Theoretical and practical aspects of linear and nonlinear model order reduction techniques

by

Dmitry Missiuro Vasilyev

B.S. Physics,
Saint Petersburg State Polytechnical University, Russia, 1998
M.S. Physics,
Saint Petersburg State Polytechnical University, Russia, 2000

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

February 2008

Author . . . .
Department of Electrical Engineering and Computer Science
December 20, 2007

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

Accepted by . . . .
Professor Terry P. Orlando
Chairman, Department Committee on Graduate Students

Theoretical and practical aspects of linear and nonlinear model order reduction techniques

by

Dmitry Missiuro Vasilyev

Submitted to the Department of Electrical Engineering and Computer Science
on December 20, 2007, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Model order reduction methods have proved to be an important technique for accelerating time-domain simulation in a variety of computer-aided design tools. In this study we present several new techniques for model reduction of the large-scale linear and nonlinear systems.

First, we present a method for nonlinear system reduction based on a combination of the trajectory piecewise-linear (TPWL) method with truncated-balanced realizations (TBR). We analyze the stability characteristics of this combined method using perturbation theory.

Second, we describe a linear reduction method that approximates TBR model reduction and takes advantage of sparsity of the system matrices or available accelerated solvers. This method is based on AISIAD (approximate implicit subspace iteration with alternate directions) and uses low-rank approximations of a system's gramians. This method is shown to be advantageous over the common approach of independently approximating the controllability and observability gramians, as such independent approximation methods can be inefficient when the gramians do not share a common dominant eigenspace.

Third, we present a graph-based method for reduction of parameterized RC circuits. We prove that this method preserves stability and passivity of the models for nominal reduction. We present computational results for large collections of nominal and parameter-dependent circuits.

Finally, we present a case study of model reduction applied to electroosmotic flow of a marker concentration pulse in a U-shaped microfluidic channel, where the marker flow in the channel is described by a three-dimensional convection-diffusion equation. First, we demonstrate the effectiveness of the modified AISIAD method in generating a low order models that correctly describe the dispersion of the marker in the linear case; that is, for the case of concentration-independent mobility and diffusion constants. Next, we describe several methods for nonlinear model reduction when the diffusion and mobility constants become concentration-dependent.

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

Acknowledgments

First and foremost, I would like to thank my advisor Professor Jacob White. Without his guidance this thesis would have been impossible. I really appreciate his insight into numerous aspects of numerical simulation, his enthusiasm, wisdom, care and attention. Learning from him was truly an invaluable experience.

I am very grateful to my thesis committee members, Professors Luca Daniel and Alex Megretski for their help and suggestions throughout my academic life at MIT.

Also, I would like to extend my sincere appreciation to Prof. Karen Willcox, Prof. John Wyatt, and Prof. Munther Dahleh for discussions. I would like to thank my graduate counselor John Kassakian for being supportive and cheerful.

I would like to thank Michał Rewieński, who helped me to make my first steps in model order reduction.

I would like to thank current and past students of our group for keeping a spirit of collaboration, our talks and laughs and great company: Jung Hoon Lee, Dave Willis, Carlos Coelho, Kin Cheong Sou, Xin Hu, Tom Klemas, Xin Wang, Shihhsien Kuo, Jay Bardhan, Annie Vithayathil, Brad Bond, Zhenhai Zhu, Lei Zhang, Homer, Laura. Big thanks to Dr. Yehuda Avniel, Dr. Joel Phillips from Cadence, Dr. Mark Reichelt from Ansoft and Patricio Ramirez. I am thankful to our faithful lab secretary Chadwick Collins. Thanks to students and professors of the 7th and 8th floor for their kind character! The weekly group lunch should be mentioned here, too.

Now it's time to thank all my friends from Russia and my Russian friends in the States. Thanks very much to Katya Zagryadskaya and Oleg Kovalenko, Mike and Sasha Khusid, Volodya and Olga Dashevsky, Volodya Titov, Sergey Tarasenko, Andrey Voronin, Aleksei Vishentsev, Olga Samoylova, Maksim Lvov, Nikolai Nezlobin, Sergeev Rinat, Belowsov Yurij Grigorievich and Belowsov Andre.

Special thanks to my mother-in-law Wesia who is the best and wisest mother-in-law in the world and whose care for her family is really amazing. Also to my father-in-law Włodzimierz who has very kind heart and broad knowledge with passion for science. Thanks very much to my extended family and to my father, Mikhail Lamkin, who influenced my life a lot.

Thanks to Singapore-MIT alliance, National Semiconductor, MIT Presidential fellowship and SRC for supporting my research at MIT.

Finally, I am much much more than grateful to my wonderful wife, Patrycja, for everything what she is, for her charm and personality, love and dedication. Kocham ciebie, złotko!

I dedicate this thesis to my mom, whom I owe everything I have; whose sleepless nights, labour, nerves and care guarded me during whole my life in Russia. It is truly impossible to mention all the sacrifices this woman made (with but the smile on her face) for me being fed, nurtured and educated.

