in this case the signal dies out quite fast. In the second case the input signal has a higher amplitude and we expect that this plus in energy is enough to activate some more diodes.

One of the crucial points in applying this adapted POD approach is again the choice of clever training signals.

However, table 1 is very appealing. We clearly see, that the adapted POD is superior to the classical POD. Note that we are comparing cputimes of a matlab implementation where the training was done with the mentioned stepfunction.

		cputime [s]	
input	full	classical POD	adapted POD
like training	42.01	35.51	5.12
$7.5 \cos \ldots$	40.22	45.34	6.28

Table 1: Time consumption adapted POD and POD (matlab implementation).

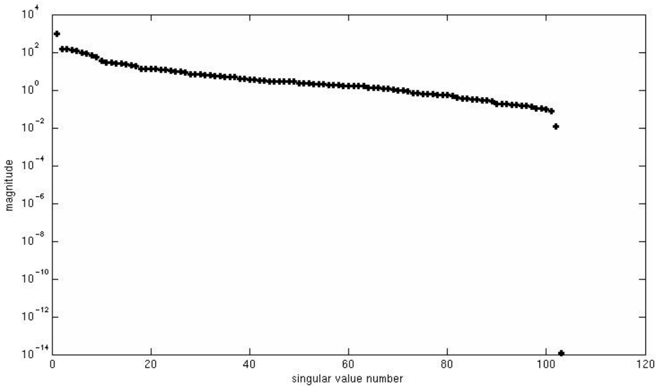

Figure 29: Inverter chain: singular values.

Summary

Resuming the different approaches and examples, we record that we have to do more tests with more realisitic problems. For the time being circuits like the inverter chain, whose characteristic it is to buffer and transport information, might not be a good example for where model order reduction can be applied. The diode chain that was used to put the adapted POD to the test seems also not be ideal because actually a large part of the system is not active at all during all the time and basically what is done is just cutting off the parts of the nonlinear functions that do not change at all.

Adapted POD and the inverter chain. We close this section with an inspection on how the adapted POD approach [40] is behaving with the inverter chain from Fig. 2. The singular values corresponding to the inverter chain are given in Fig. 29. One can clearly see that they do not decrease rapidly. However, this is one thing we expect as the signal is passing through all the stages with loosing just a little bit energy. Therefore, we also expect that the (adapted) POD approach does not yield very good results.

In a first trial we use the same reduction parameter $r = 30$ and $g = 35$ for the dimension of the reduced state space and the dimension of the nonlinear element functions, respectively. With this setup the reduced order model is not able to substitute the full nonlinear system as we see in Fig. 30. However, also with a higher dimension of both the reduced order model as well as the nonlinear function, choosing $r = 50$ and $g = 55$ we do not arrive at a corresponding model as can be seen in Fig. 31. In this case the special structure of the inverter chain causes the extraction of an unreasonable model. In Fig. 30 the input source U_{in} is connected to the gate of the very first transistor. As the input is given by a voltage, the current through this source is introduced as an additional unknown.

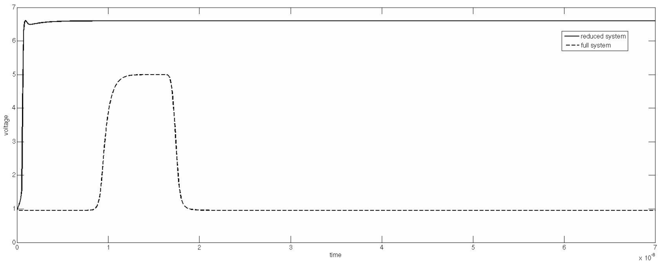

Figure 30: Inverter chain: adapted POD-resimulation, $r = 30, g = 35$, inverter 6.

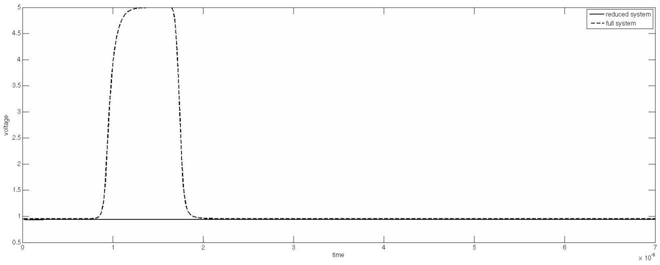

Figure 31: Inverter chain: adapted POD-resimulation, $r = 50, g = 55$, inverter 6.

