A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems

by

Michał Jerzy Rewieński
M.S., Computer Science
Technical University of Gdańsk, Poland, 1998

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2003

Author .....
Department of Electrical Engineering and Computer Science
May 23, 2003

Certified by .....
Jacob K. White
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

To my Parents,

A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems

by
Michał Jerzy Rewieński
M.S., Computer Science
Technical University of Gdańsk, Poland, 1998

Submitted to the Department of Electrical Engineering and Computer Science
on May 23, 2003, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

In this study we discuss the problem of Model Order Reduction (MOR) for a class of nonlinear dynamical systems. In particular, we consider reduction schemes based on projection of the original state-space to a lower-dimensional space e.g. by using Krylov methods. In the nonlinear case, however, applying a projection-based MOR scheme does *not* immediately yield computationally efficient macromodels. In order to overcome this fundamental problem, we propose to first approximate the original nonlinear system with a weighted combination of a small set of linearized models of this system, and then reduce each of the models with an appropriate projection method. The linearized models are generated about a state trajectory of the nonlinear system corresponding to a certain 'training' input.

As demonstrated by results of numerical tests, the obtained trajectory quasi-piecewise-linear reduced order models are very cost-efficient, while providing superior accuracy as compared to existing MOR schemes, based on single-state Taylor's expansions. In this dissertation, the proposed MOR approach is tested for a number of examples of nonlinear dynamical systems, including micromachined devices, analog circuits (discrete transmission line models, operational amplifiers), and fluid flow problems. The tests validate the extracted models and indicate that the proposed approach can be effectively used to obtain system-level models for strongly nonlinear devices.

This dissertation also shows an inexpensive method of generating trajectory piecewise-linear (TPWL) models based on constructing the reduced models 'on-the-fly', which accelerates simulation of the system response. Moreover, we propose a procedure for estimating simulation errors, which can be used to determine accuracy of the extracted trajectory piecewise-linear reduced order models. Finally, we present projection schemes which result in improved accuracy of the reduced order TPWL models, as well as discuss approaches leading to guaranteed stable and passive TPWL reduced-order models.

Thesis Supervisor: Jacob K. White
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

First, I would like to thank Professor Jacob White, my advisor for the last four years, for introducing me to the world of nonlinear simulation and modeling. Learning from him, especially learning how to ask the right questions, has proved an invaluable experience. His guidance and encouragement made this work possible.

I am also very grateful to my thesis committee members, Professors Karen Willcox and Alexandre Megretski for the numerous, fruitful, and often very animated discussions on nonlinear dynamical systems and model order reduction, which substantially deepened my understanding of this problem and its challenges, and their important suggestions which contributed to improving and broadening this dissertation.

I would like to thank Luca Daniel and Prof. John Wyatt for interesting discussions on passivity preservation for nonlinear reduced order models, as well as Prof. Jaime Peraire, Prof. Steven Senturia, and Dr. Joel Phillips from Cadence, for their valuable comments and suggestions on my research.

Special thanks to my groupmates, including but not limited to Bjärne Buchmann, Jingfang Huang, Joe Kanapka, Tom Klemas, Jing Li, Deepak Ramaswamy, Xin Wang, Dave Willis, and Zhenhai Zhu who were always available for a friendly discussion – either technical, or totally non-technical, when I was getting tired of my codes and simulations.

Also, I would like to thank Dmitry Vasilyev for a very interesting and fruitful cooperation on some of the topics in nonlinear model order reduction, during my last year at MIT.

Finally, I give thanks to Sylwia for her simple everyday understanding, patience, and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Progress in fabrication technology of integrated circuits (ICs) over the last decade resulted in an unprecedented expansion of functionality of systems built on a single chip. Simultaneously, the design of modern ICs has become substantially more complex, creating a pressing demand for new, more effective and efficient, automated verification and optimization strategies and tools, in order to avoid costly multiple redesigns.

Today's systems-on-a-chip integrate analog and digital circuitry, as well as micro-electro-mechanical systems (MEMS), leading to complicated mixed-signal, mixed-technology designs. The complicated, multi-physical descriptions of the subsystems force the designers to move to higher levels of abstraction and use inexpensive behavioral macromodels in system-level design. At the same time, however, shrinking process technologies generate a need to consider low-level physicals effects e.g. transistor-level effects, details of implementation, parasitic effects, or signal integrity [57]. Consequently, simple heuristic macromodels, such as e.g. SPICE macromodels used in the past, are often insufficiently flexible and inaccurate for computational prototyping of modern integrated circuits.

In order to provide circuit designers with models of various circuit's subsystems, which would be adequate, yet suitable for system-level simulation, sophisticated approaches for macromodel extraction from a detailed physical description of the original device need to be applied. In order to be useful, such techniques for extracting macromodels (system-level models) need to satisfy a number of requirements. The most important ones may be summarized as follows:

- The extraction techniques should generate macromodels which provide feasible, accurate enough models of the original subsystems or devices.

- The extraction process should be relatively inexpensive, and as automatic as possible, i.e. require little or no extra modeling expertise from the designers to obtain a macromodel.

- The macromodels should be inexpensive to use. – More specifically, the cost of simulation with a macromodel should be substantially lower than the cost of analogous simulation with the original model.

- The system-level models extracted from various subsystems of a system-on-a-chip should use a simple, uniform description, regardless of the original, possibly very different, descriptions of the subsystems.

The above requirements also provide the main goals of macromodeling. So far, one of the most successful approaches toward macromodel extraction, which often meets the requirements set forth above, is based on the concept of model order reduction (MOR). MOR techniques generate 'reduced order models,' based on the original, detailed description of a device or system at hand. By the term 'reduced order model' we mean a model which:

- uses a significantly reduced number of variables or states, as compared to the original model, to describe a given system,
- is relatively inexpensive to simulate and store in computer's memory – the computational complexity associated with evaluating this model is significantly smaller than the analogous complexity for the original model.
- is controlled by a limited number of relevant inputs.

Thus, a reduced order model provides a designer with a convenient abstraction, necessary to efficiently perform system-level simulation and verification. Reduced order modeling has therefore been applied in a number of engineering disciplines. In electrical engineering MOR techniques were first developed for macromodeling lumped circuits and interconnects [7], [16], [17], [19], [28], [43], [44], [52], [64], [77], [78], [82]–[84], [86], [87], [97], [100], [101], [121]. Then, those techniques were extended and applied to distributed electronic systems, full-wave electromagnetic modeling of various structures (including e.g. waveguiding structures) [8], [11], [12], [21], [53], [85], [90], [104], [111]–[113]. Another important class of MOR applications refers to modeling power grids, and power electronic circuits [70], [80], [89]. Integration of micro-electro-mechanical systems (MEMS) on a chip, created a need to construct efficient macromodels for this class of devices through reduced order modeling. Consequently, many MOR strategies have been developed for MEMS [14], [15], [27], [40]–[42], [66], [73], [76], [88], [110], [118].

Applications of model order reduction reach far from electrical engineering, and include problems in computational fluid dynamics e.g. modeling turbomachinery, supersonic flows in jet engine inlets [4], [9], [37], [49], [115]–[117], efficient modeling of biological systems [50], [51], [120], rapid thermal processing [2], problems in chemical [61] or mechanical [59] engineering.

Apart from a growing pool of applications of MOR in various disciples of science and engineering, fast progress in developing various model reduction approaches is observed. The following section briefly describes major classes of existing MOR techniques, and comments on their applicability, as well as some of their advantages and limitations.

1.1.1 Previous work on model order reduction

So far, the largest group of model order reduction algorithms applies to linear systems, or more precisely linear time-invariant (LTI) systems. Within the MOR algorithms for LTI systems one may distinguish the following most popular classes: 1) MOR algorithms based on Krylov subspace methods, 2) methods based on Hankel norm approximants and truncated balancing realiza-

tion (TBR), 3) techniques using Karhunen-Loève expansion (or Proper Orthogonal Decomposition (POD)). All of those methods apply the concept of approximating a certain part of a high-dimensional state space of the original system with a lower-dimensional state space, or, in other words, they perform projection of the original state space (cf. Chapter 2, [22], [34], [74]). The projection is achieved in a number of different ways.

The first group of MOR algorithms uses Krylov subspace methods to construct the projection bases. Krylov subspace methods provide numerically robust algorithms for generating reduced bases which include a certain number of moments of the transfer function of the original system. Consequently, the transfer function of the reduced system approximates well the original transfer function around a specified frequency, or collection of frequency points [34]. Owing to their robustness and low numerical cost, Krylov subspace algorithms proved very suitable for reduction of large-scale systems, and gained considerable popularity, especially in electrical engineering. A number of Krylov-based MOR algorithms have been developed, including techniques based on Lanczos method [11]–[13], [25], [26], [29], [30], [90], or Arnoldi algorithm [7], [8], [18] [28], [43], [44], [78], [88], [100], [101], [110], [113], [117]–[119]. The main drawbacks of those methods are, in general, lack of provable error bounds for the extracted reduced models, and no guarantees of preserving stability or passivity of the original system. Nevertheless, it has been demonstrated that if the original system has a desired structure, then both stability and passivity may be preserved with the reduced system, by exploiting the fact that congruence transformations preserve definiteness of a matrix – cf. e.g. PRIMA algorithm [67], [77], [78], see also [7], [8], [19], [24], [28], [97], [98], [101], [119]. Also, some efforts have been been undertaken to develop error estimates for Krylov-based reduced models [5], [103].

Another group of MOR algorithms suitable for the analysis of LTI systems is based on Hankel norm approximants and truncated balancing realization (TBR) [32], [36], [65], [75], [81], [85], [87], [99], [105], [121]. Unlike Krylov-based method, balanced realization methods have provable error bounds for the extracted reduced order models [32], [75], and guarantee that stability of the original system will be preserved in the reduced order model.[1] The main drawback of this class of MOR methods is high numerical cost of extracting the reduced models, associated with expensive ($O(N^3)$) solution of Lyapunov equations. Due to this fast growing complexity, applicability of TBR methods for large-scale systems is limited. In order to overcome this difficulty methods based on approximate Grammian computation have been developed [4], [10], [64], [65], [71], [87], [99], [105], [121].

Yet another approach toward linear model order reduction is based on Karhunen-Loève expansion, or equivalently Proper Orthogonal Decomposition (POD), which obtains projection bases from time- or frequency-domain snapshots [9], [37], [49], [102], [115]–[117]. If time-domain snapshots are used, generation of the reduced bases is inexpensive, but at the same time the choice of suitable inputs for time-domain simulation becomes involved. On the other hand, generating frequency-domain snapshots is more costly, but more adequate bases are obtained, and spatial symmetries of the modeled system may be used to reduce the cost of model extraction [116]. One may also exploit connections between TBR and POD, in order to efficiently obtain approximate truncated balanced realizations of the original linear system, by using proper orthogonal decomposition method [58], [65], [116].

[1] Stability preservation does not immediately imply passivity preservation. For a discussion on passivity for TBR-based models refer to [85].

The reduction methods for LTI systems may be extended to modeling linear time varying (LTV) systems, or parametrized LTI systems. MOR methods for LTV systems are based on the concept of time-varying transfer functions, and Krylov-based reduction [82], [86], [94]. The recently developed algorithms for extracting geometrically parametrized reduced order models which allow one to efficiently perform not only simulation, but also optimization, exploit similarities between frequency variables and geometric parameters, and perturbation theory to construct macromodels with Krylov subspace methods, or truncated balanced realization [1], [20], [38], [67], [113].

While model order reduction is currently a well established approach to generate low-cost macromodels for linear systems, developing effective and efficient MOR strategies for nonlinear systems remains a challenging and relatively open problem. Furthermore, due to a broad scope of applications (cf. Chapter 2 for examples) which require nonlinear macromodeling, nonlinear model order reduction constitutes an area of very intensive investigation.

A number of linear MOR techniques described above have been extended to the nonlinear case. The simplest approach for generating reduced order models for nonlinear systems is based on linearization, or polynomial (Taylor) expansion of system's nonlinearity, and subsequent application of Krylov projection [14], [16], [17], [74], [83], [84], [86], [110] (cf. also Section 2.3). Bilinearization of the nonlinearity [6], [47], [84], [86], as well as functional (Volterra) series expansion [42], [50], [51], [86], followed by a suitable projection, provide similar solutions. The main drawback of those approaches is that they generate macromodels valid only locally around the initial operating point of the nonlinear system. This is a straightforward consequence of an inherently local nature of approximation of the nonlinearity by polynomial or functional series, and limits application of the discussed MOR methods to weakly nonlinear systems, and 'small' input disturbances. Another difficulty which arises while using e.g. polynomial expansions of nonlinear operators is fast, exponential growth (with the order of the expansion) of computational and memory complexity associated with using the macromodels. Consequently, practical implementations are limited to quadratic expansions.

MOR techniques based on Proper Orthogonal Decomposition (Karhunen-Loève expansion) have also been widely used to model nonlinear systems [2], [15], [40], [41], [52], [89], [120]. Yet another group of methods generates approximate truncated balanced realizations for nonlinear systems [69], [95], often exploiting POD to find approximate Grammians [59], [61]. The main problem with this class of algorithms applied to the nonlinear case is that they generally inadequately address the problem of efficient representation of the nonlinearity in the reduced order model. Consequently, despite the reduced order of the solved system, the cost of evaluating the projected nonlinear operator often remains very high, which severely reduces performance of the corresponding reduced order models. This issue is addressed in more detail in the following chapter. Another problem with using the discussed algorithms is a high cost of generating the reduced order models, due to necessity of computing snapshots in the time domain, which in turn requires performing simulation of the original nonlinear system.

Summing up, a number of fundamental issues in reduced order modeling of nonlinear systems remains unresolved. On one hand they include overcoming the 'weak nonlinearity' constraint which severely limits the scope of applications, and on the other hand they involve addressing questions related to cost-efficient representation of system's nonlinearity and to numerical cost of generating the reduced order models.

1.2 Scope and goals of the dissertation

In this study, we will investigate the problem of model order reduction for strongly nonlinear systems. We will focus on considering dynamical systems (1st order ODEs) in state-space formulation, since it constitutes by far the most popular description for systems in various engineering disciplines.

Also, we will attempt to design MOR approaches which are cost-efficient – provide relatively inexpensive strategy for automatically extracting the reduced order model, and above all generate feasible reduced order models which are easy and inexpensive to use, and consequently may effectively and efficiently applied in system-level simulations. To this end, we will select numerical techniques which, in author's opinion, proved most effective and robust at different stages of model reduction. In order to validate the proposed MOR strategies, we will select examples of systems from engineering domain, which proved challenging for previously developed macromodeling techniques. According to the above description, the scope of this dissertation may be summarized in the following points:

- Presenting formulation of the model order reduction problem for nonlinear dynamical systems.
- Proposing a cost-efficient method of representing system's nonlinearity.
- Developing automatic, effective, and efficient state-space reduction and model extraction strategies.
- Analyzing selected properties of the extracted macromodels, including stability and passivity, and investigating the problem of estimating errors associated with reduced order modeling.
- Presenting selected application examples and validating the proposed MOR strategies for those examples through numerical simulations.
- Carrying out performance tests for the proposed MOR algorithms, and extracted macromodels of nonlinear systems.

In this context, the emerging main goal of this dissertation is developing an automatic, cost-efficient model order reduction strategy applicable to those strongly nonlinear systems, which could not be effectively treated by the previously existing MOR algorithms.

1.3 Chapter outline

We start in Chapter 2 by defining nonlinear dynamical systems in mathematical terms, and presenting examples of selected problems from the domains of electrical engineering (nonlinear analog circuits, micro-electro-mechanical systems) and computational fluid dynamics (shock dynamics problems), which are then formulated as initial value problems for nonlinear dynamical systems. Next, we describe a general model order reduction approach based on the state space projection framework, and discuss the main challenges associated with using this approach for reducing nonlinear systems. We also present strategies for generating a suitable state-space projection basis,

focusing on using popular Krylov subspace techniques (cf. e.g. [34]). Finally, we discuss some of the earlier attempts to overcome difficulties of nonlinear MOR, based on polynomial expansions of the nonlinear operator [83], and leading to reduction schemes appropriate for weakly nonlinear dynamical systems.

Chapter 3 presents an originally developed reduction technique – a Trajectory Piecewise-Linear (TPWL) MOR scheme [91], based on considering a collection of linear models along a state-space trajectory of the original nonlinear system. Also, possible methods for constructing the reduced order basis are discussed, including a novel approach in which multiple bases corresponding to different Krylov subspaces are aggregated [92]. A fast approximate simulation algorithm is also proposed, which allows one to efficiently generate TPWL reduced order models. This chapter also discusses issues related to simulating the extracted reduced order models, e.g. possible time integration schemes, or limitations of the proposed approach.

Chapter 4 provides error analysis for TPWL reduced order models. First, a procedure for *a posteriori* error estimation is developed. This procedure can be used to estimate the solution errors on the fly, during the actual simulation with a given reduced order model. Then, a method of selecting linearization points for the TPWL reduced order models, based on the *a posteriori* error analysis is proposed. Finally, we discuss the problem of error boundedness for infinite system trajectories, using *a priori* error analysis.

In Chapter 5 we focus on the problem of preserving stability and passivity with the TPWL reduced order models. The presented discussion provides insight to the problems which may arise when replacing the original nonlinear system with a trajectory piecewise-linear system. Possible approaches for ensuring stability and passivity of a TPWL system are considered.

Chapter 6 presents computational results, which validate the proposed TPWL model order reduction algorithm for the examples of nonlinear analog circuits and micromachined circuits. Results of both time-domain and periodic steady-state simulations are presented and discussed.

Chapter 7 analyzes performance of the proposed MOR strategies, the cost of basis generation, and efficiency of simulation with the reduced order models, using results of numerical tests for the implemented MOR algorithms.

Finally, in Chapter 8, we present our conclusions, and brief recommendations for future work.

Chapter 2

Model Order Reduction for nonlinear dynamical systems

In this chapter we first define the class of dynamical systems to be considered in this study, and briefly state which types of problems associated with those systems typically arise in engineering applications. Then, in order to provide further motivation and background, in Section 2.2 we present examples of systems which may be described in terms of nonlinear dynamical systems, and will be used throughout this dissertation to test the proposed MOR techniques, and illustrate problems associated with nonlinear model order reduction.

Finally, in Section 2.3 we present a general framework for Model Order Reduction, based on the concept of projecting the original state space into a low-dimensional, reduced-order space. At that point we also discuss the main challenges associated with Model Order Reduction for nonlinear systems, and describe the simplest strategies for nonlinear MOR which provide a starting point for more sophisticated algorithms proposed in the following chapters.

2.1 Formulation of the problem

In this study we focus on nonlinear dynamical systems (first order ordinary differential equations) in the following state-space form:

$$\begin{cases} \frac{dg(x)}{dt} = f(x) + B(x)u \\ y = C^T x \end{cases}, \tag{2.1}$$

where $x = x(t) \in R^N$ is a vector of states (evolving with time t) for a given system, $f : R^N \to R^N$ and $g : R^N \to R^N$ are nonlinear vector-valued functions, $B = B(x)$ is a state-dependent $N \times M$ input matrix, $u \in R^M$ is an input to the system, C is an $N \times K$ output matrix and $y \in R^K$ is the output. The first of the equations is called the state evolution equation, and the second one – the output equation. Typically, states contained in vector x have some physical meaning. They are e.g. voltages at specified circuit nodes, or values of pressure at certain grid points.

Although the above form of a dynamical system is not the most general one, it is sufficiently ample for a broad class of engineering problems. Consequently, in this thesis we will formulate model order reduction algorithms for systems in form (2.1). Possible extensions, such as including

the case of nonlinear input term ($B(x,u)$), or nonlinear output ($y = C(x)$) will be briefly discussed in Section 6.1.3. The problems associated with dynamical systems in form (2.1), most commonly encountered in engineering include:

1. Finding the response of system (2.1) to a given input signal $u = u(t)$ on a specified time interval $t \in [0,T]$ ($T > 0$), and for a given initial state of the system x_0 ($x(0) = x_0$). Mathematically, it corresponds to solving an initial value problem for the first of equations (2.1) for a given time interval and computing the output $y(t)$.

2. Finding the periodic steady state of system (2.1), for a given periodic input signal u (with period T). Mathematically, it corresponds to finding a solution $x(\tau)$ for the first of equations (2.1), for $\tau \in [0,T]$, such that $x(0) = x(T)$.

In the rest of this study we will implicitly assume that functions $f(x)$, $g(x)$, and $B(x)$ are regular enough to ensure existence and uniqueness of solutions to problems described above.

2.2 Examples of nonlinear dynamical systems

Before we turn to defining the problem of Model Order Reduction for nonlinear systems, in this section we present examples of devices from engineering domain which can be described in terms of dynamical systems in form (2.1). The presented examples show how broad of a spectrum of problems may be modeled using the framework of dynamical systems. More importantly, those examples serve us later on to validate the proposed MOR techniques and, due to their often strongly nonlinear dynamical behavior, illustrate well certain challenges associated with nonlinear model order reduction.

2.2.1 Nonlinear analog circuits

The first group of examples includes nonlinear analog circuits. In those devices the nonlinearities are introduced by various electronic elements, such as diodes, transistors, or nonlinear resistors. Due to an inherently discrete nature of lumped circuits, those systems may be directly cast into a standard state-space form (2.1) of a dynamical system.

The first of the circuit examples, considered also by Chen et al. [17], is a nonlinear transmission line circuit model shown in Figure 2-1. The circuit consists of resistors, capacitors, and diodes with a constitutive equation $i_d(v) = \exp(40v) - 1$, where v is the voltage between diode's terminals. (In the linear model, considered later on, we assume that $i_d(v) = 40v$, and in the quadratic model – $i_d(v) = 40v + 800v^2$.) For simplicity we assume that all the resistors and capacitors have unit resistance and capacitance, respectively ($r = 1$, $C = 1$). The input is the current source entering node 1: $u(t) = i(t)$, and the output is chosen to be the voltage at node 1: $y(t) = v_1(t)$. Using constitutive relations for capacitors, resistors, and diodes, as well as Kirchhoff's current law, we obtain a dynamical system:

$$\begin{cases} \frac{dx}{dt} = f(x) + Bu \\ y = C^T x \end{cases}, \tag{2.2}$$

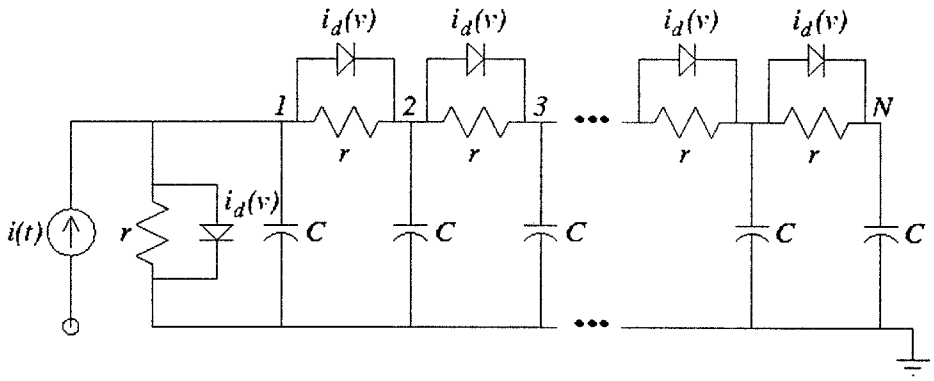

Figure 2-1: Example of a nonlinear transmission line circuit model.

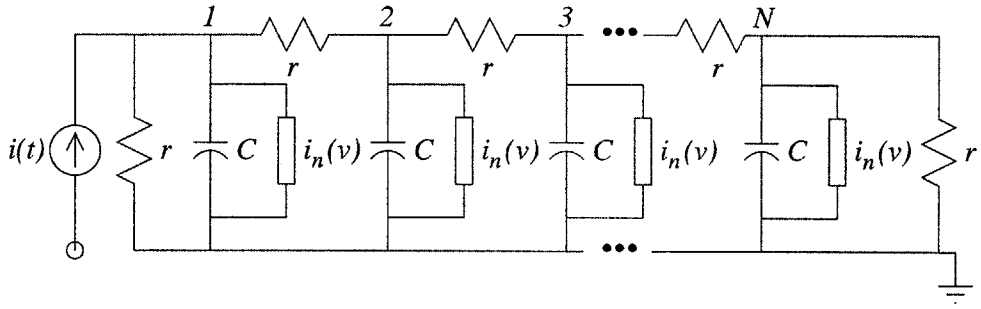

Figure 2-2: Example of a transmission line model with quadratic nonlinearity.

where $x = [v_1, v_2, \ldots, v_N]^T$ is a vectors of states, and consists of voltages v_i at subsequent nodes of the system, N is the number of nodes, $B = C = [1, 0, \ldots, 0]^T$, and

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

We also considered an example of a transmission line circuit model with a simple quadratic nonlinearity (cf. Figure 2-2), which served mainly to test the proposed error estimation procedure (cf. Chapter 4). The quadratic nonlinearity is introduced to the circuit by adding nonlinear resistors to the ground at each node:

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

where v is the voltage at the resistor terminals, g is the conductivity coefficient, and $\mathrm{sgn}(v) = 1$ if $v \geq 0$ and $\mathrm{sgn}(v) = -1$ if $v < 0$. If we take $C = r = g = 1$, then the nonlinear operator f (cf. (2.1)) takes the following form:

$$f(v) = Av - n(v), \tag{2.3}$$

where:

$$A = \begin{bmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ & & 1 & -2 \end{bmatrix}, \quad n(v) = \begin{bmatrix} \mathrm{sgn}(v_1)v_1^2 \\ \mathrm{sgn}(v_2)v_2^2 \\ \dots \\ \mathrm{sgn}(v_N)v_N^2 \end{bmatrix}, \tag{2.4}$$

and $v = [v_1, \dots, v_N]$ is the vector of states.

If the number of nodes N is large, simulating the behavior of the presented circuits becomes computationally expensive. Consequently, in order to reduce the effort associated with circuit simulation, one may consider using macromodels, instead of full order models. However, in the first circuit, a strongly nonlinear element (a diode) is connected to every node of the circuit, which makes it a challenging system for the existing MOR techniques. As shown in the following chapters, the presented transmission line circuit models provide excellent examples to test performance and accuracy of the discussed model order reduction techniques.

Another circuit example we consider in this study is an operational amplifier with differential input and output, and consisting of 70 MOSFETs, 13 resistors and 9 linear capacitors connected to 51 circuit nodes. (The netlist for this circuit in the NITSWIT circuit simulator format [54] is included in Appendix A.) Nodal analysis yields a nonlinear model of the device in form (2.1), with voltages at the circuit nodes defining a state vector. It should be noted, that if MOSFET transistor models include the effect of a nonzero gate oxide thickness (as controlled by the 'tox' parameter – cf. Appendix A), then the capacitances become nonlinear, and consequently $g(x)$ (cf. equation (2.1)) becomes a nonlinear function of x. Also, the op-amp is modeled as a multiple input system with eight inputs: 1) the differential input with signals v_{in1} and v_{in2}, 2) the auxiliary inputs v_{cmmrst}, v_{gnd}, v_{intn}, v_{intp}, v_{rst}, and v_{cmmin} used in common mode rejection testing.

The main difficulty with simulating this circuit lies *not* in the order of the system of nonlinear ODEs to be solved (which in this case equals $N = 51$), but rather in handling the multiple nonlinearities associated with each of the transistor models. Some circuit simulators use e.g. a piecewise-linear model separately for every transistor in the circuit (cf. e.g. Saber [109]). During simulation of the circuit the operating points for each of the transistors change at very different times and frequencies. Consequently, a very large number of piecewise-linear regions, and a variety of timescales need to be taken into account during a single simulation, which leads to complicated and often costly integration schemes.

In this context Model Order Reduction algorithms may serve as a tool to generate a simplified and unified description of the entire circuit, rather than its numerous components (e.g. transistors). Such a description in terms of a reduced order dynamical system can be used in conjunction with straightforward time integration schemes, and more importantly – very easily incorporated to system-level simulators. As we will show in the following chapters, the proposed trajectory piecewise-linear (TPWL) MOR approach allows one to model the discussed op-amp circuit with such a compact, easy to use macromodel, consisting of a small number of linearized models.

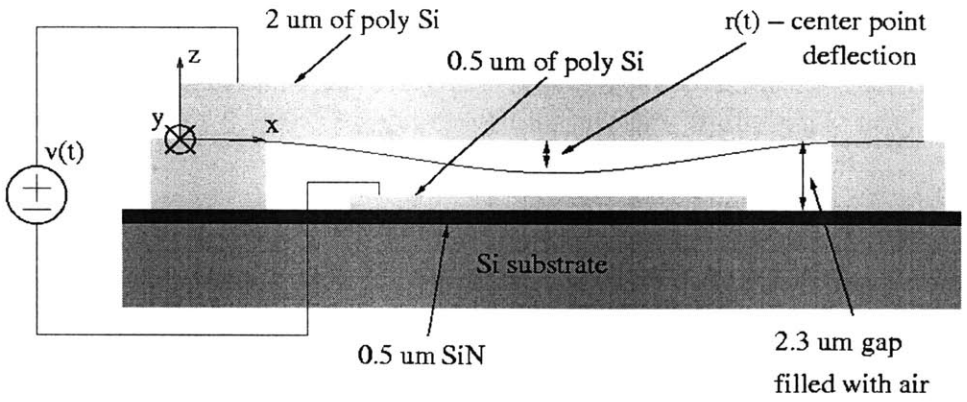

Figure 2-3: Micromachined switch (following Hung et al. [40]).

2.2.2 A micromachined device

Another example which can be modeled as a nonlinear dynamical system is the micromachined switch (fixed-fixed beam) shown in Figure 2-3. The switch consists of a polysilicon beam suspended over a silicon substrate. The beam has length $l = 610\mu$m, width $w = 40\mu$m, and height $h = 2.2\mu$m. When voltage is applied between the beam and the substrate (cf. Fig. 2-3), the beam deflects toward the silicon substrate. One should also note that a layer of air between the beam and the substrate plays an important role in the behavior of this microdevice, acting as a damper for the deflecting beam.

Since $l \gg w$ we may assume that the deflection of the beam is uniform across its width. Consequently, following Hung et al. [40], the dynamical behavior of the discussed coupled electromechanical-fluid system can be modeled with 1D Euler's beam equation, and 2D Reynolds' squeeze film damping equation given below:

$$\hat{E}I\frac{\partial^4 u}{\partial x^4} - S\frac{\partial^2 u}{\partial x^2} = F_{elec} + \int_0^w (p - p_a)dy - \rho\frac{\partial^2 u}{\partial t^2}, \tag{2.5}$$

$$\nabla \cdot ((1 + 6K)u^3 p\nabla p) = 12\mu\frac{\partial(pu)}{\partial t}, \tag{2.6}$$

where x, y, and z are as shown in Figure 2-3, $\hat{E}$ is Young's modulus, I is the moment of inertia of the beam, S is the stress coefficient, ρ is the density, p_a is the ambient pressure, μ is the air viscosity, K is the Knudsen number, $u = u(x,t)$ is the height of the beam above the substrate, and $p = p(x,y,t)$ is the pressure distribution in the air below the beam. The electrostatic force is approximated assuming nearly parallel plates and is given by $F_{elec} = -\frac{\varepsilon_0 w v^2}{2u^2}$, where v is the applied voltage.

For the considered example the following set of parameters has been used: $E = 149$GPa, $S/(hw) = -3.7$MPa, $\rho/(hw) = 2330kg/m^3$, $p_a = 1.013 \cdot 10^5$Pa, $\mu = 1.82 \cdot 10^{-5}kg/(m \cdot s)$, $\lambda = 0.064\mu m$. Also, for the system described by equations (2.5) and (2.6) we consider the following initial conditions:

$$u(x,0) \equiv u_0, \quad p(x,y,0) \equiv p_a,$$

for all $x \in [0,l]$, and $y \in [0,w]$, where $u_0 = 2.3\mu m$ is the length of the gap between the substrate and the undeflected beam, and the following boundary conditions:

$$u(0,t) \equiv u(l,t) \equiv u_0, \quad \partial p(0,y,t)/\partial x \equiv \partial p(l,y,t)/\partial x \equiv 0, \quad p(x,0,t) \equiv p(x,w,t) \equiv p_a, \tag{2.7}$$

for all $x \in [0,l]$, $y \in [0,w]$, and $t > 0$.

The following state-space representation $x = [x_1, x_2, x_3]^T$ has been selected for the physical problem described above:

$$x_1 = u, \quad x_2 = \frac{\partial(u^3)}{\partial t}, \quad x_3 = p. \tag{2.8}$$

Using the above state-space variables equations (2.5) and (2.6) may be written as follows:

$$\frac{\partial x_1}{\partial t} = \frac{x_2}{3x_1^2} \tag{2.9}$$

$$\frac{\partial x_2}{\partial t} = \frac{2x_2^2}{3x_1^3} + \frac{3x_1^2}{\rho}\left[\int_0^w (x_3 - p_a)dy + S\frac{\partial^2 x_1}{\partial x^2} - EI\frac{\partial^4 x_1}{\partial x^4}\right] - \frac{3\varepsilon_0 w}{2\rho}v^2 \tag{2.10}$$

$$\frac{\partial x_3}{\partial t} = -\frac{x_2 x_3}{3x_1^3} + \frac{1}{12\mu x_1}\nabla\left(\left(1 + 6\frac{\lambda}{x_1}\right)x_1^3 x_3 \nabla x_3\right). \tag{2.11}$$

The above equations may be written in the following compact form:

$$\frac{\partial x}{\partial t} = \mathbf{N}(x) + Bv^2, \tag{2.12}$$

where $\mathbf{N}$ is a certain nonlinear integro-differential operator, and B is a constant input vector. One should note that, although selecting state vector in form (2.8) is only one of many possible choices, it leads to a particularly simple form (2.12), with B being *constant* (i.e. independent of x or t) due to the appropriately selected state-space variables (2.8). One may also consider a 'more natural' state vector $\tilde{x} = [\tilde{x}_1, \tilde{x}_2, \tilde{x}_3]^T$, where:

$$\tilde{x}_1 = u, \quad \tilde{x}_2 = \frac{\partial u}{\partial t}, \quad \tilde{x}_3 = p,$$

which then leads to a system in the following form:

$$\frac{\partial \tilde{x}}{\partial t} = \tilde{\mathbf{N}}(\tilde{x}) + \tilde{B}(\tilde{x})v^2. \tag{2.13}$$

Both forms (2.12) and (2.13) were used to test the proposed model order reduction approaches. In order to solve system (2.12) or (2.13) we perform spatial discretization of the computational domain, using the following grid in the $\hat{x} - \hat{y}$ plane: $x^i = i \cdot \Delta x$, $y^j = j \cdot \Delta y$, for $i = 0, \ldots, (n+1)$ and $j = 0, \ldots, (m+1)$, where $x^0 = 0$, $x^{n+1} = l$, $y^0 = 0$, $y^{m+1} = w$, $\Delta x = l/n$, and $\Delta y = w/m$. Applying a standard finite difference scheme to equations (2.9)-(2.11) yields a large nonlinear dynamical system in form (2.1) of order $N = 2n + mn$. (Please refer to Appendix B for a detailed formulation

of the discrete system.) In our tests we take $n = 40$, and $m = 20$, which leads to a system of order $N = 880$. For the considered example we also select our output $y(t) = u(l/2)$, i.e. as a height of the center of the deflected beam – cf. Figure 2-3.

The microswitch problem has been selected as one of the main test examples for model order reduction algorithms proposed in this dissertation due to a number of factors. Firstly, because of a complicated structure of the nonlinearity, and a relatively large number of equations to be solved (after discretization) this device is expensive to model using straightforward time integration of the original model. Secondly, the ability to model the dynamics of the pull-in, which is a strongly nonlinear effect observed during operation of the discussed microswitch, is of particular interest to MEMS engineers. Finally, the microswitch example proved very challenging for the existing MOR techniques. As illustrated in Chapter 6, some of those recently developed techniques failed to provide accurate macromodels of the original system. Summing up, many problems of nonlinear model order reduction become very visible while considering the presented micromachined switch example.

2.2.3 Fluid dynamics problems

In this study we also consider problems associated with fluid dynamics. In particular, we focus on two examples of shock movement in a fluid, which is an inherently nonlinear phenomenon.

We start by considering a model problem based on 1D Burgers' equation:

$$\frac{\partial U(x,t)}{\partial t} + \frac{\partial f(U(x,t))}{\partial x} = g(x), \tag{2.14}$$

where U is the unknown conserved quantity (mass, density, heat etc.), $f(U) = 0.5U^2$ and in this example, $g(x) = 0.02\exp(0.02x)$. The initial and boundary conditions used with the above PDE are:

$$U(x,0) \equiv 1, \quad U(0,t) = u(t),$$

for all $x \in [0,l]$, and $t > 0$, where u is the incoming flow, and l is the length of the modeled region. Discretizing U with respect to x yields $\underline{U} = [U_1, \ldots, U_N]^T$, where $U_i = U(x_i)$, and $x_i = i\Delta x$ ($\Delta x = l/N$, where N is the number of grid points). Applying Godunov's scheme for approximating $\partial/\partial x$ [39] to (2.14), we obtain the following equation at node $i > 0$:

$$\frac{dU_i}{dt} = -\frac{1}{\Delta x}\left(F_{i+1/2} - F_{i-1/2}\right) = g(x_i),$$

where:

$$F_{i+1/2} = \begin{cases} \min_{U\in[U_i,U_{i+1}]} f(U) & U_i < U_{i+1} \\ \max_{U\in[U_i,U_{i+1}]} f(U) & U_i > U_{i+1} \end{cases}.$$

Please note that for $f(U) = 0.5U^2$, and if all U_i's are positive, then $F_{i+1/2} = 0.5U_i^2$. Discretizing (2.14) with respect to x, and incorporating the boundary conditions results in the following dynamical system:

$$\frac{d\underline{U}}{dt} = F(\underline{U}) + G + Bu^2,$$

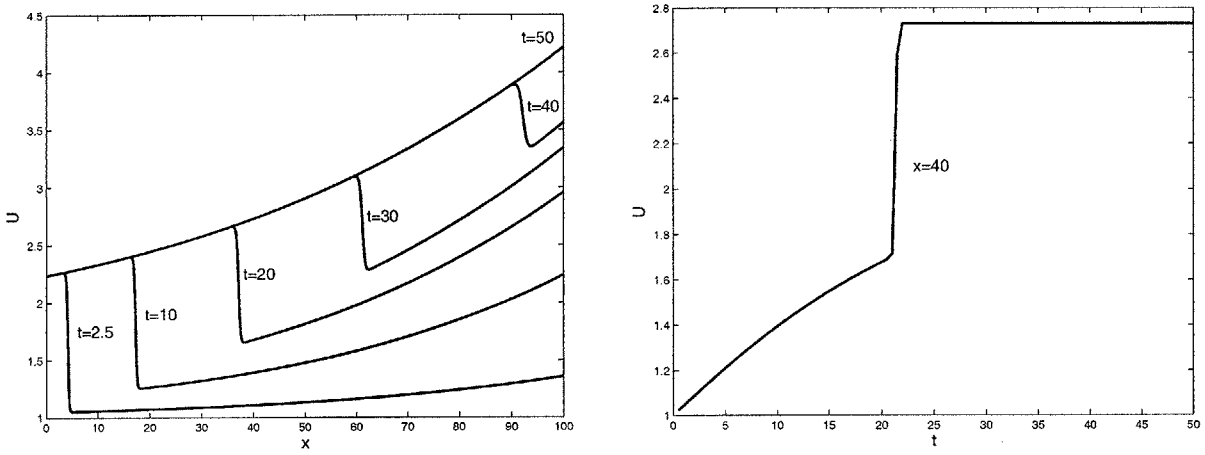

Figure 2-4: The left graph shows snapshots of the moving shock. The right graph shows the shock passing through a specified location at $x = 40$.

where $G = 0.02[\exp(0.02x_1)\ldots\exp(0.02x_N)]^T$, $B = [(1/(2\Delta x))\,0\ldots0]^T$, and:

$$F(\underline{U}) = \frac{1}{\Delta x}\begin{bmatrix} -0.5U_1^2 & & & \\ & 0.5(U_1^2 - U_2^2) & & \\ & & \ddots & \\ & & & 0.5(U_{N-1}^2 - U_N^2) \end{bmatrix}.$$

Figure 2-4 shows the shock movement, as modeled by Burgers' equation (2.14), for $l = 100$, $N = 1000$, $\Delta x = 0.1$, and the incoming flow $u(t) \equiv \sqrt{5}$.