Contents

List of Figures

List of Tables

Notation

The following notation is used throughout this thesis:

A^T, A^*, A^{-1}	Matrix transpose, complex-conjugate transpose, inverse
$\exists, \forall, \rightarrow, \in$	There exist, for all, converges to, is element of
n	Original order (state dimensionality of the original system)
q	Reduced order (state dimensionality of the reduced system)
$\mathbb{R}^n, \mathbb{R}^{n\times n}$	Space of all real vectors of length n, space of real matrices $n \times n$
$x \in \mathbb{R}^n$	state vector of the original system
$z \in \mathbb{R}^q$	state vector of the reduced system
k, m	Number of output signals and number of input signals
$u(t) \in \mathbb{R}^m,\ y(t) \in \mathbb{R}^k$	vector of input signals, vector of output signals
s, j	Complex frequency, imaginary unity
$\Re, \Im$	Real part, imaginary part
A^r	The superscript r denotes the matrix of a reduced system.
(A, B, C, D)	Linear state-space dynamical system (1.7)
$G(s)$	Transfer function of a linear dynamical system (1.6)
G, C	Conductance matrix, capacitance matrix (in Chapter 6)
U, V	Projection matrices (Section 2.2)
$\lambda_i(A), \Lambda(A)$	i-th eigenvalue of A, spectrum of A
$\sigma_1, \dots \sigma_n$	Hankel singular values of the system (Page 44)
P, Q	System's controllability and observability gramians (Section 2.2.3)
colsp	Column span (also called *range*) of a matrix
$\mathrm{diag}(x_1, \dots, x_k)$	Diagonal (as well as block-diagonal) matrix with diagonal $x_1, \dots x_k$.
$\mathcal{K}_\nu(A, B)$	Krylov subspace; $\mathcal{K}_\nu(A, B) = \mathrm{colsp}\{B, AB, A^2B, \dots A^{\nu-1}B\}$
$\subseteq$	Subset
$\left\{ \frac{\partial f}{\partial x}\Big\|_{x_0} \right\}$	Jacobian matrix (matrix of derivatives) of f at the point x_0
$\\| \cdot \\|$	2-norm of a vector or a matrix
$\mathrm{cond}(A)$	Matrix condition number
$qr(A), svd(A)$	QR-decomposition of A, singular value decomposition of A

Chapter 1

Introduction

Any intelligent fool can make things bigger, more complex, and more violent.
It takes a touch of genius, and a lot of courage to move in the opposite direction.

- Albert Einstein

The topic of model order reduction has a lot of interpretations. In general, the terms similar to "reduced-basis approximation" or "dimensionality reduction" can be found virtually in any engineering discipline. In this work we consider approximations of *continuous-time dynamical systems*.

Model order reduction algorithms, broadly speaking, aim at approximating a "complex system" (in the sense, which will be described later) by a "simpler system", while preserving, as much as possible, input-output properties of this system.

In this thesis, we focus on several methods of model reduction for linear and nonlinear dynamical systems.

1.1 Motivations for model reduction

In this section, we outline several applications of MOR techniques.

1.1.1 Compact macromodels for system-level simulation and optimization

There is a strong need for obtaining compact dynamical models for accelerated simulation of complex interconnected systems, such as integrated circuits (ICs). In such systems it is usually possible to obtain a large discretized model of each subsystem from first principles, however simulating the overall system using these big models of subsystems is computationally infeasible. Therefore, first each subsystem must be approximated by a smaller dynamical system.

In various design problems some parameters of the physical system need to be found, in order to optimize performance of a device. In such cases the need for a *parameterized reduced model* arises. Employing such models significantly accelerates system-level optimization, since during any optimization cycle the model is being evaluated many times, for different values of design parameters.

1.1.2 Real-time control systems

Model reduction is also essential when an active control is being used in a feedback system. Very frequently the result of an optimal feedback design is a high-order dynamical system, which is expensive to implement. Model reduction can help in reducing the complexity of such controller.

1.2 Background and problem formulation

1.2.1 Dynamical systems

In general, a *dynamical system* is a mapping from the space of *input signals* to the space of *output signals*. By the term *signal* we mean a real vector-valued function of a *time variable* [59]. Dynamical systems can be divided into two main classes, based on the nature of this time variable:

- Discrete-time (DT) system, where the time variable is a set of integers.

- Continuous-time (CT) systems, where the time variable is a set of real numbers.

In this work we consider solely model reduction problem for continuous-time systems. In general, when a continuous-time system is being simulated on a computer, it is always converted to a discrete-time system by means of *time discretization*:

$$\hat{u}[i] \approx u(i\tau),\ \hat{x}[i] \approx x(i\tau),\ \hat{y}[i] \approx y(i\tau),$$

where $u(t), x(t)$ and $y(t)$ are the continuous time input, state and output (for the case of a state-space system, more details on the state-space models below), while $\hat{u}[i], \hat{x}[i]$ and $\hat{y}[i]$ are the corresponding sampled (discretized) approximations. However, very frequently not only a time-step τ can change during the simulation, but also the discretization algorithm. Therefore, both the input and the output of model reduction routines for physical-domain simulations are generally continuous-time dynamical systems[1].

In this work we consider solely *time-invariant (TI)* systems, namely the systems which do not explicitly depend on the time variable: if an input signal $u(t)$ produces an output $y(t)$, than for input $u(t+\delta)$ the system will produce output $y(t+\delta)$. Here and for the rest of this work, $u(t)$ denotes an input signal and $y(t)$ denotes an output signal.

Another assumption which will be implicitly made throughout this work is the *causality* of the systems. We assume that the system's output at any time is completely determined by past and current values of inputs, and is independent of the future input values.