Not taking into account the biasing source U_{op}, the matrix $\mathbf{B}$ in fact is the vector $(0, \ldots, 0, 1)^T \in \mathbb{R}^n$. The transistor models do not allow for a current through gate. Therefore the current through the voltage source is constantly zero. This leads to a projection matrix $\mathbf{V}$ where the last column is constant zero. Finally, projection yields $\mathbf{V}^T\mathbf{B} \equiv \mathbf{0}$. This means, that the reduced model will ignore the input $\mathbf{u}(t)$.

The application of POD to DAE systems is addressed in [13]. In [3] the application of least-squares techniques instead of Galerkin projection is proposed. The most promising technique to overcome the problem of eliminating input signals is to adapt structure preserving techniques like SPRIM [14] and IOPOR [45].

8 Implementation in a circuit simulator

In this section we give an outline of requirements and ideas for implementing nonlinear MOR techniques in a circuit simulator (e.g., Pstar) and introducing it in the design flow. We give some details for every phase of the process and focus mainly on trajectory piecewise-linearization techniques.

In most of the cases at least for a certain time interval a transient simulation of the original system is needed, given a training input. Hence, the transient implementation needs to be adapted to provide the data for construction of the linearized models.

Selection of linearization points. The procedure described in section 3.2 for determining the linearization points can be implemented in the transient algorithm without many changes. The states and Jacobians are usually available at every time point and if a linearization point is selected, the corresponding states and undecomposed Jacobians need to be stored.

Reduction of linear submodels. If an algorithm for reduction of linear systems is already available in the implementation, it can be used for reducing the linear submodels as well. If not available, algorithms for linear model order reduction can be implemented as a new functionality, and should be implemented in such a way that they can be used for other MOR purposes as well.

Simulation of piecewise-linear models. This part will probably have the most impact on the existing transient implementation. Although all information for the weighting procedure is available (current state and linearization points), accurate determination and combination of the linear models may influence big parts of the code. Also combinations with other time integration schemes like multirate may introduce additional difficulties. Special care must be taken if the piecewise-linear model is just a submodel of a bigger circuit. In a hierarchical simulator (Pstar), the linearized model is ideally treated as a submodel.

Storage and reuse of linearized models. As also discussed in section 5.4 it is not possible to synthesize the piecewise-linear model as a netlist, like in the realization of reduced order models of linear circuits. Although one can think of a piecewise-linear netlist (a netlist for every linear model), it is still not clear how the circuit simulator would deal with this. The most pragmatic option is probably to store the piecewise-linear model in binary format, i.e., just store the system matrices of the linear (and reduced) models and the linearization points. This new type of circuit block can then be regarded as a black-box submodel and reused in the design flow as any other circuit block. The simulator must be adapted to be able to deal with such blocks. This idea can also be used for reduced order models of linear circuits (synthesis is then no longer necessary).

Netlist grammar. The grammar of the netlist language needs to be adapted to enable nonlinear model order reduction. A change parameter, or maybe a new type of simulation, can be defined to enable construction of reduced order models (including parameters for the linearization and reduction process). To enable resimulation, the language must be extended with support for piecewise-linear circuit blocks.

9 Conclusions and further work

9.1 Conclusions

Examples of nonlinear systems arise in almost all applications in electrical engineering: the presence of a single transistor or diode already makes the circuit behavior nonlinear. More concrete examples are phase locked loops (PLLs), that usually contain voltage controlled oscillators (VCOs) and transmission lines with nonlinear elements such as diodes. Especially the latter can be good candidates for nonlinear model order reduction techniques: they usually have a limited number of inputs and outputs and show smooth behavior that is suitable for reduction. Other systems, such as inverter chains, show more digital behavior and are much more difficult for model reduction techniques, since the behavior cannot be described by a few dominant states. Here other techniques such as multirate may be better alternatives.

Like in the case of reduction of linear systems, it is for nonlinear systems not only the goal to reduce the number of states. The main objective is to construct a reduced model in the sense that it can be reused at much lower computational costs. For example, one would like to have a decrease in the simulation time for transient analysis.