The second problem refers to modeling shock dynamics for a supersonic flow of air in a 2D section of a jet engine inlet. Figure 2-5 shows the engine inlet geometry, along with Mach contours, for an example of a flow (in a steady state, with freestream Mach number $M = 2.2$). The incoming supersonic flow is subject to perturbations, due to e.g. atmospheric air density disturbance. If this disturbance is significant enough, it may cause the shock to move toward the inlet, a phenomenon known as 'unstart' [60]. In order to model this effect inviscid and unsteady Euler equations are used. Finite-volume discretization of those equations leads to a large, nonlinear dynamical system in the form

$$\frac{dF(\underline{U})}{dt} = -R(\underline{U}, \dot{m}, d), \tag{2.15}$$

where $\underline{U} = [U_1, \ldots, U_N]$ is a state vector, and $U_i = [\rho^i, q^i, q^i_\perp, H^i]$ defines a state of the system at node i, where ρ^i is the air density, q^i and $q^i_\perp$ are the streamwise and normal components of the flow velocity, respectively, and H^i is the total enthalpy, for node i. Also: $\dot{m}$ specifies bleed mass fluxes, d contains the inlet disturbances, F and R are nonlinear vector-valued functions. A detailed description of the mathematical formulation, boundary conditions, and discretization is out of the scope of this dissertation, and may be found in [60].

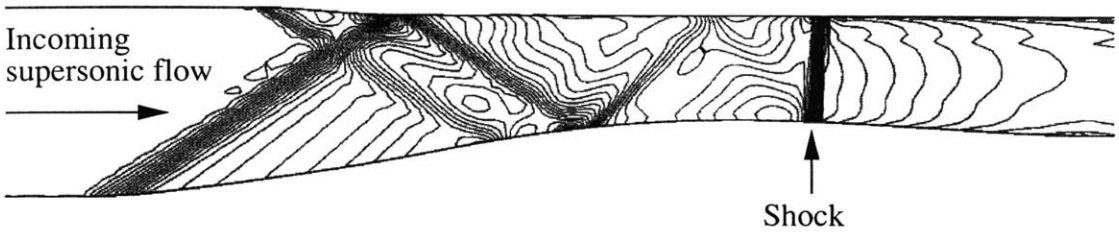

Figure 2-5: The engine inlet geometry and Mach contours for a supersonic flow (following Lassaux [60]).

State evolution equation (2.15) is complemented by an output equation in the form: $y = C(\underline{U})$, where C is a (nonlinear) function, and y is the output. For our example y is the average Mach number at the throat.

One should note that discretization considered in [60] leads to a system of order $N = 11730$. Consequently, simulation of nonlinear system (2.15) becomes very costly. Moreover, linearization of (2.15) (e.g. at the steady state) leads to a model which is unable to reproduce shock movement (cf. Chapter 6). Therefore, in order to model the unstart phenomenon efficiently, nonlinear MOR approach needs to be used.

2.3 Projection framework – challenges of nonlinear MOR

In this section we discuss one of the most widely used general frameworks for Model Order Reduction of systems described in state-space form – the projection framework. We also point out the main difficulties associated with using this approach in the nonlinear case.

Let us consider again nonlinear dynamical system (2.1):

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

where x is a vector of states. Suppose the above system is of order N, i.e. vector x contains N states. Projection-based model order reduction techniques aim at generating a system of order $q \ll N$ (i.e. involving state vectors of order q) which would approximate (in some sense) the original system through selecting a 'useful' or 'relevant' subspace in the original state-space of R^N. This is achieved by constructing a suitable $N \times q$ matrix V, whose columns span a certain subspace (and define a basis in the reduced-order state space), and approximating all states x from the original state space by reduced-order states z:

$$x \simeq \hat{x} = Vz. \tag{2.16}$$

In order to construct equations for the reduced system, we consider a residual $r = (f(Vz) + B(x)u - dg(Vz)/dt)$, and require this residual to be orthogonal to a reduced order space, spanned by columns of a suitably selected matrix W, i.e.:

$$W^T r = 0. \tag{2.17}$$

This condition leads to a reduced order system of equations in the following form:

$$\begin{cases} \frac{d}{dt}[W^T g(Vz(t))] = W^T f(Vz(t)) + W^T B(Vz(t))u(t) \\ y(t) = C^T Vz(t). \end{cases} \tag{2.18}$$

where $z \in R^q$. Please note that throughout this thesis we assume that $f, g : R^N \to R^N$, and consequently W is an $N \times q$ matrix. When $W = V$, which is often the case, condition (2.17) means that the residual error, due to restricted representation of x, should be orthogonal to the reduced state space.

System (2.18) is a reduced order system, in the sense that the unknown is a vector of order q, and the number of state evolution ODEs to be solved also equals q. If f and g are linear functions, i.e. $f(x) = Fx$, and $g(x) = Gx$, and the input term is state independent, i.e. $B(\cdot) \equiv B$, then dynamical system (2.18) will transform into:

$$\begin{cases} W^T GV \frac{dz(t)}{dt} = W^T FVz(t) + W^T Bu(t) \\ y(t) = (V^T C)^T z(t) \end{cases} . \tag{2.19}$$

Note that $W^T GV$ and $W^T FV$ are both $q \times q$ matrices. If M is a number of inputs, and K is the number of outputs, then $W^T B$ is a $q \times M$ matrix, and $V^T C$ is a $q \times K$ matrix. If the number of inputs and outputs is small, e.g. $M, K \leq q$, which is typically the case, then evaluating any of the terms from left or right hand side of (2.19) will require at most $O(q^2)$ operations. Also, simulation of the reduced-order linear system (2.19) will be inexpensive. If we use backward Euler time stepping scheme, and the number of steps equals p, then the cost of simulating (2.19) will be at most $O(pq^3)$. It is crucial to note that this cost does not depend on the size of the original system N, but only on the size of the reduced system q, which is essential for achieving a substantial reduction in the simulation time. Summing up, in the linear case reducing the order of the system *automatically* implies reducing the numerical cost associated with evaluating/simulating the system at hand. In other words, order of a model is a suitable measure of complexity for linear dynamical systems.

This in turn implies that, in the linear case, by applying model order reduction we immediately satisfy the main goal of macromodeling – substantial reduction of numerical cost of simulating a given system. The only issue which needs to be considered is constructing reduced order bases V and W, such that the reduced order system will provide a feasible approximation of the original system.

In the nonlinear case the situation is much more involved. Although, in system (2.18) the number of equations is reduced to q, and the unknown vector of states is of order q, this does *not* imply that system (2.18) is inexpensive to simulate. Let us assume that evaluation of $f(x)$ takes $O(N^\alpha)$ operations, where $\alpha \geq 1$. (In a typical case we assume that the cost of evaluating a nonlinear function $f(x) : R^N \to R^N$ to be at least linear in terms of N.) Let us now consider the cost of evaluating the corresponding term – $W^T f(Vz)$ in the reduced order system. Except for a few specific forms of f (cf. e.g. [2]) computation of this term is performed in the following 3 steps: 1) computing $x = Vz$, 2) evaluating the full-order nonlinear function $\tilde{x} = f(x)$, and 3) computing $W^T \tilde{x}$. This three-step procedure requires $O(N^\alpha + 2Nq)$ operations, and is actually costlier than evaluating $f(x)$. Also, in order to simulate system (2.18) one needs to compute a Jacobian $W^T J_f(z)V$ (where J_f is the Jacobian of f) at every Newton iteration, which requires $O(Nq)$ operations, provided J_f is sparse, i.e. the number of its nonzero elements is of $O(N)$, rather than $O(N^2)$. Consequently,

in this case the cost of simulating reduced order model (2.18) will typically be dominated by the cost of evaluating the nonlinear term $W^T f(Vz)$, and its Jacobian. If using backward Euler time-stepping scheme, the discussed simulation cost will be roughly $O(pk(Nq+N^\alpha+q^3))$, where p is the number of timesteps, k is an average number of Newton iterations per timestep, and the term q^3 refers to the cost of factorizing the reduced order Jacobian. Unlike in the linear case the simulation cost depends on the order of the original system N, which will adversely affect the simulation performance.

This brief analysis also shows that in the nonlinear case reduction of the order of a system at hand does *not* automatically imply reduction of numerical cost associated with simulating the reduced order model, which is one of the most profound differences between linear and nonlinear model order reduction. This also follows from a somewhat more general observation, that order of a model is not a valid measure of complexity of a nonlinear system.

Consequently, in order to obtain efficient, low-cost reduced order models for nonlinear systems the following two issues need to be addressed by a MOR technique:

1. Constructing low-order projection bases W and V, which approximate 'useful' parts of the state-space.

2. Applying low-cost, yet feasible approximate representation of system's nonlinearity (associated with $f(x)$ and $g(x)$).

The first of the issues – developing suitable projection bases for linear systems – has received a considerable attention, leading to successful approaches based e.g. on Proper Orthogonal Decomposition (Karhunen-Loève expansion) (cf. also Chapter 1). At the same time, the problem of finding cost-efficient representations of nonlinearity has been widely overlooked or ignored. Many of the MOR techniques presented in literature (cf. e.g. [40], [59], [89]) treat system in form (2.18) as their final reduced order model, and do not analyze the cost associated with evaluating that model.

In this study we will attempt to propose an approach which will address both issues mentioned above. Moreover, we will focus on numerical techniques which perform model extraction at a low cost, and therefore are well suited for reducing large-scale dynamical systems.

In the next section we briefly describe the simplest approaches for 1) obtaining a low-cost representation of the nonlinearity, 2) constructing a reduced order projection basis, which will provide a necessary introduction for a new strategy toward nonlinear model order reduction presented in Chapter 3.

2.3.1 Nonlinearity representation based on polynomial expansion

In order to extract efficient reduced order models for nonlinear systems, one needs to find representations for $f(\cdot)$ and $g(\cdot)$ which allow approximate evaluation of $W^T f(V\cdot)$ at a cost significantly lower than $O(N)$, where N is the size of the original system.

The simplest such approach is based on a polynomial expansion of the nonlinearity, or more specifically on Taylor's expansion about some (usually initial or equilibrium) state x_0. We expand f (following [82]) as:

$$f(x) = f(x_0) + A_0(x-x_0) + \frac{1}{2}W_0(x-x_0)\otimes(x-x_0) + \ldots, \tag{2.20}$$

where $\otimes$ is the Kronecker product, and A_0 and W_0 are, respectively, the Jacobian and the Hessian of $f(\cdot)$ evaluated at the initial state x_0. Analogously, we expand g in Taylor's series about x_0.

If we only take a linear approximation of g, then:

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

where G_0 is the Jacobian of g at x_0. Consequently, approximate evaluation of $d(W^T g(V\cdot))/dt$ becomes inexpensive – it requires $O(q^2)$ operations. If we also take only the constant and linear terms from expansion (2.20), and substitute those expansions in (2.18), we will obtain the following reduced order model (cf. also [14], [17], [83], and [110]):

$$\begin{cases} G_{0r}\frac{d\bar{z}}{dt} = W^T f(x_0) + A_{0r}\bar{z} + B_{0r}u \\ y = C_r^T \bar{z} \end{cases}, \tag{2.21}$$

where $A_{0r} = W^T A_0 V$ (Jacobian of f) and $G_{0r} = V^T G_0 V$ (Jacobian of g) are $q \times q$ matrices, $B_{0r} = V^T B(x_0)$, $C_r = V^T C$, and $V\bar{z} = x - x_0$. Including also the quadratic term from the expansion of f yields the following quadratic reduced order model [83][1]:

$$\begin{cases} G_{0r}\frac{d\bar{z}}{dt} = V^T f(x_0) + A_{0r}\bar{z} + \frac{1}{2}W_{0r}(\bar{z}\otimes\bar{z}) + B_{0r}u \\ y = C_r^T \bar{z} \end{cases}, \tag{2.22}$$

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

In the linearized reduced model (2.21), the reduced Jacobian matrices A_{0r} and G_{0r} are typically dense (even if the original, nonreduced Jacobians are sparse and structured), and therefore are represented explicitly. Consequently, the cost of approximately computing $W^T f(Vz)$ and $d(W^T g(Vz))/dt$ terms in model (2.21), and the cost of storing the reduced matrices G_{0r} and A_{0r} will be $O(q^2)$. For the quadratic model (2.22), this cost will equal $O(q^3)$. The cost of simulating models (2.21) or (2.22) is therefore independent of N, and keeping in mind that q is typically of order 10, it will be very low.

Nevertheless, memory and computational costs for models based on Taylor's expansions will grow exponentially with the number of nonlinear terms included. Also, already construction of the reduced order Hessian (the quadratic term) may become involved for some applications, since generating the explicit, nonreduced Hessian matrix may be difficult or impossible for large N, due to $O(N^3)$ cost of storing the matrix. In such cases implicit representation of the Hessian, or numerical approximation of the quadratic term based on directional derivative of Jacobians need to be used - cf. [16].

Consequently, although the discussed method may be extended to higher orders [83], it is limited in practice to cubic expansions. For instance, if we consider quartic expansion of order $q = 10$, then the memory storage requirement exceeds $q^5 = 100,000$ elements. In addition, the computational cost of evaluating the quartic approximation to $V^T f(V\cdot)$ is $O(q^5)$. Also, if the quartic term is

[1] An alternative formulation of the quadratic reduced order model based on Taylor's expansion is presented in [17]. According to our experience, both formulations give almost identical results.

not given in a simple analytic form, its accurate numerical evaluation will be virtually impossible.

At the same time, using only two or three terms from Taylor's expansion is often insufficient in order to adequately represent nonlinearity of a given system. Also, expansions like (2.20) are inherently local, i.e. (2.20) provides suitable approximation of $f(x)$ only in some neighborhood of the expansion point x_0, which is typically an equilibrium point. If the operating point of the original nonlinear system varies significantly from x_0, using this approximation becomes very inaccurate. Consequently, application of model order reduction approach based on Taylor's expansion of the nonlinearity about a single state of the system is limited to weakly nonlinear systems, and/or limited ranges of input signals. As shown in Chapter 6 they fail to provide accurate models for examples of systems described earlier in Section 2.2.

Apart from the Taylor series approach, another popular technique for representing system's nonlinearity is based on bilinear approximation. For description of bilinearization procedures for nonlinear systems, and reduction of bilinear realizations please refer to [6], [83], and [86]. In general, bilinearization techniques share common limitations with methods based on Taylor's expansions discussed above.

2.3.2 State-space projection techniques

A large number of different numerical techniques for constructing reduced order projection bases have been developed over the last decade (cf. Chapter 1). In this section however, we present only a single such algorithm, based on the concept of a Krylov subspace, and in its simplest version valid for LTI systems. We select this particular method, since it provides a good starting point for discussing the projection methods for nonlinear systems proposed in Chapter 3.

Krylov projection algorithms are one of the most widely used techniques for constructing reduced order bases for large-scale LTI dynamical systems (cf. Chapter 1), particularly in the electrical engineering domain. One of the main reasons for their popularity is low computational cost, and ease of generating the projection basis from the original state-space model.

Let us consider the following system:

$$\begin{cases} E\frac{dx}{dt} = Ax + Bu \\ y = C^T x \end{cases}. \tag{2.23}$$

In the most basic version of the projection algorithm we take $W = V$ (cf. equation (2.19)), where V is an orthonormal basis in the following q-th order Krylov subspace:

$$K_q(A^{-1}E, A^{-1}B) = \text{span}\left\{A^{-1}B, A^{-1}EA^{-1}B, \ldots, (A^{-1}E)^{q-1}A^{-1}B\right\}. \tag{2.24}$$

A numerically robust way of generating an orthonormal basis in the above Krylov subspace is based on the Arnoldi algorithm. A simple version of this algorithm, applying the Modified Gram-Schmidt (MGS) orthogonalization, is given on the next page.

The cost of executing the MGS-Arnoldi algorithm includes: 1) The cost of solving the system $Av = x$. (Provided A is sparse and structured, very efficient, almost-linear complexity algorithms may be applied to solve this system.); 2) the cost of the orthogonalization, which is $O(q^2N)$. This means that for sparse and structured matrices generation of the reduced basis V may be very cost-efficient.

MGS-ARNOLDI(A,B,E,q)
Input: System matrices A, E, input vector B, order of the reduced system q;
Output: Projection basis V;
(1) $v_1 = B/\|B\|_2$
(2) **for** $i = 1$ **to** q
(3) $x = Ev_i$;
(4) Solve: $Av = x$;
(5) Orthogonalize v:
for $j = 1$ **to** i
(6) $\alpha = v_j^T v$;
(7) $v := v - \alpha v_j$;
(8) Set $v_i = v/\|v\|_2$
(9) Set: $V = [v_1, \ldots, v_q]$.

Also, suppose we use basis V, such that colspan$\{V\} = K_q(A^{-1}E, A^{-1}B)$ (cf. (2.24)) to construct the reduced order model. Then, the transfer function $H_r(s)$ for this model:

$$H_r(s) = C_r^T (sE_r - A_r)^{-1} B_r,$$

where $A_r = V^T AV$, $E_r = V^T EV$, $B_r = V^T B$, and $C_r = V^T C$, will match the first q moments of the transfer function $H(s) = C^T (sE - A)^{-1} B$ for the original model. (For a proof of this fact see [33].) This property of reduced order models based on Krylov projection proved very advantageous in many engineering applications. It implies e.g. that the reduced order model will accurately model the steady-state response of the original system. In fact, reduced models using Krylov projection are typically able to approximate correctly the response of the original system for a wide frequency range. If more advanced projection schemes are applied, e.g. multipoint Krylov schemes, using different frequency expansion points, better accuracy of the reduced order models for selected frequency intervals may be achieved.

The main drawback of projection schemes based on Krylov subspace methods is, that the resulting reduced model cannot provide correct global approximation for the original model. Therefore, Krylov-based MOR methods lack of provable error bounds. Also, in general, the extracted reduced models are *not* guaranteed to preserve stability, or passivity of the original system.

Chapter 3

Trajectory Piecewise-Linear Model Order Reduction

In the previous chapter we described the main challenges associated with model order reduction for nonlinear systems. In particular, we pointed out the issue of efficient representation of system's nonlinearity, which has a substantial impact on the performance of the extracted reduced order models. In this context we briefly analyzed a method of Taylor series expansion of the nonlinearity as the simplest representation yielding reduced order models whose complexity does not depend on the original size of the system N, but only on the size of the reduced system q. This approach, however, features a number of drawbacks. On one hand, reduced order models based on Taylor's series expansion become prohibitively expensive for high series order, due to the fact that computational and memory complexities associated with those models grows exponentially with the order of the expansion. On the other hand, simple linearized reduced order models (cf. (2.21)), although computationally inexpensive, may be applied only to weakly nonlinear systems and are usually valid for only a very limited range of inputs [110], which is a simple consequence of the fact that linearization approximates a given nonlinear system only locally, around the expansion point. This leads us to proposing a representation of system's nonlinearity, and then a model order reduction scheme, based on collection of linearizations of the original nonlinear system about suitably selected states [91]–[93].

In this chapter we first describe a representation of a nonlinear system in a form of a weighted combination of linearized models from different expansion points, briefly discuss the weighting procedure, and show how state projection framework (cf. Chapter 2) may be employed in this case to obtain a cost-efficient quasi-piecewise-linear reduced order model. Then, in Section 3.2, we propose and examine a method of generating the quasi-piecewise-linear models by selecting linearization points from a state-space trajectory of the original nonlinear dynamical system. Finally, in Section 3.3, we present a procedure for fast approximate simulation of the original nonlinear system, which allows one to inexpensively generate an approximate state-space trajectory, and consequently significantly reduce the cost of extracting the reduced order model.

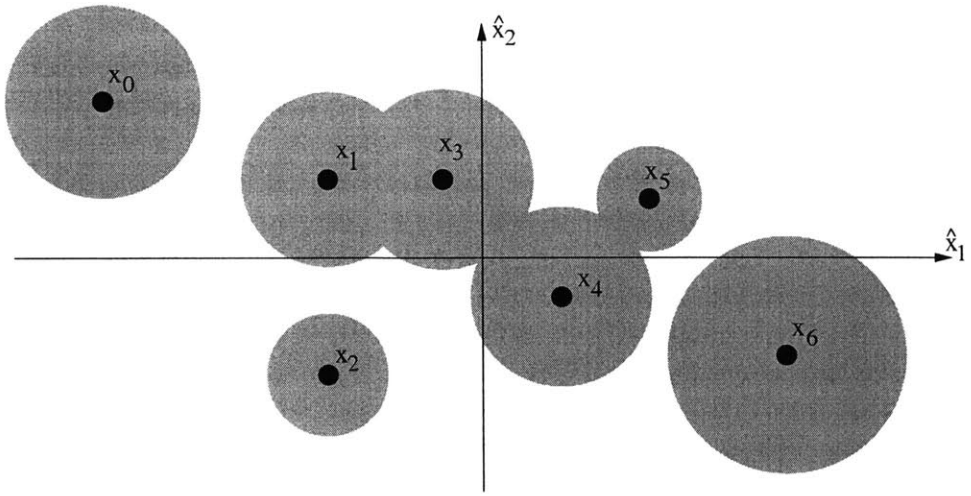

Figure 3-1: Collection of linearization points $x_0, \ldots, x_6$ in a 2D state space. For each x_i, the corresponding circle marks a region in the state space in which a linearized model about x_i 'suitably' approximates the original nonlinear system.

3.1 Quasi-piecewise-linear representation

In this study we propose to approximate the nonlinear dynamical system in form (2.1) by a suitable combination of linear models, being linearizations of the original system generated at different points in the state space. The key idea in this approach is to consider *multiple* linearizations about suitably selected states of the system, instead of relying on a single expansion about the initial or equilibrium state, in order to overcome the 'weak nonlinearity' limitation.

To start our discussion, let us assume we have generated s linearized models of the nonlinear system (2.1), with expansions about states $x_0, \ldots, x_{s-1}$:

$$\frac{d}{dt}(g(x_i) + G_i(x - x_i)) = f(x_i) + A_i(x - x_i) + B_i u, \tag{3.1}$$

where x_0 is the initial state of the system, A_i and G_i are the Jacobians of $f(\cdot)$ and $g(\cdot)$, respectively, evaluated at states x_i, and $B_i = B(x_i)$. Each of the above models approximates adequately the original nonlinear system in some neighborhood of its expansion point x_i (cf. Figure 3-1). By constructing a dynamical system which includes all those linearized models we potentially obtain a model which will provide an adequate approximation of the original nonlinear system over a larger region in its state space.

One possible approach to construct such a system is to take a weighted combination of linearized models (3.1):

$$\frac{d}{dt}\left[\sum_{i=0}^{s-1} \tilde{w}_i(x)(g(x_i) + G_i(x - x_i))\right] = \sum_{i=0}^{s-1} \tilde{w}_i(x)\left(f(x_i) + A_i(x - x_i) + B_i u\right), \tag{3.2}$$

where $\tilde{w}_i(x) \in R$ are state-dependent weights. (We assume that all the weights are nonnegative, and satisfy a normalization condition, such as e.g. $\sum_{i=0}^{s-1} \tilde{w}_i(x) = 1$, for all x, which means that in (3.2) we take a convex combination of the linearized models.)

Before we discuss the choice of the weighting procedure (cf. Section 3.1.1), and a strategy of selecting suitable linearization points x_i (cf. Section 3.2) let us consider the problem of constructing a reduced order model from system (3.2). Note that system (3.2) is still nonlinear. Nevertheless, since the nonlinearity is introduced through scalar weights, an efficient reduced order representation may be derived. Assuming we have already generated q-th order bases V and W (cf. (2.16) and (2.17)) projecting (3.2) yields:

$$\frac{d}{dt}\left[\sum_{i=0}^{s-1} \tilde{w}_i(Vz)(W^T g(x_i) + W^T G_i(Vz - x_i))\right]$$

$$= \sum_{i=0}^{s-1} \tilde{w}_i(Vz)\left(W^T f(x_i) + W^T A_i(Vz - x_i) + W^T B_i u\right). \tag{3.3}$$

Note that evaluating either left or right hand side of the above equation will be inexpensive (in a sense that it will require less than $O(N)$ operations), provided computing $\tilde{w}_i(Vz)$ is inexpensive, and $s \ll N$. In fact, instead of weights $\tilde{w}_i$ which take a nonreduced state vector as an argument, we may use a new set of weighting functions $w_i(z)$, which depend on the reduced order state vector z. This will typically reduce the cost of evaluating the set of weights from $O(sN)$ to $O(sq)$. (For a broader discussion see the following section.) Replacing $\tilde{w}_i(Vz)$ with $w_i(z)$ in (3.3) yields the following reduced order model:

$$\begin{cases} \frac{d}{dt}((\sum_{i=0}^{s-1} w_i(z)G_{ir})z + \delta \cdot w(z)) = (\sum_{i=0}^{s-1} w_i(z)A_{ir})z + \gamma \cdot w(z) + (\sum_{i=0}^{s-1} w_i(z)B_{ir})u \\ y = C_r z \end{cases}, \tag{3.4}$$

where:

$$G_{ir} = W^T G_i V, \quad A_{ir} = W^T A_i V, \quad B_{ir} = W^T B_i, \quad C_r = C^T V,$$

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

$$\delta = \left[W^T(g(x_0) - G_0 x_0), \ldots, W^T(g(x_{s-1}) - G_{s-1}x_{s-1})\right],$$

and $w(z) = [w_0(z) \ldots w_{s-1}(z)]^T$ is a vector of weights, such that $\sum_{i=0}^{s-1} w_i(z) = 1$ for all z. Assuming that the cost of computing $w_i(z)$ is $O(q)$, and the number of system inputs (M), and outputs (K) is relatively small, i.e. $M, K \leq q$, the cost of evaluating any of the terms in model (3.4) will require at most $O(sq^2)$ operations. Note that for the weighting procedure discussed in the following section, typically only one or two of w_i's are non-zero at a time, and consequently model evaluation time is rather insensitive to s. In any case, if the number of linearization points is kept small, in a sense that $s \approx q$, then the complexity associated with (3.2) will not exceed the complexity of a quadratic reduced order model (2.22) (cf. Section 2.3.1). This means that system (3.4) is characterized both by reduced order and reduced complexity, as compared to the original nonlinear system, and therefore provides an attractive framework to construct reduced models for nonlinear dynamical systems. Also, as shown in Section 3.1.2, specific time-stepping schemes may be used with model (3.4) in order to perform inexpensive time-domain simulations.

3.1.1 Discussion of the weighting procedure

In this Section we discuss the problem of computing weights $\tilde{w}_i$ and w_i used in models (3.2) and (3.4). Our goal is to find procedures for computing a distribution of weights, such that system's nonlinearity will be adequately approximated by the resulting convex combination of the linearized models in some neighborhood of the linearization points. Furthermore, the cost of computing the weights needs to be low. In particular, we impose the complexity of computing the collection of weights w_i for the reduced order model not to exceed $O(sq)$, where s is the number of linearized models, and q is the order of the linearized system.

Let us first consider the problem of approximating e.g. $f(x)$ from equation (2.1) with a linearized model from a given state x_i: $\tilde{f}(x) = f(x_i) + A_i(x - x_i)$, where A_i is the Jacobian of f evaluated at x_i. Then:

$$f(x) - \tilde{f}(x) = \left(\int_0^1 (1-s) W(x + s(x - x_i)) ds \right) (x - x_0) \otimes (x - x_0),$$

where W is the Hessian of f. This clearly implies that, for $x \in B(x_i, \varepsilon)$, where $\varepsilon > 0$

$$\|f(x) - \tilde{f}(x)\| \leq \frac{1}{2} \sup_{x \in B(x_i, \varepsilon)} \|W(x)\| \|(x - x_0)\|^2. \tag{3.5}$$

The above means that, provided the norm of Hessian is bounded in the neighborhood of x_i, the error due to a linear approximation to f grows at most quadratically with the growing distance between the expansion point x_i, and the evaluation point $x \in B(x_i, \varepsilon)$. It also suggests that one may use information on the distances between x and x_i to compute the distribution of weights. If information on the norms of Hessian in the neighborhoods of the linearization points x_i is not available, we may use an intuitive heuristics following from (3.5), which says that the 'dominant' linear model used in (3.2) should be the one corresponding to expansion point x_j which is the closest to the current state x of the system. In other words, we take

$$\tilde{w}_i = \tilde{w}_i(\|x - x_0\|, \ldots, \|x - x_{s-1}\|), \tag{3.6}$$

such that, if $j = \text{argmin}_{i=0,\ldots,(s-1)} \|x - x_j\|$, then $\tilde{w}_j = \max_{i=0,\ldots,(s-1)} \{\tilde{w}_i\}$.

The question which arises now is how to construct weights w_i for the reduced order system (3.4). Using (3.6) one could take:

$$w_i = \tilde{w}_i(\|Vz - x_0\|, \ldots, \|Vz - x_{s-1}\|). \tag{3.7}$$

However, evaluation of w_i would then require computing Vz, which involves $O(Nq)$ operations, and would be too costly. In order to lower this cost, instead of using distances $\|Vz - x_i\|$ in the nonreduced state space, one may consider the distances between the reduced order state z, and projections z_i of the linearization points x_i onto the reduced state space. The projections z_i may be taken as least-square approximations of x_i in basis V, i.e.:

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

where V is an orthonormal projection basis. One should note that when we move from the original

to the reduced order state space the geometric relations may be deformed. In particular, relative distances between subsequent linearization points may be different in the original and reduced spaces. In order to overcome this problem one can make sure that all linearization points can be represented exactly in the reduced order space, i.e. $x_i \in \text{colspan}\{V\}$, for all i. Then, for any z

$$\|Vz - x_i\|_2 = \|Vz - Vz_i\|_2 = \left((Vz - Vz_i)^T (Vz - Vz_i)\right)^{1/2} = \left((z - z_i)V^T V(z - z_i)\right)^{1/2} = \|z - z_i\|_2.$$

That means that the Cartesian distances between linearization points x_i and x's representable in the reduced order space (i.e. such that $x = Vz$ for some z) are preserved after orthonormal projection.

There may be various weighting procedures which select the 'dominant' linearized model based on the information about distances $\|z - z_i\|_2$. Below we present one which proved efficient for the considered application examples:

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

where β is a positive constant. In our implementation of the algorithm we took $\beta = 25$. In this way we obtain a weighting procedure such that the distribution of weights changes rather rapidly as the current state z evolves in the state space, i.e. once e.g. z_j becomes the point closest to z, then weight w_j almost immediately becomes 1. This provides a rationale for referring to model (3.4) as to a *piecewise-linear* reduced order model of nonlinear system (2.1) (cf. Figure 3-2 for sample shapes of the weights computed with the discussed weighting procedure). However, please note that model (3.4) is not piecewise-linear in a strict sense, since weights w_i, although rapidly changing, still remain continuous and smooth functions of their arguments.

As already mentioned, the above weighting procedure provides only one possible choice. Many other options are available, although in general it seems challenging to determine which one will yield most adequate approximations for a broad class of nonlinear functions, especially if available information consists only of locations of the expansion points x_i, and the corresponding data for the linearized models (i.e. Jacobians and values of $f(x_i)$).

One of more specific questions refers, e.g. to the value of coefficient β in the above weighting procedure. If β is large then the weights change relatively rapidly, and otherwise they have mild slopes. If we consider a quadratic nonlinearity, larger values of β seem to be more adequate. This is motivated by the following observation. Let us take a 1D function in form $f(x) = ax^2 + bx + c$, and approximate this function by a combination of two linearized models from points x_0 and x_1:

$$\tilde{f}(x) = w_0(x)(c + bx + 2ax_0x - ax_0^2) + w_1(x)(c + bx + 2ax_1x - ax_1^2),$$

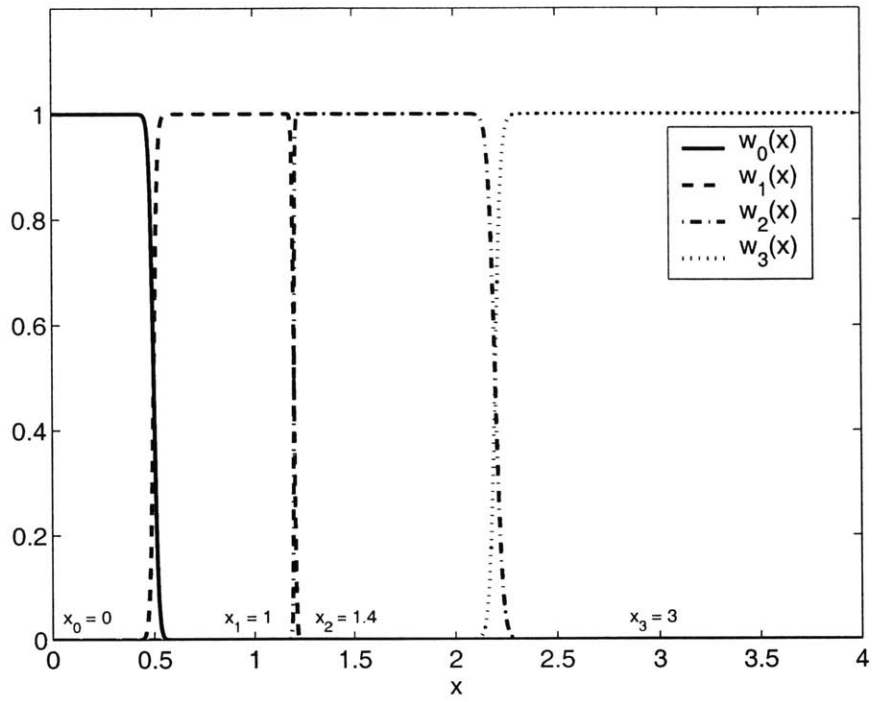

Figure 3-2: Shapes of the weights for a sample distribution of 4 linearization points in 1D.

where $w_0(x) + w_1(x) = 1$, for all x. Then the error between f and $\tilde{f}$ equals:

$$e(x) = f(x) - \tilde{f}(x) = aw_0(x)(x - x_0)^2 + aw_1(x)(x - x_1)^2.$$

We may now consider two extreme choices of the weights – the first one:

$$w_0(x) = \begin{cases} 1 & x \in [x_0, (x_0 + x_1)/2] \\ 0 & x \in [(x_0 + x_1)/2, x_1] \end{cases},$$

and $w_1(x) = 1 - w_0(x)$, and the second:

$$w_0(x) = \frac{x_1 - x}{x_1 - x_0}, \quad w_1(x) = \frac{x - x_0}{x_1 - x_0}.$$

If we now compute $E = \int_{x_0}^{x_1} e(x)dx$ then in the first case we will obtain $E = a(x_1 - x_0)^3/12$, and in the second case – $E = a(x_1 - x_0)^3/6$. This indicates that using rapidly changing weights will in this case reduce the approximation error. Clearly, for different sorts of nonlinearities the situation may be substantially different.

Apart from the case of quadratic nonlinearities, in which rapidly changing weights may be advantageous, a more general argument in their favor may also be given. Weights generated using the algorithm presented above lead to quasi-piecewise-linear approximations of the original nonlinear function, that is 'most of the time' model (3.4) actually reduces to a certain linear model. This may allow one to predict or control more easily some properties of that model. For example, if we know that all the Jacobians A_{ir} are stable (Hurwitz) matrices, then in the regions where only a

single weight is nonzero, the "A" matrix for system (3.4) is clearly Hurwitz. On the other hand, in the regions with multiple nonzero weights, the associated "A" matrix may not be stable, since a convex combination of stable matrices may not be a stable matrix. This means that those regions may potentially give rise to undesirable system's instabilities. Nevertheless, note that the discussed procedure still does not guarantee stability of the resulting quasi-piecewise-linear model.

The weighting algorithm discussed in this section is a simple heuristic with limited justification. Yet, it requires no extra knowledge on the system, and as shown in Chapters 6 and 7, provides an effective and efficient approach for the considered application examples.

A slightly more sophisticated weighting algorithm may be developed if norms of the Hessian of f may be computed or estimated during model extraction. Suppose we are able to compute norms of W (Hessian of f) at the linearization points:

$$\tilde{n}_0 = \|W(x_0)\|_2, \tilde{n}_1 = \|W(x_1)\|_2, \ldots, \tilde{n}_{s-1} = \|W(x_{s-1})\|_2.$$

We may define $n_i = \tilde{n}_i / (\max_{j=0,\ldots s-1}\{\tilde{n}_i\})$. Then, instead of distances $d_i = \|z - z_i\|_2$ which serve to compute the weights w_i, we may take scaled distances $\hat{d}_i = (n_i)^{1/2} d_i$, for $i = 0, \ldots, s-1$, and plug them into the weighting procedure presented above. By selecting such scaled distances (which is motivated by formula (3.5)), we include information on the curvature of f, which determines the range of applicability of a given linearized model, into the weighting procedure. Unfortunately, very often computation of norms of the Hessian of f is not possible, or too expensive. In this case the coefficients $\tilde{n}_i$ may be taken as e.g. lower bounds for $\|W(x_i)\|_2$, which in turn can be approximately computed using directional derivatives of the Jacobians of f:

$$\tilde{n}_i = \frac{\|(J(x_i + \varepsilon(x - x_i)) - J(x_i)) \cdot (x - x_i)\|_2}{\varepsilon \|x - x_i\|_2^2}, \tag{3.8}$$

where J is the Jacobian of f, ε is a small positive number (e.g. $\varepsilon = 1 \cdot 10^{-6}$), and x is a state different than x_i (e.g. it may be taken as a state near x_i, taken from a trajectory of system (2.1) passing through x_i). One may note that computing the above estimates is straightforward, and can be easily performed during extraction of the reduced order model.