There are several most frequently used model descriptions for CT time-invariant systems (sorted by increasing complexity of analysis)[88]:

1. Memoryless dependence:
$$y(t) = f(u(t)) \tag{1.1}$$

[1] Quite frequently MOR routines use conversions to DT model to perform certain computations.

2. An ordinary differential equation (ODE) form, also called a *state-space* form:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = g(x(t), u(t)) \end{cases}, \quad x(t) \in \mathbb{R}^n \tag{1.2}$$

   Here a time-dependent vector $x(t)$ called *state* summarizes all the past inputs $u(t)$ needed to evaluate future outputs $y(t)$ of the system. In certain cases, state-space models are written in a more general form:

$$\begin{cases} \frac{dh(x)}{dt} = f(x(t), u(t)) \\ y(t) = g(x(t), u(t)) \end{cases}, \quad x(t) \in \mathbb{R}^n, \quad \begin{array}{l} h(\cdot) : \mathbb{R}^n \to \mathbb{R}^n \\ f(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \\ g(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^k \end{array} \tag{1.3}$$

   Such descriptions arise, for example, from simulation of electrical circuits with nonlinear capacitors and/or nonlinear inductors[2].

3. A delay-differential equation (DDE), (also called time-delay system):

$$\begin{cases} \dot{x}(t) = f(x(t), x(t-\tau_1), \ldots, x(t-\tau_v), u(t)) \\ y(t) = g(x(t), x(t-\tau_1), \ldots, x(t-\tau_v), u(t)) \end{cases} \tag{1.4}$$

4. A system of (nonlinear) partial differential equations (PDE):

$$\begin{cases} \mathcal{F}(x, \frac{\partial x}{\partial t}, \frac{\partial x}{\partial w_1}, \ldots, \frac{\partial x}{\partial w_v}, w_1, \ldots, w_v, u) = 0 \\ y = \mathcal{G}(x, w_1, \ldots, w_v, u) \end{cases}, \quad x(w_1, \ldots, w_v, t) \in \mathbb{R}^\nu \tag{1.5}$$

   Here the state x is a vector-valued function of v continuous variables $w_1, \ldots, w_v$ and time; $\mathcal{F}$ and $\mathcal{G}$ are vector-valued nonlinear operators.

[2] In general, for some choice of functions $f(\cdot)$ and $g(\cdot)$ with certain initial conditions, the equation (1.3) may not have a solution (or may have infinitely many of them). These "troublesome cases" appear for such states x when the Jacobian of g (matrix of derivatives) $\left\{\frac{\partial g}{\partial x}\right\}$ becomes singular, and therefore effectively some equations represent a constraint rather than the implicit differential equation. We are considering the equation (1.3) with implicit assumption that nonlinearities in $g(\cdot)$ are weak enough to make its Jacobian singular for practical values of x.

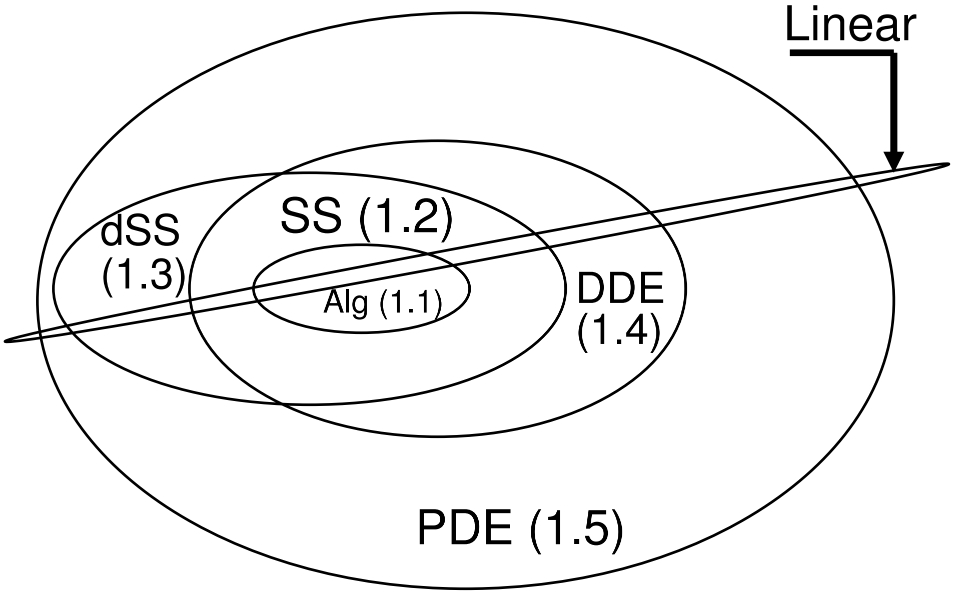

Figure 1-1: The set relationships between different dynamical system descriptions

All of the above descriptions define a certain subspace in a (linear) space of all possible dynamical systems (for a given number of input and output signals), and each can be *linear* or *nonlinear*. The above mentioned descriptions, along with the appropriate *initial conditions* for non-algebraic relationships, uniquely identify a dynamical system. However, there is a redundancy in the descriptions via differential equations, for example, in the descriptions (1.3, 1.4) we can always change variables as $x = V\tilde{x}$, using any square nonsingular matrix $V \in \mathbb{R}^{n\times n}$ and obtain a system with completely identical input-output behavior.