Piecewise-linearization in combination with linear model order reduction techniques in principle offers both reduction in number of states and in simulation time. According to the experiments described in this report, however, trajectory piecewise-linearization (TPWL [32]) is not robust enough, most likely because of the procedure to determine the linearization points.

One clear result for TPWL is that it is (much) more accurate and robust to determine the weights using state vectors in the original state space, i.e., reduced state vectors must be projected back to the original full state space before determining the weights. Numerical experiments confirmed that this leads to more accurate results.

Another important issue is related to the reuse of piecewise-linear models. Since the piecewise-linear model is constructed using a single training input, this training input needs to chosen with care. Our experiments, however, indicate that even with an input close to the training input, simulation of the piecewise-linear models may become inaccurate. This also puts doubts on the robustness of current linearization schemes. For both TPWL and POD based approaches, already a change in amplitude of the input signal (compared to the training input) may result in an inaccurate resimulation.

We stress that from a theoretical viewpoint, piecewise-linearization clearly shows potential; in the next section we propose some points for further research.

Concerning implementation in a circuit simulator, the most impact will be in the transient algorithm. For the reduction of linear models, one can rely on existing linear model order reduction schemes, if already available.

Besides the TPWL method the adapted POD approach should be analyzed more carefully. The obvious advantage of this approach is that nonlinearity is conserved explicitly, leading to more accurate results, which can be seen in the less wiggling trajectories. However, we stated many serious questions concerning especially the region of trust of this method and the robustness of missing point estimation.

9.2 Further work

Topics for further work are:

- Test the piecewise-linear algorithms for other, industry relevant, examples, with specific training inputs. The current implementation seems not robust enough, but more examples are needed to get a definite insight and to find improvements for the linearization point and weight determination. Also include the output function in the linearization/reduction scheme.
- Decoupling of linear and nonlinear parts of large circuits: the effects of reducing linear and/or nonlinear parts of large systems separately need to be better understood.
- Reduction of parameterized nonlinear systems: in many real-life cases the original nonlinear system has a (small or large) number of parameters that can be changed in the design. For such a system, at least the presence of the dominant parameters in a reduced order model is required. Starting point for further research could be [7]. In [20] MOR techniques are used to improve efficiency of the adjoint transient sensitivity analysis.
- Simulation and reduction of very large scale circuits: if systems are so large that even simulation (which is required for construction of reduced order models) is no longer feasible within reasonable time, there is need for a different class of reduction methods. Combinations with multirate time integration are an option. One could also consider replacing parts of the system by reduced order models, or to use approximate simulation algorithms as suggested in [31, Section 3.3].
- For linear systems, structure preserving projection methods IOPOR [45] and SPRIM [14] have recently been developed to preserve especially the input-to-state and state-to-output maps (the B and C matrices). This might also be of interest for projection methods for nonlinear systems, since

also here one wants eventually to insert the reduced model back into the original system, by connecting at the terminals. In principle, the projection-splitting ideas of [45, 14] can be reused for POD and TPWL based approaches.

A more detailed issue related to the previous remark is whether structure preserving methods should also be used during the reduction of the linearized models that arise in TPWL.

Another question related to TPWL is how accurate the reduced models for the linearized systems should be. How does the number of matched moments/poles/Hankel singular values affect the accuracy of the reduced nonlinear system?

References

[1] Athanasios C. Antoulas. *Approximation of Large-Scale Dynamical Systems.* SIAM, 2005.

[2] P. Astrid. *Reduction of process simulation models: a proper orthogonal decomposition approach.* PhD thesis, Technische Universiteit Eindhoven, 2004.

[3] P. Astrid and A. Verhoeven. Application of least squares mpe technique in the reduced order modeling of electrical circuits. In *Proceedings of the 17th Int. Symp. MTNS*, pages 1980–1986, 2006.

[4] P. Astrid, S. Weiland, K. Willcox, and A. C. P. M. Backx. Missing point estimation in models described by proper orthogonal decomposition. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, volume 2, pages 1767–1772, 2004.

[5] Peter Benner. Model reduction using center and inertial manifolds. http://www-user.tu-chemnitz.de/∼benner/workshops/BBC-MOR/benner.pdf, 2006. Berlin-Braunschweig-Chemnitz Workshop: Recent Advances in Model Reduction.

[6] G. Berkooz, P. Holmes, and J. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25:539–575, 1993.