We have tested this approach for the example of the micromachined switch, described in Section 2.2.2. We have generated a quasi-piecewise-linear reduced order model (in form (3.4)) of order $q = 29$, consisting of 23 linearized models located on a trajectory of system (2.1) corresponding to a step 9-volt input voltage (cf. Section 3.2 for description of model generation strategy). Then, we tested the discussed reduced order model for an 8.5-volt step input voltage. We computed the relative error in the output (which is the deflection of the center of the beam) as: $e = \|\underline{y} - \underline{y}_r\|_2 / \|\underline{y}\|_2$, where $\underline{y}_r = [y_r(0), y_r(\Delta T), \ldots, y_r(T)]$ is a vector of outputs at subsequent timesteps ($\Delta T = 0.001$ms, and $T = 1$ms) computed with the reduced order model, and y is a respective reference output computed with the full order nonlinear model. If we used regular distances d_i to compute weights with the procedure described above, the error e equaled 3.5%. If scaled distances $\tilde{d}_i$ were used, this error was lowered to 2.4%. In order to find scaling coefficients we used estimates of the Hessian norms computed with (3.8). Although, in this case using Hessian norm estimates (3.8) proved effective, reliablity of this approach is not yet well established.

3.1.2 Time stepping schemes for quasi-piecewise-linear models

One may note that, since weights w_i in (3.4) depend on z, this system still consists of nonlinear ODEs. Consequently, in general, time domain integration of (3.4) would require solving a nonlinear system of equations at each step. If we apply backward Euler scheme, finding the solution for each subsequent time step requires a few Newton iterations, which in turn involves computing Jacobians of w_i, and solving a linear system at each iteration.

We have found that very often, instead of a strict backward Euler scheme, a simplified time stepping method can be used. The idea is to first compute the weight distribution for an estimate of z to be computed at a specific state. Then, we can substitute the fixed values of the weights to (3.4), and solve a linear system to find z. Suppose we want to compute z at timestep t_j. This can be achieved by using the following procedure:

1. Compute an estimate z^e for $z(t_j)$. To this end one may use any of the methods for finding an initial guess for Newton's algorithm. The two simplest approaches include taking $z^e = z(t_{j-1})$, or using linear extrapolation:

$$z^e = z(t_{j-1}) + (z(t_{j-1}) - z(t_{j-2}))\frac{t_j - t_{j-1}}{t_{j-1} - t_{j-2}}.$$

2. Find $w_0(z^e), \ldots, w_{s-1}(z^e)$.

3. Substitute the above values of weights to the first of equations (3.4):

$$\frac{d}{dt}((\sum_{i=0}^{s-1} w_i(z^e)G_{ir})z + \delta \cdot w(z)) = (\sum_{i=0}^{s-1} w_i(z^e)A_{ir})z + \gamma \cdot w(z) + (\sum_{i=0}^{s-1} w_i(z^e)B_{ir})u.$$

   The above equation becomes now a linear ODE.

4. Apply any discrete time-stepping scheme, and find z by solving a single linear system of equations.

Please note that the above scheme is substantially less expensive than e.g. a full backward Euler scheme, since it does not require computing derivatives of the weighting functions, and provides a solution in a single linear solve (single Newton iteration). Also, in the case when solution z obtained from the linear solve is in a very different region, from the one predicted by z^e, one may shrink the timestep Δt, and repeat the computations. This will prevent problems related to large timestep instability, similar to the ones observed for the forward Euler scheme. Summing up, we have used the time-stepping algorithm extensively, and found it very effective for all the considered application examples.

3.2 Generation of TPWL models

So far we have not discussed the how to generate quasi-piecewise-linear models given by (3.4). More specifically, we have not yet answered the two fundamental questions: 1) How to select the collection of linearization points x_i? 2) How to construct the reduced order projection bases V and W? (cf. equation (3.4)). The following sections address those questions in detail.

3.2.1 Selecting linearization points through simulation

We start our discussion on selecting a collection linearization points for quasi-piecewise-linear models, given by equations (3.4) with repeating a basic fact that linearization of f from state x_i is an accurate approximation for the original nonlinear function at some given state x, provided x is 'close enough' to x_i, i.e. $||x-x_i|| < \varepsilon$, or x lies within an N-dimensional ball of radius ε and centered at x_i. Consequently, in order to obtain a good global approximation of f, it is obviously desirable to cover the entire N-dimensional state space with such balls, thereby assuring that any state is within ε of a certain linearization point (and a corresponding linearized model). The problem is that the number of balls will grow exponentially with order of the state space N. For example, the number of radius 0.1 balls required to fill a 1000-dimensional unit hypercube will equal roughly 10^{1000}. Generating and storing in computer's memory such a large number of models would be inefficient, if not impossible.

Since it is computationally infeasible to cover the entire N-dimensional state space with suitable linear models, we propose to generate a collection of models along a single, fixed trajectory of the system. (The idea of using a collection of linearized models along e.g. an equilibrium manifold or a given trajectory has also been used in design of gain-scheduled controllers for nonlinear systems – cf. [96], [106].) We generate this trajectory by performing a single simulation of the nonlinear system for a fixed 'training' input. (Instead of an exact simulation of the original nonlinear system we may perform an approximate, fast simulation – cf. Section 3.3.) Given a training input signal $u(t)$, initial state x_0, and a finite number of models to be generated s ($s > 1$), the proposed algorithm for generating the collection of linearized models may be summarized in the following steps:

1. Generate a linearized model about the initial state x_0, and set $i = 0$;

2. Simulate the nonlinear system while

$$\min_{0 \le j \le i} \left(\frac{||x - x_j||}{||x_j||} \right) < \delta,$$

   for some $\delta > 0$, i.e. while the current state x is close enough to any of the previous linearization points;

3. Generate a new linearized model about $x_{i+1} = x$, and set $i := i + 1$;

4. If $i < (s-1)$ return to step 2.

This procedure is also depicted in Figure 3-3. Provided the weighting algorithm described in Section 3.1.1 is used, the above procedure generates a quasi-piecewise-linear model of the original nonlinear system in form (3.4), which from now on we will call a Trajectory PieceWise-Linear (TPWL) model.

It should be stressed at this point that the quasi-piecewise-linear approach resulting from the discussed procedure is different from methods presented e.g. in [23] or [46], where piecewise-linear approximations of individual elements of a circuit (e.g. diodes or transistors) are considered, and a very large collection of linear models is used. In our algorithm, piecewise-linear approximation applies to a trajectory of the *entire* nonlinear system, and therefore the number of linearized models may be kept small.

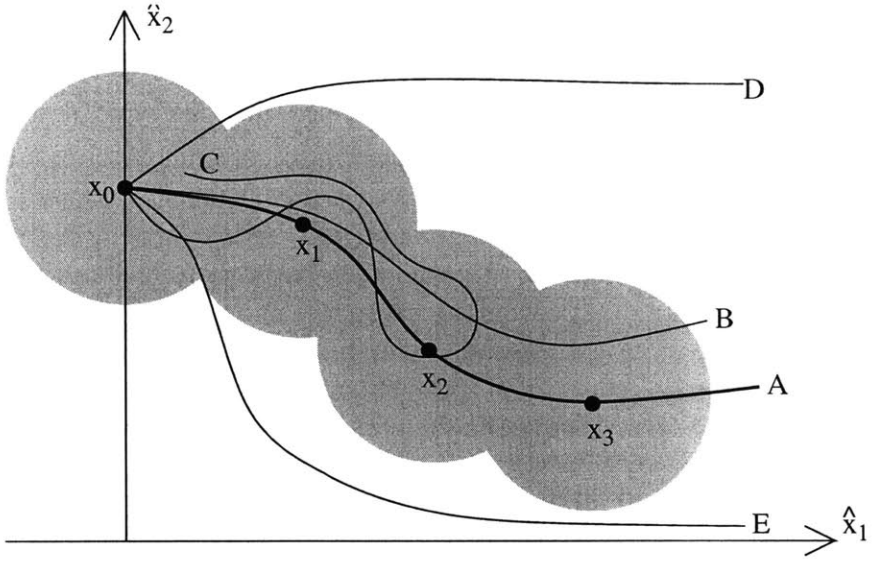

Figure 3-3: Generation of the linearized models along a trajectory of a nonlinear system in a 2D state space. Trajectory 'A' is called the 'training' trajectory.

As illustrated in Figure 3-3, the procedure proposed above allows one to 'cover with models' only the part of the state space located along the 'training' trajectory (curve A). Let us assume that the reduced order model (3.4) is composed of linear models generated along this trajectory. If a certain system's trajectory, corresponding to a given input signal u, lies within the region of the state space covered by these models (or, in other words, stays sufficiently close to all linearization points), we expect that the constructed piecewise-linear model (3.4) will suitably approximate the input/output behavior of the original nonlinear system (cf. curves B and C).[1] It should also be stressed at this point that, although the considered trajectory stays close to the states visited by the 'training' trajectory in the state space, the corresponding input signal can be very different in terms of dynamics and frequency content from the 'training' input. In other words, we may apply a TPWL model for inputs which are significantly different from the 'training' input, provided the corresponding trajectories stay in the region of the state space covered by the linearized models (cf. curve C and results in Chapter 6). This case is also illustrated in Figure 3-4, which shows computational results for the example of a micromachined switch (cf. Section 2.2.2). This figure presents the system response to a 7-volt step input voltage, computed with a TPWL reduced model of order $q = 41$, generated with an 8-volt step input training voltage. (The model was extracted using the fast simulation algorithm proposed in Section 3.3.) We should stress that, in fact, the input to the system is the squared input voltage $u(t) = v^2(t)$. One may note that the obtained output signal approximates very accurately the output signal computed with the full nonlinear model of the device (the curves on the graphs overlap almost perfectly). In this case the TPWL model provides significantly more accurate results than the linear or quadratic models based on expansions about the single state (the initial state of the system).

A different situation occurs when the input signal causes the trajectory to leave the region

[1]The additional rationale for this observation is that in typical situations the dimensions of observable and controllable spaces of a dynamical system are much lower than the dimension of its state space. (This is expected to be true for the examples of nonlinear dynamical systems presented in Section 2.2.)

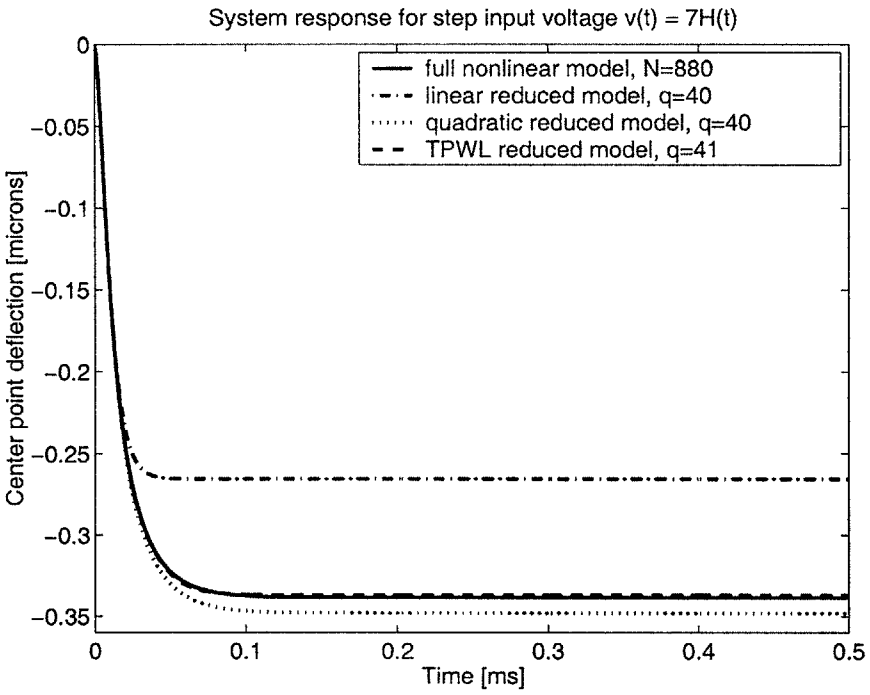

Figure 3-4: Comparison of the system response (micromachined switch example) computed with linear, quadratic, and TPWL reduced order models ($q = 40$ and $q = 41$) to a step input voltage $u(t) \equiv 7$. The TPWL model was generated for an 8-volt step input voltage.

covered by the linearized models (cf. curves D and E in Figure 3-3). Then, the TPWL model (3.4) will most likely *not* provide significantly better approximation to the nonlinear system than a simple linear reduced model (2.21). This situation has been illustrated in Figure 3-5. Due to a significant difference in scales (amplitudes) between the 'training' input ($u(t) \equiv 7^2$) and the testing input ($u(t) \equiv 9^2$) the piecewise-linear model is no longer able to reproduce accurately the response of the nonlinear system. Now, if we generate a TPWL model with a 9-volt training input (cf. Figure 3-6), then this model is able to reproduce accurately the nonlinear response. One should note that in this case the TPWL is able to accurately model the dynamics of a strongly nonlinear pull-in effect (the beam is pulled down to the substrate), which is of particular importance in applications [40]. One may note from the graph that the linear model is not able to reproduce this phenomenon at all, while the quadratic model is unable to reproduce the correct dynamics. Still, this example shows that if a TPWL model is to be used for inputs with very different scales, one should consider more complicated schemes of generating the linearized models, based e.g. on multiple training inputs.

There is another important issue concerning the above procedure for selecting linearization points from a state-space 'training' trajectory of the original system. In order to ensure appropriate quality of approximation of f, and to be able to reproduce nonlinear effects in the behavior of a dynamical system, the linearization points should be changed 'frequently enough' during the proposed piecewise-linear simulation. This is controlled by a constant parameter δ in the procedure presented above. In our first approach this parameter was set arbitrarily. However, it was found that a proper choice of δ depends significantly e.g. on the amplitude of the input signal $u(t)$.

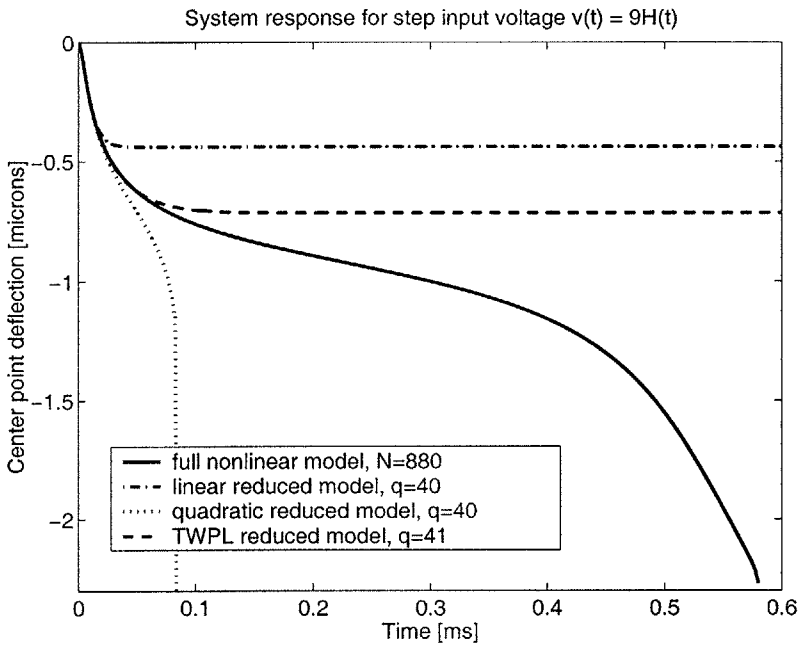

Figure 3-5: Comparison of the system response (micromachined switch example) computed with linear, quadratic, and TPWL reduced order models ($q = 40$ and $q = 41$) to a step input voltage $u(t) \equiv 9$ ($t > 0$). The TPWL model was generated for a 7-volt step input voltage.

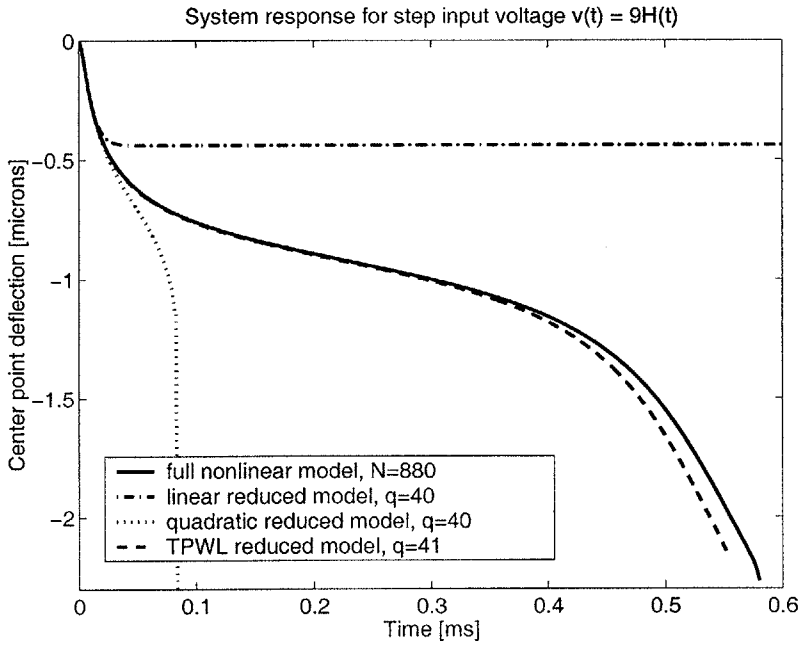

Figure 3-6: Comparison of the system response (micromachined switch example) computed with linear, quadratic, and TPWL reduced order models ($q = 40$ and $q = 41$) to a step input voltage $u(t) \equiv 9$ ($t > 0$). The TPWL model was generated for a 9-volt step input voltage.

Therefore, in order to achieve better robustness of the discussed procedure the choice of δ should be as automatic as possible. A simple procedure for determining an appropriate value of δ in a less arbitrary way is the following. First, for a given input signal, we perform a reduced order simulation of the linearized dynamical system, with linearization about the initial state, to find the final (steady state) vector x_T. Although, in most cases, x_T will *not* be the correct steady state for our nonlinear dynamical system, it will give us information about the scale of changes between the initial and the final state:

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

(If $x_0 = 0$ we may take $d = ||x_T||$.) It is clear that in order to capture any nonlinear effects one has to select the value of δ such that $\delta < d$. We have found that in practical situations it is usually enough to select $\delta = d/5$ or $\delta = d/10$. Still, also this approach suffers from a certain level of arbitrariness, and cannot be used if a given input signal does not lead the linearized dynamical system to a steady state. In the following chapter we discuss a procedure for selecting linearization points based on error estimates, which provides a more grounded approach to this problem.

Apart from a procedure for selecting linearization points, the other necessary element for generating a Trajectory Piecewise-Linear (TPWL) reduced order model is a method for constructing reduced order projection bases V and W. The following sections describe some of the possible approaches, focusing on Krylov subspace algorithms.

3.2.2 Constructing the reduced order basis – a simple algorithm

The simplest approach for generating the reduced order basis V employs Krylov subspace projection algorithm well known from linear MOR, and described in Section 2.3.2. This algorithm may be used also in the nonlinear case, particularly if we start our simulation at an equilibrium point of the original system (which commonly occurs in applications). Then, we generate a basis V in a suitable Krylov subspace corresponding to a linearized model from the initial, equilibrium state of the system x_0 ($f(x_0) = 0$). Furthermore, since we start the simulation at state x_0, we also incorporate this state to the projection basis. Finally we take $W = V$. The procedure for constructing the reduced order basis $V = [v_1, \ldots v_q]$, where $v_i \in R^N$, is summarized in the following three steps:

1. Generate linearization of dynamical system (2.1) about the initial, equilibrium state x_0:

$$\begin{cases} G_0 \frac{dx(t)}{dt} = A_0(x - x_0) + B_0 u(t) \\ y(t) = C^T x(t) \end{cases}, \tag{3.9}$$

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

$$K_l(A_0^{-1}G_0, A_0^{-1}B_0) = \text{span}\{A_0^{-1}B_0, \ldots, (A_0^{-1}G_0)^{l-1}A_0^{-1}B_0\}, \tag{3.10}$$

   using the Arnoldi algorithm [110] (or block Arnoldi algorithm [88] if the number of inputs $M > 1$).

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

3. Take V as a union of $\tilde{V}$ and v_{lM+1}: $V = [\tilde{V}; v_{lM+1}]$.

The final size of the reduced basis V equals $q = lM + 1$. Note that the above choice of basis V ensures that l moments of the transfer function for the *reduced order* linearized model match l moments of the transfer function for the original linearized model (3.9) [34], [83]. Moreover, the last two steps of the algorithm ensure that the term corresponding to the initial condition in the transfer function for the full order linearized model[2] will be exactly matched by a respective term in the transfer function for the reduced order linearized model, provided the simulation starts also at x_0. Exact representation of the initial state also ensures that we may correctly start the reduced order simulation of the nonlinear system. Note also that if $x_0 \in \text{colspan}(\tilde{V})$, then clearly steps 2 and 3 become unnecessary.

The above algorithm provides one of the least expensive approaches for generating a reduced order basis (with complexity of $O(q^2N) + O(qN)$ if A_0 is sparse and structured). However, in the nonlinear context, its main drawback is that it uses only the linearized model from the initial, equilibrium state of the system. This may adversely influence quality of the resulting TPWL reduced model, and force increment in size q of that model in order to achieve satisfactory accuracy.

3.2.3 Constructing the reduced order basis – an extended algorithm

The simple algorithm for generating the reduced order basis presented above, which constructed a Krylov subspace only at the initial, equilibrium state of the system, can be extended in order to include in the reduced order basis Krylov subspaces corresponding to other linearization points located along the training trajectory [92]. This extension of the reduced basis is motivated by the following fact. As already mentioned in the previous chapter (Section 2.3.2), if we use basis V spanning Krylov subspace (3.10) to construct a linear reduced order model, then the first l moments of the transfer function for this reduced order linear model match the first l moments of the transfer function for the original linearized model (3.9). Consequently, important dynamical features of the nonreduced linearized model are preserved by the reduced order linear model [34], [83]. Since in the TPWL model we use a collection of reduced order linear models, taking a union of bases in Krylov subspaces corresponding to subsequent linearized models as a reduced order basis V will ensure that, for every resulting reduced order linear model, the first few moments of its transfer function will match the first few moments of the transfer function of the corresponding *nonreduced* linearized model. Consequently, we may expect that important dynamical properties for each of the linearized models will be preserved after the projection process.

Two important technical details arise during the construction of the union of bases at different linearization points. Firstly, if we linearize about a non-equilibrium state and/or start our simulation at a non-equilibrium state, the following terms contribute to the dynamics of the system: 1) $B_i u$ input term, 2) $(f(x_i) - A_i x_i)$ term (which can be treated as an additional input), where x_i is a given linearization point (cf. (3.12)). Note that if x_i is not an equilibrium (or if it is an equilibrium, but the initial condition does not equal x_i), the second term no longer corresponds to a simple shift in the output. Consequently, bases corresponding to two different Krylov subspaces $K_l(A_i^{-1}G_i, A_i^{-1}B_i)$, and $K_l(A_i^{-1}G_i, A_i^{-1}(f(x_i) - A_i x_i))$ (cf. (3.10)) need to be constructed. Secondly,

[2] If simulation is started at x_0 this term equals $C^T x_0/s$, and corresponds to a constant shift in the output.

we need to eliminate redundant (e.g. almost parallel) vectors (which may appear after we take a union of a collection of bases) from the reduced basis. To this end we may apply e.g. singular value decomposition (SVD) algorithm, and discard vectors corresponding to the smallest singular values. Using the above motivation, we developed an extended algorithm for generating the projection basis, which may be summarized in the following steps:

1. Set $V_{agg} = [\,]$, $i = 0$;

2. Repeat until the 'training' simulation is completed:

   (a) Consider linearization of dynamical system (2.1) about state x_i:

   $$\begin{cases} G_i \frac{dx(t)}{dt} = f(x_i) + A_i(x(t) - x_i) + B_i u(t) \\ y(t) = C^T x(t) \end{cases}, \tag{3.11}$$

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

   $$\text{span}\{V_1\} = K_{q_i}(A_i^{-1}G_i, A_i^{-1}B_i),$$
   $$\text{span}\{V_2\} = K_{q_i}(A_i^{-1}G_i, A_i^{-1}(f(x_i) - A_i x_i)),$$

   using the Arnoldi algorithm [110];[3]

   (b) Take $\tilde{V} = [V_1\ V_2\ x_i]$;

   (c) Set $V_{agg} := [V_{agg}\ \tilde{V}]$;

3. Orthogonalize the columns of the aggregate basis V_{agg} using SVD algorithm and construct the final reduced order basis V which includes orthogonalized columns of V_{agg} corresponding to singular values larger than some $\varepsilon_2 > 0$.

One may note that the above method is more expensive than the simple algorithm presented earlier, since we need to generate two orthogonal bases at every linearization point. To this end we need to perform LU factorization of Jacobians A_i at every linearization point x_i. In the simple approach, presented in Section 3.2.2, this had to be done only once. The extended algorithm also requires an additional SVD step.

Nevertheless, since we generate a 'richer' basis we expect that it will more adequately approximate the original state space. One may argue though that the above method may generate models of significantly larger order than the simple algorithm. In fact the situation is often the opposite. As shown in Section 7.2, the extended basis generation algorithm has potential to extract suitable, accurate reduced bases with a lower order than the simple algorithm which uses single linearization about the initial state.

[3] One may also use block Arnoldi algorithm [88] to construct a basis in Krylov subspace $K_{q_i}(A_i^{-1}G_i, A_i^{-1}[B_i|(f(x_i) - A_i x_i)])$, spanning the same range of vectors as the union of bases V_1 and V_2.

3.2.4 Beyond Krylov-based reduction schemes

In this study we focus on reduction approaches based on Krylov subspace methods, which dominated MOR for linear systems in electrical engineering applications. Nevertheless, also methods based on truncated balanced realization (TBR) have recently gained much interest in the engineering community, due to their superior accuracy, as compared to Krylov-based methods, guaranteed error bounds, and stability preservation property for the linear reduced order models.

Truncated balanced realization algorithm can also be used to generate reduced order bases for TPWL models in form (3.4) for nonlinear dynamical systems. Similarly as in the case of Krylov-based methods, a few different strategies toward generating the projection bases V and W can be considered.

In the simplest approach we may apply a standard TBR algorithm to obtain a balancing transformation for the linearization of system (2.1) at the initial state x_0. In an extended approach we may include subsequent linearization points in the projection bases, as well as generated truncated balancing transformations for linearizations of system (2.1) at different states, located along the training trajectory. In this way we clearly obtain a richer basis, which includes 'most controllable' and 'most observable' parts of the state space for an entire collection of linearized models which form a TPWL model. In order to perform the projection the multiple bases corresponding to different linearization points need to be aggregated into a single basis using a biorthonormalization algorithm.

Due to high complexity of the TBR algorithm ($O(N^3)$, where N is the original size of the system), and problems with ill-conditioning of Lyapunov equations, reduction algorithms based solely on TBR are limited in practice to systems with a few hundred unknowns. In order to allow effective and efficient reduction of larger systems one may use a two-step hybrid reduction, first applied for linear systems in [45]. In this reduction approach a standard Krylov subspace method is first used to reduce the original TPWL model to a medium-sized model (e.g. of order $q_i \approx 100$), and then TBR is used to further compress that model. Detailed description of TBR-based projection in the context of trajectory piecewise-linear model for nonlinear dynamical systems may be found in [107]. The results presented in the quoted paper indicate that applying TBR-based projection, or hybrid two-step projection may lead to reduced models with superior accuracy as compared to models based solely on Krylov subspace reduction.

3.3 Fast generation of TPWL models – approximate simulation algorithm

One may note that the proposed method of generating TPWL models of nonlinear dynamical systems requires performing simulation of the original nonlinear system (2.1), which may be very costly, due to the original size of the problem. In order to reduce the computational effort associated with reduced model extraction, we note that it is unnecessary to compute the *exact* trajectory for the 'training' input in order to generate a collection of linearized models. In fact it suffices to compute an *approximate* trajectory, and take linearization points from that trajectory. As long as the approximate trajectory stays close to the exact trajectory, the collection of linearized models corresponding to points from an approximate trajectory will 'cover' the suitable region in the state space. In order to find an approximate trajectory of the original nonlinear dynamical system (2.1)

we may perform a much faster approximate simulation. Our idea is to use reduced order linearized systems instead of the full order nonlinear system to perform the simulation. Note that in the algorithm from Section 3.2.1, we generate subsequent linearized models as we go along the training trajectory. Therefore, at a given linearization point we may generate a linearized system, reduce its order, and then simulate it to reach to the next linearization point. This idea is summarized in the following steps of a fast simulation algorithm:

1. Take $i = 0$, set x_0 to be the initial state;
2. While $i < s$ do:
   (a) Using projection basis V construct a reduced-order model of dynamical system (2.1), linearized about state x_i:

$$\begin{cases} V^T G_i V \frac{dz}{dt} = V^T A_i V z + V^T (f(x_i) - A_i x_i) + V^T B_i u \\ y = C^T V z \end{cases}, \tag{3.12}$$

   where z is a reduced order approximation of state vector x ($x \approx Vz$). This step requires computation of the Jacobians A_i and G_i (at $x = x_i$) in the non-reduced state space;
   (b) Simulate reduced order linear dynamical system (3.12), i.e. compute $z(t)$ for subsequent time steps $t = t_j$, while the state $Vz(t_j)$ is close enough to the initial state x_i ($x_i \neq 0$), i.e. when:

$$||Vz(t_j) - x_i||/||x_i|| < \delta,$$

   where δ is an appropriately selected constant (cf. discussion at the end of Section 3.2.1);
   (c) Take the next linearization point $x_{i+1} = Vz(t_j)$, $i := i + 1$.

Note that in the above algorithm the projection basis V can be picked in a number of different ways. For instance, it may be generated only once, at the initial state of the system, using the approach from Section 3.2.2. Alternatively, at each linearization point we may generate a different basis e.g. in a Krylov subspace corresponding to a linearized model at this point. Clearly, the second approach will typically yield more accurate simulations, at an additional cost of generating multiple projection basis.

As shown in Chapter 7 using the above approximate simulation algorithm can dramatically improve performance of the MOR procedures. It has also been found that it can serve one as an accurate standalone simulation tool. Nevertheless, special care should be taken when using the discussed approach. For instance, it may happen that the affine function $V^T A_i V z + V^T (f(x_i) - A_i x_i$ from the right hand side of the first of equations (3.12) has a zero close to the linearization point x_i. Then, for a zero input signal, the linearized model will tend to the zero of the mentioned affine function, rather than the actual equilibrium of the nonlinear system. Consequently, although the operating state will not change significantly, the simulation error will rapidly grow. Therefore, in order to use the above procedure as a reliable simulation tool, one needs to be able to estimate the error due to approximations introduced by the reduced order linearized models. Those issues are in focus of the next chapter.

Chapter 4

Error analysis for TPWL models

One of the crucial problems of model order reduction is assessing quality of the extracted reduced order models. In the linear case, this assessment is done most often in the frequency domain by comparing transfer functions of the original linear system and the reduced order system. For instance, for techniques based on Hankel model order reduction or truncated balanced realization global error bounds can be given in terms of L^∞ norm of the difference between the two transfer functions [32], [72]. On the other hand, reduction methods based on Padé approximation and Krylov subspace algorithms generate models with transfer functions which provide only local approximations for the transfer function of the original system.

In the nonlinear case, since frequency domain description of systems in terms of transfer functions is not available, we compare time-domain responses of the original and reduced order systems. A number of different questions may be asked in order to determine quality of approximation of a given nonlinear system by a reduced order TPWL system. First, we may look for the error between solutions computed with the original and TPWL models for up to a specific time T, and for a given finite input signal $u : [0,T] \to R^M$. In this case, as shown in the next section, *a posteriori* error estimation may provide us with precise answers, at least for certain classes of systems.

If we are also interested in global approximation properties of TPWL models, we should consider infinite time domain input signals, and ask if an error bound between the corresponding outputs (or states) for the original and TPWL models exists and can be given. In order to approach this problem, *a priori* error analysis presented later in this chapter may be applied.

4.1 A posteriori error estimation

A posteriori error analysis aims at obtaining an error bound for a solution which has already been computed with an approximate model. In other words, one's goal is to compute a bound between the solutions (e.g. vectors of states, or outputs) obtained with the approximate model and the original ('exact') model, from an initial time t_0 up to a given time T, using information on the computed approximate solution up to time T. In this section we present a method which *a posteriori* estimates the error of solving (2.1) with a trajectory piecewise-linear model (3.4). More precisely, we derive an iterative formula for an error bound of $\|x(t) - Vz(t)\|_2$, where $x(t)$ is a solution of (2.1) at time $0 < t < T$, and $z(t)$ is solution of (3.4) at the same time, both corresponding to the same initial condition and the same input signal $u(t)$.

The following derivation of the error bound is based on the assumption that the original nonlinear function f (cf. (2.1)) is negative monotone, i.e. [114]:

$$\exists_{\lambda>0}\forall_{x,y}(x-y)^T(f(x)-f(y)) \leq -\lambda(x-y)^T(x-y). \tag{4.1}$$

The above assumption is satisfied by a number of nonlinear systems, including e.g. a certain class of nonlinear analog circuits [114]. (For instance, nonlinear function f given by (2.3), for the nonlinear transmission line model with quadratic resistors, is negative monotone.) Also, one may easily note that if f is negative monotone then system (2.1) is L_p stable for any admissible p, provided $g(x) = Gx$, where G is a symmetric positive definite (SPD) matrix.

For simplicity we will also assume in this chapter that $g(x)$ from equation (2.1) is an identity transformation, and that $B(x) \equiv B$. Still, the following derivations may be easily extended for the case of an arbitrary invertible transformation $g(x)$ (with appropriately modified assumptions), and a state-dependent input matrix $B(x)$.

We will look for an estimate of $||\delta x(t)||_2$, where: $\delta x(t) = x(t) - \hat{x}(t) = x(t) - Vz(t)$, and $x(t)$ and $z(t)$ are solutions at time t of (2.1) and (3.4), respectively (for the same initial condition x_o[1], and the same input signal u). From (2.1) and (3.4) we have:

$$\dot{x} = f(x) + Bu,$$

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

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

$$\dot{\delta x} = f(x) - f(\hat{x}) + h(z) + (I - VV^T)Bu, \tag{4.2}$$

where

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

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

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

where $\theta_i(s,z) = x_i + s(Vz-x_i)$, and $W(\cdot)$ is the Hessian of f: $W(\cdot) = [w_{ij}^k] \in R^{N\times N\times N}$, and $w_{ij}^k = \frac{\partial^2 f_i(\cdot)}{\partial x_i \partial x_j}$. Then $h(z)$ becomes:

$$h(z) = \sum_{i=0}^{s-1} w_i(z)\left\{\left[(I-VV^T)(A_iVz + f(x_i)) + VV^T A_i V z_i - A_i x_i\right]\right.$$
$$\left. + \left[\int_0^1 (1-s)W(\theta_i(s))ds \cdot (Vz-x_i)\otimes(Vz-x_i)\right]\right\}. \tag{4.3}$$

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

Left multiplying (4.2) by δx^T, and applying property (4.1) and Schwartz inequality gives:

$$\frac{d}{dt}(\|\delta x\|_2^2) \leq -\lambda\|\delta x\|_2^2 + \|\delta x\|_2\|h(z) + (I - VV^T)Bu\|_2. \tag{4.4}$$

Let us now consider time interval $[t_i, t_{i+1}]$. Suppose we know $||\delta x(t_i)||_2$. Then, applying Comparison Lemma [35] to differential inequality (4.4) yields:

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

$$+\|\delta x(t_i)\|_2\exp(-\lambda(t - t_i)), \tag{4.5}$$

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

1. At initial time t_0 set
$$e(t_0) = \|I - VV^T\|_2\|x_o\|_2,$$
where x_o is a known initial condition;

2. For $i = 1, 2, \ldots$ iteratively compute

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

$$+e(t_{i-1})\exp(-\lambda(t_i - t_{i-1})). \tag{4.6}$$

Clearly, $\|\delta x(t_0)\|_2 = \|(I - VV^T)x_o\|_2 \leq e(t_0)$, and also it follows from (4.5) that $e(t_i) \geq \|\delta x(t_i)\|_2$ for every $t_1, t_2, \ldots$, i.e. $e(t_i)$ provides a desired error bound. In practice, we replace the supremum in the above formula by a maximum over a discrete set of timesteps between t_{i-1} and t_i, corresponding to a certain numerical time integration scheme. (If t_i are the same as subsequent integration steps, then we take a maximum of the two values at the ends of the considered time interval.) Clearly, this method of evaluating the supremum implicitly assumes that neither $h(z(t))$, nor $u(t)$ behave pathologically between subsequent integration timesteps.

The main challenges associated with using the above scheme are related to: 1) Finding λ (compare (4.1)) which should be as precise as possible. (Quality of the error estimates heavily depends on this parameter, therefore one could consider using different λ's in different regions of the state space, if at all possible and computationally feasible.); 2) Finding estimates of $||h(z(t))||_2$, given by (4.3), which typically requires estimating $||W(x)||_2$ – the norm of the Hessian of f.

Moreover, note that in general the cost of evaluating $h(z)$ given by (4.3) can be as high as the cost of evaluating the initial nonlinear model, which would make using the above error estimation procedure impractical. However, in many cases we may compute the discussed estimates at a significantly lower cost. If we apply the aggregate reduced order basis V, described in Section 3.2.3, one may easily note that $Vz_i = x_i$, for every i. Furthermore, if we include $(f(x_i) - A_ix_i)$ $(\forall_i)$ in the reduced basis V, then $VV^T(f(x_i) - A_ix_i) = (f(x_i) - A_ix_i)$ (cf. (4.3)) and $||h(z)||_2$ may be

estimated as follows:

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

Then, we may replace (4.6) with

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

$$\left. + ||(I - VV^T)B||_2 ||u(\tau)||_2 \right] (1 - \exp(-\lambda(t_i - t_{i-1}))) + e(t_{i-1}) \exp(-\lambda(t_i - t_{i-1})). \quad (4.7)$$

One should note that since the values of norms $||(I - VV^T)A_i V||_2$ (for every i) and $||(I - VV^T)B||_2$ can be computed during construction of the reduced model, the cost of evaluating (4.7) is $O(sq)$ only. This means that error estimation may be performed 'on the fly', along with the reduced order simulation, without increasing the complexity of the fast TPWL solver.

In order to verify the proposed method of estimating simulation errors for TPWL reduced order models, we considered a simple test example of a transmission line model with quadratic nonlinearity at every node, introduced in Section 2.2, and shown in Figure 2-2. In our test the original number of nodes $N = 100$, and $C = r = g = 1$. It may easily be proven that function f describing system's nonlinearity (given by equation (2.3)) is negative monotone, provided all v_i are nonnegative at all times (which is satisfied if the input current $i(t) \geq 0$ for all t).