When modeling physical devices, the first-principle descriptions are usually described in the form (1.5), and by using a *spatial discretization* methods (for example, finite-difference, finite-volume, finite-element) or by other kinds of methods they are converted to either (1.2) or (1.3).

The encapsulation scheme of the classes above is depicted on figure 1-1.

1.2.2 Stability of a dynamical system

For an arbitrary dynamical system there are two major kinds of stability notions: *internal stability* and *external stability.* The internal stability considers trajectories of an *autonomous* system $\dot{x}(t) = f(x(t))$, i.e. system without any inputs and outputs; this way, it is a property of internal dynamics of the system. External stability concerns with how much the system amplifies signals. One of such notions is a *BIBO-stability* (BIBO stands for "bounded input - bounded output"). The system is BIBO-stable if and only if any bounded input signals will necessarily produce bounded output signals.

In our work we will consider mostly the internal stability of systems locally near *equilibrium states.* An equilibrium state is a state that the autonomous system can maintain for an infinite time. There are many notions of local internal stability, among which we will be mostly concerned with two:

Definition 1.2.1 (Lyapunov stability)**.** *An equilibrium state* x_0 *is called* Lyapunov-stable*, if* $\forall \epsilon > 0$*,* $\exists \delta > 0$*, such that if* $\|x(t_0) - x_0\| < \delta$*, then* $\|x(t) - x_0\| < \epsilon$*,* $\forall t > t_0$*.*

A stronger notion is the following:

Definition 1.2.2 (Asymptotic stability)**.** *An equilibrium state* x_0 *is called* asymptotically stable *if it is stable in sense of Lyapunov and in addition* x_0 *is attractive, i.e.* $\exists \gamma > 0$ *such that if* $\|x(t_0) - x_0\| < \gamma$*, then* $x(t) \to x_0$*, as* $t \to \infty$*.*

A frictionless pendulum with x_0 corresponding to the resting state is an example of a Lyapunov-stable system around x_0, but not asymptotically stable. A pendulum with friction is an asymptotically stable system around such x_0.

The system $\dot{x}_1 = ux_1,\ y = x_1$ has any point x_1 as an internal equilibrium; it is Lyapunov-stable around any x_1, but not asymptotically stable. It is not BIBO stable, either.

1.2.3 Passivity of the dynamical model

There is a number of problem-dependent constraints, apart from stability, which are required for some models of real devices. One such constraint, which is very important for circuit simulation, is called *passivity.* A model of a circuit is passive if it doesn't generate energy. This notion ultimately depends on the nature of the input and output signals. For example, if the dynamical model represents a passive linear subcircuit[3] where input signals are port currents and output signals are port voltages, the passivity constraint would require such a system to be *positive-real* [87]. We will have more to say about such constraints in the following Chapters.

1.2.4 Linear dynamical models

The subset of linear time-invariant (LTI) dynamical systems within each class described in (1.1 - 1.5) is pretty obvious: one just needs to restrict all of the functions $g(\dots)$, $f(\dots)$, $h(\dots)$, $\mathcal{F}(\dots)$, $\mathcal{G}(\dots)$ to be linear with respect to their arguments. In addition, since we are considering only causal systems, we are assuming zero initial conditions of all state variables at $t = -\infty$.

An essential property of any linear dynamical system is its *transfer function* $G(s)$, which describes an algebraic relation between input and output signals in the Laplace domain:

$$Y(s) = G(s)U(s), \quad U(s) = \mathcal{L}(u(t)) \triangleq \int_{-\infty}^{\infty} e^{-st}u(t)dt, \quad Y(s) = \mathcal{L}(y(t)), \qquad (1.6)$$

where $U(s)$ and $Y(s)$ denote Laplace transforms (vector-valued functions of a complex frequency s) of the input and output signals, respectively. Alternatively, one can think of a system's transfer function as a Laplace transform of system's *impulse response.* Strictly speaking, a given transfer function may correspond to several different dynamical systems (depending on the region of convergence), however if one assumes that the system is causal, specifying a transfer function uniquely specifies

[3] A passive subcircuit is a part of a circuit, which does not contain any kind of energy sources.

the linear dynamical system[4].

1.2.5 Model reduction methods - problem setup

As it is shown on the figure 1-1, the most general model description is based on partial differential equations (PDE). When simulating physical devices with dynamics, the first-principle descriptions are usually of this kind. However, such a description usually cannot be directly simulated on a computer. Only systems in the form (1.2), (1.3) or (1.4) can be simulated using generic ODE or DDE solvers [73, 80]. Therefore, *spatial discretization* is required. However, obtaining a compact discretized dynamical model for a given PDE and geometry is usually very difficult, unless the system has a very simple geometry. There exist several model reduction methods which aim at producing compact macromodels directly from a PDE description of, for example, transmission lines [24]; however such methods are targeted at very particular devices. All generic discretization algorithms, such as the *finite-element method* and the *finite-difference method*, usually produce large dynamical system descriptions in the form of a state-space (1.2) or descriptor state space (1.3).