[7] B. N. Bond and L. Daniel. A piecewise-linear moment-matching approach to parameterized model-order reduction for highly nonlinear systems. *IEEE Trans. CAD Int. Circ. Syst.*, 26(12):2116–2129, December 2007.

[8] B. N. Bond and L. Daniel. Stabilizing schemes for piecewise-linear reduced order models via projection and weighting functions. In *IEEE/ACM International Conference on Computer-Aided Design, 2007*, pages 860–867, 2007.

[9] J. Carr. *Applications of Center Manifold Theory.* Springer-Verlag, 1981.

[10] Kwang Lim Choi. *Modelling and simulation of embedded passives using rational functions in multi-layered substrates.* PhD thesis, Georgia Institute of Technology, 1999.

[11] P. G. Ciarlet, W. H. A. Schilders, and E. J. W. ter Maten, editors. *Numerical Methods in Electromagnetics*, volume XIII of *Handbook of Numerical Analysis.* Elsevier, North Holland, 2005.

[12] A. Demir, A. Mehrotra, and J. Roychowdhury. Phase noise in oscillators: a unifying theory and numerical methodsfor characterization. *IEEE Trans. Circ. Syst. I*, 47(5):655–674, May 2000.

[13] Falk Ebert. A note on POD model reduction methods for DAEs. Preprint 06-343, TU Berlin, Inst. f. Mathematik, 2006.

[14] Roland W. Freund. SPRIM: Structure-Preserving Reduced-Order Interconnect Macromodeling. In *Technical Digest of the 2004 IEEE/ACM International Conference on Computer-Aided Design, Los Alamitos, California*, pages 80 – 87. IEEE Computer Society Press, 2004.

[15] E. Gad and M. Nakhla. Efficient model reduction of linear periodically time-varying systems via compressed transient system function. *IEEE Trans. Circ. Syst.–1*, 52(6):1188–1204, June 2005.

[16] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences).* The Johns Hopkins University Press, October 1996.

[17] Michael Günther, Uwe Feldmann, and E. Jan W. ter Maten. Modelling and discretization of circuit problems. In *[11]*, pages 523–650. Elsevier B.V., 2005.

[18] K. C. Hall, J. P. Thomas, and E. H. Dowell. Proper orthogonal decomposition technique for transonic unsteady aerodynamic flows. *AIAA Journal*, 38(10):1853–1862, 2000.

[19] M. Hinze and S. Volkwein. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control. In P. Benner, V. Mehrmann, and D. Sorensen, editors, *Dimension Reduction of Large-Scale Systems*, Lecture Notes in Computational and Applied Mathematics, pages 261–306. 2005.

[20] Zoran Ilievski, H. Xu, Arie Verhoeven, E.J.W. ter Maten, W.H.A. Schilders, and R.M.M. Mattheij. Adjoint Transient Sensitivity Analysis in Circuit Simulation. In G. Ciuprina and D. Ioan, editors, *Scientific Computing in Electrical Engineering – SCEE 2006*, volume 11 of *Mathematics in Industry*, pages 183 – 189. The European Consortium for Mathematics in Industry, Springer-Verlag Berlin Heidelberg, 2007.

[21] Tudor C. Ionescu and Jacquelien M. A. Scherpen. Positive Real Balancing for Nonlinear Systems. In G. Ciuprina and D. Ioan, editors, *Scientific Computing in Electrical Engineering – SCEE 2006*, volume 11 of *Mathematics in Industry*, pages 153 – 159. The European Consortium for Mathematics in Industry, Springer-Verlag Berlin Heidelberg, 2007.

[22] Roxana Ionutiu, Sanda Lefteriu, and Athanasios C. Antoulas. Comparison of Model Order Reduction with Applications to Circuit Simulation. In G. Ciuprina and D. Ioan, editors, *Scientific Computing in Electrical Engineering – SCEE 2006*, volume 11 of *Mathematics in Industry*, pages 3 –

24. The European Consortium for Mathematics in Industry, Springer-Verlag Berlin Heidelberg, 2007.

[23] Kazufumi Ito and Karl Kunisch. Reduced order control based on approximate inertial manifolds. *Linear Algebra Appl.*, 415:531 – 541, 2006.