The value of λ (cf. (4.1)) may then be taken as: $\lambda = -1 \cdot \max\{\lambda_i \in \sigma(A)\}$, where $\sigma(A)$ is the spectrum of matrix A, given by (2.4). For $N = 100$, $\lambda = 9.67 \cdot 10^{-4}$. We also have that $||W(x)||_2 = 2g = 2$, for all x. Knowing λ and $||W(x)||_2$ we are ready to use formula (4.7) to compute the error bounds.

In a numerical test we generated a reduced order TPWL model of order $q = 25$ (with $s = 16$ linearization points), and simulated both the original nonlinear system and its TPWL reduced order model, with the input current $i(t)$ equal to unit step. (It should be stressed that q is relatively large, as compared to N for this example, and therefore it may be inefficient to use the extracted TPWL model in practice. Still, this reduced model provides useful insight while considering the problem of error estimation.) The actual error $||\delta x||_2$ and its estimate were computed at every timestep. Figure 4-1 shows comparison of the actual error and its estimate for the considered case. One may note that formula (4.7) gives reasonable bounds for the error of approximating the original nonlinear system with a TPWL reduced order model.

One should note that the error estimation procedure described above may be used not only to assess errors in solutions computed with an existing TPWL reduced order model, but also to improve the algorithm for generating the TPWL models (or, more precisely, the algorithm for selecting subsequent linearization points). In a simple algorithm for selecting linearization points described in Section 3.2.1, during the 'training' simulation subsequent linearization points are selected using a simple geometric criterion: if the current state is 'far enough' from all previous linearization points, then it becomes the next linearization point. Instead of this geometric criterion one may use a measure based on error estimates (which exploit additional information on the nonlinear system at hand) to select a collection of linearization points. This idea is developed in the following section.

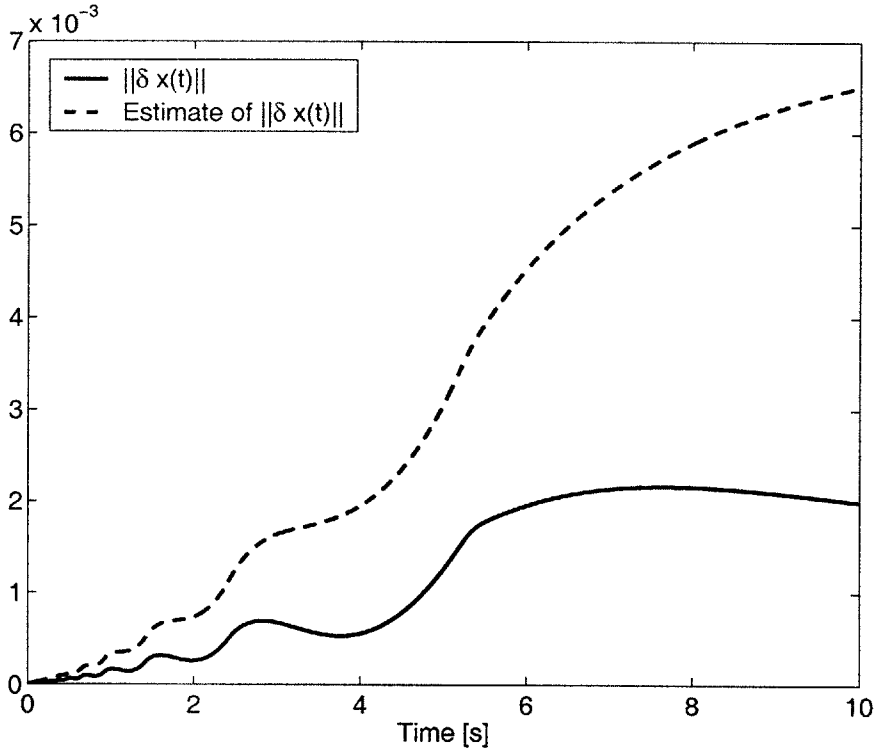

Figure 4-1: A posteriori error bounds for a TPWL reduced order model of a nonlinear transmission line model with quadratic resistors.

4.1.1 Selecting linearization points based on error bounds

As already mentioned, error bounds computed 'on the fly' with a procedure described in the previous section may be used to select a collection of linearization points more effectively, while extracting a TPWL reduced order model.

Below we present a procedure which uses fast approximate simulation described in Section 3.3, and *a posteriori* error bounds derived in the previous section to select a collection of linearization points $x_0, \ldots, x_{s-1}$:

1. Generate a linearized model about the initial state x_0; Set $i = 0$ and the initial error estimate $e(0) = 0$;

2. Simulate the nonlinear system while

$$\left(\frac{\|e(t_k)\|_2}{\|Vz\|_2}\right) < \varepsilon,$$

where

$$e(t_k) = \frac{1}{2\lambda} \sup_{\tau \in [t_{k-1}, t_k]} \left[\sup_{x \in T(Vz(t_{k-1}), Vz(t_{k_i}))} \|W(x)\|_2 \|Vz(\tau) - x_i\|_2^2 + \right.$$

$$\left. \|(I - VV^T)(A_i Vz(\tau) + Bu(\tau))\|_2 \right] (1 - \exp(-\lambda(t_k - t_{k-1}))) +$$

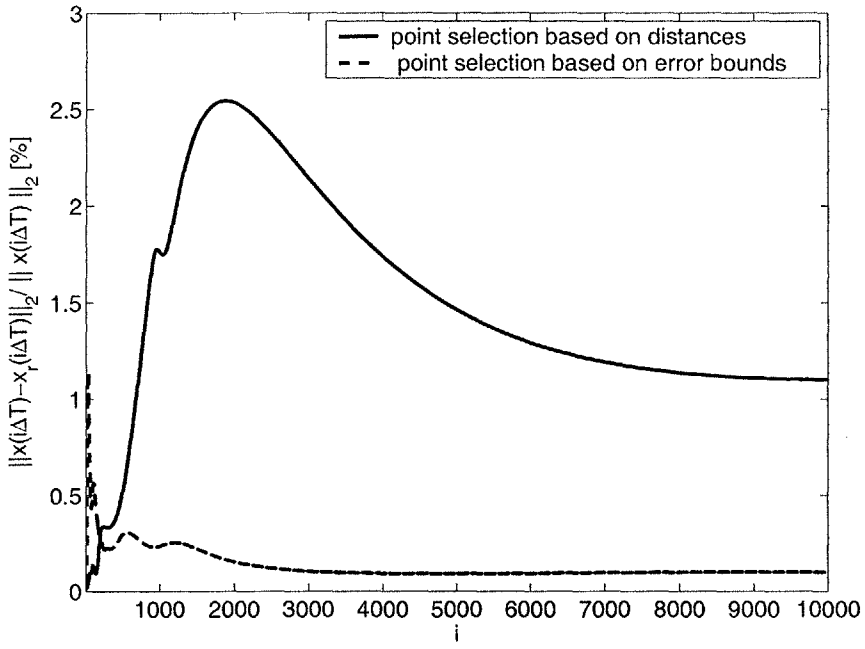

Figure 4-2: Comparison of relative errors for two different reduced order TPWL models. x and x_r are vectors of states computed with full nonlinear, and reduced-order TPWL models, respectively.

$$+e(t_{k-1})\exp(-\lambda(t_k - t_{k-1})), \tag{4.8}$$

t_k and t_{k-1} are the current and previous timesteps, respectively, $z(t_k)$ is the current reduced order state, and $T(\alpha,\beta)$ denotes a trajectory of the simulated system between states α and β;

3. Generate a new linearized model about $x_{i+1} = Vz(t_k)$; Set $e(t_k) = 0$, $i := i+1$;

4. If $i < (s-1)$ return to step 2.

One may also develop a similar procedure if exact, full-order nonlinear simulation is performed to find the training trajectory.

We have tested the above procedure again for the example of a transmission line circuit model with quadratic resistors, introduced in Section 2.2. First, we used a simple procedure for selecting linearization points (cf. Section 3.3), based solely on distances between the points in the state space, and obtained a TPWL model of order $q = 27$ with $s = 15$ linearizations. Then, we applied the above procedure based on error bounds to select linearization points. We have taken $\varepsilon = 0.18$ to obtain a TPWL model (of order $q = 27$) again with 15 linearizations. In order to compare the quality of both TPWL models we tested them for a step input voltage with amplitude 1, and computed the relative errors. The results of this comparison are shown in Figure 4-2. One may note that, on average, the relative error for a TPWL model obtained with the procedure discussed above in this section is significantly lower than the error for the TPWL model obtained with a simple algorithm of selecting linearization points. As one would expect, taking into account the impact of system's nonlinearity, results in a better selection of linearization points.

4.2 Error boundedness and a priori error estimation

In this section we briefly discuss the problem of global approximation properties of TPWL models. One of general questions is: does there exist a global bound for an error between the solution computed with the full nonlinear model and the reduced order TPWL model, valid for all times, and any (infinite) input signal u?

This question brings a number of issues which need to be considered. Firstly, if one looks at the *a posteriori* error bound, given by (4.5), one may immediately note that a global error bound follows from this inequality provided: 1) the input signal $u(t)$ is bounded for any time (which implies that an infinite signal u is bounded in terms of L^∞ norm.), and 2) the trajectory of the reduced order TPWL system is also bounded. (We also need to assume that Hessian $W(x)$ is bounded for all x.) Please note that inequality (4.5) has been derived under assumption that f was negative definite, which in turn implies that the original nonlinear system was L_p stable (and, in particular also BIBO stable.) Consequently, a trajectory of the original system corresponding to a bounded input is also bounded. Unfortunately, we do not know if a trajectory of the TPWL system corresponding to a bounded input is also bounded, or in other words whether the reduced order TPWL model preserve input-output stability of the original nonlinear system. This question is discussed in detail in Chapter 5.

Secondly, if we take an input signal leading to a corresponding unbounded trajectory of the original nonlinear system, then the discussed global error bound for the TPWL model will not exist in a vast majority of cases. This is motivated by the fact that any TPWL model, by our assumption, consists of a finite number of linearized models. Consequently, if the distance between the linearization points and the current state of the nonlinear system grows infinitely, then none of the linearized models will be able to approximate correctly the original nonlinear system for that trajectory. This can be clearly seen if one tries to derive an *a priori* error bound, as described below. For simplicity we will compare just the original nonlinear model

$$\dot{x} = f(x) + Bu, \tag{4.9}$$

with a full-order TPWL model

$$\dot{\hat{x}} = \sum_{i=0}^{s-1} \tilde{w}_i(\hat{x}) \left[f(x_i) + A_i(\hat{x} - x_i) \right] + Bu. \tag{4.10}$$

The first of the equations may be transformed as follows:

$$\dot{x} = f(x) + Bu = \sum_{i=0}^{s-1} (\tilde{w}_i(\hat{x}) f(x)) + Bu = \sum_{i=0}^{s-1} \tilde{w}_i(\hat{x}) \left\{ \left[f(x_i) + A_i(x - x_i) \right] + \right.$$

$$\left. \left[\int_0^1 (1-s) W(x + s(x - x_i)) ds \cdot (x - x_i) \otimes (x - x_i) \right] \right\} + Bu. \tag{4.11}$$

Subtracting equation (4.10) from (4.11) yields:

$$\dot{\delta x} = \left(\sum_{i=0}^{s-1} \tilde{w}_i(\hat{x}) A_i \right) \delta x + \sum_{i=0}^{s-1} \tilde{w}_i(\hat{x}) \int_0^1 (1-s) W(x + s(x - x_i)) ds \cdot (x - x_i) \otimes (x - x_i), \tag{4.12}$$

where $\delta x = x - \hat{x}$. We now make use of the following theorem:

Theorem [108]: Given a linear time-varying differential equation:

$$\dot{x} = A(t)x(t) + v(t),$$

where $t \geq 0, x(t), v(t) \in R^N, A(t) \in R^{N \times N}$, and both $v(\cdot)$ and $A(\cdot)$ are piecewise-continuous, $\|x(t)\|$ is bounded as follows:

$$\|x(t)\| \leq \exp\left\{\int_{t_0}^{t} \mu_{ind}[A(\tau)]d\tau\right\}\|x(t_0)\| + \int_{t_0}^{t} \exp\left\{\int_{\tau}^{t} \mu_{ind}[A(s)]ds\right\}\|v(\tau)\|d\tau,$$

for $t \geq t_0 \geq 0$, where t_0 is the initial time, and $\mu_{ind}(\cdot)$ denotes a matrix measure induced by an R^N norm $\|\cdot\|$.

For proof of the above theorem see [108]. Note that the above inequality based on matrix measures will give tighter bounds than inequalities derived using Gronwall's inequality. In our case we take $A(t) = \left(\sum_{i=0}^{s-1} \tilde{w}_i(\hat{x}(t))A_i\right)$, since $\hat{x}(t)$ is a fixed, although possibly unknown trajectory in the state space. Applying the above theorem to error equation (4.12), and keeping in mind that $\delta x(t_0) = 0$ yields:

$$\|\delta x(t)\|_{1,2,\infty} \leq \int_{t_0}^{t} \exp\left\{\int_{\tau}^{t} \mu_{ind1,2,\infty}[\sum_{i=0}^{s-1} \tilde{w}_i(\hat{x}(s))A_i]ds\right\} \times$$

$$\left\|\sum_{i=0}^{s-1} \tilde{w}_i(\hat{x}(\tau)) \int_0^1 (1-s)W(x(\tau) + s(x(\tau) - x_i))ds \cdot (x(\tau) - x_i) \otimes (x(\tau) - x_i)\right\|_{1,2,\infty} d\tau. \qquad (4.13)$$

Since:

$$\mu_{ind}\left[\sum_{i=0}^{s-1} \tilde{w}_i(\hat{x}(s))A_i\right] \leq \sum_{i=0}^{s-1} \tilde{w}_i(\hat{x}(s))\mu_{ind}[A_i],$$

and

$$\left\|\int_0^1 (1-s)W(x+s(x-x_i))ds \cdot (x-x_i) \otimes (x-x_i)\right\| \leq \frac{1}{2}\left(\sup_x \|W(x)\|\right)\|x-x_i\|^2,$$

inequality (4.13) may be further transformed as follows:

$$\|\delta x(t)\|_{1,2,\infty} \leq \int_{t_0}^{t} \exp\left\{\int_{\tau}^{t} \sum_{i=0}^{s-1} \tilde{w}_i(\hat{x}(s))\mu_{ind1,2,\infty}[A_i]ds\right\} \times$$

$$\sum_{i=0}^{s-1}\left[\tilde{w}_i(\hat{x}(\tau))\frac{1}{2}\left(\sup_x \|W(x)\|_{1,2,\infty}\right)\|x(\tau) - x_i\|_{1,2,\infty}^2\right] d\tau. \qquad (4.14)$$

Moreover, since $0 \leq \tilde{w}_i(\hat{x}) \leq 1$ and $\sum_{i=0}^{s-1} \tilde{w}_i(\hat{x}) = 1$, for every $\hat{x}$, we have:

$$\|\delta x(t)\|_{1,2,\infty} \leq \int_{t_0}^{t} \exp\left\{\int_{\tau}^{t} \mu_{\max} ds\right\} \frac{1}{2} \sup_x \|W(x)\|_{1,2,\infty} \max_{i\in\{0,\ldots,(s-1)\}} \|x(\tau) - x_i\|_{1,2,\infty}^2 d\tau =$$

$$\frac{1}{2} \sup_x \|W(x)\|_{1,2,\infty} \int_{t_0}^{t} \exp\{\mu_{\max}(t-\tau)\} \max_{i\in\{0,\ldots,(s-1)\}} \|x(\tau) - x_i\|_{1,2,\infty}^2 d\tau \leq$$

$$\frac{1}{2} \sup_x \|W(x)\|_{1,2,\infty} \sup_{\tau\in[t_0,t]} \left[\max_{i\in\{0,\ldots,(s-1)\}} \|x(\tau) - x_i\|_{1,2,\infty}^2\right] \frac{1}{-\mu_{\max}} [1 - \exp(\mu_{\max}(t-t_0))], \quad (4.15)$$

where $\mu_{\max} = \max_{i\in\{0,\ldots,(s-1)\}} \mu_{ind1,2,\infty}[A_i]$.

From the above it follows that the error δx will be bounded for all times $t \geq t_0$, provided: 1) the trajectory $x(t)$ is bounded for all times, and 2) $\mu_{\max} < 0$. Note that for a given matrix $A = [a_{ij}]$ we have:

$$\mu_{ind1} = \max_j\{a_{jj} + \sum_{i\neq j} |a_{ij}|\}, \quad \mu_{ind2} = \lambda_{\max}[A^H + A]/2, \quad \mu_{ind\infty} = \max_i\{a_{ii} + \sum_{j\neq i} |a_{ij}|\}.$$

Consequently, one may immediately infer from the above analysis that, for a bounded input signal (and a stable original nonlinear model), δx will be bounded for all times in the following cases: 1) All A_i's are symmetric and strictly stable; 2) All A_i's are strictly diagonally dominant, with all diagonal elements being negative. Note that if we were comparing solutions of the original nonlinear system and the *reduced* TPWL model, then the above conclusions hold, provided we replace Jacobians A_i with $VV^T A_i$.

One may continue derivation (4.15) in order to replace the term

$$\sup_{\tau\in[t_0,t]} \left[\max_{i\in\{0,\ldots,(s-1)\}} \|x(\tau) - x_i\|^2\right],$$

with a term which depends only on $\sup_{\tau\in[t_0,t]} \|u(\tau) - \tilde{u}(\tau)\|$, where u is the testing input, and $\tilde{u}$ is the training input, used to generate a given TPWL model. This can be achieved by considering the training trajectory $\tilde{x}$, which satisfies the following equation:

$$\dot{\tilde{x}} = f(\tilde{x}) + B\tilde{u}. \quad (4.16)$$

Subtracting (4.16) from (4.9) gives:

$$\dot{\overline{\delta x}} = f(x) - f(\tilde{x}) + B(u - \tilde{u}),$$

where $\overline{\delta x} = x - \tilde{x}$. Assuming additionally that f is Lipschitz continuous, i.e. $\exists_L \forall_{x_1,x_2} \|f(x_1) - f(x_2)\| \leq L\|x_1 - x_2\|$, the above equation yields:

$$\|\overline{\delta x}(t)\| \leq L \int_{t_0}^{t} \|\overline{\delta x}(\tau)\| d\tau + \int_{t_0}^{t} \|B\| \|u(\tau) - \tilde{u}(\tau)\| d\tau, \quad (4.17)$$

for all $t \geq t_0$.[2] Applying Gronwall's inequality to (4.17) gives the following bound:

$$\|x(t) - \tilde{x}(t)\| = \|\overline{\delta x}(t)\| \leq \frac{\|B\|}{L} \sup_{\tau \in [t_0, t]} \|u(\tau) - \tilde{u}(\tau)\| \left[\exp(L(t - t_0)) - 1\right].$$

Then, we have:

$$\sup_{\tau \in [t_0, t]} \left[\max_{i \in \{0, \ldots, (s-1)\}} \|x(\tau) - x_i\|^2 \right] \leq \left[\sup_{\tau \in [t_0, t]} \|x(\tau) - \tilde{x}(\tau)\| + \sup_{\tau \in [t_0, t]} \left[\max_{i \in \{0, \ldots, (s-1)\}} \|\tilde{x}(\tau) - x_i\| \right] \right]^2 \leq$$

$$\leq \left[\frac{\|B\|}{L} \sup_{\tau \in [t_0, t]} \|u(\tau) - \tilde{u}(\tau)\| \left[\exp(L(t - t_0)) - 1\right] + \sup_{\tau \in [t_0, t]} \left[\max_{i \in \{0, \ldots, (s-1)\}} \|\tilde{x}(\tau) - x_i\| \right] \right]^2. \quad (4.18)$$

The above inequality may be applied in (4.15) to obtain an *a priori* error bound. Since $\|\tilde{x}(\tau) - x_i\|$ is known *a priori* ($\tilde{x}$ is a training trajectory), for every i and τ, this bound will only depend on the distance between the testing and training inputs $\|u(\tau) - \tilde{u}(\tau)\|$. Unfortunately, the discussed bound will typically be very conservative, especially in the case of stable systems. This is a consequence of applying Lipschitz condition in the derivation of (4.18), yielding an estimate of $\|x(\tau) - x_i\|^2$ which always grows exponentially with time. Therefore, practical importance of the considered *a priori* bound will be limited.

Furthermore, even knowing that δx is bounded, i.e. that trajectories of the original nonlinear, stable system, and the respective TPWL reduced order model, corresponding to the same bounded input signal do not diverge may not be enough. Often, we need to know whether the response of a given TPWL model to a finite energy signal will also have finite energy. This leads us to a question of whether a TPWL model preserves the property of L_2 (or more generally L_p) stability of the original nonlinear system. Those issues are discussed in the following chapter.

[2] Above we have implicitly assumed that the training and testing trajectories begin at the same initial state.

Chapter 5

Stability and passivity preservation with TPWL models

In this chapter we analyze if, or under what conditions trajectory piecewise-linear reduced order models preserve stability and/or passivity of the original nonlinear system. The property of preserving stability and passivity by a reduced order model plays a crucial role, firstly in the context of controlling the approximation error, and secondly if reduced order models are used as part of a larger system, composed of many interconnected subsystems. If one replaces an original passive model of a given subsystem (e.g. a model of an interconnect) with a reduced order model, which is not guaranteed to be stable, then simulation of the entire system (with the reduced model) may yield nonphysical solutions. For instance, it may happen that the reduced model of the subsystem will become an active element, i.e. will act as a (nonexistent) energy source. Such situation will be especially common if feedback loops exist in the modeled system.

In order to prevent the situation when a reduced order model of a passive system exhibits nonphysical behavior (exhibits an instability, or generates energy), a number of approaches for extracting stable and passive reduced order models have been proposed, mostly in the context of linear circuits and interconnects [67], [77], [78], see also [7], [8], [19], [24], [28], [97], [98], [101], [119]. One of the most widely used ideas, exploited in algorithms such as PRIMA [77], [78] is that congruence transformations preserve definiteness of a matrix. Consequently, given a class of stable linear dynamical systems with negative definite system matrices, the corresponding reduced order systems (obtained with congruence transformations) also have negative definite system matrices, and therefore are stable. Below, we will build an analogy between negative definite linear dynamical systems, and nonlinear systems with nonlinearities satisfying a 'negative definiteness' condition, and then explore how this analogy can be used to ensure stability and passivity of trajectory piecewise-linear reduced order models.

5.1 Stability analysis

In this section we consider the problem of stability of TPWL reduced order models. First, we present some background in stability theory. Subsequently, we briefly describe the problem of stability preservation for linear systems, and then turn to discussing nonlinear systems and TPWL models. Finally, we present conditions which are sufficient to ensure L_p (input-output) stability of

TPWL models, analyze the assumptions imposed on the original nonlinear system, and describe approaches for enforcing stability of a system, e.g. based on modifying the weighting procedure.

5.1.1 Background

This paragraph gives definitions of asymptotic, exponential, and L_p (input-output) stability, and presents a few basic facts from stability theory which will be useful later on. In order to define asymptotic and exponential stability we consider an autonomous (unforced, time invariant) system

$$\dot{x} = f(x), \tag{5.1}$$

where $f : R^N \to R^N$ is a globally Lipschitz continuous function. We also assume throughout this section that system (5.1) has an equilibrium point at $x = 0$, i.e. $f(0) = 0$. One should note that this assumption does not make our discussion or results less general, since if system in form (5.1) has an equilibrium point at $x_0 \neq 0$, then we may perform a change of variables: $\tilde{x} = x - x_0$, and analyze the following equivalent system: $\dot{\tilde{x}} = \tilde{f}(\tilde{x})$ (where $\tilde{f}(\tilde{x}) \equiv f(x)$ for any x), which clearly has an equilibrium at the origin.

Asymptotic stability: System (5.1) has a globally asymptotically stable equilibrium point at $x = 0$ if:

1. For each $\varepsilon > 0$, there is $\delta > 0$ such that, for any initial condition x_0, if $\|x_0\| < \delta$, then $\|x(t)\| < \varepsilon$ for every $t > t_0$, where $x(t)$ is the solution of (5.1) with initial condition $x(t_0) = x_0$, and
2. $\lim_{t\to\infty} x(t) = 0$.

Exponential stability: System (5.1) has a globally exponentially stable equilibrium point at $x = 0$, if there exist constants $k > 0$ and $\gamma > 0$, such that, for any initial condition x_0,

$$\|x(t)\| \leq k\|x_0\|e^{-\gamma(t-t_0)},$$

where $x(t)$ is the solution of (5.1) with initial condition $x(t_0) = x_0$.

Clearly, if $x = 0$ is a globally exponentially stable equilibrium point, then it is also a globally asymptotically stable equilibrium point for system (5.1). Also, in the linear case, i.e. if $f(x) = Ax$, asymptotic stability implies exponential stability.

In order to define L_p stability we consider a nonautonomous (forced) system with an additional output equation:

$$\begin{cases} \dot{x} = f(x,u) \\ y = h(x,u) \end{cases}, \tag{5.2}$$

where $f(\cdot,\cdot)$ is Lipschitz continuous, and $h(\cdot,\cdot)$ is continuous. We define space L_p as follows:

$$L_p = \left\{ v : \|v\|_{L_p} < \infty \right\},$$

where $v : [t_0, \infty) \rightarrow R^N$ is a vector-valued time signal, t_0 is the initial time, and

$$\|v\|_{L_p} = \left(\int_{t_0}^{\infty} \|v(t)\|^p dt \right)^{1/p},$$

where $\|\cdot\|$ is some norm in R^N. An extended space L_{pe} is defined as:

$$L_{pe} = \left\{ v : v_\tau \in L_p, \forall_{\tau \geq t_0} \right\},$$

where v_τ is a truncation of v:

$$v_\tau(t) = \begin{cases} v(t) & t_0 \leq t \leq \tau \\ 0 & t > \tau \end{cases}.$$

L_p stability: System (5.2) is L_p stable for some $p \in [1, \infty]$, if there exist nonnegative constants γ and β such that, for all input signals u, and for all $\tau \geq t_0$, if $u \in L_{pe}$ then

$$\|y_\tau\|_{L_p} \leq \gamma \|u_\tau\|_{L_p} + \beta,$$

where y is the output signal of (5.2) corresponding to the input signal u.

The following fact, presented in [48] links exponential stability with L_p stability:

Theorem 1: Suppose f is continuously differentiable, the Jacobian matrices $[\partial f / \partial x]$ and $[\partial f / \partial u]$ are bounded, and $h(x,u)$ satisfies the condition:

$$\forall_{x,u} \|h(x,u)\| \leq \eta_1 \|x\| + \eta_2 \|u\|,$$

for some nonnegative constants η_1, η_2. If $x = 0$ is a globally exponentially stable equilibrium point of the unforced system: $\dot{x} = f(x,0)$, then system (5.2) is L_p stable for any initial condition x_0, and any $p \in [1, \infty]$.

Summarizing, provided f and h satisfy some basic continuity conditions, global exponential stability of the origin for the autonomous system, implies L_p stability of the corresponding forced system. Consequently, in the following sections we will focus on finding conditions for exponential stability of the origin for an unforced TPWL system, which will guarantee L_p stability for the corresponding forced TPWL system. Another useful theorem is the following version of the classical Lyapunov stability theorem.

Theorem 2 (Lyapunov): Let $x = 0$ be an equilibrium point for system (5.1). If there exists a continuously differentiable function $V : R^N \rightarrow R$, and positive constants k_1, k_2, k_3, and c such that

$$k_1 \|x\|^c \leq V(x) \leq k_2 \|x\|^c, \tag{5.3}$$

$$\dot{V}(x) = [\partial V(x) / \partial x] \cdot f(x) \leq -k_3 \|x\|^c, \tag{5.4}$$

for all x, then $x = 0$ is a globally exponentially stable equilibrium point for system (5.1).

Proofs of the theorems cited above may be found in the book by Khalil [48].

5.1.2 Stability of linear systems

Before we turn to the problem of stability preservation for nonlinear systems let us first examine the linear case. Consider an unforced linear system

$$\dot{x} = Ax, \tag{5.5}$$

where A is a nonsingular $N \times N$ matrix with real elements. Then $x = 0$ is a unique, isolated equilibrium point for this system, and $x = 0$ is the globally asymptotically (exponentially) stable equilibrium point, if and only if matrix A is Hurwitz, i.e. if all eigenvalues λ_i satisfy condition $\mathrm{Re}(\lambda_i) < 0$. Now, let us consider the corresponding reduced order linear system, obtained from (5.5) using an orthonormal projection:

$$\dot{z} = V^T AVz, \tag{5.6}$$

where V is a real $N \times q$ orthonormal matrix with rank q. Again, system (5.6) will have an asymptotically (exponentially) stable equilibrium point at $x = 0$, if and only if matrix $V^T AV$ is Hurwitz. Therefore, the fundamental question is to find conditions under which matrix $V^T AV$ will be Hurwitz, provided A is Hurwitz. Since all eigenvalues of A have negative real parts A cannot be positive definite. Consequently, we have two cases:

1. A is negative definite, i.e. for any $x \neq 0$, we have $(x^*)^T Ax < 0$. (This condition also means that the quadratic form on the left hand side of the inequality takes only real values and implies that all eigenvalues of A are real and negative.) Clearly, then matrix $V^T AV$ is also negative definite and Hurwitz, since it is obtained from matrix A by a congruence transformation, which preserves matrix definiteness. For any $z \neq 0$, we may take $x = Vz$ and then, since A is negative definite, we have
$$(z^*)^T V^T AVz = (x^*)^T Ax < 0.$$
This implies that 0 is an asymptotically stable equilibrium for system (5.6). In this case for any projection basis V (assuming V is real and full rank) the corresponding reduced order system (5.6) preserves stability of the original system (5.5).

2. A is indefinite. (Note, this implies that A is nonsymmetric, since any symmetric Hurwitz matrix is negative definite.) In this case we cannot guarantee that for any real, full rank matrix V matrix $V^T AV$ will be Hurwitz. Only for some special choices of V (e.g. if every column of V is an eigenvector of A) matrix $V^T AV$ will remain Hurwitz.

To sum up, if we consider a general reduced order basis V, we can guarantee that $x = 0$ will be an asymptotically stable equilibrium point for system (5.6) only if A is negative definite. Otherwise, the stability will be preserved only for certain choices of V.

5.1.3 Stability of projected nonlinear systems

Now, let us consider an unforced nonlinear system $\dot{x} = f(x)$. Recall that in the linear case, the reduced order linear system preserved stability of the original linear system for any projection basis V provided system matrix A was negative definite. In the nonlinear case we introduce a

condition on f analogous to negative definiteness:

$$\exists_{\lambda>0}\forall_x \; x^T f(x) \leq -\lambda x^T x. \tag{5.7}$$

Clearly, we also assume that $f(0) = 0$. Note that the above condition is significantly weaker than the assumption of negative monotonicity, introduced in Chapter 5.[1]. One may also note that the following fact is true:

Fact 1 Given Lipschitz continuous function f, such that $f(0) = 0$, for any initial condition x_0, and for any $t > t_0$, the solution $x(t)$ of the Cauchy problem:

$$\dot{x} = f(x), \quad x(t_0) = x_0 \tag{5.8}$$

satisfies the inequality

$$\|x(t)\|_2 \leq \|x(t_0)\|_2 e^{-\lambda(t-t_0)}, \tag{5.9}$$

for some $\lambda > 0$ (independent of x_0), if and only if condition (5.7) holds with the same λ.

Proof: ($\Leftarrow$) Let us consider problem (5.8). Then, we have

$$x^T\dot{x} = x^T f(x) \leq -\lambda x^T x.$$

Since $2x^T\dot{x} = d(\|x\|_2^2)/dt = 2\|x\|_2 \cdot d(\|x\|_2)/dt$, the above yields:

$$\frac{d(\|x\|_2)}{dt} \leq -\lambda\|x\|_2.$$

Using Comparison Lemma [35] we obtain condition (5.9).
($\Rightarrow$) Since $x(t)$ is a solution of (5.8) it is also a solution the following Cauchy problem:

$$\tfrac{1}{2}\tfrac{d(\|x\|_2^2)}{dt} = x^T f(x), \quad \|x(t_0)\|_2^2 = \|x_0\|_2^2 \cdot$$

From the above and condition (5.9) we obtain:

$$\|x_0\|_2^2 + 2\int_{t_0}^{t} x^T(s) f(x(s))ds = \|x(t)\|_2^2 \leq \|x_0\|_2^2 e^{-2\lambda(t-t_0)},$$

for any $t \geq t_0$, and for any x_0. Simplifying the above yields:

$$\int_{t_0}^{t} x^T(s) f(x(s))ds \leq \frac{\|x_0\|_2^2}{2}\left(e^{-2\lambda(t-t_0)} - 1\right).$$

Substituting $t = t_0 + \varepsilon$ ($\varepsilon > 0$) and dividing both sides of the above inequality by ε gives:

$$\frac{1}{\varepsilon}\int_{t_0}^{t_0+\varepsilon} x^T(s) f(x(s))ds \leq \frac{\|x_0\|_2^2}{2\varepsilon}\left(e^{-2\lambda\varepsilon} - 1\right).$$

[1] f is negative monotone if $\exists_{\lambda>0}\forall_{x,y} \; (x-y)^T f(x-y) \leq -\lambda(x-y)^T(x-y)$.

Since f is Lipschitz continuous, taking the limit of both sides of the above inequality for $\varepsilon \to 0^+$ yields condition (5.7) with $x = x_0$ (but x_0 was arbitrary). This concludes the proof. ■

Using the above fact we easily conclude that condition (5.7) implies that the origin is an exponentially stable equilibrium point for system $\dot{x} = f(x)$, provided $f(0) = 0$ and f is Lipschitz continuous (cf. definition of the exponential stability in Section 5.1.1). Clearly, the projected function $\hat{f}(\cdot) = V^T f(V\cdot)$ also satisfies condition (5.7) (with the same λ if V is an orthonormal basis), since for any z:

$$z^T \hat{f}(z) = z^T V^T f(Vz) = (Vz)^T f(Vz) \leq -\lambda z V^T V z = -\lambda z^T z. \tag{5.10}$$

Therefore, the projected system $\dot{z} = V^T f(Vz)$ also has an exponentially stable equilibrium point at the origin.

5.1.4 Stability of TPWL systems

Finally, let us consider an unforced reduced order TPWL system corresponding to the original nonlinear system (5.1):

$$\dot{z} = \sum_{i=0}^{s-1} w_i(z) \left[V^T f(x_i) + V^T A_i V z - V^T A_i x_i \right], \tag{5.11}$$

where $A_i = [\partial f(x_i)/\partial x]$, and x_i are the linearization points. Below we will also assume that the first linearization point $x_0 = 0$, i.e. one of the linearized models is generated at the equilibrium point of the original nonlinear system.

Before we discuss how one may ensure stability of system (5.11), or more precisely how the negative definiteness condition (5.7) can be exploited to this end, let us discuss some of the general problems which arise when using a linearization or collection of linearizations to approximate a nonlinear system. Let us first consider properties of a single linearization of (5.1) at the equilibrium point $x = 0$:

$$\dot{x} = A_0 x, \tag{5.12}$$

where A_0 is the Jacobian of f at the origin. In general, if $x = 0$ is a globally asymptotically stable equilibrium point for system (5.1), we cannot guarantee that it will also be a (globally) asymptotically stable equilibrium for system (5.12). We need to make a slightly stronger assumption on the original nonlinear system, namely that the origin is the globally *exponentially* stable equilibrium point for that system. Then $x = 0$ is also the globally exponentially ($\equiv$ asymptotically) stable equilibrium point for the linearized system (5.12) (cf. [48] for a proof of this fact). Unfortunately, exponential stability of the origin for nonlinear system (5.1) does not imply negative definiteness of matrix A_0, and consequently does not guarantee stability of the reduced linearized system $\dot{z} = V^T A_0 V z$. Nevertheless, if we assume that f satisfies condition (5.7), then the origin will be an asymptotically (exponentially) stable equilibrium point also for the reduced order linearized system $\dot{z} = V^T A_0 V z$, which follows from Fact 2 (below), Fact 1, and preservation of the matrix definiteness by congruence transformations (cf. Section 5.1.2).

Fact 2: If $f : R^N \to R^N$ satisfies condition (5.7) and $f(0) = 0$, then $x^T A_0 x \leq -\lambda x^T x$ for any x, where A_0 is the Jacobian of f at $x = 0$.

Proof: Since $f(x) = [f_1(x), \ldots, f_N(x)]^T$ then

$$x^T A_0 x = x^T \left[\nabla_x f_1(0), \ldots, \nabla_x f_N(0)\right]^T,$$

where $\nabla_x f_i(0)$ is a directional derivative of f in direction x, evaluated at 0. Since:

$$\nabla_x f_i(0) = \lim_{\varepsilon \to 0^+} \frac{f_i(0+\varepsilon x) - f_i(0)}{\varepsilon} = \lim_{\varepsilon \to 0^+} \frac{f_i(\varepsilon x)}{\varepsilon},$$

and by (5.7)

$$\varepsilon x^T f(\varepsilon x) \leq -\lambda \varepsilon^2 x^T x,$$

then

$$x^T A_0 x = \lim_{\varepsilon \to 0^+} x^T \left[\frac{f_1(\varepsilon x)}{\varepsilon}, \ldots, \frac{f_N(\varepsilon x)}{\varepsilon}\right] = \lim_{\varepsilon \to 0^+} \frac{1}{\varepsilon^2} \varepsilon x^T f(\varepsilon x) \leq \lim_{\varepsilon \to 0^+} (-\lambda x^T x) = -\lambda x^T x. \blacksquare$$

If we now consider TPWL model (5.11), the stability analysis becomes much more involved due to the following facts: 1) We consider a collection of linearized models of (5.1) at *nonequilibrium* points x_i (except for x_0); 2) At each state z we take a convex combination of those models.

Firstly, since we linearize around nonequilibrium points the following problem may occur. Suppose that, for a given $j \in \{1, \ldots, (s-1)\}$, there exists a nonempty set D in the state space, defined as follows: $D = \{z \in R^q : w_j(z) = 1\}$. If $z \in D$ then TPWL system (5.11) reduces to a single, linearized dynamical model:

$$\dot{z} = V^T f(x_j) + V^T A_j (Vz - x_j).$$

Let us now consider $\tilde{z} = (V^T A_j V)^{-1}(V^T A_j x_j - V^T f(x_j))$. It is trivial to check that if $\tilde{z} \in D$ then $\tilde{z}$ will be an equilibrium point for TPWL system (5.11). In order to illustrate this situation better let us consider the following example of a simple 1D dynamical system: $\dot{x} = -\tan(x)$, where $x \in (-\pi/2, \pi/2)$.[2] Now, let us generate a corresponding TPWL system with two linearization points $x_0 = 0$ and $x_1 = \pi/4$. It is easy to check that the resulting TPWL system takes the following form:

$$\dot{x} = w_0(x) \cdot (-1 \cdot x) + w_1(x) \cdot (-1 - 2 \cdot (x - \pi/4)). \tag{5.13}$$

The original system has a single equilibrium point at $x = 0$. In the TPWL model an 'artificial' equilibrium point may appear at $\tilde{x} = \pi/4 - 1/2$, depending on the applied weighting procedure. If we use e.g. the following weighting procedure:

$$w_1(x) = \begin{cases} 1 & |x - \pi/4| < 1/4 \\ 0 & \text{otherwise} \end{cases},$$

[2]Note that $f(x) = -\tan(x)$ satisfies condition (5.7).