The departing point of almost all general-purpose model reduction methods are state-space models (1.2) or (1.3), or their linear counterparts:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}, \qquad \begin{array}{l} x \in \mathbb{R}^n, A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m} \\ C \in \mathbb{R}^{k\times n},\ D \in \mathbb{R}^{k\times m} \end{array} \tag{1.7}$$

and

$$\begin{cases} E\dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}, \qquad \begin{array}{l} x \in \mathbb{R}^n, A, E \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m} \\ C \in \mathbb{R}^{k\times n},\ D \in \mathbb{R}^{k\times m} \end{array} \tag{1.8}$$

Summarizing, the model construction workflow for simulation can be outlined as

[4] Laplace transform of a function does not uniquely specify it's original: for example, the transform $\frac{1}{s+1}$ may correspond to either $e^{-t}h_h(t)$, or $-e^{-t}h_h(-t)$, where $h_h(\cdot)$ is the Heaviside step function. Specifying additionally the region of convergence of the Laplace transform uniquely specifies the original function. Initial conditions play similar role in the time-domain system description via ODE.

the following:

1. Given a system to be simulated, obtain a (large) state-space model by using any applicable discretization technique (for example, finite-element method, finite-difference method, etc.).

2. Reduce the number of unknowns in the state vector using model reduction.

1.2.6 Reduction accuracy measures

How can we quantify how well a reduced system "approximates" the original system? One way is to define an error signal, $e(t)$, as the difference between outputs of the reduced and the original system for the same input signal $u(t)$. We can use any measure of the resulting map $u(t) \rightarrow e(t)$ to characterize the error (cf. Figure 1-2).

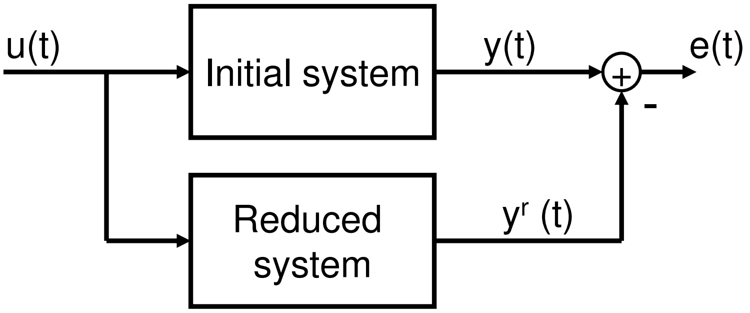

Figure 1-2: Measuring error of approximation

There are many such system measures ([18], Chapter 16); one of the most important ones is L_2-*gain*, which for linear time-invariant systems equals to H_∞ norm of the system's transfer function [53]. If such measure is used for quantification of the reduction accuracy, this would correspond to equally weighting error for all frequencies of operation. For linear systems, this would lead to the problem of H_∞-optimal reduction with error metric as in (2.21). This way, we are talking about *wide-bandwidth models*, i.e. models which should be valid in essentially large frequency span, compared to the system's dynamics. This kind of model reduction problems can be solved using balancing methods similar to the method described in Section 2.2.3. There are frequency-weighted versions of balancing methods. For an overview of such methods the reader is referred to [8].

However, very frequently the MOR accuracy is treated quite differently. As an example, when simulating passive electrical circuits it is enough to reproduce only a low-frequency behavior. For example, preserving an Elmore delay (approximate rise-time of a step response) requires a first-order accurate matching of the system's frequency response near zero frequency. Here we are concerned with *narrow-band models*. Moment-matching methods based on Krylov-subspace projections (see Section 2.2.2) are quite often used for such applications. In such cases we can link the reduction accuracy with the number of matched moments.

In various application areas other accuracy measures can be considered. For example, in Chapter 6 we are using yet another error measure (6.29), which effectively scales the error tolerance according to the magnitude of the unreduced system's response. This measure is adequate if, for example, one wants to preserve very small entries of the transfer function, or in cases where at certain frequencies the transfer function is infinitely large.

Finally, almost every application area requires the reduced models to be stable. In addition, passivity of the reduced model is also required in some cases.

1.2.7 Model complexity measures

As we have mentioned, the goal of model reduction is to approximate a "complex" dynamical system by a "simpler" one. Here by model complexity we mean the cost associated with the time of simulating the model alone or as a sub-system in a larger system. This is an informal definition of model complexity.

For linear systems such simulation cost can be treated as growing (no slower than linearly) with the dimensionality of the state vector. It is very important to note that the cost of evaluating right-hand side of (1.7) is different depending on whether matrix A is sparse or not, since majority of commercial simulators take advantage of the sparsity. From another hand, by employing the eigenvalue decomposition of A one can almost always transform A to tri-diagonal form[5]. Therefore, assuming that the number of input and output signals is not too large, the number of nonzero entries, which this model introduces into the overall Jacobian matrix[6], is proportional to the order of the system n.

For nonlinear systems an analysis of complexity is more complicated and we will describe it in Chapter 3.

1.2.8 Trade-offs and practical reduction methods

In general, there is no ideal reduction method which would solve all the problems, even for linear dynamical systems. As we have already observed for the case of the accuracy metric, different applications require different trade-offs to be chosen. In some applications the time of reduction is extremely important, so even for moderate sizes of original systems, methods like balancing (see Section 2.2.3) are too expensive. This way, the size of reduced model may be sacrificed for reduction speed. In other application areas the reduced models are being simulated and used for a long time, therefore it is extremely important to get the most compact model possible, therefore justifying more expensive reduction algorithms.

The following is a general list of such trade-off directions.