[24] S. Lall, J. Marsden, and S. Glavaski. Empirical model reduction of controlled nonlinear systems. In *Proceedings of the IFAC World Congress*, volume F, pages 473–478. International Federation of Automatic Control, 1999.

[25] M. Loève. *Probability Theory.* Van Nostrand, 1955.

[26] Marcus Meyer. *Reduktionsmethoden zur Simulation des aeroelastischen Verhaltens von Windkraftanlagaen.* PhD thesis, TU Braunschweig, 2003.

[27] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17 – 32, 1981.

[28] J. R. Phillips. Projection-based approaches for model reduction of weakly nonlinear, time-varying systems. *IEEE Trans. CAD Int. Circ. Syst.*, 22(2):171–187, February 2003.

[29] R. Pinnau. Model reduction via proper orthogonal decomposition. In W. Schilders, H. van der Vorst, and J. Rommes, editors, *Model order reduction: theory, applications, and research aspects*, pages 95–109. Springer, 2008.

[30] M. Rathinam and L. R. Petzold. A new look at proper orthogonal decomposition. *SIAM J. Numer. Anal.*, 41(5):1893–1925, 2003.

[31] M. J. Rewieński. *A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems.* PhD thesis, Massachusetts Institute of Technology, 2003.

[32] M. J. Rewieński and J. White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Trans. CAD Int. Circ. Syst.*, 22(2):155–170, February 2003.

[33] R. D. Russel, D. M. Sloan, and M. R. Trummer. Sum numerical aspects of computing inertial manifolds. *SIAM Journal of Scientific Computing*, 14(1):19 – 43, 1993.

[34] V. Savcenco. *Multirate Numerical Integration for Ordinary Differential Equations.* PhD thesis, Universiteit van Amsterdam, 2008.

[35] Jacquelien M. A. Scherpen. *Balancing for nonlinear systems.* PhD thesis, University of Twente, 1994.

[36] L. Sirovich. Turbulence and the dynamics of coherent structures part i-iii. *Quarterly of App. Math.*, 45(3):561–571, 573–582, 583–590, 1987.

[37] Michael Striebel. *Hierarchical Mixed Multirating for Distributed Integration of DAE Network Equations in Chip Design.* Number 404 in Fortschritt-Berichte VDI Reihe 20. VDI, 2006.

[38] Dmitry Vasilyev, Michal Rewieński, and Jacob White. A TBR-based Trajectory Piecewise-Linear Algorithm for Generating Accurate Low-order Models for Nonlinear Analog Circuits and MEMS. In *Design Automation Conference*, pages 490 – 495, June 2003.

[39] A. Verhoeven. *Redundancy reduction of IC models by multirate time-integration and model order reduction.* PhD thesis, Technische Universiteit Eindhoven, 2008.

[40] Arie Verhoeven, Jan ter Maten, Michael Striebel, and Robert Mattheij. Model order reduction for nonlinear IC models. CASA-Report 07-41, TU Eindhoven, 2007. to appear in IFIP2007 conference proceedings.

[41] A.J. Vollebregt, Tamara Bechtold, Arie Verhoeven, and E.J.W. ter Maten. Model Order Reduction of Large Scale ODE Systems: MOR for ANSYS versus ROM Workbench. In G. Ciuprina and D. Ioan, editors, *Scientific Computing in Electrical Engineering – SCEE 2006*, volume 11 of *Mathematics in Industry*, pages 175 – 182. The European Consortium for Mathematics in Industry, Springer-Verlag Berlin Heidelberg, 2007.

[42] T. Voß, A. Verhoeven, T. Bechtold, and ter Maten J. Model Order Reduction for Nonlinear Differential Algebraic Equations in Circuit Simulation. In L.L. Bonilla, M. Moscoso, G. Platero, and J. M. Vega, editors, *Progress in Industrial Mathematics at ECMI 2006*. Springer-Verlag, 2007.

[43] Thomas Voß. Model reduction for nonlinear differential algebraic equations. Master's thesis, University of Wuppertal, 2005.

[44] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40(11):2323–2330, November 2002.

[45] Fan Yang, Xuan Zeng, Yangfeng Su, and Dian Zhou. Rlcsyn: Rlc equivalent circuit synthesis for structure-preserved reduced-order model of interconnect. *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, May 2007.

Chemnitz Scientific Computing Preprints – ISSN 1864-0087