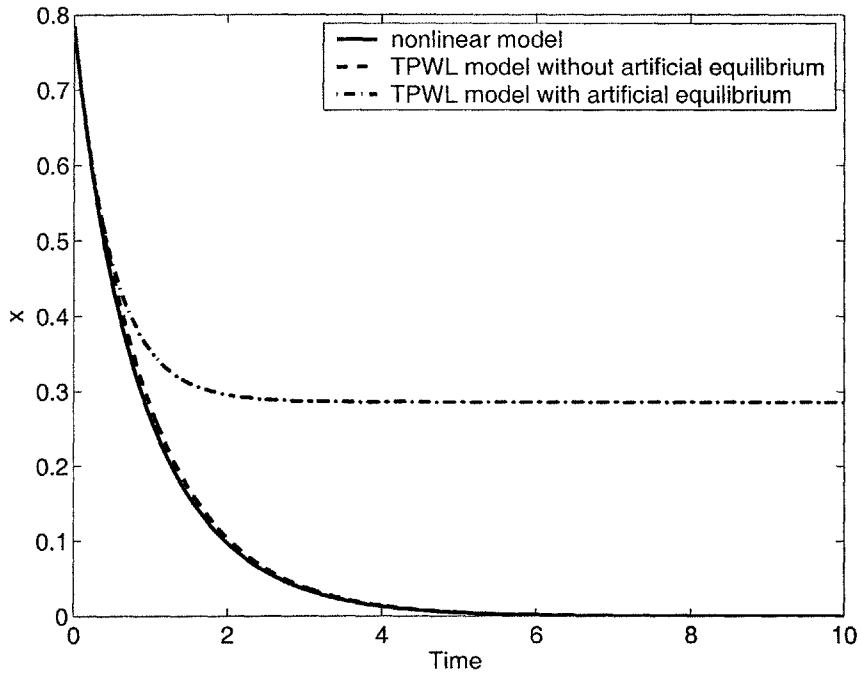

Figure 5-1: The effect of artificial equilibrium in a 1D TPWL system.

and $w_0(x) = 1 - w_1(x)$, then for the initial condition $x(0) = \pi/4$ the solution $x(t)$ of (5.13) will monotonically tend to 0 as $t \to \infty$. On the other hand, if we apply the following weights:

$$w_1(x) = \begin{cases} 1 & |x - \pi/4| < 3/5 \\ 0 & \text{otherwise} \end{cases},$$

and $w_0(z) = 1 - w_1(z)$, then the solution of (5.13) (with the same initial condition $x(0) = \pi/4$) will monotonically tend to the artificial equilibrium point at $(\pi/4 - 1/2)$ as $t \to \infty$ (cf. Figure 5-1).

This example shows that TPWL models may introduce some numerical artifacts, such as additional equilibria, which will usually be very undesirable. Clearly, if such artificial equilibrium exists, the TPWL model will not have a globally asymptotically stable equilibrium point at $x = 0$. Nevertheless, as also illustrated in the above example, by an appropriate choice of weights, one may eliminate artificial equilibria. In the following sections we will discuss how weights can be used to ensure that a given TPWL system does not have such additional equilibria.

Another issue is that, without any extra assumptions of the nonlinear operator f, we only know that $A_0 = \partial f/\partial x(0)$ is Hurwitz. We do not know anything about properties of all other Jacobians A_i $(i = 1, \ldots, (s-1))$. Moreover, even if we knew that all A_i were Hurwitz, then a convex combination of those matrices does not have to be Hurwitz. Also, even if f satisfies condition (5.7), then, except for A_0, none of the Jacobians A_i is guaranteed to be negative definite or stable.

The above mentioned problems indicate that, in general, reduced order TPWL models will not preserve stability of the original nonlinear system, even if f from the original nonlinear system is negative definite. Still, one may attempt to find conditions, to be satisfied by a TPWL model, which will be sufficient to guarantee its stability. Note that, as found in the previous section, if $f(\cdot)$ is negative definite, then also $V^T f(V\cdot)$ is negative definite, implying exponential stability of

the origin for the corresponding reduced order system. One may therefore consider an analogous condition for a TPWL system:

$$z^T \cdot \left(\sum_{i=0}^{s-1} w_i(z) \left[V^T f(x_i) + V^T A_i (Vz - x_i) \right] \right) \leq -k_3 z^T z, \tag{5.14}$$

where k_3 is some positive constant. In view of Fact 1, the above condition implies exponential stability of the origin for system (5.11) (provided w_i's are Lipschitz continuous). It also has advantages which will be described later in this section. At this point let us only note that it does ensure that system (5.11) has an equilibrium only at $x = 0$, since by (5.14) the right hand side of (5.11) must be nonzero for all $z \neq 0$.

The above stability condition may also be derived using Lyapunov theorem (Theorem 2 from Section 5.1.1). The theorem states that the origin is guaranteed to be a globally exponentially stable equilibrium point for system (5.11), provided we can find a Lyapunov function $V(z)$ satisfying conditions (5.3) and (5.4) with $f(z) = \sum_{i=0}^{s-1} w_i(z) \left[V^T f(x_i) + V^T A_i (Vz - x_i) \right]$. The simplest function we may consider is a quadratic function $V(z) = \frac{1}{2} z^T z$. Clearly, for this function condition (5.3) is satisfied automatically. Since $\partial V(z)/\partial z = z^T$, condition (5.14) directly follows from (5.4).

The most important question is whether it is always possible to construct a TPWL model, such that condition (5.14) will be satisfied. A related question is whether, for given x_i and A_i, it is possible to find weights $w_i(z)$ such that inequality (5.14) will hold for all z, and some constant $k_3 > 0$. The following simple fact gives a positive answer, for the case of negative definite matrix A_0:

Fact 3: If matrix A_0 is negative definite then, for given x_i and A_i, where $x_0 = 0$, we can find weights $w_i(z)$ such that condition (5.14) will hold for any z.

Proof: Take the following weights: $w_0(z) \equiv 1$, and $w_1(z) \equiv \ldots \equiv w_{s-1}(z) \equiv 0$. Then condition (5.14) reduces to: $z^T V^T A_0 V z \leq -k_3 z^T z$, since $x_0 = 0$ and $f(x_0) = 0$. We also have:

$$\sup_{z \neq 0} \frac{z^T V^T A_0 V z}{z^T z} = \sup_{z \neq 0} \frac{(Vz)^T A_0 V z}{(Vz)^T V z} = \sup_{\|Vz\|=1} (Vz)^T A V z \leq \sup_{\|x\|=1} x^T A_0 x.$$

Since hypersphere $S = \{x : \|x\| = 1\}$ is a compact set, the supremum on the right hand side of the above inequality is attained, and we can define $k_3 = -\max_{\|x\|=1} x^T A_0 x$. A_0 is negative definite, i.e. $x^T A_0 x < 0$ for $x \in S$, and therefore $k_3 > 0$. Consequently

$$\sup_{z \neq 0} \frac{z^T V^T A_0 V z}{z^T z} \leq -k_3, \tag{5.15}$$

which ends the proof. ■

Consequently, provided A_0 is negative definite we can always find a weighting procedure such that the discussed TPWL system will have an exponentially stable equilibrium point at the origin. Still, the above proof raises a question whether such weighting procedure will generate a feasible

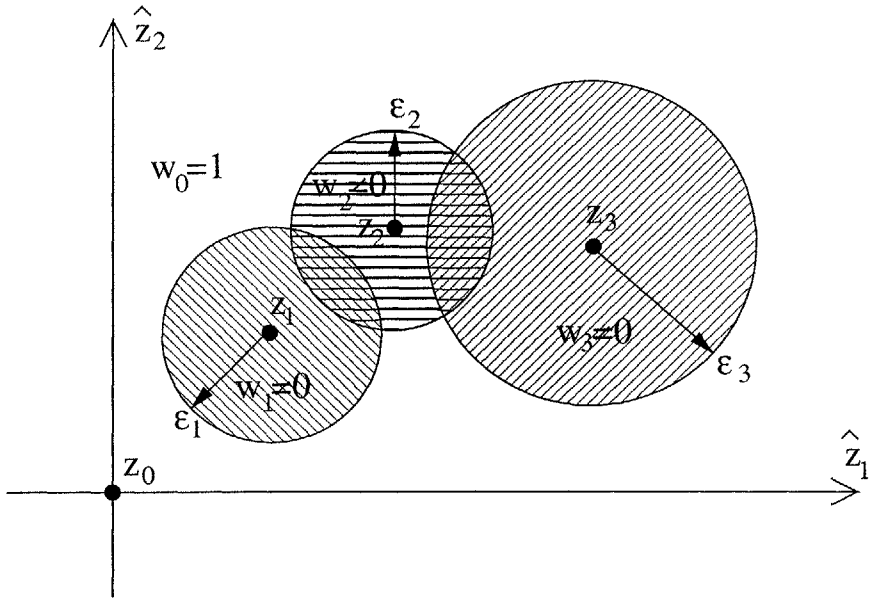

Figure 5-2: Admissible weight distribution in a stability-preserving weighting procedure.

TPWL model of the original nonlinear system. The worst case scenario is when, in order to preserve stability, the weighting procedure implies using only a single linearized model at the origin. In this case the TPWL model would be no more accurate than a simple linear reduced order model. Therefore we may ask the following question: Is it possible to generate a weighting procedure such that, for all z, condition (5.14) is satisfied *and*, for every i, there exists ε_i, such that, for all z, if $z \in B_{\varepsilon_j}(z_j)$ (i.e. if z is in a ball with radius ε_j centered at z_j, where $z_j = V^T x_j$ is the projection of linearization point x_j) for some particular $j = 0, \ldots, (s-1)$ then $w_j(z) = 1$? In other words, is there a stability-preserving weighting procedure which will also select the 'optimal' linearized model (in particular – the model which is the closest to the current state of the system z), provided z is close enough to the linearization points? As proved below, the answer to this question is positive, provided nonlinear function f satisfies condition (5.7).

Theorem 3 Suppose $f : R^N \to R^N$ is Lipschitz continuous, $f(0) = 0$, and f satisfies condition (5.7). Suppose also that all linearization points x_i can be represented *exactly* in an orthonormal basis V, i.e. $x_i = V z_i$, for $i = 1, \ldots, (s-1)$. Then, the origin is an exponentially stable equilibrium point for TPWL system (5.11) if, for all $i = 1, \ldots, (s-1)$, $w_i(z) = 0$ for $z \in (R^q \backslash B_{\varepsilon_i}(z_i))$, where

$$\varepsilon_i \leq \left(\sqrt{\frac{\tilde{\lambda}^2}{W^2} + \frac{2\tilde{\lambda}}{W}\|x_i\|_2} - \frac{\tilde{\lambda}}{W} \right), \tag{5.16}$$

and $\tilde{\lambda}$ is any positive constant such that $\tilde{\lambda} < \lambda$, $W = \sup_x \|W(x)\|_2$, $W(x)$ is the Hessian of f evaluated at x, and $B_{\varepsilon_i}(z_i) = \{z \in R^q : \|z - z_i\| \leq \varepsilon_i\}$ (cf. Figure 5-2).

Proof: We will prove that assumptions of the theorem on the weights and function f imply condition (5.14). Since, for every z $\sum_{i=0}^{s-1} w_i(z) = 1$, and $x_i = V z_i$ this condition may be written in

the following form:

$$\left(\sum_{i=0}^{s-1} w_i(z) z^T \left[V^T f(x_i) + V^T A_i V(z - z_i)\right]\right) \leq -k_3 \sum_{i=0}^{s-1} (w_i(z) z^T z).$$

Therefore, the condition which clearly implies (5.14) is the following:

$$\exists_{\lambda > k_3 > 0} \forall_{i=0,\ldots,(s-1)} \forall_{z \in D_i} \; z^T \left(V^T f(x_i) + V^T A_i V(z - z_i)\right) \leq -k_3 z^T z, \tag{5.17}$$

where $D_i = \{z : w_i(z) \neq 0\}$. First, we note that if $z \in D_0$ then, since A_0 is negative definite (cf. Fact 2), we have:

$$z^T \left(V^T f(x_0) + V^T A_i V(z - z_0)\right) = z^T V^T A_0 V z \leq -\lambda z^T V^T V z = -\lambda z^T z < -k_3 z^T z. \tag{5.18}$$

Therefore condition (5.17) needs to be considered only for $i = 1, \ldots (s-1)$. If we subtract term $z^T V^T f(Vz)$ from both sides of (5.17), and use (5.7), we will obtain the following condition equivalent to (5.17):

$$\exists_{\lambda > \tilde{\lambda} > 0} \forall_{i=1,\ldots,(s-1)} \forall_{z \in D_i} \; z^T \left(V^T f(x_i) + V^T A_i V(z - z_i) - V^T f(Vz)\right) \leq \tilde{\lambda} z^T z. \tag{5.19}$$

Since:

$$V^T f(Vz) = V^T f(x_i) + V^T A_i (Vz - x_i) + \int_0^1 (1-s) V^T W(x_i + s(Vz - x_i)) ds (Vz - x_i) \otimes (Vz - x_i),$$

for all i and z, condition (5.19) may be expressed in the following equivalent form:

$$\exists_{\lambda > \tilde{\lambda} > 0} \forall_{i=1,\ldots,(s-1)} \forall_{z \in D_i} \; z^T \left(-\int_0^1 (1-s) V^T W(\theta_i(s,z)) ds (V(z - z_i)) \otimes (V(z - z_i))\right) \leq \tilde{\lambda} z^T z,$$

where $\theta_i(s,z) = x_i + s(Vz - x_i)$. Below we write two subsequently stronger conditions implying the above inequality:

$$\exists_{\lambda > \tilde{\lambda} > 0} \forall_{i=1,\ldots,(s-1)} \forall_{z \in D_i} \; \left\| \int_0^1 (1-s) V^T W(x_i + s(Vz - x_i)) ds (V(z - z_i)) \otimes (V(z - z_i)) \right\|_2 \leq \tilde{\lambda} \|z\|_2,$$

$$\exists_{\lambda > \tilde{\lambda} > 0} \forall_{i=1,\ldots,(s-1)} \forall_{z \in D_i} \; \frac{1}{2} W \|z - z_i\|_2^2 \leq \tilde{\lambda} \|z\|_2, \tag{5.20}$$

where $W = \sup_x \|W(x)\|_2$, and we used the fact that $\|V^T\|_2 = \|V\|_2 = 1$, since V is an orthonormal basis. Suppose now we consider a ball $B_{\varepsilon_i}(z_i)$ for a given i, and $D_i \subset B_{\varepsilon_i}(z_i)$. This means that $w_i(z) \neq 0$ only if $\|z - z_i\|_2 < \varepsilon_i$. If we impose that

$$\forall_{i=1,\ldots,(s-1)} \forall_{z \in D_i} \; \frac{1}{2} W \varepsilon_i^2 \leq \tilde{\lambda} \|z\|_2,$$

for some $\lambda > \tilde{\lambda} > 0$, then condition (5.20) will be satisfied. Since $\|z\|_2 \geq \|z_i\|_2 - \varepsilon_i \geq \|x_i\|_2 - \varepsilon_i$

for every $i = 1, \ldots, (s-1)$, we may write an even stronger condition:

$$\forall_{i=1,\ldots,(s-1)} \ \varepsilon_i^2 \leq \frac{2\tilde{\lambda}}{W}(\|x_i\|_2 - \varepsilon_i).$$

Solving the above inequality for ε_i yields condition (5.16). Therefore, as we have shown condition (5.16) implies condition (5.14), which proves that the origin is an exponentially stable equilibrium point for TPWL system (5.11). ■

One may note that, for any $i = 1, \ldots, (s-1)$, condition (5.16) implies that $\varepsilon_i < \|x_i\|_2$. It is also clear that we should take $\tilde{\lambda}$ closest to λ in order to obtain the loosest bound for ε_i. Condition (5.16) also tells us that as λ grows (i.e. when the trajectories of the original nonlinear system converge faster to the origin – cf. Fact 1), we may take larger radii ε_i in the TPWL model, which is intuitively correct. Also, if f is smoother, i.e. if second derivatives (represented by W) are smaller we may take larger radii.

A trivial observation is that condition (5.14) implies that a given TPWL system has no non-zero equilibria. It is interesting to note that for 1D systems an opposite implication holds, i.e. if f is continuous and negative definite, and a 1D TPWL system has no non-zero equilibria, then this system is also negative definite, and therefore has an exponentially stable equilibrium at the origin. Such implication most likely does not hold for higher-dimensional systems.

Also note that Theorem 3 provides a set of guidelines for selecting subsequent linearization points. The discussed theorem defines state-space regions around the linearization points x_i (or, more precisely, balls with radii ε_i, centered at x_i), which are admissible support regions for the corresponding weighting functions w_i. Note that if distances between x_i become large enough, those regions may become disjoint (cf. Figure 5-2), which means that in order to enforce stability we may need to set w_0 to 1 in the regions located between the linearization points. This in turn may adversely affect accuracy of a given TPWL model. In order to avoid such situation one may select subsequent linearization points in such a way that e.g. $\|x_{i-1} - x_i\|_2 < (\varepsilon_{i-1} + \varepsilon_i)$, for every i.

5.1.5 Computing stability-preserving weights

Furthermore, Theorem 3 tells us that, provided f satisfies condition (5.7), we can select an 'optimal' linearized model provided current state z of the TPWL model is close enough to some linearization point z_i, or more precisely provided z is in ball centered at z_i with radius ε_i, satisfying condition (5.16) for some i. Still, condition (5.16) is only a sufficient one, and it is significantly stronger than condition (5.14) (with $x_i = Vz_i$). Consequently, in practice we may use (5.14) instead of (5.16) in order to compute stability-preserving weights.

Below we present an example of a weighting procedure, based on condition (5.14), which guarantees that the origin will remain an exponentially stable equilibrium point for the reduced order TPWL model. The procedure assumes we already have another algorithm which computes 'optimal' (in some sense) weights $\tilde{w}_i(z)$ based on the information about the current state of the system.

Given current state z of TPWL system (5.11):

1. Compute $\tilde{w}_i(z)$ and take $w_i(z) = \tilde{w}_i(z)$ for $i = 0, \ldots, (s-1)$.

2. Set

$$I^- = \{i \in \{0,\ldots,(s-1)\} : z^T \left[V^T f(x_i) + V^T A_i V(z-z_i)\right] < 0\},$$
$$I^+ = \{i \in \{0,\ldots,(s-1)\} : z^T \left[V^T f(x_i) + V^T A_i V(z-z_i)\right] > 0\}.$$

3. While

$$(s_1 = z^T \cdot \left(\sum_{i=0}^{s-1} w_i(z) \left[V^T f(x_i) + V^T A_i V(z-z_i)\right]\right)) > (s_2 = -kz^T z),$$

where k is a fixed small positive constant, do:

(a) Set $j = \text{argmax}_{i \in I^-}\{w_i(z) : w_i(z) < 1\}$.

(b) Set $k = \text{argmin}_{i \in I^+}\{w_i(z) : w_i(z) > 0\}$.

(c) Set $\delta = \min\{(1 - w_j(z)), w_k(z), \delta_1\}$, where

$$\delta_1 = \frac{s_1 - s_2}{z^T \cdot (V^T f(x_k) + V^T A_k V(z-z_k) - V^T f(x_j) - V^T A_j V(z-z_j))}.$$

(d) Set $w_k(z) := w_k(z) - \delta$, and $w_j(z) := w_j(z) + \delta$.

The above procedure gradually transfers weight from models for which

$$z^T \left[V^T f(x_i) + V^T A_i V(z-z_i)\right] > 0,$$

which is undesirable from the point of view of stability, into models for which an opposite condition is satisfied. The weight is transferred first from the models which are 'least optimal' to the models which are 'most optimal' for given z. It is also important to note that Fact 2 guarantees that the above procedure will always finish. (In the worst case the entire weight will be put on the linearized model from the origin.) Please also note that the above procedure may be modified in such a way, that condition (5.14) will be satisfied with $k_3 = \lambda$.

Fact 2 also tells us that the above procedure will guarantee stability even if f does not satisfy condition (5.7). The only assumption we need is that A_0 is a negative definite matrix. Nevertheless, this weighting procedure will most likely generate more feasible models if f satisfies condition (5.7), as stated in Theorem 3. There is yet another issue which needs to be considered: if ε_i from Theorem 3 are small as compared to the distances between the linearization points then, most likely, the above procedure will select an 'optimal' model only if the current state of the system is very close to some linearization point, and otherwise select the 0-th model. A solution to this problem would be putting the linearization points closer to each other, as suggested in the previous section. For the example of transmission line circuit model with quadratic nonlinearities, it has been found that such approach leads to a reasonable number of suitably distributed linearization points.

Modifying weights w_i is just one possible approach for enforcing negative definiteness of TPWL system (5.11), which in turn implies exponential stability of its equilibrium at the origin. In a

different approach one may add a dissipative term $(-\kappa(z)z^T z)$ to the right hand side of (5.11). The value of $\kappa(z)$ can be then selected in such a way that condition

$$z^T \cdot \left(\sum_{i=0}^{s-1} w_i(z) \left[V^T f(x_i) + V^T A_i (Vz - x_i) - \kappa(z) z \right] \right) \leq -k_3 z^T z \tag{5.21}$$

holds for all z. Again, one may specify $k_3 = \lambda$, in order to mimic condition (5.7) for the original nonlinear function f (or condition (5.10) for the corresponding projected function).

5.1.6 L_p stability of TPWL models

Let us now consider the following forced TPWL system:

$$\begin{cases} \dot{z} = \sum_{i=0}^{s-1} w_i(z) \left[V^T f(x_i) + V^T A_i V z - V^T A_i x_i) \right] + V^T B u \\ y = C^T V z \end{cases} . \tag{5.22}$$

Using Theorem 1 which links exponential stability with L_p stability, and Theorem 3, we formulate the following corollary:

Corollary 1: Suppose $f : R^N \to R^N$ is Lipschitz continuous and differentiable, $f(0) = 0$, and f satisfies condition

$$\exists_{\lambda > 0} \forall_x \; x^T f(x) \leq -\lambda x^T x.$$

Suppose also, that all linearization points x_i can be represented *exactly* in orthonormal basis V, i.e. $x_i = V z_i$ for $i = 1, \ldots, (s-1)$. Then, system (5.22) is L_p stable for any initial condition x_0, if for all $i = 1, \ldots, (s-1)$, $w_i(z) = 0$ for $z \in (R^q \backslash B_{\varepsilon_i}(z_i))$, where

$$\varepsilon_i \leq \left(\sqrt{\frac{\tilde{\lambda}^2}{W^2} + \frac{2\tilde{\lambda}}{W} \|x_i\|_2} - \frac{\tilde{\lambda}}{W} \right), \tag{5.23}$$

and $\tilde{\lambda}$ is any positive constant such that $\tilde{\lambda} < \lambda$, $W = \sup_x \|W(x)\|_2$, $W(x)$ is the Hessian of f evaluated at x, and $B_{\varepsilon_i}(z_i) = \{z \in R^q : \|z - z_i\| \leq \varepsilon_i\}$.

Proof: This corollary follows directly from Theorems 1 and 3. We only need to check that assumptions of Theorem 1 hold. First, the appropriate Jacobians are bounded, since $\|B\|_2$, and $\|A_i\|_2$ are finite, and $[\partial w_i(x)/\partial x] \equiv [0]$ provided $\|x\|_2$ is large enough, since we consider only a finite number of models (s), and ε_i must be finite for all $i = 1, \ldots, (s-1)$. Moreover, for all z, $\|y(z)\|_2 = \|C^T V z\|_2 \leq \|C^T V\|_2 \|z\|_2$. Therefore, assumptions of Theorem 1 are satisfied. ■

5.2 Passivity analysis

In this section we present a basic result on passivity of TPWL reduced order models, which follows from the stability analysis presented in the previous sections. Let us first define passivity for a nonlinear dynamical system:

Passivity: System (5.2) is passive if there exists a continuously differentiable positive semidefinite function $V : R^N \rightarrow R$ such that

$$\forall_{x \in R^N} \forall_{u \in R^M} \; u^T y \geq \frac{\partial V}{\partial x} f(x,u) + \delta y^T y,$$

for some $\delta \geq 0$. It $\delta > 0$ then we call system (5.2) output strictly passive.

Theorem 4: Suppose we consider the following nonlinear system:

$$\begin{cases} \dot{x} = f(x) + Bu \\ y = C^T x \end{cases} . \tag{5.24}$$

Then, the above system is output strictly passive provided $C = B$, and

$$\exists_{\lambda > 0} \forall_x \; x^T f(x) \leq -\lambda x^T x.$$

Proof: We consider a simple quadratic function $V(x) = x^T x$. This function is clearly positive semidefinite. We compute:

$$u^T y - \frac{\partial V}{\partial x}(f(x) + Bu) = y^T u - x^T f(x) - x^T Bu =$$

$$y^T u - x^T f(x) - x^T Cu = -x^T f(x) \geq \lambda x^T x \geq 0.$$

This means that system (5.24) is passive. Furthermore, since $y^T y = x^T CC^T x \leq \|CC^T\|_2 x^T x$ then from the above we obtain:

$$u^T y \geq \frac{\partial V}{\partial x}(f(x) + Bu) + \lambda x^T x \geq \frac{\partial V}{\partial x}(f(x) + Bu) + \frac{\lambda}{\|CC^T\|_2} y^T y.$$

Consequently, system (5.24) is output strictly passive. ■

Theorem 4 immediately implies the following corollary:

Corollary 2: Suppose assumptions of Corollary 1 are satisfied. Then the trajectory piecewise-linear system (5.22) is output strictly passive provided $C = B$.

Proof: Since assumptions of Corollary 1 hold, then condition (5.14) is satisfied – cf. proof of Theorem 3. This means that assumptions of Theorem 4 are satisfied for $f = g$, where

$$g(z) = \sum_{i=0}^{s-1} w_i(z) \left[V^T f(x_i) + V^T A_i V z - V^T A_i x_i) \right] .$$

From Theorem 4 it follows immediately that system (5.22) is output strictly output passive. ■

Chapter 6

Application and validation of the TPWL MOR algorithm

Theoretical analysis, presented in Chapters 4 and 5 offers important, yet sometimes limited insight to properties of the proposed Trajectory Piecewise-Linear (TPWL) model order reduction strategy. Therefore, in this chapter we focus on discussing results of numerical experiments with TPWL reduced order models, extracted for examples of strongly nonlinear systems described in Section 2.2. Those computational results provide additional insight to some of the features of the discussed MOR strategy, validate it for the considered application examples, and present its potential.

6.1 Transient simulations

This section focuses on presenting results of time-domain simulations for the considered examples of nonlinear systems. The main goal of the performed tests was to find out whether the proposed TPWL MOR technique really generates *a model* of the original nonlinear system. Recall that a TPWL reduced order model, consisting of a collection of linearized models is obtained by simulating the original system for a *given* training input signal, and then by selecting linearization points from the corresponding training trajectory. In order to show that we have indeed generated a model of the original system we should verify that it gives correct outputs not only for the input it was generated with (i.e. the training input), but also for other inputs. This verification was done experimentally for different application examples (nonlinear analog circuits and a micromachined device) by performing simulations of the TPWL reduced order models for selected testing input signals.

Results of transient simulations presented in this section also illustrate some of the difficulties which arise in reduced order TPWL modeling. We use examples of the micromachined device and fluid flow problems to briefly discuss those difficulties, mainly related to the projection stage of the TPWL MOR approach.

6.1.1 Modeling analog circuits

First, we considered the nonlinear transmission line circuit model (shown in Figure 2-1, in Section 2.2) with $N = 100$ nodes, and generated its reduced order TPWL model of order $q = 10$ using a step

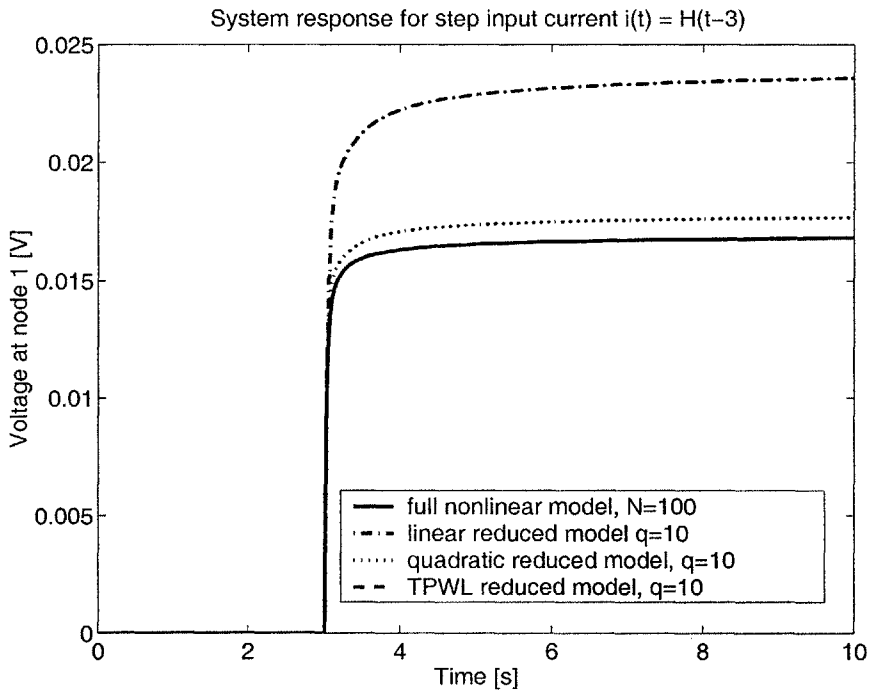

Figure 6-1: Comparison of the system response (nonlinear transmission line circuit model) computed with linear, quadratic, and TPWL reduced order models (of order $q = 10$) for the step input current $i(t) = H(t-3)$ ($H(t) \equiv 1$ for $t > 3$ and $H(t) \equiv 0$ for $t < 3$). The TPWL model was generated using a unit step input current.

training input $i(t) = H(t-3)$. For this example, the linearization point changed 4 times, therefore our model used 5 reduced order Jacobians A_{0r}, ..., A_{4r}. The reduced order model was tested for different inputs, including the step input used to generate it. Figure 6-1 shows the computed transient for the step input (the training input).

Figures 6-2 and 6-3 show the reduced order simulation results for a cosinusoidal input, and an exponentially decaying input, respectively. In all the cases the output voltages obtained with the TPWL reduced order model accurately approximate the reference voltages (the curves overlap almost perfectly).

The results shown verify that a TPWL reduced order model, generated for a step training input current, can be successfully used to model responses of the original nonlinear circuit for inputs which are very different from the step signal, i.e. cosinusoidal and exponentially decaying signals.

Figures 6-1, 6-2, and 6-3 also provide a comparison of the proposed TPWL reduced order model with linear and quadratic reduced models, generated using methods described in [17], [83] and [110] (cf. also Section 2.3.1). It is apparent from the graphs that the TPWL reduced order model gives significantly more accurate results than the linear and quadratic reduced order models using Taylor's expansion about the initial state. One may note that, for the case of the step input current, the linear and quadratic reduced order models are unable to accurately compute the steady state of the system. It should be stressed at this point that all the models (linear, quadratic and TPWL) were of the same order and, moreover, applied the same projection basis V (obtained with the procedure described in Section 3.2.2).

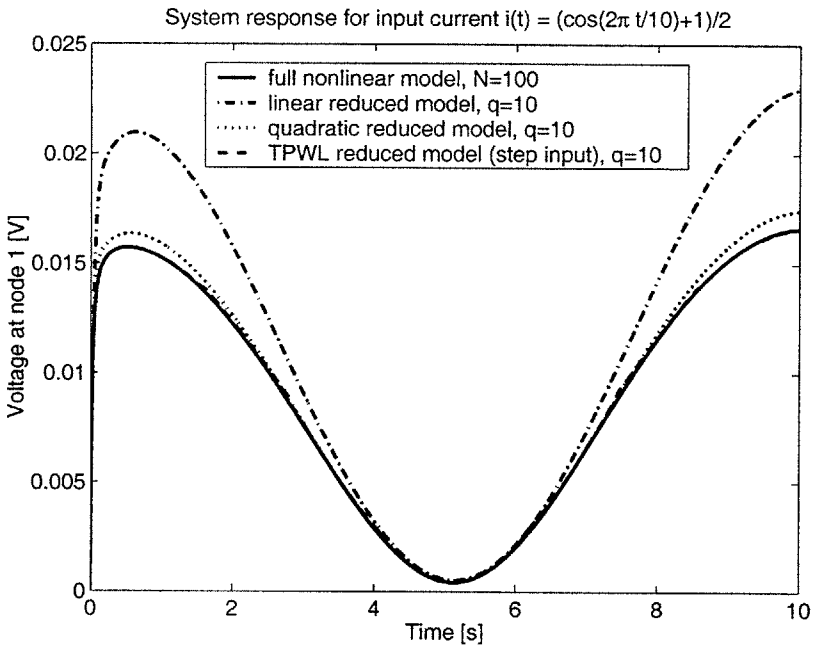

Figure 6-2: Comparison of the system response (nonlinear transmission line circuit model) computed with linear, quadratic, and TPWL reduced order models, for the input current $i(t) = (\cos(2\pi t/10) + 1)/2$. The TPWL model was generated using a unit step input current.

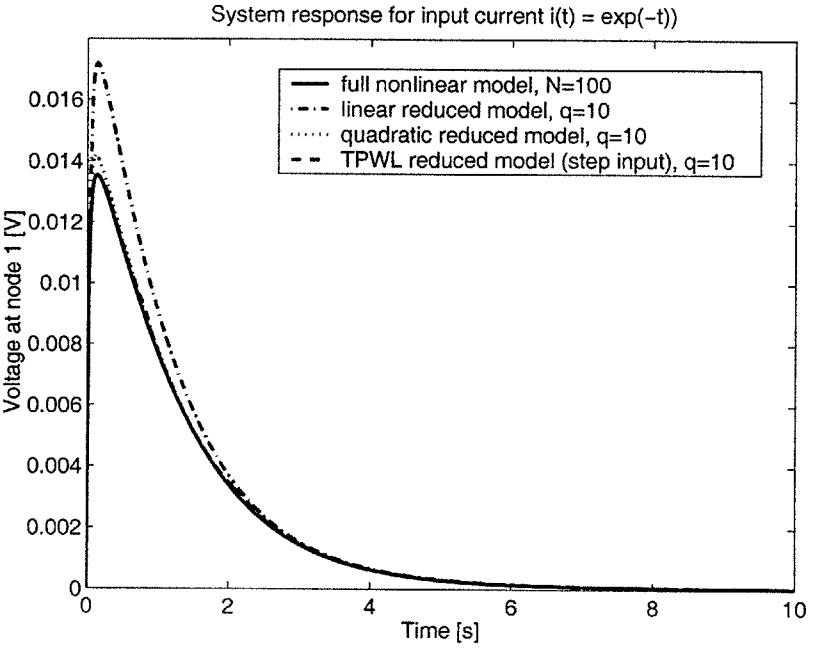

Figure 6-3: Comparison of system response (nonlinear transmission line circuit model) computed with linear, quadratic, and TPWL reduced order models, for the input current $i(t) = \exp(-t)$. The TPWL model was generated using a unit step input current.

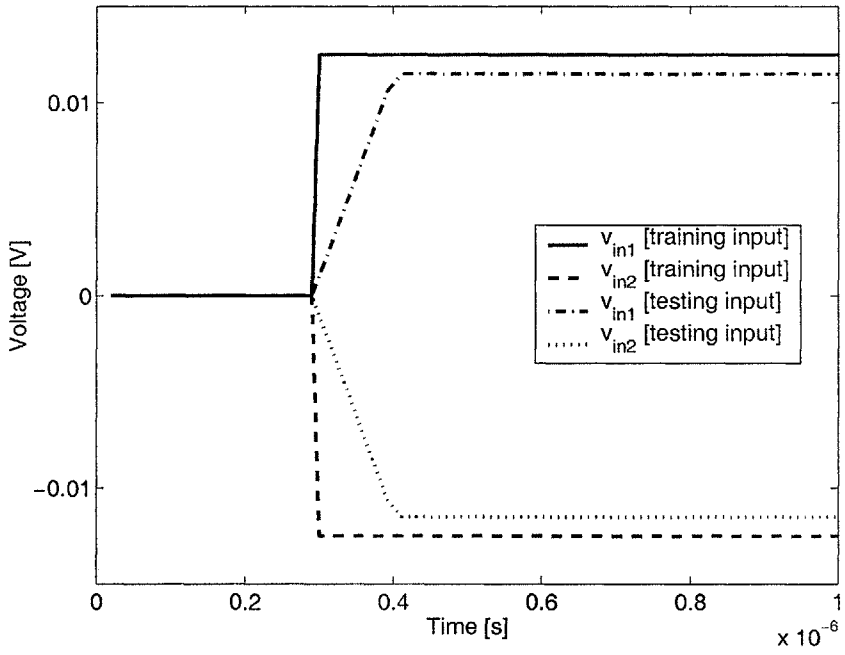

Figure 6-4: Op-amp input signals v_{in1} and v_{in2}.

The discussed example of the nonlinear transmission line circuit model has also been used in [6] to test a model order reduction approach based on Carleman bilinearization. In [6], a bilinear model of order $q = 17$ has been generated for the original circuit with $N = 100$ nodes. The graph included in the cited paper, showing system's response to a step input signal clearly indicates inferior accuracy of the bilinear model of order $q = 17$, as compared to a TPWL model of order $q = 10$ (cf. Figure 6-1).

Next, we considered the operational amplifier example, described in Section 2.2. The examined circuit had $N = 51$ nodes and eight inputs: 1) the differential input with input signals v_{in1} and v_{in2}, 2) the auxiliary inputs v_{cmmrst}, v_{gnd}, v_{intn}, v_{intp}, v_{rst}, and v_{cmmin} used in common mode rejection testing. We considered two full-order models of the op-amp: 1) simplified – employing linearized capacitance models for MOSFETs (Then, $g(x) = Gx$ – cf. equation (2.1).); 2) regular – employing nonlinear capacitance models (i.e. with a state-dependent capacitance matrix $G(x) = dg(x)/dx$). In both cases the full nonlinear simulations were performed using NITSWIT circuit simulator, described in [54] and [55]. In order to generate the reduced order TPWL models we applied the following set of training inputs:

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

$v_{in2} = -v_{in1}$ (cf. Figure 6-4), and auxiliary input signals shown in Figure 6-5.