- Speed of the reduction process
- Quality of reduction (getting a smaller model for given accuracy)
- Reliability of the method (for example, preservation of stability, error bounds)
- Region of applicability (i.e. frequency range where error is small)
- Generality of the reduction method (exploring symmetry of matrices etc.)

[5] If matrix A is diagonalizable and has real spectrum, then by change of variables we can transform the system to A in a diagonal form. If A is diagonalizable and has complex-conjugate pairs of eigenvalues, we can represent each complex-conjugate pair with 2×2 block on the diagonal. In the general case of non-diagonalizable matrices, a *real Jordan form* can be used.

[6] Here we consider a case where the system to be reduced is a part of some big nonlinear network to be simulated. Sparsity of the derivative matrix (Jacobian) of this big system is very important for accelerating such simulation. All of the nonzero entries in matrices A, B, C and D of the sub-system we are reducing will be typically introduced into this global Jacobian.

As an example, Krylov-subspace methods are fast, but do not have stability and error guarantees for general systems. The TICER method described in the Chapter 6 exploits the fact that the underlying system is an RC network, and performs a narrow-band approximation, but on the positive side it is very fast and preserves passivity of the model.

The reduction problem can be posed in two alternative ways:

- Find the most compact model for a given accuracy metric
- Find the best approximation among all systems of a given complexity, which usually corresponds to the order.

Solving the first problem requires either an *a-priori* error bound, or iterative search over the space of reduced orders. Alternatively, a heuristic can be employed based on large number of training examples.

Many model reduction methods assume a particular reduced order, therefore they fall into the second category. Such methods are quite frequently used as an inner loop for the first problem. With this in mind, one of the desirable properties of such MOR methods is *incrementality*, i.e. ability to re-use computations while iterating over reduced orders.

1.3 Thesis contributions and outline

The next two Chapters are of an introductory nature and provide an in-depth overview of the model reduction methods for linear (Chapter 2) and nonlinear (Chapter 3) dynamical systems. The rest of the dissertation contains the following major contributions:

- In the Chapter 4 we analyze applicability and properties of the TBR-based TPWL nonlinear reduction method. We also provide perturbation analysis of the TBR reduction in order to assess the stability of the reduced models.
- In the Chapter 5 we present a fast approximation of the TBR linear model reduction method called modified AISIAD.

- In the Chapter 6 we present a graph-based linear reduction method for parameterized RC circuits.

- In the Chapter 7 we provide a case study for the linear and nonlinear models of a microfluidic channel.

Conclusions end the dissertation.

Chapter 2

Linear model reduction methods – an overview

Everything that can be invented has been invented.

- Charles H. Duell, Commissioner, U.S. Office of Patents, 1899.

The problem of linear model reduction can be stated as following: given a system in the form (E, A, B, C, D) as in (1.8), obtain a reduced system $(E^r, A^r, B^r, C^r, D^r)$, which approximates the original system.

Applying Laplace transform to the both sides in (1.8), yields the following expression for the transfer function $G(s)$, which is defined in (1.6):

$$G(s) = D + C(sE - A)^{-1}B. \tag{2.1}$$

That is, the transfer function is a $k \times m$ matrix, where each element is a rational function of s. Therefore, the problem of linear model reduction turned into the approximation of one matrix of rational functions by another matrix of rational functions, with smaller state-space representation.

One should note that if the matrix E in (1.8) is nonsingular, then the model reduction for the system (1.8) can be (formally) cast as a model reduction for the system $(E^{-1}A, E^{-1}B, C, D)$ in (1.7), however this transformation has two possible disadvantages. First, matrices E and A might be sparse, but matrix $E^{-1}A$ may be dense; therefore manipulations with such a model may be much more expensive. Second, computing an inverse of matrix E may lead to numerical round-off errors.

All model reduction methods can be divided into two major classes:

1. Projection-based methods

2. Non projection-based methods

In this Chapter we describe some of the most popular generic linear model reduction methods. We should note however that there is a variety of specialized methods targeted at a particular subclasses of systems, for example, RLC circuits etc. We start with some necessary background from linear systems theory.

2.1 Properties of linear dynamical systems

Though arbitrary linear time-invariant systems can exhibit even more complex behavior, the class of systems that are represented by a state space (1.8) having a transfer function (2.1) is of paramount importance to our analysis[1]. There are two important sub-classes of this class, namely *proper* and *stable* systems.

2.1.1 Proper systems

Definition 2.1.1. *The system (1.8) is called* **proper** *if all elements of the transfer function matrix $G(s)$ in (2.1) are proper rational functions of s, i.e. the degrees of the numerator polynomials are not greater than the degrees of the corresponding denominator polynomials. The system is called* **strictly proper** *if the numerator degrees are strictly less than the denominator degrees.*

In other words, a proper system is a state-space system, whose transfer function doesn't infinitely grow as $s \to j\infty$.

A sufficient condition for the system (1.8) to be proper is that matrix E is non-singular. The reverse is not necessarily true: under certain conditions a singular E results in a proper system.

[1] As before, in addition to the system's ODE (1.8) we assume causality

In general, a transfer function (2.1) with singular E can be represented as a sum of a proper transfer function and a matrix of polynomials in s:

$$G_{(E,A,B,C,D)} = G^{p.r.}(s) + \sum_{i>0} G_i s^i, \quad G_i \in \mathbb{R}^{k\times m}, \tag{2.2}$$

where $G^{p.r.}(s)$ is a matrix of proper rational functions of s.