For the case of the system with a linearized capacitance matrix we obtained a TPWL model of order $q = 18$ (with 35 linearization points, and 8 inputs), and for the case of the system with nonlinear capacitance we obtained a model of order 34 (with 29 linearization points). As one may note, due to an increased complexity of the applied MOSFET device models, the order of the

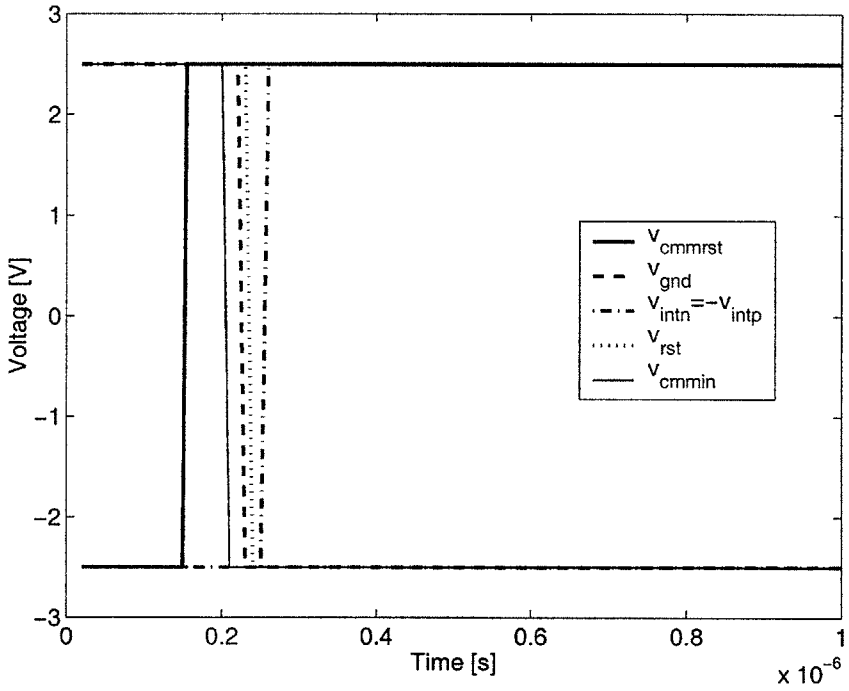

Figure 6-5: Auxiliary inputs for the op-amp circuit.

resulting reduced order model is higher in the second case.

The obtained reduced order TPWL models were then tested for the following piecewise-linear input (cf. Figure 6-4):

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

and $v_{in2} = -v_{in1}$. Figures 6-6 and 6-7 show a comparison of the transients computed with NITSWIT, and with the reduced order TPWL models for one (of the two) output nodes of the amplifier. One may note excellent agreement of the output signals for both cases, which indicates that suitable reduced order TPWL models of the original systems have been extracted. The results also indicate that the proposed MOR method may be successfully used for multiple input systems with state-dependent matrix $G(x)$. It is important to point out that not only do the TPWL models have a lower order than the original system, but also they are much easier to use. Since a TPWL model consists of a weighted combination of linear models, the time-stepping is very straightforward. In a simplified backward Euler time-stepping scheme (cf. Section 3.1.2) we compute weights w_i (cf. equation (3.4)) e.g. using the previous state of the system, or a predictor of the next state and then, assuming that these weights are fixed, we find the state at the next timestep by performing only a single Newton update (i.e. solving a low order linear system of equations). In a more sophisticated time stepping scheme, one can account for derivatives of $w_i(x)$, which is also straightforward, since the weights are simple scalar functions. In a regular simulator, if using backward Euler scheme, one needs to compute a number of Newton updates for the full order nonlinear system at each timestep, which is considerably more complex.

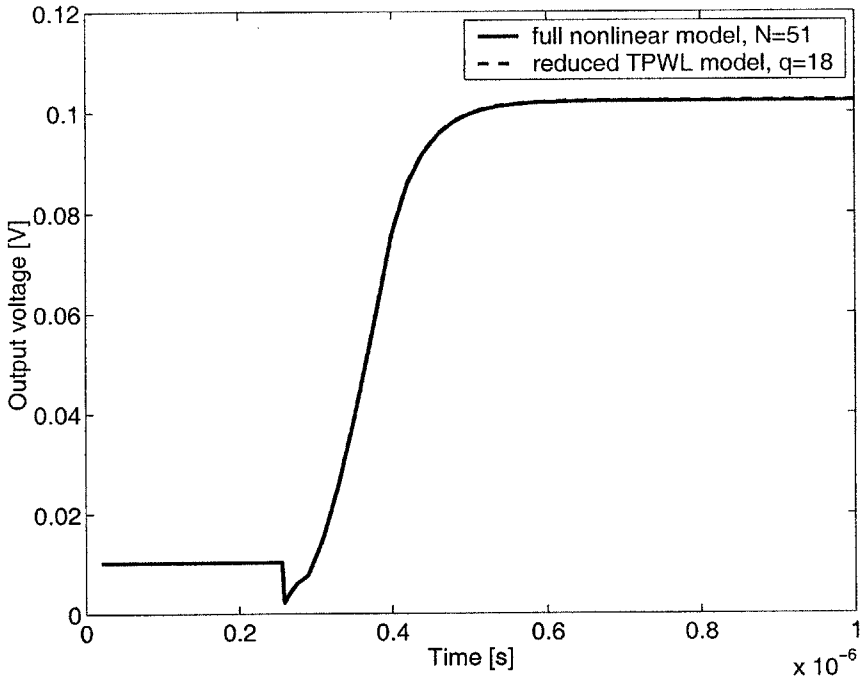

Figure 6-6: Comparison of the output voltage (op-amp example, simplified linearized capacitance models), computed with a reduced order TPWL model and NITSWIT circuit simulator.

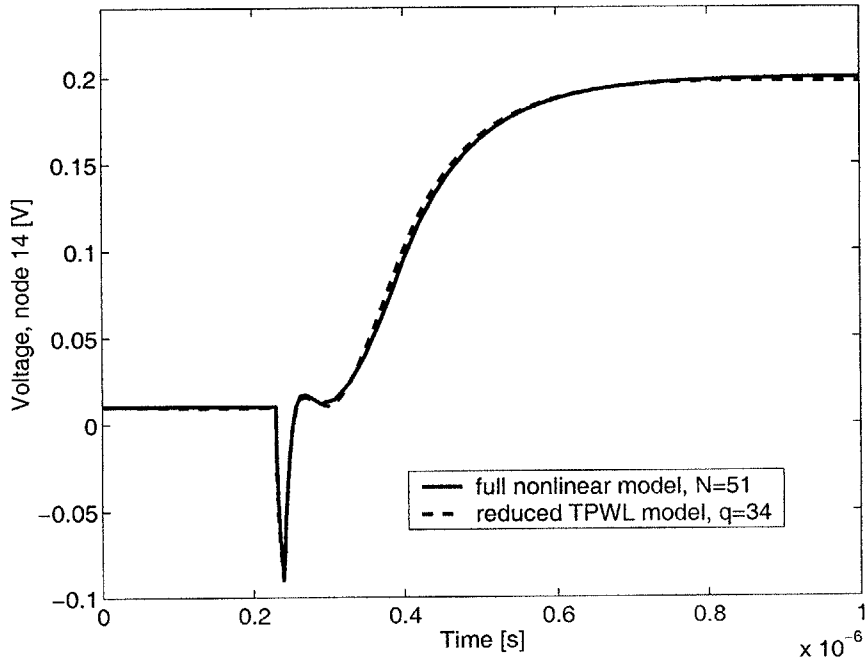

Figure 6-7: Comparison of the output voltage (op-amp example, regular nonlinear capacitance models), computed with a reduced order TPWL model and NITSWIT circuit simulator.

6.1.2 Modeling MEMS

In order to validate the proposed TPWL MOR approach also for the example of the micromachined switch described in Section 2.2, we performed analogous tests as in the previous section. For this example we considered two different algorithms of generating the reduced order basis: the simple one, described in Section 3.2.2, and the extended one, introduced in Section 3.2.3. In the first case the reduced model of order $q = 41$ (with $s = 9$ linearization points) was generated for an 8-volt step training input voltage. In the second case, we generated a TPWL reduced model of order $q = 26$ (with $s = 11$ linearization points), for a 9-volt step training input voltage. Both models were tested for a cosinusoidal input with a 7-volt amplitude. Figures 6-8 and 6-9 show results of the simulation. The transients obtained with the TPWL models match very accurately the reference result obtained with the full nonlinear model of order $N = 880$. Once again, it results that a TPWL model may be used for inputs which are dynamically different from the training input. Note that in this case the testing input signal, not only had a very different frequency content, but also a different amplitude, as compared to the training signal. The graphs also show that the TPWL models offer superior accuracy as compared to linear and quadratic reduced order models. Also note that the described TPWL reduced order model of order $q = 26$ was also successfully used to perform periodic steady state simulations, as shown in Section 6.2.

Yet another test has been performed for the example of the micromachined switch packed in air under low pressure of $p_0 = 0.01$atm. In this case, the layer of air provides a much lighter damping for the beam, and consequently oscillations of the beam occur for a step input voltage (cf. Figure 6-10).

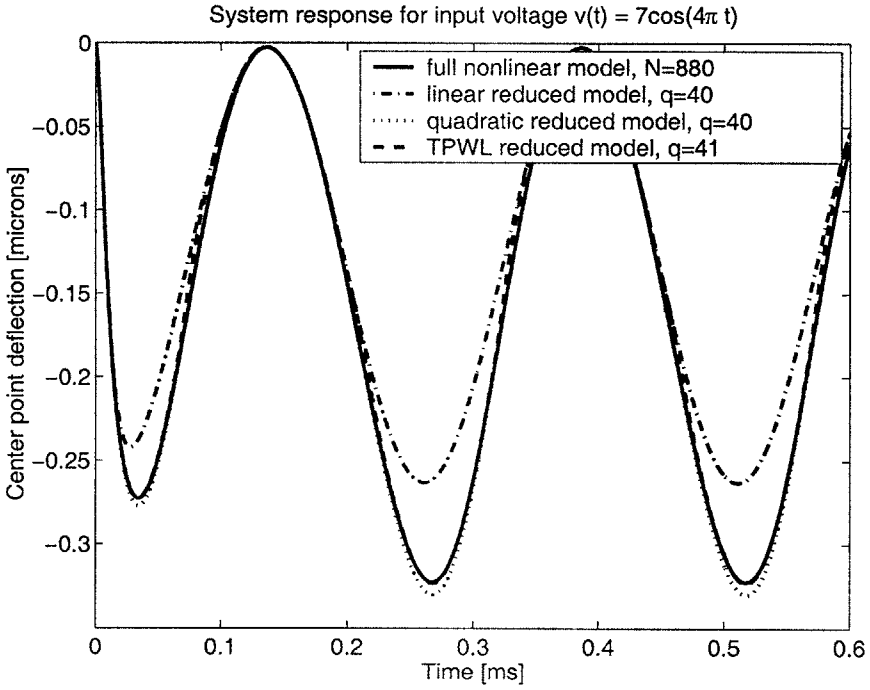

Figure 6-8: Comparison of the system response (micromachined switch example) computed with linear, quadratic, and TPWL reduced order models (of order $q = 40$ and $q = 41$) for the input voltage $u(t) = 7\cos(4\pi t)$. The TPWL model was extracted for an 8-volt step input voltage.

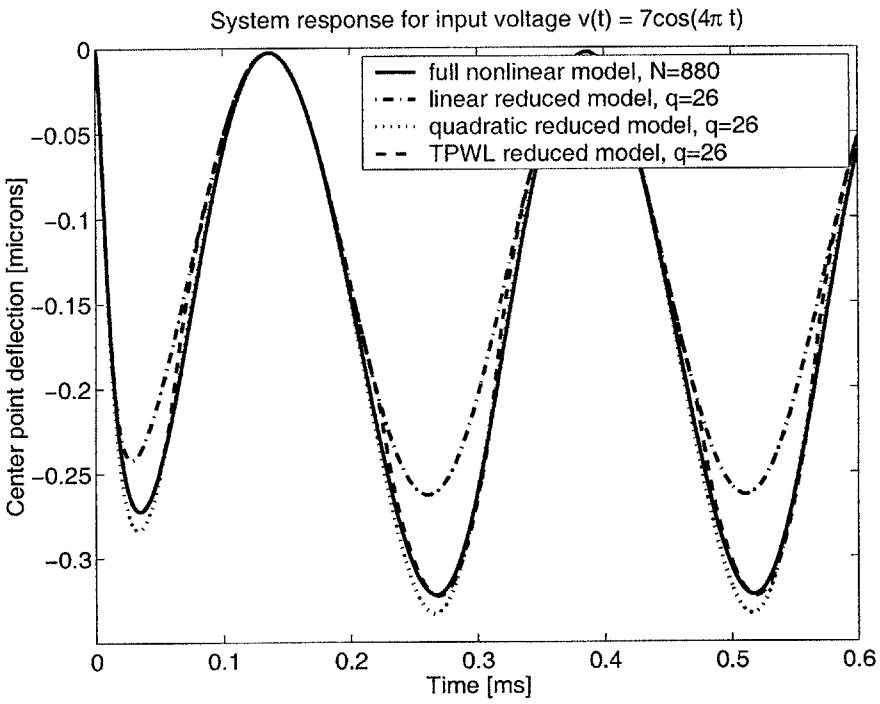

Figure 6-9: Comparison of the system response computed using the linear, quadratic, and TPWL reduced order models, for the input signal $u(t) = (7\cos(4\pi t))^2$.

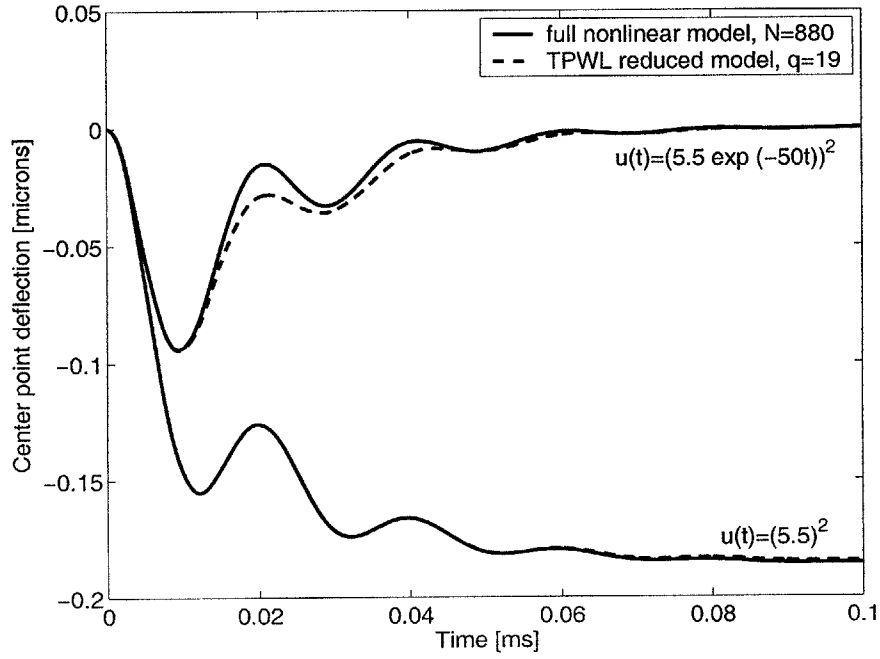

Figure 6-10: Comparison of the system response (for air pressure $p_0 = 0.01$atm) computed with the full order model, and a TPWL model of order $q = 19$.

In this case a TPWL reduced model of order $q = 19$ (number of linearizations $s = 12$) has been generated using the extended algorithm for constructing the projection basis. The training input was a 5.5-volt step input voltage. The TPWL model was then tested for $u(t) \equiv 5.5^2$, and $u(t) = (5.5\exp(-50t))^2$ input signals. As shown in Figure 6-10, for the first input excellent agreement with the full nonlinear model is observed, while for the second input some error in the computed transient occurs.

Further results validating TPWL models for the micromachined switch example have already been presented in Section 3.2.1, and also can be found in Section 7.2. The mentioned results also show some limitations of TPWL models. In particular, it has been found that when the amplitudes of the testing inputs differed too much from the amplitude of the training input, the corresponding TPWL model was unable to provide adequate accuracy.

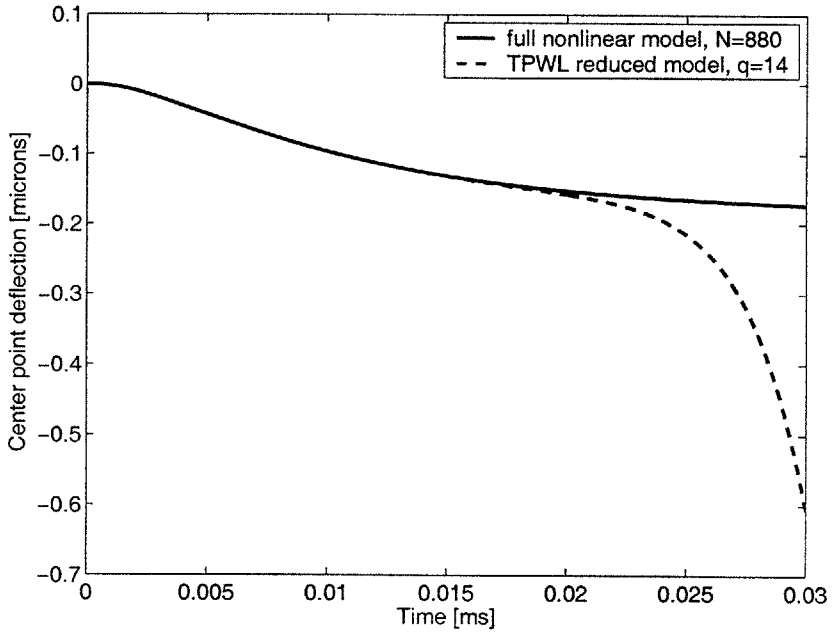

Figure 6-11: Instability in a TPWL model of order $q = 14$. Both training and testing input signals equaled a 5-volt step voltage.

Also, serious problems with stability of TPWL models have been observed for some values of q. Although the original nonlinear system is clearly stable and dissipative (passive), some of the TPWL reduced order models show very unstable behavior. For instance, a TPWL model of order $q = 14$ (with 11 linearizations, generated and tested for a step 5.5-volt input voltage) is unstable (the solution tends to $-\infty$), as shown in Figure 6-11. This instability is due to the fact that, for some orders q of the reduced TPWL models, either some or all of the reduced Jacobians A_{ir} are unstable. As checked, all the full-order Jacobians A_i are Hurwitz, and they 'lose' stability during the reduction process. This occurs because the original Jacobians A_i, although stable are nonsymmetric and indefinite. Consequently, the corresponding reduced order Jacobians, obtained with congruence transforms (and Krylov-based reduction) are not guaranteed to be stable. It is also interesting to note that if full-order Jacobians are used in a TPWL model, instead of the reduced ones, we obtain a correct, stable transient. This clearly indicates that the main problem is ensuring stability of the reduced Jacobian matrices. So far, this problem remains open. It appears though

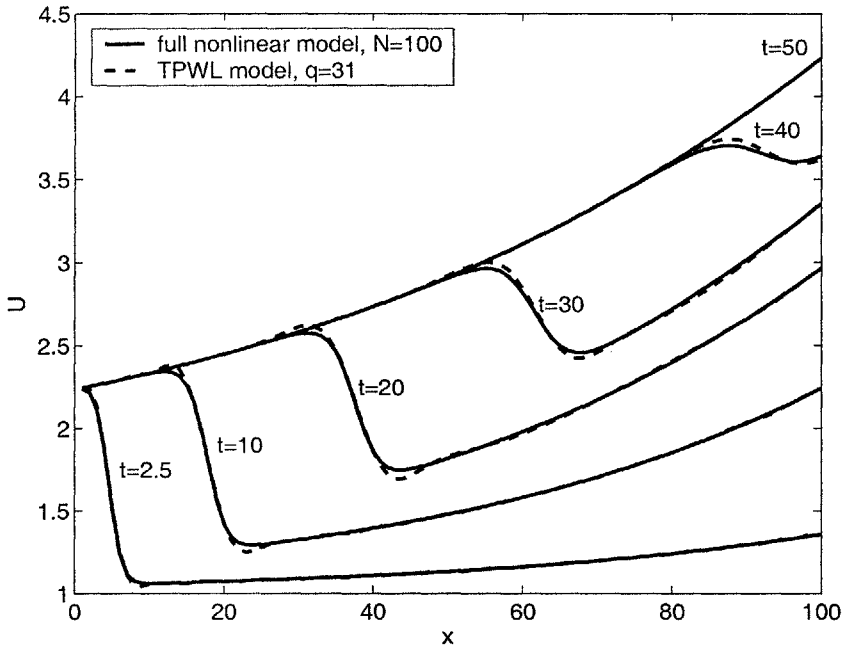

Figure 6-12: Shock movement in 1D, $N = 100$.

that it may be solved only by applying some new reduced order projection scheme. Please note that strategies for stabilizing TPWL systems presented in Chapter 5 cannot be applied in this case, since the original system's nonlinear function does *not* satisfy negative definiteness condition (5.7).

6.1.3 Modeling fluid flow problems

The other two application examples, introduced in Section 2.2.3 are associated with shock movement in a fluid, which requires using nonlinear models. The first of the examples refers to shock propagation in 1D, as modeled by Burgers' equation. We considered discretized systems of order $N = 100$ (length of the region $l = 100$, $\Delta x = 1$), and $N = 1000$ (length of the region $l = 100$, $\Delta x = 0.1$). We have generated TPWL reduced order models of order $q = 31$ (with $s = 21$ linearized models) for the first case, and $q = 38$ (with $s = 24$ linearized models) for the second case. Then, we tested our models for a steady incoming flow $u(t) \equiv \sqrt{5}$. The results of the reduced order simulations and their comparison to the respective results obtained with full order model are shown in Figures 6-12 and 6-13. One may note that for $N = 100$ the reduced order TPWL model is able to accurately approximate both shock propagation and its shape. However, for $N = 1000$, while dynamics of the shock is modeled correctly, its shape cannot be represented accurately. Due to a very large slope associated with the shock a phenomenon similar to Gibbs' effect occurs if a reduced order model is used. We have observed that if a full-order TPWL model is used instead the discussed effect disappears. This indicates that the problem lies in an inability to represent a shape with a large slope in the reduced order basis, similar to an inability of accurately representing a step function by a small number of Fourier modes. The problem persisted for all the considered projection bases, generated with different MOR strategies, and it appears to be difficult to overcome without applying additional solution smoothening techniques.

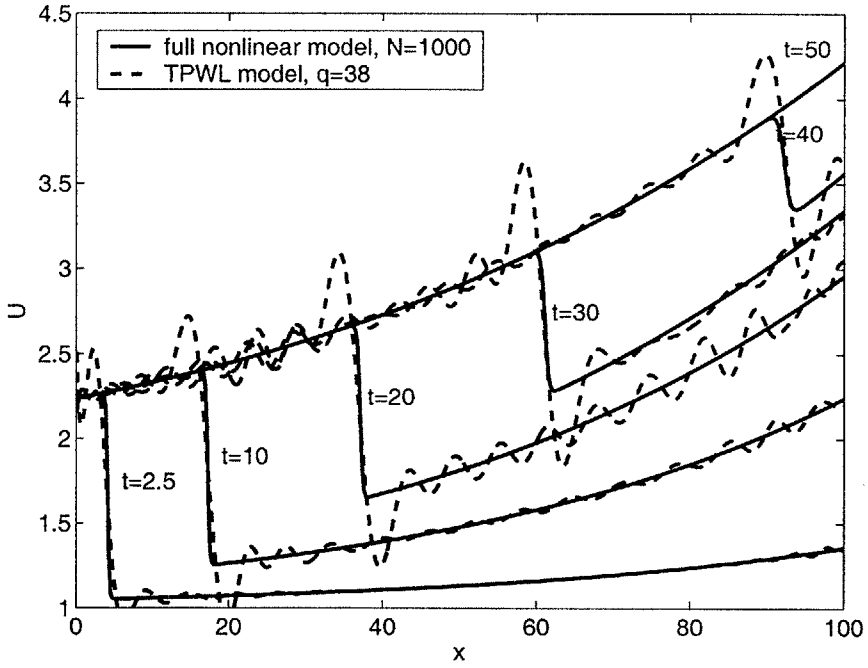

Figure 6-13: Shock movement in 1D, $N = 1000$.

Next, we considered the problem of modeling shock dynamics in the jet engine inlet described in Section 2.2.3. In order to apply the piecewise-linear representation introduced in Section 3.1 we select linearizations of nonlinear dynamical system (2.15) at states U_i and for zero input disturbances $\dot{m} = d = 0$[1]:

$$E_i \frac{d\underline{U}}{dt} = -R(\underline{U}_i, 0, 0) - A_i(\underline{U} - \underline{U}_i) - B_{mi}\dot{m} - B_{di}d, \tag{6.1}$$

where $E_i = dF(\underline{U}_i)/d\underline{U}$, $A_i = \partial R(\underline{U}_i, 0, 0)/\partial \underline{U}$, $B_{mi} = \partial R(\underline{U}_i, 0, 0)/\partial \dot{m}$, and $B_{di} = \partial R(\underline{U}_i, 0, 0)/\partial d$. Since the output, selected as the average Mach number at the throat, is also a nonlinear function of the state $\underline{U}$, we also consider linearizations of the output equation $y = C(\underline{U})$ about states $\underline{U}_i$:

$$y = C(\underline{U}_i) + C_i(\underline{U} - \underline{U}_i), \tag{6.2}$$

where $C_i = dC(\underline{U}_i)/d\underline{U}$. Taking a weighted combination of equations (6.1) and (6.2) leads to a TPWL model in the form:

$$\left(\sum_{i=0}^{s-1} \tilde{w}_i(x) E_i\right) \frac{d\underline{U}}{dt} = \sum_{i=0}^{s-1} \tilde{w}_i(x) \left[-R(\underline{U}_i, 0, 0) - A_i(\underline{U} - \underline{U}_i) - B_{mi}\dot{m} - B_{di}d\right]$$

$$y = \sum_{i=0}^{s-1} \tilde{w}_i(x) \left[C(\underline{U}_i) + C_i(\underline{U} - \underline{U}_i)\right]. \tag{6.3}$$

[1]Note that, since in the original nonlinear system $R(\cdot)$ is a nonlinear function of the inputs, we need to consider linearization both at a given state U_i and for fixed values of the inputs.

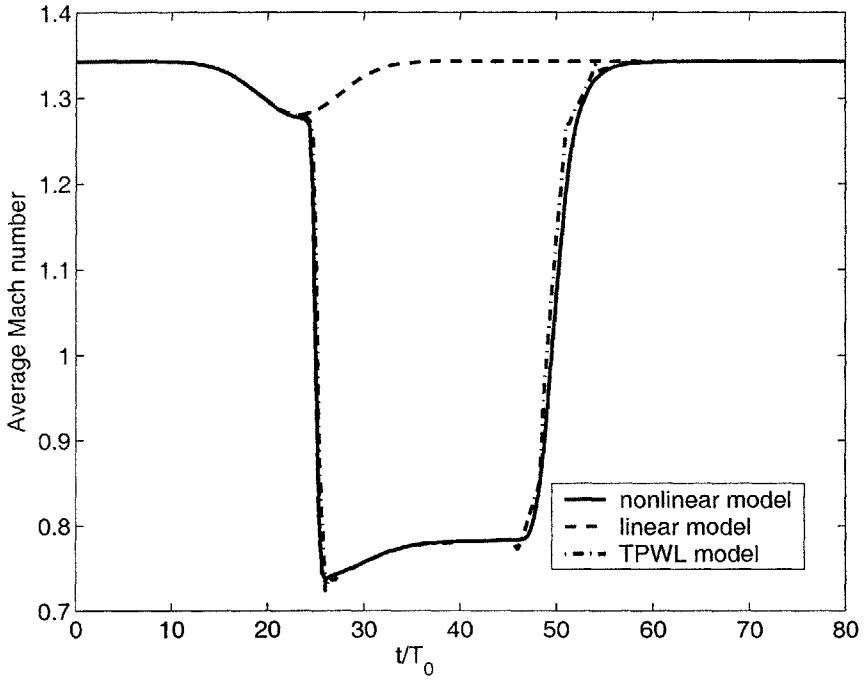

Figure 6-14: Average Mach number at the throat of the engine inlet vs. time, computed with full-order nonlinear, linear, and TPWL models. $N = q = 11730$.

Note that the form of the above TPWL system is slightly more general the form of TPWL systems discussed so far (cf. e.g. equation (3.4)), due to the nonlinear output equation.

In order to find out whether model (6.3) can be used to simulate strongly nonlinear behavior of the original nonlinear system, we first generated a *full-order* TPWL model ($q = N = 11730$). To this end we applied the following input signals: 1) inlet density disturbance:

$$d = -\beta \exp(-g(t - t_p/f_0)^2),$$

where $\beta = 0.03$, $g = 0.03f_0^2$, $t_p = 20$, and $f_0 = 3.4255$; 2) bleed mass flux $\dot{m} = 0$. For the above inlet density disturbance, shock movement at the throat is observed. Consequently, the operating state of the modeled nonlinear system changes very substantially, and linearized models generated at the steady state of the system are unable to capture the strongly nonlinear behavior associated with shock propagation. The above input has been used first as a training input, to generate a TPWL model of order 11730, with $s = 28$ linearizations, and then as a testing input. The results of the simulation are shown in Figure 6-14. Clearly, the TPWL model provides very good approximation of the response for the full nonlinear system, while the linearized model is unable to capture the effect of the moving shock.

Problems begin when we try to perform reduction of the TPWL system (6.3) using a projection basis generated with Krylov-based methods. According to our experience the reduced order models generated with projection algorithms presented in Section 3.2 fail to provide accurate ap-

proximations of the transient shown in Figure 6-14 for all tested orders. Analogous observations are reported in [60] for the linearized reduced models extracted with Krylov subspace methods. In the cited reference, Lassaux claims that the main cause of the observed problems with order reduction is the discrete formulation of the dynamical system which incorporates algebraic relations defining the boundary conditions as separate equations to system (6.1) (which also causes matrices E_i to be singular). In [60], a 'condensed' formulation of system (6.1) is proposed, in which equations related to boundary conditions are eliminated, and a change of variables is performed. If Krylov reduction is applied to the 'condensed' formulation of system (6.1), one obtains reduced models which correctly approximate the original system around the linearization state, as reported in [60] and also found during our own numerical tests. Unfortunately, the change of variables leading to the 'condensed' formulation, varies from one linearization point to another. Consequently, state representations vary across the collection of linearization points, which makes application of quasi-piecewise-linear scheme (in the form presented in Chapter 3) impossible. In order to be able to use this collection of linearized models, with each one employing a different state representation and a different projection, one needs to perform suitable translations of states during the piecewise-linear simulation. This approach is currently under investigation, in the context of the presented problem of supersonic flow in the jet engine inlet.

6.2 Periodic steady state simulations

In order to further validate the TPWL reduced order models we performed a series of periodic (sinusoidal) steady state simulations. In this section we present a comparison of periodic steady states computed with the full order nonlinear models and the reduced order TPWL models. The tests were performed for the examples of a nonlinear transmission line circuit model (cf. Figure 2-1), the operational amplifier, and the micromachined switch, described in Section 2.2.

In the first series of tests, we computed the sinusoidal steady state for the nonlinear transmission line model with $N = 1500$ nodes, excited with the input current $i(t) = (\cos(2\pi t) + 1)/2$. The simulation was performed using a simple, fixed-timestep shooting method [56]. Figure 6-15 shows the computed sinusoidal steady state output of the system in the time domain (output $y(t) = v_1(t)$, where $v_1(t)$ is the voltage at node 1 of the circuit). One may note that the result obtained with the reduced order model of order $q = 30$ closely matches the reference result.

The frequency domain analysis has also been performed for the sinusoidal steady state output signal $y(t)$ shown in Figure 6-15. We computed a complex DFT of the discrete output signal $\hat{y}(n) = y(n\Delta t)$. The first four Fourier coefficients c_k (not normalized) are shown in Table 6.1. One may note that the sinusoidal steady state $y(t)$ obtained with the discussed reduced order model matches closely up to the $3rd$ harmonic of the reference sinusoidal steady state for the considered nonlinear transmission line model. This result suggests that the extracted reduced order models may be used to analyze second-order effects like harmonic distortion.

We have performed analogous tests for the micromachined switch example. We used a TWPL reduced order model of order $q = 26$, generated for a 9-volt step input voltage to find the sinusoidal steady state of the micromachined switch. To this end we applied the shooting method with our reduced model, and then computed harmonics of the resulting periodic signal. Table 6.2 compares the first three harmonics computed with the reduced order model and the full nonlinear model. Also for this application example one notes good agreement in the computed harmonics.

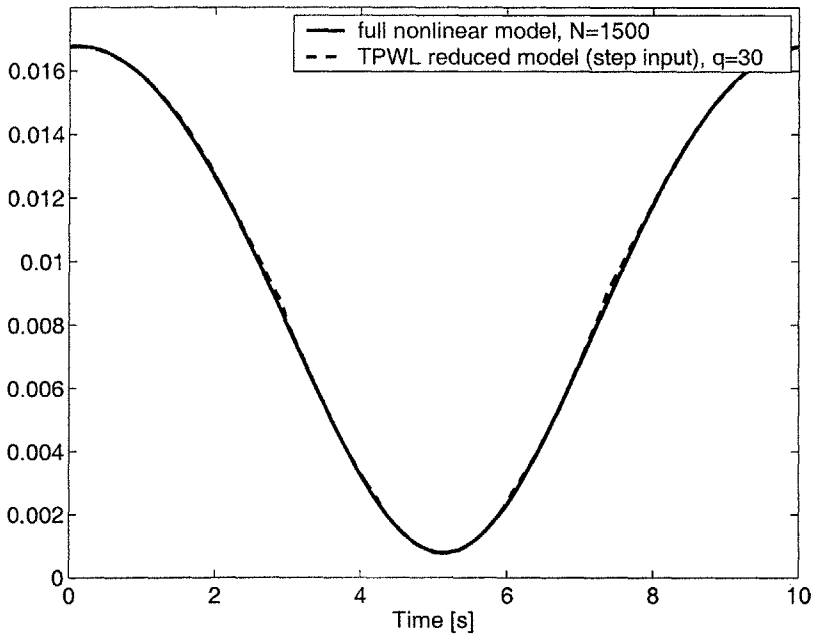

Figure 6-15: Comparison of the sinusoidal steady state (nonlinear transmission line circuit model) computed using the full nonlinear and the reduced TPWL model, for the input signal $i(t) = (\cos(2\pi t) + 1)/2$.

Table 6.1: Comparison of the subsequent harmonics of the sinusoidal steady state, computed using the full order nonlinear model, and the reduced order TPWL model.

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

Table 6.2: Comparison of the subsequent harmonics of the sinusoidal steady state, computed using the full order nonlinear model, and the reduced order TPWL model. The input signal to the system was $u(t) = (9\cos(\pi t))^2$.

Harmo -nics	Full nonlinear model	Reduced order TPWL model	Error [%]
dc	1.9587	1.9526	0.4
1st	-0.1967+0.0351i	-0.1935+0.0352i	1.6
2nd	-0.0289+0.0283i	-0.0263+0.0263i	8.2
3rd	0.0004+0.0143i	-0.0017+0.0155i	17.6

Table 6.3: Comparison of the main intermodulation harmonics of the sinusoidal steady state, computed using the full order nonlinear model, and the reduced order TPWL model ($q = 39$).

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

Table 6.4: Comparison of the main intermodulation harmonics of the sinusoidal steady state, computed using the full order nonlinear model, and the reduced order TPWL model ($q = 35$).

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

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

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

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

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

where $f_1 = 1$MHz and $f_2 = 1.2$MHz, and the spectrum of the computed sinusoidal steady state was extracted using DFT. Tables 6.3 and 6.4 show the complex amplitudes of the main intermodulation products and the driving harmonics obtained with the TPWL reduced order models, and the full

nonlinear model of the considered op-amp. The error shown in the tables is the relative error of the computed complex amplitude. The results indicate that the TPWL reduced models are able to effectively reproduce the intermodulation distortion effects in the considered case. At the same time, comparison between the two tables shows that the computed amplitudes of intermodulation products vary significantly depending on the applied TPWL model, which suggests that in this case we are close to the limits of applicability of the considered macromodels.

Chapter 7

Performance of TPWL MOR algorithms

In this chapter we consider performance of the proposed MOR technique, based on selecting a collection of linearizations from a state-space trajectory of the original nonlinear system. First, we discuss efficiency of both TPWL model extraction and time-domain simulations with the reduced order models. Then, we compare accuracy of TPWL reduced models, extracted with two different projection bases (generated with algorithms presented in Sections 3.2.2 and 3.2.3, respectively). Also, efficiency of model extraction in both cases is considered. Finally, we discuss performance and accuracy of the fast piecewise-linear simulator, introduced in Section 3.3.

7.1 Model extraction and time domain simulations

First, we considered efficiency of TPWL model extraction for the example of the nonlinear transmission line circuit model, described in Section 2.2. The original system had order $N = 1500$, and the corresponding reduced TPWL model of order $q = 30$ (with $s = 21$ linearization points) was extracted for a 1-volt step training input using the fast approximate simulator (cf. Section 3.3). Also, linear and quadratic reduced order models of the same order were extracted (with the expansion point at the initial equilibrium state). All the reduced models were tested for the same input signal, equal to the training input signal. Table 7.1 shows a comparison of execution times both for model extraction, and reduced-order simulation for all the discussed MOR techniques.

Table 7.1: Comparison of the times of model extraction, and reduced order simulation for the linear, quadratic, and TPWL MOR techniques. The original problem had size $N = 1500$. All the reduced model had size $q = 30$.

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

Table 7.2: Comparison of the times of model extraction, and reduced order simulation for the linear, quadratic, and TPWL MOR techniques ($N = 880$, $q = 26$).

MOR method	Model extraction time [s]	Simulation time [s]
linear	14.8	1.0
quadratic	3712.3	33.3
TPWL	293.5	8.3

One should note that all the extraction algorithms, and the reduced order solvers were implemented in Matlab. The tests were performed on a Linux workstation with Pentium III Xeon processor. One may note that performance for linear and TPWL MOR algorithms is comparable, while generation of the quadratic reduced model is significantly more expensive, due to the costly reduction of the Hessian matrix, which requires q^2 computations of the matrix-vector product $W(x \otimes x)$, where W is a full order $N \times N^2$ Hessian matrix (usually represented implicitly – cf. [17], or Section 3.1.1).

From the table it also follows that simulating the TPWL reduced model is more expensive than simulating the linear model, which is due to the fact that in a TPWL model it is necessary to compute a weighted combination of the Jacobians at every timestep. Nevertheless, the TPWL solver again outperforms the quadratic solver of the same order. One should note that for all the reduced order solvers the simulation time is much shorter than the corresponding simulation time for the full-order nonlinear solver, which equals 9573 seconds, or about 160 minutes (for Matlab implementation).

Analogous tests have been performed for the micromachined switch example. In this case the original nonlinear system had order $N = 880$, and the extracted reduced linear, quadratic, and TPWL models had order $q = 26$. The TPWL model was extracted for a 9-volt step input voltage, and the number of linearizations equaled 26. Table 7.2 shows model extraction times, as well as reduced order simulation times for this case. One may note that the performance pattern for the discussed MOR algorithms, and the corresponding reduced order solvers is analogous as in the previous example.

Let us briefly analyze the memory requirements for the TPWL reduced order solver. As already discussed in Section 3.1, the memory complexity for TWPL reduced models is $O(sq^2)$, where s is the number of linearization points. Consequently, this memory cost is roughly s times larger than the cost for the linear reduced order simulator (which is $O(q^2)$). The memory cost of the quadratic reduced order solver is $O(q^3)$ (the reduced order Hessian must be stored explicitly as a matrix), so if $s \approx q$, then the memory requirements for the TPWL solver are approximately the same as for the quadratic solver. For the examples of the nonlinear transmission line and the micromachined switch $s = 21 \approx (2/3)q$, or $s = 26 = q$, so in those cases the amount of memory used by the TPWL algorithm was comparable to the amount of memory used by the quadratic solver.

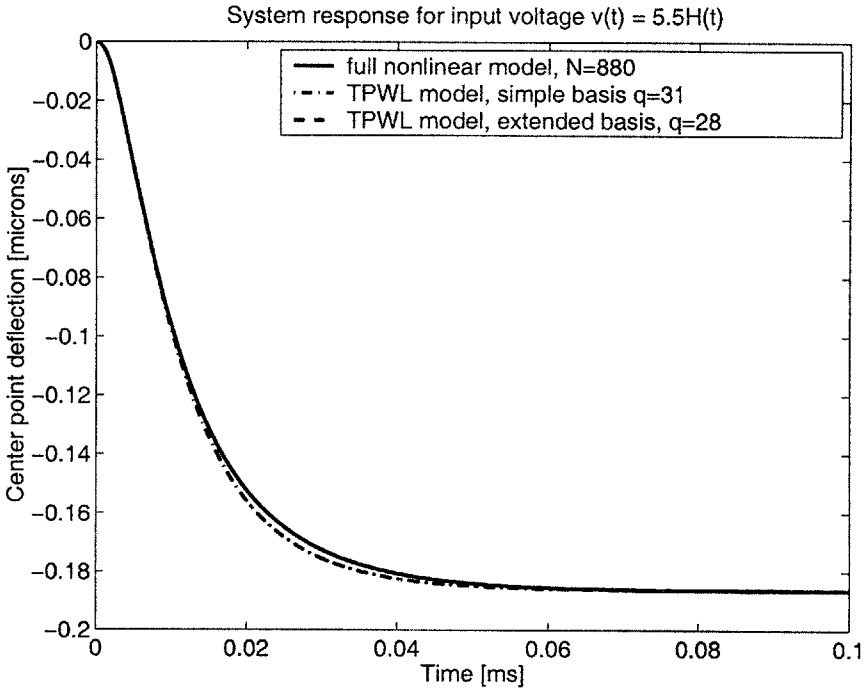

Figure 7-1: Comparison of the system response (micromachined switch example) computed using the TPWL models extracted with the simple and extended reduced order bases. In both cases the models were generated for a 5.5-volt step input voltage.

7.2 Performance of the extended algorithm for generating the reduced order basis

This section presents computational results comparing two algorithms for generating the reduced order basis – a simple one, presented in Section 3.2.2, and the extended one, introduced in Section 3.2.3. Figure 7-1 shows the deflection of the center of the micromachined fixed-fixed beam computed using the two considered methods. In both MOR methods a 5.5-volt step input voltage was used as a 'training' input, and the number of linearization points equaled 6. For the simple algorithm the order of the reduced model equaled $q = 31$. In the extended algorithm, a basis of order 7 ($= 2 \cdot q_i + 1$, $q_i = 3$ for every i – cf. Step 2(a) in the algorithm from Section 3.2.3) was generated at each of the linearization points. Then, the size of the aggregate basis V_{agg} has been reduced from $42 = 6 \cdot 7$ to 28 using the SVD algorithm. One may note that the TPWL model of order $q = 28$, generated with the extended reduced order basis, gives significantly more accurate results than the TPWL model extracted with a simple basis. (On the graph, the dashed line overlaps perfectly with the solid line.) In order to obtain the desired accuracy with the model extracted with the simple basis, the order of the basis needs to be increased to $q = 41$ in the considered case (cf. Figure 7-2).

Yet in a different test (cf. Figure 7-3) we simulated the pull-in effect for the micromachined switch example. Again in this case the MOR method employing the extended reduced basis provides the best accuracy among the considered MOR techniques. It also generates a model with the lowest order.

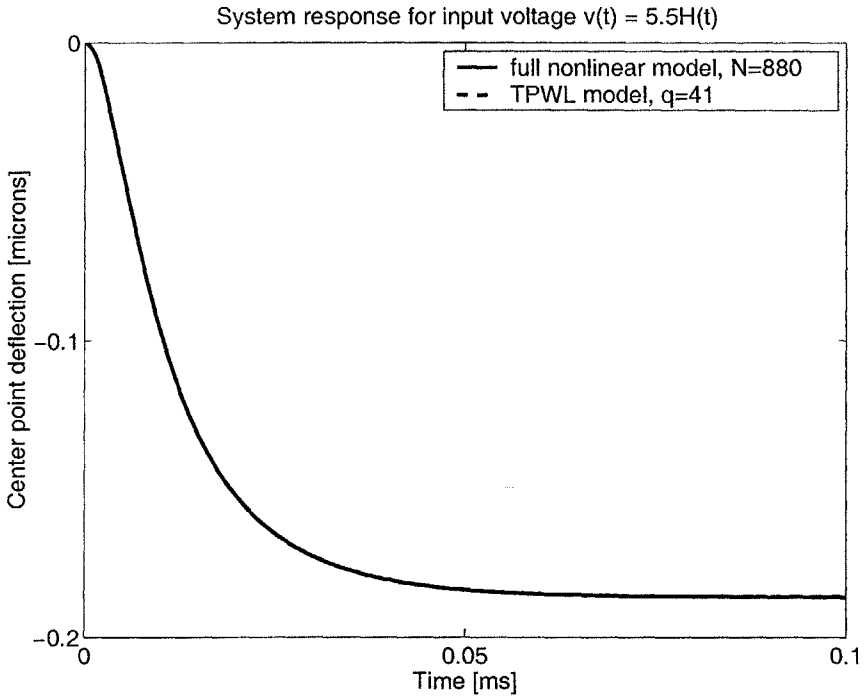

Figure 7-2: Comparison of the system response (micromachined switch example) computed using the full nonlinear simulator, and the TPWL reduced order model extracted with the simple reduced order basis. The model of order q=41 was generated for a 5.5-volt step input voltage.

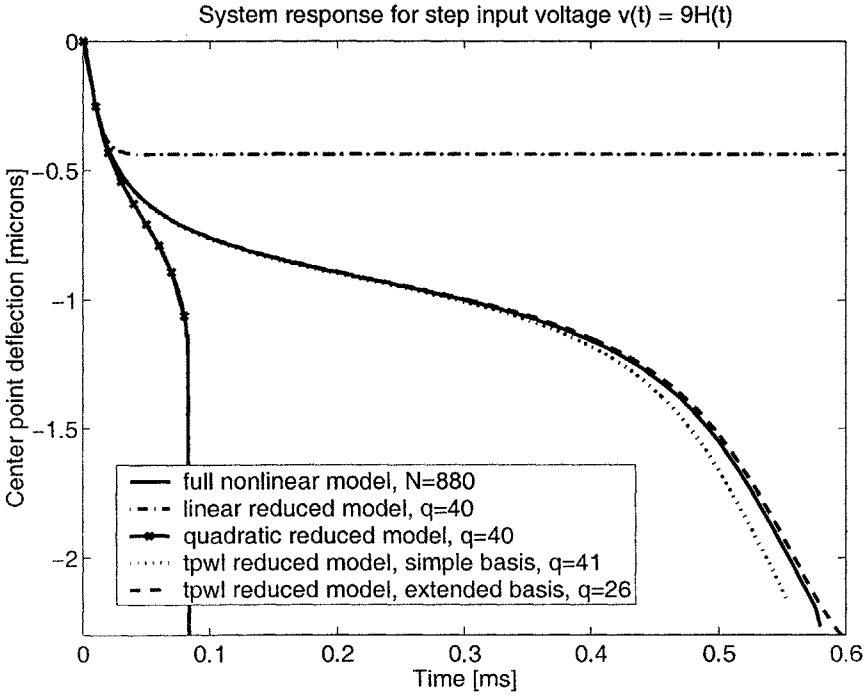

Figure 7-3: Comparison of the system response (micromachined switch example) computed with different MOR algorithms. The TPWL models were generated for a 9-volt step input voltage.

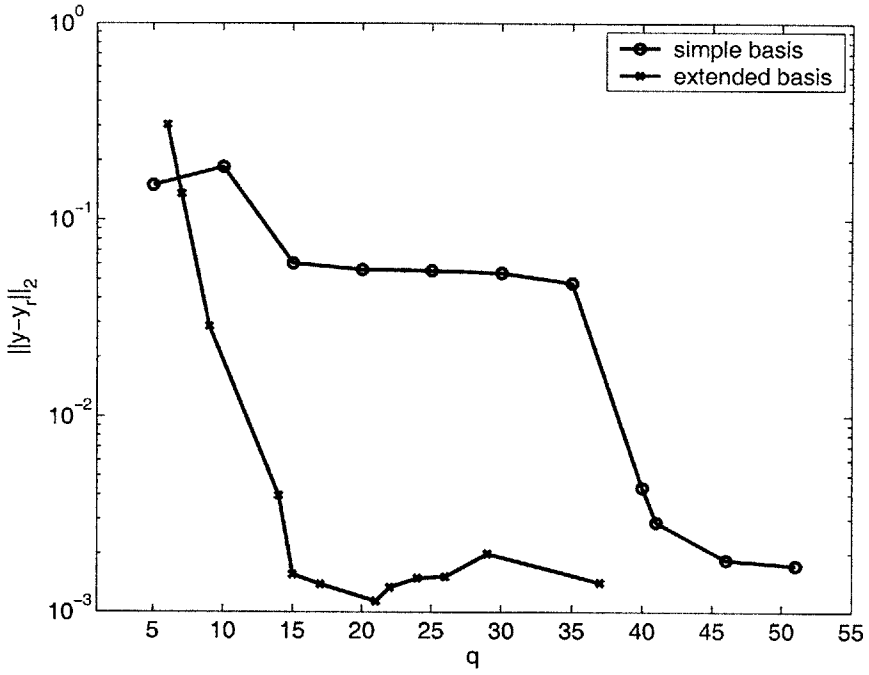

Figure 7-4: Comparison of the error in the output vs. the order q of the reduced order TPWL models, generated with simple and extended projection bases.

Please note that for the extended algorithm the size of the basis can be kept small since we generate very low order bases at each linearization point. On the other hand, in the 'simple' approach we generate the projection basis at a single point, but then it needs to have a much higher order, if desired level of accuracy is to be reached. In order to more precisely compare relative accuracy of TPWL models extracted with simple and extended projection bases we examined the error $e = \|\underline{y} - \underline{y}_r\|_2$, where $\underline{y}_r = [y_r(0), y_r(\Delta T), \ldots, y_r(T)]$ is a vector of outputs at subsequent timesteps obtained with a reduced order TPWL model, and $\underline{y}$ is the corresponding vector obtained with a full order nonlinear model. This error was computed for TPWL reduced models for the micromachined switch, having different orders q. All the compared models have been trained and tested for the same 5.5-volt step input signal, and used the same number of linearization points ($s = 11$), in order to obtain a fair comparison. The results are shown in Figure 7-4, and clearly indicate that the 'extended' algorithm generates projection bases which lead to more accurate TPWL reduced models. It can also be observed from the graph in Figure 7-4 that once a certain order of the model is reached, the accuracy of the model does not improve (and does not tend to 0, while we increase the order of the model). The non-zero 'limit error' is due to the applied piecewise-linear approximation of system's nonlinearity. This error can be lowered only if we improve quality of the TPWL approximation, e.g. by increasing the number of linearization points.

Let us briefly compare the cost of extracting a TPWL model, using the 'simple' and 'extended' algorithms. On one hand, the approximate training simulation (with fast simulator presented in Section 3.3) is typically cheaper if we use the 'extended' algorithm, since the linearized models generated 'on the fly,' during the piecewise-linear simulation have lower orders. On the other hand, if LU factorization of Jacobians is performed to construct Krylov subspace(s), then the 'extended' algorithm requires factorizing a full-order Jacobian at each linearization point, while the 'simple'

Table 7.3: Comparison of the times of model extraction, and reduced order simulation for the TPWL MOR algorithms using two different methods of generating the reduced order basis. The tests were run for the micromachined switch example. The original problem size equaled $N = 880$.

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

algorithm needs to perform such factorization only once – at the initial state, where the projection basis is generated. The cost associated with Jacobian factorization typically dominates over the simulation cost, rendering the 'extended' algorithm computationally more costly. This is illustrated in Table 7.3, which shows model extraction times for the discussed algorithms, for selected values of q. The times were measured on a Linux workstation with Pentium 4, 2-GHz processor.

7.3 Performance of the fast piecewise-linear simulator

The algorithm for approximately generating the training trajectory for TPWL model extraction proposed in Section 3.3, may also be used as a standalone fast simulator for nonlinear dynamical systems. Effectiveness of this simulator has already been indirectly verified by the quality of the extracted TPWL reduced models, as shown in previous sections and chapters. In this Section we present a few selected simulation results which confirm high accuracy of solutions computed with the discussed fast simulator, which performs a series of simulations of linear reduced order models generated 'on the fly'.

First, we used the proposed fast simulator to compute voltages induced in the nonlinear transmission line, in response to a step input current $i(t) \equiv H(t-3)$, where $H(\cdot)$ is the Heaviside (step) function. To this end, we used nonlinear transmission line circuit model with $N = 1500$ nodes, shown in Figure 2-1 (cf. Section 2.2.1). Figure 7-5 presents the output voltage $v_1(t)$ (voltage at node 1) for time from 0 to 10 seconds, computed using full order linear and quadratic models, as well as the proposed fast piecewise-linear simulator. The reference result is computed with a simulator using the full order nonlinear circuit model. In the simulation the number of timesteps was 1000 ($T = 10$, $\Delta T = 0.01$), the fast piecewise-linear simulator used the reduced basis of order $q = 30$ generated at the initial state (the original problem size was $N = 1500$), and we used 21 different linear models during the fast simulation. As Figure 7-5 clearly shows, the output voltage obtained with the fast simulator matches very well the reference result (the curves overlap almost perfectly). Table 7.4 shows the relative error between the vector of voltages at node 1 at subsequent timesteps: $\underline{v}_1 = [v_1(0), v_1(\Delta T), \ldots, v_1(T)]$ computed with linear, quadratic and TPWL simulators and the corresponding reference voltages $\underline{v}_{1,ref}$, obtained with the full nonlinear model of the circuit. It is apparent that the proposed fast piecewise-linear simulator computes a very ac-

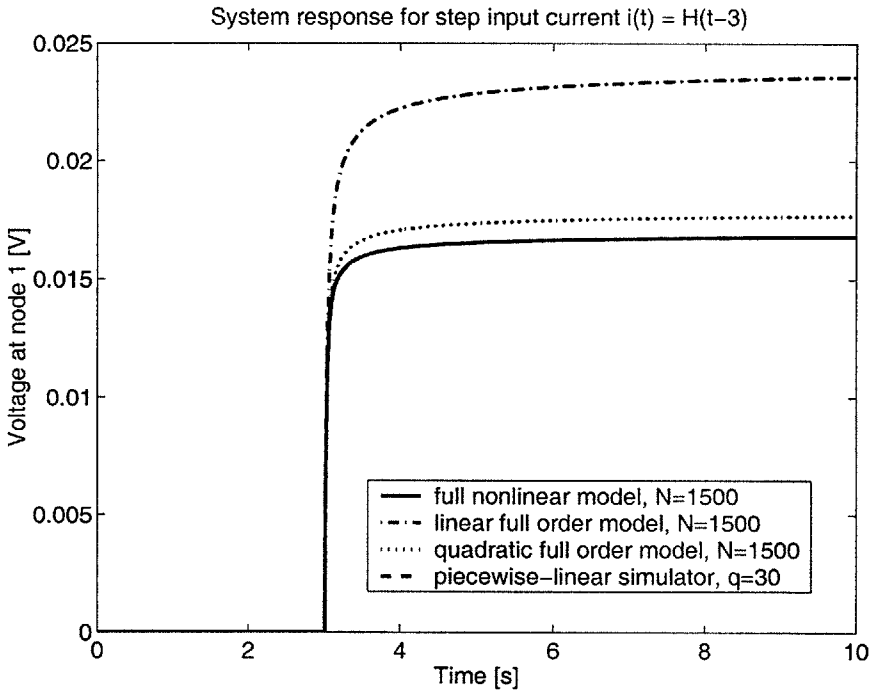

Figure 7-5: Comparison of the system response (nonlinear transmission line circuit model) computed with full-order linear, quadratic, and nonlinear models, and the fast piecewise-linear simulator. The input signal was a step current $i(t) = H(t-3)$ ($H(t) \equiv 1$ for $t > 3$ and $H(t) \equiv 0$ for $t < 3$). $N = 1500$.

Table 7.4: Quality of solutions obtained with linear, quadratic, and fast piecewise-linear simulators.

Model	$\frac{\|\underline{v}_1 - \underline{v}_{1,ref}\|_2}{\|\underline{v}_{1,ref}\|_2}$
linear, $N = 1500$	0.384
quadratic, $N = 1500$	0.049
TPWL, $N = 1500$, $q = 30$	0.003

curate solution (with adequately matched steady state), easily outperforming linear and quadratic simulations. We also compared simulation times for the proposed fast piecewise-linear simulator and the full nonlinear simulator, for the example of nonlinear transmission line circuit model with $N = 1500$ and $N = 100$ nodes, excited with three different input signals. The results are shown in Table 7.5. In order to ensure appropriate accuracy, the order of the reduced basis used by the fast piecewise-linear simulator equaled $q = 30$ (for $N = 1500$), and $q = 10$ (for $N = 100$). Both simulators were implemented in Matlab. Note that high performance implementations will most likely give significantly lower absolute execution times, and may change the relative performance of the two algorithms. The tests were performed on a Linux workstation with Pentium III Xeon processor. It is apparent that for either small or large original problem sizes N the piecewise-linear simulator was significantly faster than the full nonlinear solver. For $N = 1500$ a 100-fold acceleration in computation time was achieved.

Table 7.5: Comparison of simulation times for the full order nonlinear simulator, and the proposed fast piecewise-linear reduced order simulator.

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

7.3.1 Using a posteriori error estimates in fast simulation

In Section 4.1.1 we proposed a procedure for selecting subsequent linearization points (located on the training trajectory) based on *a posteriori* error bounds. This procedure may again be used as a separate simulator, alternative to the one considered above (and presented in Section 3.3). The difference between these two simulators is that the first one (called from now on 'simple' fast simulator) uses only the information on distances between the states in order to decide whether the next linearized model should be generated, while the other one (called from now on 'extended' fast simulator) makes this decision based on the estimated error in the solution computed at a given timestep. In the 'simple' approach, an arbitrary parameter controls how frequently linearized models are changed, while in the 'extended' one, linearized models change so frequently, to ensure that the relative error of the solution does not exceed a prescribed value. This means that in the 'extended' approach one may guarantee quality of the computed solutions. Unfortunately the 'extended' fast simulator, described in Section 4.1.1, may only be used for systems with negative monotone nonlinearities, and if estimates of Hessian norms are available.

The transmission line circuit model with quadratic resistors (cf. Figure 2-2) is an example of a system, for which those conditions are satisfied. Consequently, we used it to compare the two fast simulators discussed above. Also note that while in the previous section the fast simulator used projection basis generated at the initial state, both 'simple' and 'extended' fast simulators examined in this section generate a new basis (in a suitable Krylov subspace) for every new linearization point (cf. Section 3.2.3).

In the first test we computed values of the output of our system $y_{ri} = y_r(i\Delta T)$, for $\Delta T = 0.01$ and $i = 1, \ldots, 10000$, in response to a step input current $i(t) \equiv 1$. Both 'simple' and 'extended' fast simulators used 16 reduced linearized models of order 7 to perform this simulation. For the 'extended' simulator, the number of linearizations used corresponded to the maximum estimated relative error $\varepsilon = 0.1$ (cf. Section 4.1.1). The relative errors (with respect to the full nonlinear model) in the solutions computed with both fast simulators are shown in Figure 7-6. One may note from the graph that the 'extended' simulator distributes linearization points (identified by sharp notches) much more evenly during the simulation, as compared to the 'simple' simulator. This leads to overall lower levels of the errors. Also note that this error does not exceed the

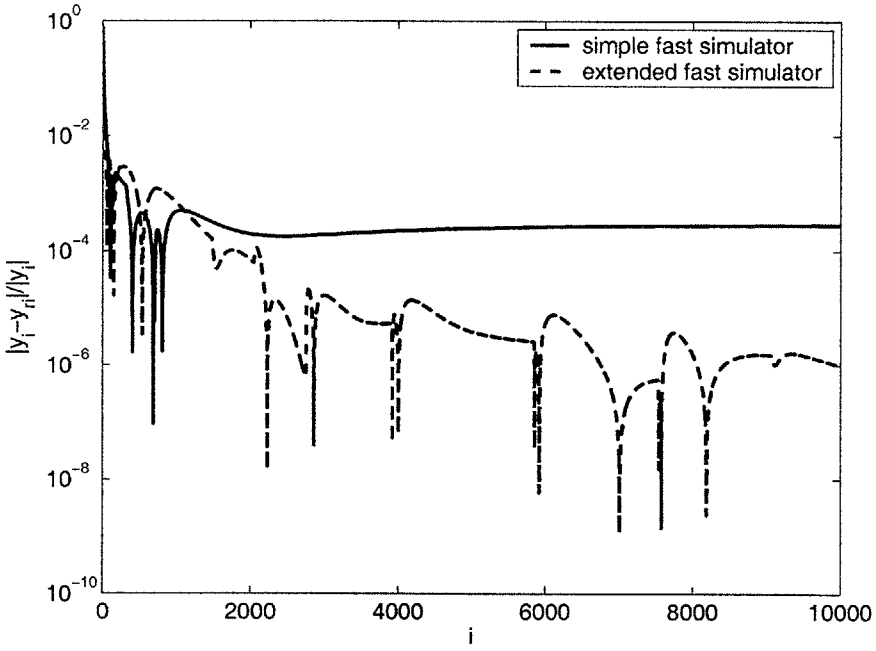

Figure 7-6: Comparison of simulation errors for the 'simple' and 'extended' fast simulators, for $u(t) \equiv 1$.

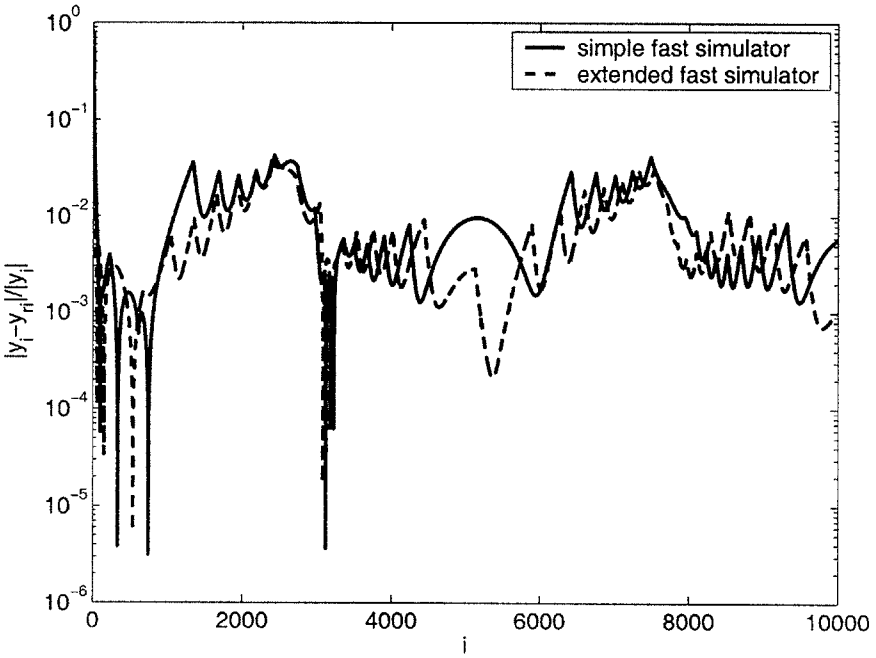

Figure 7-7: Comparison of simulation errors for the 'simple' and 'extended' fast simulators, for $u(t) = (\cos(\pi(i-1)/5000))^2$.

prescribed value of $\varepsilon = 0.1$. Nevertheless, since the number of linearization points is the same in both simulators, the 'simple' simulator outperforms the 'extended' one in the beginning of the simulation. Consequently, if we compute

$$e = \|\underline{y} - \underline{y}_r\|_2 / \|\underline{y}\|_2, \tag{7.1}$$

where $\underline{y} = [y_1, \ldots, y_{10000}]$ contains values of output computed with fast simulators, and $\underline{y}_r$ – the corresponding values computed with the full nonlinear model, we will obtain that $e = 0.53$ for the 'simple' simulator, and $e = 0.52$ for the 'extended' simulator.

In the second test we used input current $i(t) = (\cos(\pi t/50))^2$ ($t \in [0, 100]$). Both fast simulators used 45 reduced linearized models of order 7 to perform this simulation. For the 'extended' simulator, the number of linearizarions used corresponded to the maximum estimated relative error $\varepsilon = 0.05$. Figure 7-7 shows relative errors of the solutions computed with the 'simple' and 'extended' simulators. One may note that in this case both simulators distribute linearization points rather evenly. However, if one computes e, given by (7.1), then for the 'simple' simulator equals $e = 0.26$, and for the 'extended' one – $e = 0.15$. This means that the 'extended' model is able to select the linearization points in a more optimal manner.

We also compared simulation times for both 'simple' and 'extended' algorithms. The tests were performed on a Linux workstation with Pentium 4, 2-GHz processor. For the first input signal, simulation with the 'simple' algorithm took 3.5 seconds, and with the 'extended' algorithm – 6.1 seconds. For the second input signal, the simulation times equaled 3.9 and 6.7 seconds, respectively. The full nonlinear simulation took 57.2 seconds, for both input signals.

Chapter 8

Conclusions

In this dissertation we studied the problem of constructing reduced order models for nonlinear dynamical systems, which mathematically describe a broad and important class of devices and phenomena in all branches of engineering. We focused on addressing the following aspects of nonlinear model order reduction:

- modeling strongly nonlinear systems in which system's operating point varies significantly with time,
- cost-efficient representation of nonlinearity in a reduced order system.

The main motivation behind this choice of topics is lack of MOR methods which would adequately approach the important problems mentioned above. Firstly, the majority of existing nonlinear MOR algorithms are based on polynomial or functional expansions about a single state of a given system, and consequently cannot be applied to modeling systems in which a strong nonlinearity drives the system through a series of different operating points. Unfortunately, single-state expansion limits very considerably the scope of application of those methods. Secondly, very often existing MOR strategies overlook the problem of computational complexity associated with using the extracted reduced order model. Unlike in the linear case, order reduction (e.g. corresponding to a certain state-space projection) of a nonlinear system does *not* usually imply a commensurate reduction of numerical cost associated with solving/evaluating the reduced order system. Significant simulation speedups may be achieved only if order reduction is accompanied by a cost-efficient representation of system's nonlinearity. Finally, some of the MOR methods are characterized by high cost of extracting the reduced order model, which effectively eliminates them from being used for larger dynamical systems.

Consequently, during the entire process of constructing a new MOR strategy for nonlinear dynamical systems we took a cost-driven approach, which manifested itself through:

- proposing a simple, easy to handle approximation of system's nonlinearity, based on a weighted combination of a relatively small collection of linearized models,
- developing procedures for a generating reduced order projection basis, exploiting inexpensive Krylov subspace methods,

- constructing an algorithm for approximate simulation of the original system (based on reduced linear models generated 'on the fly'), to reduce the costs of model extraction.

In order to be able to account for strongly nonlinear effects in the reduced order model, the idea of selecting linearization points from a state-space trajectory of the original nonlinear system has been proposed, leading to a Trajectory Piecewise-Linear (TPWL) model order reduction strategy.

Properties of the proposed TPWL MOR approach, as well as its applicability, have been investigated primarily by performing suitable numerical experiments. The TPWL MOR algorithm was applied to extracting models for a few selected examples of devices which are composed of strongly nonlinear elements and/or demonstrate strongly nonlinear input-output behavior, and proved challenging for the existing MOR approaches. The examples included a micromachined switch, nonlinear transmission line circuit models, an operational amplifier, and a jet engine inlet operating at supersonic speeds. In particular, the micromachined switch example and the nonlinear transmission line circuit model were used by a number of researchers to test different MOR approaches, based on polynomial expansions of the nonlinearity, bilinearization, and Karhunen-Loève expansions. As reported in literature, reduced order models generated by existing MOR techniques were unable to capture strongly nonlinear effects in the behavior of the discussed systems, were lacking accuracy (in modeling either transient or steady-state behavior), or were offering insufficient computational speedup.

Results of numerical experiments, presented mostly in Chapter 6, indicate that the TPWL MOR method outperforms previously used MOR algorithms based on polynomial expansions about a single state of a system (e.g. linear, quadratic reduced models), often in a number of aspects. Models extracted with the TPWL method were found to be able to capture strongly nonlinear effects (such as pull-in effect for the micromachined switch example, or shock movement for the supersonic fluid flow problems), and adequately simulate both transient and steady-state responses for systems with a large number of strongly nonlinear elements (e.g. for the nonlinear transmission line models, or the op-amp example). The discussed models were able to compute adequately the first few harmonics, and the main intermodulation products of a periodic steady state in a nonlinear system, indicating that they may be effectively used to find distortion in nonlinear circuits, such as op-amps. However, it has also been found that the applied reduced order projection bases have difficulties with adequately representing some of the shocks.

As shown in Chapter 7, TPWL MOR algorithms and the corresponding reduced order models are characterized by high performance, and low computational complexity. Thanks to the proposed fast approximate simulation algorithm, and the applied reduced order projection techniques, extraction of TPWL models can be performed very efficiently, often at a significantly lower cost than extraction of e.g. quadratic macromodels. Memory resources required to store a TPWL reduced order model are also modest, and comparable to those necessary to store a quadratic reduced order model. Finally, as shown through performance tests, simulation of a reduced order TPWL system is very fast, due to efficient representation of the nonlinearity, and the applied time-stepping scheme which uses estimates of the next state to compute the weights at a given timestep. Summing up, the results of computational experiments clearly indicate that using TPWL reduced order

models results in very large speedups in computation time, and requires very modest computer resources. They confirm that the developed TPWL MOR strategy adequately addresses important problems of nonlinear model order reduction, including macromodeling of strongly nonlinear systems, and providing an efficient representation of system's nonlinearity. Consequently, the main goal of this dissertation has been achieved.

Computational tests performed within this study also aimed at investigating limitations of TPWL model order reduction approach. Recalling that a given TPWL model is associated with a training trajectory corresponding to a certain training input, a question which arises is establishing a class or range of inputs for which a given TPWL model will be sufficiently accurate. Giving a general, yet practical answer to this question is difficult, since e.g. *a priori* error analysis (which ultimately gives a bound for the error between solutions computed with the TPWL model and the original nonlinear system, expressed in terms of e.g. the distance between the testing and training input signals) typically yields very conservative bounds (cf. Chapter 4). In order to practically assess the range of applicability of a given TPWL system, numerical tests were performed. Results of those tests, presented in Chapters 3 and 6, suggest that often a TPWL model can provide an adequate model of the original nonlinear system even for a testing input signal which is dynamically very different from the training input signal (e.g. a sinusoidal signal vs. a step signal), for which this model has been generated. The mentioned results provide therefore an important argument in favor of TPWL MOR approach.

Although practical importance of *a priori* error estimates is limited, we have found that *a posteriori* error bounds may be effectively used to assess quality of the computed solutions during the actual simulation with a reduced order TPWL model. In Chapter 4, a procedure for computing such error bounds 'on the fly' has been derived for the case of systems with negative monotone nonlinearity. It has also been shown that this procedure may be effectively applied to compute realistic error bounds for an example of a nonlinear analog circuit.

Another aspect of MOR, which may become particularly important if using reduced order models to replace subsystems in larger, interconnected systems refers to the property of preserving stability and passivity of the original nonlinear system. Chapter 5 presents discussion of those problems, focusing on negative definiteness as the simplest property which may help one to ensure stability of TPWL reduced order systems. In that chapter some basic results, stating under which condition a TPWL reduced order model is guaranteed to be stable, are presented. Those results provide a starting point for developing strategies for constructing guaranteed stable and passive reduced models.

Yet another issue studied in this dissertation refers to making the model extraction process as automatic as possible. In the initial version of the proposed TPWL MOR algorithm a few arbitrarily set parameters are used. The most important one controls 'how often' the subsequent linearization points will be picked, which determines the final number of the linearized models in a TPWL reduced order system. In an effort to eliminate this arbitrary parameter, in Chapter 4 we proposed a procedure for selecting linearization points based on *a posteriori* error analysis, valid for negative monotone nonlinearities. As shown using a computational example, the proposed procedure,

apart from making the model extraction more automatic, leads to an improved distribution of the linearization points. In the thesis we have also suggested possible enhancements of accuracy of TPWL models, e.g. we presented a weighting procedure which exploited information on Hessians of the nonlinear function.

Previous chapters also present problems which were observed while performing simulations with TPWL reduced order models for given application examples, and could not be resolved so far. They include problems with stability of TPWL models for the case of nonsymmetric, indefinite nonlinear operators (e.g. for the micromachined switch example), as well as difficulties with reduced order representation of shock shapes, and inadequate projection basis generated by Krylov subspace methods for the problem of the jet engine inlet. The above problems suggest that further investigation is needed in the following topics:

- stability and passivity preservation, while reducing stable systems with nonlinear operators which do *not* satisfy the negative definiteness condition,
- error analysis for the case of non-negative monotone nonlinearities,
- reduced order projection strategies for TPWL models, based on methods other than Krylov subspace algorithms, e.g. TBR or Fourier Model Reduction (FMR).

Possible extensions of the TPWL MOR approach proposed in this study also include:

- applying a different projection basis for each of the linearized models (and performing necessary translations of states between different representations in reduced order subspaces),
- generating TPWL models using multiple training trajectories,
- optimizing the choice of linearization points,
- automatic order selection based on error analysis,
- considering weighted combinations of higher order models, obtained with polynomial expansions at different states of the system.

Some of the topics mentioned above have not yet been considered, while others are currently investigated by researchers following on this initial study, which gives promise for further development of reduction algorithms based on the concept of trajectory quasi-piecewise-linear approximations for nonlinear dynamical systems.