2.1.2 Stability of linear systems

The following results are well-known in literature [18], p. 137:

Theorem 2.1.1 (Asymptotic stability of linear systems). *The causal system* (A, B, C, D) *is asymptotically stable around* $x_0 = 0$ *if and only if all eigenvalues of matrix* A *have strictly negative real parts.*

Theorem 2.1.2 (Marginal stability of linear systems). *The causal system* (A, B, C, D) *is Lyapunov-stable around* $x_0 = 0$ *if and only if all real parts of eigenvalues of* A *are nonpositive. In addition, all pure imaginary eigenvalues of* A *should have their algebraic multiplicities equal to their geometric multiplicities.*

As it can be seen, determination of the (internal) stability of any LTI system (A, B, C, D) is quite simple by computing eigenvalues of matrix A. For systems in descriptor form (E, A, B, C, D) asymptotic stability is guaranteed if and only if all finite generalized eigenvalues of the pair (E, A) lie in the open left-half plane [78].

2.2 Projection-based linear MOR methods.

Projection-based model reduction methods are by far the most widely used MOR methods [8]. In addition, projection-based methods generalize naturally to handle nonlinear systems.

Consider a dynamical system (E, A, B, C, D) as in (1.8). A reduced model can be obtained by (formally) changing the variables $x = Uz, \quad U \in \mathbb{R}^{n\times q}$, and projecting the residual in the first equation of (1.8) onto the column span of some matrix $V \in \mathbb{R}^{n \times q}$:

$$\begin{cases} V^T E U \dot{z} = V^T A U z + V^T B u \\ y = C U z + D u \end{cases}, \qquad z \in \mathbb{R}^q, \tag{2.3}$$

The equation (2.3) can be viewed as a reduced-order system $(V^T EU, V^T AU, V^T B, CU, D)$ of order q.

An essential property of the projection-based methods is the fact that the transfer function of the reduced system depends only on the column spans of the projection matrices U and V (Proposition 6 in [44], p. 23).

All projection-based methods, such as Krylov-subspace methods, proper orthogonal decomposition, truncated balanced realizations and other kinds are essentially constructing such matrices V and U according to certain considerations.

One should note that, in general, if projection matrices U and V are dense, and the original matrices A and E are sparse, then the system matrices $V^T EU$ and $V^T AU$ will be dense, and therefore the simulation of a reduced model might take longer than simulation of the original model, unless a real eigenvalue decomposition is employed for the reduced system.

The most widely used general classes of projection-based MOR methods are the following:

1. Proper Orthogonal Decomposition (POD) methods

2. Krylov-subspace and shifted Krylov-subspace methods

3. Balancing-based methods

The first two kinds of methods are generally fast and can take advantage of the sparsity of the system matrices. On the negative side, such methods are generally less reliable and are less accurate for a given order than the balancing-based methods. The latter class of methods, in general, have an $O(n^3)$ computational complexity, even for sparse models. On the positive side however, these methods possess a-priori error and stability guarantees; they generally produce more compact models.

Below we provide the most important features and descriptions of the most popular of these algorithms.

2.2.1 Proper Orthogonal Decomposition methods

This family of methods was discovered independently in many application areas, hence there are many names (POD, Karhunen-Loéve decomposition, Principal Components Analysis (PCA)) which refer to the same idea: construction of the projection basis $U = V$ from the orthogonalized[2] snapshots of the state vector at different time-points $\{x(t_1), \dots x(t_q)\}$ during simulation of some *training input* (see [11] and references therein).

The benefits of such approach are the following:

- One can re-use an existing solver in order to extract the snapshots.
- Simple to implement.
- In practice works quite reliably.
- Has a straightforward generalization for nonlinear systems.
- Fast; can take advantage of the sparsity of A and E or fast solvers.

The major drawback of this family of methods is, in general, absence of any accuracy and stability guarantees. It is not known *a-priori* how many snapshots are enough to guarantee a certain accuracy level. Also note that only information about how input signals "populate" the state space is being used: no information about the output matrix C is utilized in the construction of projection matrices. To say it another way, no information about observability is used.

There is another flavor of POD called *frequency-domain POD* [41], where the snapshots correspond to some frequencies of interest: $x_1 = (s_1 E - A)^{-1} B, \dots, x_q = (s_\nu E - A)^{-1} B$. This method stands in between POD and Krylov-subspace methods: as we shall see further, using such projection leads to the matching of frequency

[2] SVD is usually employed to eliminate "almost linearly dependent" snapshots

response of the original and reduced systems at frequencies $s_1, \dots, s_\nu$. Finally, the paper [40] extends this idea and incorporates observability measures into reduction. Similar idea was described in [65].

2.2.2 Krylov Subspace methods, also known as Rational Krylov subspace methods

In this Section, as well as in the previous one, we consider a system (1.8), where matrix E may be singular.

In order to introduce Krylov-subspace methods [30, 50, 9, 62, 29], we need the notion of *transfer function moments* of the system.