Appendix A

Netlist for the op-amp example

Following is the input file for the NITSWIT circuit simulator, which contains the netlist for the op-amp example, along with definitions of the input signals. For a description of the file format please refer to [68].

```
global 0
cdif 14 15 c c = 4pf
cccmm1 14 702 c c = 1pf
cccmm2 15 702 c c = 1pf
;***********************************************
vcmmdc 902 0 dc v = 10.2mv
mrr2 902 700 r r = 1
vcmmrst 903 0 pwl t0=0 v0=-2.5 t1=150ns v1=-2.5 t2=155ns v2=2.5
mrr3 903 610 r r = 1
mpcmm1 14 610 702 100 pch w = 14u l = 3u
mpcmm2 15 610 702 100 pch w = 14u l = 3u
mrra 911 601 r r = 1
vin1 911 0 pwl t0=0 v0=0 t1=290ns v1=0 t2=400ns v2=11.5mv t3=10000ns v3=11.5mv
vin2 901 0 pwl t0=0 v0=0 t1=290ns v1=0 t2=400ns v2=-11.5mv t3=10000ns v3=-11.5mv
mrr1 901 602 r r = 1
vgnd 904 0 pwl t0=0 v0=2.5 t1=220ns v1=2.5 t2=230ns v2=-2.5 t3=10000ns v3=-2.5
mrr4 904 605 r r = 1
vintn 905 0 pwl t0=0 v0=-2.5 t1=250ns v1=-2.5 t2=260ns v2=2.5 t3=10000ns v3=2.5
mrr5 905 606 r r = 1
vintp 906 0 pwl t0=0 v0=2.5 t1=250ns v1=2.5 t2=260ns v2=-2.5 t3=10000ns v3=-2.5
mrr6 906 607 r r = 1
vrst 907 0 pwl t0=0 v0=2.5 t1=230ns v1=2.5 t2=240ns v2=-2.5 t3=10000ns v3=-2.5
mrr7 907 608 r r = 1
vcmmin 908 0 pwl t0=0 v0=2.5 t1=200ns v1=2.5 t2=210ns v2=-2.5 t3=10000ns v3=-2.
mrr8 908 609 r r = 1
cin1 601 1 c c = 8.84pf
cin2 602 2 c c = 8.84pf
cint1 1 603 c c = 1.105pf
cint2 2 604 c c = 1.105pf
mngm1 1 609 0 200 nch w = 20u l = 3u
mngm2 2 609 0 200 nch w = 20u l = 3u
mngnd1 603 605 0 200 nch w = 20u l = 3u
mngnd2 604 605 0 200 nch w = 20u l = 3u
mnint1 603 606 15 200 nch w = 20u l = 3u
mnint2 604 606 14 200 nch w = 20u l = 3u
mpint1 603 607 15 100 pch w = 20u l = 3u
mpint2 604 607 14 100 pch w = 20u l = 3u
mnrst1 10 608 14 200 nch w = 20u l = 3u
mnrst2 11 608 15 200 nch w = 20u l = 3u
;***********************************************
;* first stage **** low gain **** hi bw *******
;***********************************************
mn10in1 3 1 7 200 ncc w = 200u l = 3u ad = 900p as = 900p
mn10in2 4 2 7 200 ncc w = 200u l = 3u ad = 900p as = 900p
mp10ld1 200 8 3 100 pch w = 22u l = 5u ad = 81p as = 81p
mp10ld2 200 8 4 100 pch w = 22u l = 5u ad = 81p as = 81p
mp10pu1 3 9 100 100 pch w = 72u l = 5u ad = 162p as = 162p
mp10pu2 4 9 100 100 pch w = 72u l = 5u ad = 162p as = 162p
mn10fol1 100 3 5 200 nch w = 132u l = 3u ad = 300p as = 300p
mn10fol2 100 4 6 200 nch w = 132u l = 3u ad = 300p as = 300p
;* current sources ***************************
mn10dpbs 7 201 200 200 nch w = 219u l = 5u ad = 1296p as = 1296p
mn10fbs1 5 201 200 200 nch w = 72u l = 5u ad = 324p as = 324p
mn10fbs2 6 201 200 200 nch w = 72u l = 5u ad = 324p as = 324p
mn10pubs 8 201 200 200 nch w = 72u l = 5u ad = 324p as = 324p
mn10csbs 9 0 8 200 nch w = 132u l = 3u ad = 300p as = 300p
mp10pubs 9 9 100 100 pch w = 36u l = 5u ad = 162p as = 162p
;***********************************************
coffset1 5 10 c c = 4pf
coffset2 6 11 c c = 4pf
```

```
roffset1 10 0 r r = 1e13
roffset2 11 0 r r = 1e13
;*********************************************
;** 2nd stage ** medium gain ******************
;*********************************************
mndifpr1 12 10 19 200 nch w = 100u l = 5u ad = 450p as = 450p
mndifpr2 13 11 19 200 nch w = 100u l = 5u ad = 450p as = 450p
mpdifpr1 16 10 21 100 pch w = 100u l = 5u ad = 450p as = 450p
mpdifpr2 17 11 21 100 pch w = 100u l = 5u ad = 450p as = 450p
mpcurbs1 12 101 100 100 pch w = 192u l = 5u ad = 900p as = 900p
mpcurbs2 13 101 100 100 pch w = 192u l = 5u ad = 900p as = 900p
mpcurcs1 14 102 12 100 pch w = 96u l = 5u ad = 450p as = 450p
mpcurcs2 15 102 13 100 pch w = 96u l = 5u ad = 450p as = 450p
mncurbs1 16 201 200 200 nch w = 96u l = 5u ad = 450p as = 450p
mncurbs2 17 201 200 200 nch w = 96u l = 5u ad = 450p as = 450p
mncurcs1 14 202 16 200 nch w = 48u l = 5u ad = 225p as = 225p
mncurcs2 15 202 17 200 nch w = 48u l = 5u ad = 225p as = 225p
mndifbs 19 209 200 200 nch w = 210u l = 10u ad = 900p as = 900p
mpdifbs 21 109 100 100 pch w = 420u l = 10u ad = 1800p as = 1800p
;*********************************************
;** cmm feedback ******************************
;*********************************************
mpcmmf1a 16 702 20 100 pch w = 30u l = 5u ad = 270p as = 270p
mpcmmf1b 17 702 20 100 pch w = 30u l = 5u ad = 270p as = 270p
mpcmmf2a 30 700 20 100 pch w = 30u l = 5u ad = 270p as = 270p
mpcmmf2b 30 700 20 100 pch w = 30u l = 5u ad = 270p as = 270p
mncmmm2 30 30 200 200 nch w = 28u l = 5u ad = 112p as = 112p
mncmmm1a 16 30 200 200 nch w = 14u l = 5u ad = 70p as = 70p
mncmmm1b 17 30 200 200 nch w = 14u l = 5u ad = 70p as = 70p
mpcmmbs 20 101 100 100 pch w = 96u l = 5u ad = 450p as = 450p
;*********************************************
;** hi swing biasing **************************
;*********************************************
ibias1 100 202 i i = 40ua
ibias2 100 203 i i = 40ua
mnxbs1 202 202 204 200 nch w = 48u l = 5u ad = 225p as = 225p
mnxbs2 204 202 200 200 nch w = 10u l = 6u ad = 90p as = 90p
mnxbs3 201 201 200 200 nch w = 48u l = 5u ad = 225p as = 225p
mnxbs4 100 203 201 200 nch w = 48u l = 5u ad = 225p as = 225p
mnxbs5 205 201 200 200 nch w = 48u l = 5u ad = 225p as = 225p
mnxbs6 203 202 205 200 nch w = 48u l = 5u ad = 225p as = 225p
mnxbs7 206 201 200 200 nch w = 48u l = 5u ad = 225p as = 225p
mnxbs8 102 202 206 200 nch w = 48u l = 5u ad = 225p as = 225p
mnxbs9 207 201 200 200 nch w = 48u l = 5u ad = 225p as = 225p
mnxbs10 103 202 207 200 nch w = 48u l = 5u ad = 225p as = 225p
mnxbs11 208 201 200 200 nch w = 48u l = 5u ad = 225p as = 225p
mnxbs12 109 202 208 200 nch w = 48u l = 5u ad = 225p as = 225p
mnxbs13 209 209 200 200 nch w = 105u l = 10u ad = 900p as = 900p
;*********************************************
mpxbs1 102 102 104 100 pch w = 96u l = 5u ad = 450p as = 450p
mpxbs2 104 102 100 100 pch w = 20u l = 6u as = 180p as = 180p
mpxbs3 101 101 100 100 pch w = 96u l = 5u ad = 450p as = 450p
mpxbs4 200 103 101 100 pch w = 96u l = 5u ad = 450p as = 450p
mpxbs5 105 101 100 100 pch w = 96u l = 5u ad = 450p as = 450p
mpxbs6 103 102 105 100 pch w = 96u l = 5u ad = 450p as = 450p
mpxbs7 106 101 100 100 pch w = 96u l = 5u ad = 450p as = 450p
mpxbs8 209 102 106 100 pch w = 96u l = 5u ad = 450p as = 450p
mpxbs9 109 109 100 100 pch w = 210u l = 10u ad = 1800p as = 1800p
;*********************************************
;** supply ************************************
;*********************************************
vdd 909 0 dc v = 2.5v
 mrr9 909 100 r r = 1
vss 910 0 dc v = -2.5v
 mrrb 910 200 r r = 1

;*********************************************
;** n-channel mosfet: *************************
;*********************************************
model nch nmos vto = 0.9 kp = 30.0u gamma = 1.36 phi = 0.747 cgso = 520p cgdo =
520p tox = 50n ld = 400n cj = 320u mj = 0.5 cjsw = 900p mjsw = 0.33 js = 100u

model ncc nmos vto = 0.9 kp = 40.0u gamma = 1.36 phi = 0.747 cgso = 520p cgdo =
520p tox = 50n ld = 400n cj = 320u mj = 0.5 cjsw = 900p mjsw = 0.33 js = 100u

;*********************************************
;** p-channel mosfet: *************************
;*********************************************

model pch pmos vto = -0.9 kp = 15.0u gamma = 0.59 phi = 0.66 cgso = 400p cgdo =
400p tox = 50n ld = 500n cj = 200u mj = 0.5 cjsw = 450p mjsw = 0.33 js = 100u

relax2 options stop = 1000ns  askparam=1 dosteady=0 nralpha=0.3 dodc=1 gmin=1e-9 newjacob=1
```

Appendix B

Derivation of the discretized system for the micromachined switch example

In order to discretize the problem described by equations (2.9)-(2.11) and boundary conditions (2.7) we introduce the following regular grid over the problem domain:

$$\begin{array}{l} x^i = i\Delta x, \quad i = 0,\ldots,(n+1) \\ y^j = j\Delta y, \quad j = 0,\ldots,(m+1) \end{array}, \tag{B.1}$$

where $\Delta x = l/(n+1)$, $\Delta y = w/(m+1)$, l is the length of the beam, w is the width of the beam, and n and m are the number of grid points inside the computational domain, in the x and y directions, respectively. Then, the state variables (2.8) are approximated by the vectors:

$$\begin{array}{rcl} x_1(t) & \sim & \underline{x}_1(t) = [x_1^1(t),\ldots,x_1^n(t)]^T \\ x_2(t) & \sim & \underline{x}_2(t) = [x_2^1(t),\ldots,x_2^n(t)]^T \\ x_3(t) & \sim & \underline{x}_3(t) = [x_3^{11}(t),\ldots,x_3^{1m}(t),\ldots,x_3^{nm}]^T \end{array}, \tag{B.2}$$

where $x_1^i(t) = x_1(x^i,t)$, $x_2^i(t) = x_2(x^i,t)$, and $x_3^{ij}(t) = x_3(x^i,y^j,t)$. Note that the values of the state variables at the boundaries have not been included in the above discrete representation. Therefore, the dimension of the obtained state space $\underline{x} = [\underline{x}_1,\underline{x}_2,\underline{x}_3]^T$ equals $2n+mn$. Based on the above finite-dimensional representation of the state space, the following discrete analogues of equations (2.9)-(2.11) may be written:

$$\frac{d\underline{x}_1}{dt} = \frac{1}{3}\frac{\underline{x}_2}{\underline{x}_1^2}, \tag{B.3}$$

$$\frac{d\underline{x}_2}{dt} = \frac{2}{3}\frac{\underline{x}_2^2}{\underline{x}_1^3} + \frac{3}{\rho}\underline{x}_1^2 \circ \left[\underline{\underline{INT}}(\underline{x}_3 - p_a) + S(\underline{\underline{D2}}(\underline{x}_1 - u_0) - EI(\underline{\underline{D4}}(\underline{x}_1 - u_0))\right] - \frac{3\varepsilon_0 w}{2\rho}v^2, \tag{B.4}$$

$$\begin{aligned} \frac{d\underline{x}_3}{dt} &= -\widehat{diag}\left\{\frac{\underline{x}_2}{3\underline{x}_1^3}\right\}\underline{x}_3 + \widehat{diag}\left\{\frac{\underline{x}_1 + 4\lambda}{4\mu} \circ \underline{\underline{\tilde{D}_x}}(\underline{x}_1 - u_0)\right\}\left(\underline{x}_3 \circ (\underline{\underline{D_x}}(\underline{x}_3 - p_a))\right) \\ &+ \widehat{diag}\left\{\frac{\underline{x}_1^2 + 6\lambda\underline{x}_1}{12\mu}\right\}\left[\underline{x}_3 \circ (\underline{\underline{LAP}}(\underline{x}_3 - p_a)) + (\underline{\underline{D_x}}(\underline{x}_3 - p_a))^2 + (\underline{\underline{D_y}}(\underline{x}_3 - p_a))^2\right]. \end{aligned} \tag{B.5}$$

Note that the following shortcut notation has been applied in the above formulae:

$$\begin{aligned}
\underline{a} \circ \underline{b} &\equiv \left[a^1 b^1, \ldots, a^k b^k\right]^T, \\
\frac{\underline{a}}{\underline{b}} &\equiv \left[a^1/b^1, \ldots, a^k/b^k\right]^T, \\
\frac{c}{\underline{b}} &\equiv \left[c/b^1, \ldots, c/b^k\right]^T, \\
\underline{a}^n &\equiv \left[(a^1)^n, \ldots, (a^k)^n\right]^T, \\
\underline{a} + c &\equiv \left[a^1 + c, \ldots, a^k + c\right]^T,
\end{aligned}$$

$$\widehat{diag}\left(\begin{bmatrix} a^1 \\ \vdots \\ a^n \end{bmatrix}\right) = \begin{bmatrix} \overbrace{\begin{matrix} a^1 & & \\ & \ddots & \\ & & a^1 \end{matrix}}^{m} & & \\ & \ddots & \\ & & \underbrace{\begin{matrix} a^n & & \\ & \ddots & \\ & & a^n \end{matrix}}_{m} \end{bmatrix}_{mn \times mn},$$

where k is the order of vectors $\underline{a} = [a^1, \ldots, a^k]^T$ and $\underline{b} = [b^1, \ldots, b^k]^T$, and c is a scalar. The discrete matrix operators in formulae (B.3)-(B.6) are given by:

$$\underline{\underline{D_y}} = \frac{1}{2\Delta y} \begin{bmatrix} \overbrace{\begin{matrix} 1 & & \\ -1 & \ddots & \\ & \ddots & 1 \\ & & -1 \end{matrix}}^{m} & & \\ & \ddots & \\ & & \underbrace{\begin{matrix} 1 & & \\ -1 & \ddots & \\ & \ddots & 1 \\ & & -1 \end{matrix}}_{m} \end{bmatrix}_{mn \times mn},$$

$$\underline{\underline{INT}} = \Delta y \begin{bmatrix} \overbrace{1 \ldots 1}^{m} & & \\ & \ddots & \\ & & \underbrace{1 \ldots 1}_{m} \end{bmatrix} \Bigg\} n, \quad \underline{\underline{\tilde{D}_x}} = \frac{1}{2\Delta x} \begin{bmatrix} 1 & & \\ -1 & \ddots & \\ & \ddots & 1 \\ & & -1 \end{bmatrix}_{n \times n},$$

$$\underline{\underline{D2}} = \frac{1}{\Delta x^2}\begin{bmatrix} -1 & 1 & & & \\ 1 & -2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & -2 & 1 \\ & & & 1 & -1 \end{bmatrix}_{n\times n}, \quad \underline{\underline{D4}} = \frac{1}{\Delta x^4}\begin{bmatrix} 7 & -4 & 1 & & \\ -4 & 6 & \ddots & \ddots & \\ 1 & \ddots & \ddots & \ddots & 1 \\ & \ddots & \ddots & 6 & -4 \\ & & 1 & -4 & 7 \end{bmatrix}_{n\times n},$$

$$\underline{\underline{D_x}} = \frac{1}{2\Delta x}\begin{bmatrix} \overbrace{\begin{matrix} -1 & & \\ & \ddots & \\ & & -1 \end{matrix}}^{m} & & 1 & & \\ & & & \ddots & \\ -1 & & & & 1 \\ & \ddots & & & \\ & & \ddots & & \\ & & & -1 & \underbrace{1}_{m} \end{bmatrix}_{mn\times mn},$$

$$\underline{\underline{LAP}} = \frac{1}{\Delta y^2}\begin{bmatrix} \overbrace{\begin{matrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{matrix}}^{m} & & & \\ & \ddots & & \\ & & \underbrace{\begin{matrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{matrix}}_{m} & \end{bmatrix}_{mn\times mn}$$

$$+ \frac{1}{\Delta x^2}\begin{bmatrix} \overbrace{\begin{matrix} -1 & & \\ & \ddots & \\ & & -1 \end{matrix}}^{m} & & 1 & & \\ & & & \ddots & \\ 1 & & -2 & & \ddots \\ & \ddots & & \ddots & & \ddots \\ & & \ddots & & -2 & & 1 \\ & & & \ddots & & 1 & \\ & & & & \ddots & & \ddots \\ & & & & 1 & \underbrace{\quad}_{m} & 1 \end{bmatrix}_{mn\times mn},$$

Note that the above discrete operators implicitly include appropriate boundary conditions, given by (2.7). The discrete output equation may be written in the following form:

$$y = \underline{\underline{C}}^T \underline{x}, \qquad \underline{\underline{C}} = [\overbrace{0,\ldots,0}^{[n/2]-1},1,0,\ldots,0]^T_{1\times(m+2)n}. \tag{B.6}$$

Bibliography

[1] E. Acar, L. T. Pileggi, S. R. Nassif, "A linear-centric simulation framework for parametric fluctuations," in *Proceedings of the 2002 Design, Automation and Test in Europe Conference and Exhibition*, 2002, pp. 568-75.

[2] H. Aling, R. L. Kosut, A. Emami-Naeini, J. L. Ebert, "Nonlinear Model Reduction with Application to Rapid Thermal Processing," in *Proceedings of the 35th IEEE Conference on Decision and Control*, vol. 4, 1996, pp. 4305-10.

[3] A. Ariffin, N. Munro, "Representation of a class of nonlinear systems with parameter-dependent linear approximations," in *Proceedings of the UKACC International Conference on Control '98*, vol. 1, 1998, pp. 537-42.

[4] S. Azou, P. Brehonnet, P. Vilbe, L. C. Calvez, "A New Discrete Impulse Response Gramian and its Application to Model Reduction," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 533-7, 2000.

[5] Z. Bai, R. D. Slone, W. T. Smith, Q. Ye, "Error bound for reduced system model by Padé approximation via the Lanczos process," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 2, pp. 133-41, 1999.

[6] Z. Bai, "Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems," *Applied Numerical Mathematics*, vol. 43, no. 1-2, pp. 9-44, 2002.

[7] I. Balk, "Arnoldy based passive model order reduction algorithm," in *Proceedings of the IEEE Conference on Electrical Performance of Electronic Packaging*, 2000, pp. 251-4.

[8] I. Balk, "On a Passivity of the Arnoldi Based Model Order Reduction for Full-Wave Electromagnetic Modeling," *IEEE Transactions on Advanced Packaging*, vol. 24, no. 3, pp. 304-8, 2001.

[9] G. Berkooz, P. Holmes, J. Lumley, "The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows," *Annual Review of Fluid Mechanics*, vol. 25, pp. 539-75, 1993.

[10] W. T. Beyene, J. E. Schutt-Aine, "Krylov subspace-based model-order reduction techniques for circuit simulations," in *Proceedings of the 39th Midwest Symposium on Circuits and Systems*, vol. 1, 1996, pp. 331-4.

[11] J. E. Bracken, D.-K. Sun, Z. J. Cendes, "S-Domain Methods for Simultaneous Time and Frequency Characterization of Electromagnetic Devices," *IEEE Transactions on Microwave Theory and Techniques*, vol. 46, no. 9, pp. 1277-90, 1998.

[12] A. C. Cangellaris, L. Zhao, "Rapid FDTD Simulation Without Time Stepping," *IEEE Microwave and Guided Wave Letters*, vol. 9, no. 1, pp. 4-6, 1999.

[13] M. Celik, A. C. Cangellaris, "Simulation of Multiconductor Transmission Lines Using Krylov Subspace Order-Reduction Techniques," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 5, pp. 485-96, 1997.

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

[15] J. Chen, S-M. Kang, "Model-order reduction of nonlinear MEMS devices through arclength-based Karhunen-Loève decomposition," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 2, 2001, pp. 457-60.

[16] Y. Chen, "Model Order Reduction for Nonlinear Systems", M.S. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1999.

[17] Y. Chen, J. White, "A Quadratic Method for Nonlinear Model Order Reduction," in *Proceedings of the International Conference on Modeling and Simulation of Microsystems*, 2000, pp. 477-80.

[18] M. Chou, J. White, "Efficient Reduced-Order Modeling for the Transient Simulation of Three-dimensional Interconnect," in *Proceedings of the 1995 IEEE/ACM International Conference on Computer-Aided Design*, 1995, pp. 40-4.

[19] L. Daniel, A. Sangiovanni-Vincentelli, J. White, "Techniques for including dielectrics when extracting passive low-order models of high speed interconnect," in *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, 2001, pp. 240-4.

[20] L. Daniel, C. S. Ong, S. C. Low, K. H. Lee, J. White, "Geometrically parameterized interconnect performance models for interconnect synthesis," in *Proceedings of 2002 International Symposium on Physical Design*, 2002, pp. 202-7.

[21] L. Daniel, J. Phillips, "Model Order Reduction for Strictly Passive and Causal Distributed Systems," in *Proceedings of the Design Automation Conference*, 2002, pp. 46 -51.

[22] C. de Villemagne, R. E. Skelton, "Model reduction using a projection formulation," *International Journal of Control*, vol. 46, pp. 2141-69, 1987.

[23] C. T. Dikmen, M. M. Alaybeyi, S. T, A. Atalar, E. Sezer, M. A. Tan, R. A. Rohrer, "Piecewise Linear Asymptotic Waveform Evaluation for Transient Simulation of Electronic Circuits," in *Proceedings of the International Symposium on Circuits and Systems*, vol. 2, 1991, pp. 854-7.

[24] I. M. Elfadel, D. D. Ling, "Zeros and Passivity of Arnoldi-Reduced-Order Models for Interconnect Networks," in *Proceedings of the 34th Design Automation Conference*, 1997, pp. 28-33.

[25] P. Feldmann, R. W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 5, pp. 639-49, 1995.

[26] R. W. Freund, P. Feldmann, "Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm," in *Proceedings of the 1996 IEEE/ACM International Conference on Computer-Aided Design*, 1996, pp. 280-7.

[27] L. D. Gabbay, J. E. Mehner, S. D. Senturia, "Computer-Aided Generation of Nonlinear Reduced-Order Dynamic Macromodels – I: Non-Stress-Stiffened Case," *Journal of Microelectromechanical Systems*, vol. 9, no. 2, pp. 262-9, 2000.

[28] E. Gad, A. Dounavis, M. Nakhla, R. Achar, "Passive model order reduction of multiport distributed interconnects," in *Proceedings of the 37th Design Automation Conference*, 2000, pp. 526-31.

[29] K. Gallivan, E. Grimme, P. Van Dooren, "Asymptotic waveform evaluation via a Lanczos method," *Applied Mathematics Letters*, vol. 7, no. 5, pp. 75-80, 1994.

[30] K. Gallivan, E. Grimme, P. Van Dooren, "Padé Approximation of Large-Scale Dynamic Systems with Lanczos Methods," in *Proceedings of the 33rd Conference on Decision and Control*, vol. 1, 1994, pp. 443-8.

[31] E. Gallopoulos, Y. Saad, "Efficient solution of parabolic equations by Krylov approximation methods," *SIAM Journal of Scientific and Statistical Computing*, vol. 13, no. 5, pp. 1236-64, 1992.

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

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

[34] E. J. Grimme, "Krylov Projection Methods for Model Reduction," Ph.D. dissertation, University of Illinois at Urbana-Champaign, IL, 1997.

[35] J. K. Hale, *Ordinary Differential Equations*, John Wiley and Sons Inc., New York, 1969.

[36] Y. Halevi, "Frequency Weighted Model Reduction via Optimal Projection," *IEEE Transactions on Automatic Control*, vol. 37, no. 10, pp. 1537-42, 1992.

[37] K. C. Hall, J. P. Thomas, E. H. Dowell, "Reduced-order modelling of unsteady small-disturbance flows using a frequency-domain proper orthogonal decomposition technique," in *Proceedings of the 37th Aerospace Sciences Meeting and Exhibit*, AIAA Paper 99-0655, 1999.

[38] P. Heydari, M. Pedram, "Model reduction of variable-geometry interconnects using variational spectrally-weighted balanced truncation," in *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, 2001, pp. 586-91.

[39] C. Hirsch, *Numerical Computation of Internal and External flows*, Wiley, New York, 1988.

[40] E. Hung, Y.-J. Yang, S. D. Senturia, "Low-Order Models For Fast Dynamical Simulation of MEMS Microstructures," in *Proceedings of the IEEE International Conference on Solid State Sensors and Actuators (Transducers '97)*, vol. 2, 1997, pp. 1101-04.

[41] E. S. Hung, S. D. Senturia, "Generating Efficient Dynamical Models for Microelectromechanical Systems from a Few Finite-Element Simulation Runs," *IEEE Journal of Microelectromechanical Systems*, vol. 8, no. 3, pp. 280-9, 1999.

[42] M. Innocent, P. Wambacq, S. Donnay, H. A. C. Tilmans, W. Sansen, H. De Man, "An Analytic Volterra-Series-Based Model for a MEMS Variable Capacitor," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 1, pp. 124-31, 2003.

[43] M. Kamon, N. A. Marques, L. M. Silveira, J. White, "Generating Reduced Order Models via PEEC for Capturing Skin and Proximity Effects," in *Proceedings of the 6th Topical Meeting on Electrical Performance of Electronic Packaging*, 1997, pp. 259-62.

[44] M. Kamon, N. A. Marques, L. M. Silveira, J. White, "Automatic Generation of Accurate Circuit Models of 3-D Interconnect," *IEEE Transactions on Components Packaging, and Manufacturing Technology – Part B*, vol. 21, no. 3, pp. 225-40, 1998.

[45] M. Kamon, F. Wang, J. White, "Generating nearly optimally compact models from Krylov-subspace based reduced-order models," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 4, pp. 239-48, 2000.

[46] R. Kao, M. Horowitz, "Timing analysis for Piecewise Linear Rsim," *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, vol. 13, no. 12, pp. 1498-512, 1994.

[47] F. Karray, T. A. W. Dwyer, III, "On the nonlinear system identification of a class of bilinear dynamical models," in *Proceedings of the 30th IEEE Conference on Decision and Control*, vol. 1, 1991, pp. 778-82.

[48] H. K. Khalil, *Nonlinear systems*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 1996.

[49] T. Kim, "Frequency-Domain Karhunen-Loève Method and Its Application to Linear Dynamic Systems," *AIAA Journal*, vol. 36, no. 11, pp. 2117-23, 1998.

[50] M. J. Korenberg, "Identifying Nonlinear Difference Equation and Functional Expansion Representations: the Fast Orthogonal Algorithm," *Annals of Biomedical Engineering*, vol. 16, pp. 123-42, 1988.

[51] M. J. Korenberg, I. W. Hunter, "The Identification of Nonlinear Biological Systems: Volterra Kernel Approaches," *Annals of Biomedical Engineering*, vol. 24, pp. 250-68, 1996.

[52] M. E. Kowalski, J.-M. Jin, "Karhunen-Loève Based Model Order Reduction of Nonlinear Systems," in *Proceedings of the IEEE Antennas and Propagation Society International Symposium*, vol. 1, 2002, pp. 552-5.

[53] L. Kulas, M. Mrozowski, "Reduced Order Models in FDTD," *IEEE Microwave and Wireless Components Letters*, vol. 11, no. 10, pp. 422-4, 2001.

[54] K. Kundert, J. White, A. Sangiovanni-Vincentelli, "A mixed frequency-time approach for finding the steady-state solution of clocked analog circuits," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, 1988, pp. 6.2/1-6.2/4.

[55] K. Kundert, J. White, A. Sangiovanni-Vincentelli, "A mixed frequency-time approach for distortion analysis of switching filter circuits," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 2, pp. 443-51, 1989.

[56] K. S. Kundert, "Introduction to RF Simulation and Its Application," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 9, pp. 1298-319, 1999.

[57] K. Kundert, H. Chung, D. Jefferies, G. Lamant, E. Malavasi, F. Sendig, "Design of Mixed-Signal Systems-on-a-Chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 12, pp. 1561-71, 2000.

[58] S. Lall, J. E. Marsden, S. Glavaski, "Empirical model reduction of controlled nonlinear systems," in *Proceedings of the 14th International Federation of Automatic Control World Congress*, vol. 6, 1999, pp. 473-8.

[59] S. Lall, J. E. Marsden, S. Glavaski, "A subspace approach to balanced truncation for model reduction of nonlinear systems," *International Journal of Robust and Nonlinear Control*, vol. 12, pp. 519-35, 2002.

[60] G. Lassaux, "High-Fidelity Reduced-Order Aerodynamic Models: Application to Active Control of Engine Inlets," M. S. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2002.

[61] K. S. Lee, Y. Eom, J. W. Chung, J. Choi, D. Yang, "A control-relevant model reduction technique for nonlinear systems," *Computers and Chemical Engineering*, vol. 24, pp. 309-315, 2000.

[62] D. J. Leith, W. E. Leithead, "Analysis of Gain-Scheduled and Nonlinear Systems by Velocity-Based Linearisation Families," in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 1, 1997, pp. 424-9.

[63] S. P. Levitan, J. A. Martinez, T. P. Kurzweg, A. J. Davare, M. Kahrs, M. Bails, D. M. Chiarulli, "System Simulation of Mixed-Signal Multi-Domain Microsystems With Piecewise Linear Models," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 139-54, 2003.

[64] J.-R. Li, J. White, "Efficient model reduction of interconnect via approximate system Gramians," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 1999, pp. 380-3.

[65] J.-R. Li, "Model Reduction of Large Linear Systems via Low Rank System Gramians," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2000.

[66] Y. C. Liang, W. Z. Lin, H. P. Lee, S. P. Lim, K. H. Lee, D. P. Feng, "A neural-network-based method of model reduction for the dynamic simulation of MEMS," *Journal of Micromechanics and Microengineering*, vol. 11, pp. 226-33, 2001.

[67] Y. Liu, L. T. Pileggi, A. J. Strojwas, "Model order-reduction of RC(L) interconnect including variational analysis," in *Proceedings of the 36th Design Automation Conference*, 1999, pp. 201-6.

[68] A. Lumsdaine, M. Silveira, J. White, "SIMLAB User's Guide," Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA.

[69] X. Ma, J. A. De Abreu-Garcia, "On the Computation of Reduced Order Models of Nonlinear Systems Using Balancing Technique," in *Proceedings of the 27th Conference on Decision and Control*, 1988, pp. 1165-6.

[70] D. Maksimovic, A. M. Stankovic, V. J. Thottuvelil, G. C. Verghese, "Modeling and simulation of power electronic converters," *Proceedings of the IEEE*, vol. 89, no. 6, pp. 898-912, 2001.

[71] P. Mavrikis, R. B. Vinter, "Trajectory-Specific Model Reduction," in *Proceedings of the 36th Conference on Decision and Control*, 1997, pp. 3323-8.

[72] A. Megrteski, in lecture notes for "Multivariable Control Systems," Massachusetts Institute of Technology, 2002.

[73] J. E. Mehner, L. D. Gabbay, S. D. Senturia, "Computer-Aided Generation of Nonlinear Reduced-Order Dynamic Macromodels – II: Stress-Stiffened Case," *Journal of Microelectromechanical Systems*, vol. 9, no. 2, pp. 270-8, 2000.

[74] A. A. Mohammad, J. A. De Abreu-Garcia, "A transformation approach for model order reduction of nonlinear systems," in *Proceedings of the 16th Annual Conference of IEEE Industrial Electronics Society*, vol. 1, 1990, pp. 380 -3.

[75] B. C. Moore, "Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction," *IEEE Transactions on Automatic Control*, vol .26, no. 1, pp. 17-32, 1981.

[76] T. Mukherjee, G. K. Fedder, D. Ramaswamy, J. White, "Emerging Simulation Approaches for Micromachined Devices," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 12, pp. 1572-89, 2000.

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

[78] A. Odabasioglu, M. Celik, L. Pileggi, "PRIMA: Passive Reduced-Order Interconnect Macromodeling Algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 8, pp. 645-54, 1998.

[79] A. Odabasioglu, M. Celik, L. Pileggi, "Practical considerations for passive reduction of RLC circuits," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 1999, pp. 214-9.

[80] P. A. Parrilo, S. Lall, F. Paganini, G. C. Verghese, B. C. Lesieutre, J. E. Marsden, "Model reduction for analysis of cascading failures in power systems," in *Proceedings of the 1999 American Control Conference*, vol. 6, 1999, pp. 4208-12.

[81] L. Pernebo, L. M. Silverman, "Model Reduction via Balanced State Space Representations," *IEEE Transactions on Automatic Control*, vol. 27, no. 2, pp. 382-7, 1982.

[82] J. R. Phillips, "Model Reduction of Time-Varying Linear Systems Using Approximate Multipoint Krylov-subspace Projectors," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 1998, pp. 96-102.

[83] J. R. Phillips, "Automated extraction of nonlinear circuit macromodels," in *Proceedings of the Custom Integrated Circuit Conference*, 2000, pp. 451-4.

[84] J. R. Phillips, "Projection frameworks for model reduction of weakly nonlinear systems," in *Proceedings of the 37th Design Automation Conference*, 2000, pp. 184-9.

[85] J. Phillips, L. Daniel, L. M. Silveira, "Guaranteed passive balancing transformations for model order reduction," in *Proceedings of the 39th Design Automation Conference*, 2002, pp. 52-7.

[86] J. R. Phillips, "Projection-Based Approaches for Model Reduction of Weakly Nonlinear, Time-Varying Systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 171-87, 2003.

[87] P. Rabiei, M. Pedram, "Model order reduction of large circuits using balanced truncation," in *Proceedings of the ASP-DAC Design Automation Conference*, vol. 1, 1999, pp. 237-40.

[88] D. Ramaswamy, "Automatic Generation of Macromodels for MicroElectroMechanical Systems (MEMS)," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2001.

[89] M. Rathinam, L. R. Petzold, "An iterative method for simulation of large scale modular systems using reduced order models," in *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 5, 2000, pp. 4630-5.

[90] R. F. Remis, P. M. van den Berg, "A Modified Lanczos Algorithm for the Computation of Transient Electromagnetic Wavefields," *IEEE Transactions on Microwave Theory and Techniques*, vol. 45, no. 12, pp. 2139-49, 1997.

[91] M. Rewieński, J. White, "A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices," in *Proceedings of the International Conference on Computer-Aided Design*, 2001, pp. 252-7.

[92] M. Rewieński, J. White, "Improving Trajectory Piecewise-Linear Approach to Nonlinear Model Order Reduction for Micromachined Devices Using an Aggregated Projection Basis," in *Proceedings of the 5th International Conference on Modeling and Simulation of Microsystems*, 2002, pp. 128-31.

[93] M. Rewieński, J. White, "A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 155-70, 2003.

[94] J. Roychowdhury, "Reduced-order modelling of time-varying systems," in *Proceedings of the Asia and South Pacific Design Automation Conference*, vol. 1, 1999, pp. 53-6.

[95] J. M. A. Scherpen, "Balancing for nonlinear systems," *Systems and Control Letters*, vol. 21, pp. 143-53, 1993.

[96] J. S. Shamma, "Analysis and Design of Gain Scheduled Control Systems," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1988.

[97] B. N. Sheehan, "ENOR: Model Order Reduction of RLC Circuits Using Nodal Equations for Efficient Factorization," in *Proceedings of the 36th Design Automation Conference*, pp. 17-21, 1999.

[98] B. N. Sheehan, "Projective convolution: RLC model-order reduction using the impulse response," in *Proceedings of the 1999 Design, Automation and Test in Europe Conference and Exhibition*, 1999, pp. 669-73.

[99] S. Sidi-Ali-Cherif, K. M. Grigoriadis, "Efficient model reduction of large scale systems using Krylov-subspace iterative methods," *International Journal of Engineering Science*, vol. 41, pp. 507-20, 2003.

[100] L. M. Silveira, M. Kamon, J. White, "Efficient Reduced-Order Modeling of Frequency-Dependent Coupling Inductances associated with 3-D Interconnect Structures," in *Proceedings of the European Design and Test Conference*, 1995, pp. 534-8.

[101] L. M. Silveira, M. Kamon, I. Elfadel, J. White, "A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 1996. pp. 288-94.

[102] L. Sirovich, "Turbulence and the Dynamics of Coherent Structures. Part 1: Coherent Structures," *Quarterly of Applied Mathematics*, vol. 45, no. 3, pp. 561-71, 1987.

[103] R. D. Slone, "Removing the frequency restriction on Padé Via Lanczos with an error bound," in *Proceedings of the 1998 IEEE International Symposium on Electromagnetic Compatibility*, 1998, pp. 202-7.

[104] R. D. Slone, R. Lee, J.-F. Lee, "Multiport Galerkin Asymptotic Waveform Evaluation for Model Order Reduction of Frequency Domain FEM Electromagnetic Radiation Problems," *IEEE Transactions on Antennas and Propagation*, vol. 49, no. 10, pp. 1504-13, 2001.

[105] D. C. Sorensen, A. C. Antoulas, "The Sylvester equation and approximate balanced reduction," *Linear Algebra and its Applications*, vol. 351-352, pp. 671-700, 2002.

[106] D. J. Stilwell, W. J. Rugh, "Interpolation Methods for Gain Scheduling," in *Proceedings of the 37th IEEE Conference on Decision and Control*, vol. 3, 1998, pp. 3003-8.

[107] D. G. Vasilyev, M. Rewieński, J. White, "A TBR-based Trajectory Piecewise-Linear Algorithm for Generating Accurate Low-order Models for Nonlinear Analog Circuits and MEMS," in *Proceedings of the 40th Design Automation Conference*, 2003 (in press).

[108] M. Vidyasagar, *Nonlinear systems analysis*, 2nd ed., Prentice Hall, Englewood Cliffs, NJ, 1993.

[109] M. Vlach, "Modeling and simulation with Saber," in *Proceedings of the Third Annual IEEE ASIC Seminar and Exhibit*, 1990, pp. T/11.1-T/11.9.

[110] F. Wang, J. White, "Automatic model order reduction of a microdevice using the Arnoldi approach," in *Proceedings of the International Mechanical Engineering Congress and Exposition*, 1998, pp. 527-30.

[111] J. M. Wang, Q. Yu, E. S. Kuh, "Coupled Noise Estimation for Distributed RC Interconnect Model," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, 1999, pp. 664-8.

[112] J. M. Wang, C.-C. Chu, Q. Yu, E. S. Kuh, "On Projection-Based Algorithms for Model-Order Reduction of Interconnects," *IEEE Transactions on Circuits and Systems–I: Fundamental Theory and Applications*, vol. 49, no. 11, pp. 1563-85, 2002.

[113] D. S. Weile, E. Michielssen, "Analysis of Frequency Selective Surfaces Using Two-Parameter Generalized Rational Krylov Model-Order Reduction," *IEEE Transactions on Antennas and Propagation*, vol. 49, no. 11, pp. 1539-49, 2001.

[114] J. K. White, A. Sangiovanni-Vincentelli, *Relaxation Techniques for the Simulation of VLSI Circuits*, Kluwer Academic Publishers, Norwell, MA, 1987.

[115] K. E. Willcox, J. D. Paduano, J. Peraire, K. C. Hall, "Low order aerodynamic models for aeroelastic control of turbomachines," in *Proceedings of the American Institute of Aeronautics and Astronautics Structures, Structural Dynamics, and Materials Conference*, vol. 3, 1999, pp. 2204-14.

[116] K. Willcox, J. Peraire, "Balanced Model Reduction via the Proper Orthogonal Decomposition," *AIAA Journal*, vol. 40, no. 11, pp. 2323-30, 2002.

[117] K. Willcox, J. Peraire, J. White, "An Arnoldi approach for generation of reduced-order models for turbomachinery," *Computers and Fluids*, vol. 31, no. 3, pp. 369-89, 2002.

[118] Y.-J. Yang, M. Kamon, V. L. Rabinovich, C. Ghaddar, M. Deshpande, K. Greiner, J. R. Gilbert, "Modeling gas damping and spring phenomena in MEMS with frequency dependent macro-models," in *Proceedings of the 14th IEEE International Conference on Micro Electro Mechanical Systems*, 2001, pp. 365-8.

[119] Q. Yu, J. M. L. Wang, E. S. Kuh, "Passive Multipoint Moment Matching Model Order Reduction Algorithm on Multiport Distributed Interconnect Networks," *IEEE Transactions on Circuits and Systems–I: Fundamental Theory and Applications*, vol. 46, no. 1, pp. 140-60, 1999.

[120] Y. Zhang, M. A. Henson, Y. G. Kevrekidis, "Nonlinear model reduction for dynamic analysis of cell population models," *Chemical Engineering Science*, vol. 58, pp. 429-45, 2003.

[121] D. Zhou, W. Li, W. Cai, N. Guo, "An efficient balanced truncation realization algorithm for interconnect model order reduction," in *Proceedings of the 2001 IEEE International Symposium on Circuits and Systems*, vol. 5, 2001, pp. 383-6.