Definition 2.2.1. *Let's consider a linear system description (E, A, B, C, D) in (1.8). The transfer function moments $G^{(0)}(s_0), G^{(1)}(s_0), \dots$ at the frequency point s_0 are defined as terms in the Taylor series of the transfer function $G(s)$ near the point s_0:*

$$G(s) = G^{(0)}(s_0) + G^{(1)}(s_0)(s - s_0) + G^{(2)}(s_0)(s - s_0)^2 + \dots.$$

This way, the moments are directly related to the matrices of derivatives of the transfer function:

$$G^{(k)}(s_0) = \frac{1}{k!}\frac{d^k}{ds^k} G(s)\Big|_{s=s_0},$$

and for the state-space realization (E, A, B, C, D) in (1.8), we can take the derivative in (2.1):

$$G^{(k)}(s_0) = C\Big((A - s_0E)^{-1}E\Big)^k (A - s_0E)^{-1}B, \quad k > 0$$

The following theorem provides the basis for all Krylov-subspace methods (see [30], p. 35):

Theorem 2.2.1 (Moment Matching via Projections). *If*

$$\mathcal{K}_l\{(s_0E - A)^{-1}E, (s_0E - A)^{-1}B\} \subseteq \operatorname{colsp}\{U\}, \tag{2.4}$$

and

$$\mathcal{K}_p\{(s_0E - A)^{-T}E^T, (s_0E - A)^{-T}C^T\} \subseteq \text{colsp}\{V\}, \tag{2.5}$$

provided the matrix $(s_0E - A)$ *is invertible, then*

$$G^{(k)}(s_0) = G^{r,k}(s_0), \quad k = 0, \ldots, (l + p),$$

where $G^{r,k}(s_0)$ *denotes* k^{th} *moment of the transfer function of the reduced system* $(V^TEU, V^TAU, V^TB, CU, D)$.

The meaning of this theorem is obvious: given a set of frequency points s_i, one can obtain a reduced system which matches any given number of moments of the original system by appropriately constructing the projection matrices U and V and performing the projection (provided neither of s_i is a generalized eigenvalue of the pair (E, A)).

The advantages of Krylov-subspace methods are the following:

- Simple.

- Fast; can take advantage of the sparsity of A and E or fast solvers.

- Has been extended to parameterized models [19, 38].

The drawbacks:

- In general, lack of stability and accuracy guarantees.

- The number of the vectors in the Krylov subspaces (2.4) and (2.5) is directly proportional to the number of the input and output signals, respectively.

One should note, however, that under certain assumptions stability (and even stronger properties) of a reduced system can be guaranteed, for example in case of symmetric systems (matrices $-A$ and E are symmetric positive-definite and $B = C^T$) the system reduced with $U = V$ is guaranteed to be stable [58].

2.2.3 The balanced truncation algorithm

Below we describe the most general balancing method TBR (Truncated Balanced Realization), which was first described in [10] and further developed in [61, 28, 22]. There exist several flavors of this technique, such as Positive-Real Balancing, Stochastic Balancing etc., which preserve additional properties of the original system in the reduced one. For references reader is referred to [63, 32].

In the following derivations, we will consider a system (A, B, C, D) as in (1.7), with $D = 0_{k\times m}$, since for this method (as well as for all projection-based MOR methods) matrix D is never affected in the reduction ($D^r = D$). In this Section we assume that the system is asymptotically stable, or equivalently matrix A is Hurwitz (has all eigenvalues on the open left half of a complex plane).

In order to understand the balanced truncation method, we need to introduce two characteristics of a state: *observability* and *controllability*.

L_2 norm of signals and the induced system norm

In the derivations of this Section we will need to quantify how much "larger" is a given signal with respect to another signal. Consider a signal (vector-valued function of time) $v(t)$ defined over some time period (t_1, t_2), where times t_1 and t_2 may be infinite.

Consider the following signal measure, called L_2 *norm of a signal*:

$$\|v(t)\|_2 \triangleq \sqrt{\int_{t_1}^{t_2} v^T(t)v(t)dt}. \tag{2.6}$$

A signal, which has finite L_2 norm will be referred as *square integrable*, and the set of all such signals on (t_1, t_2) will be denoted by $L^2(t_1, t_2)$. Obviously, this set is a linear space.

In this section, we will limit the set of possible input signals by $L^2(-\infty, \infty)$, therefore we can always compute L_2 norm for both input and output signals, assuming that the system is asymptotically stable.

The above mentioned signal measure gives rise to the system's measure[3]. The L_2-induced norm (also called L_2-*gain*) of any (not necessarily linear) dynamical system is defined as the maximal amplification of the input signal by the system:

$$\|G(s)\|_{i,2} \triangleq \sup_{u \in L^2(-\infty,\infty)} \frac{\|y(t)\|_2}{\|u(t)\|_2}. \tag{2.7}$$

The induced L_2 norm of an LTI system equals to the following H_∞ norm [18]:

$$\|G(s)\|_{i,2} \equiv \|G(s)\|_\infty \triangleq \sup_\omega \sigma_{\max}(G(j\omega)), \tag{2.8}$$

where $\sigma_{\max}(G(j\omega))$ refers to the maximal singular value[4] of the system's transfer function matrix evaluated at the frequency $j\omega$.

Observability

Let's consider a system $(A, B, C, 0)$ being released from some state x_0 at $t = 0$, with zero input values for $t \geq 0$. We are interested in how much energy (quantified as L_2 norm) we will observe through the system's outputs. If the energy is large, then the state will be considered "important", otherwise it can possibly be discarded.

The zero-input response of (1.7) is:

$$y(t) = Cx(t) = Cx(0)e^{At}.$$

The L_2 norm of the output signal when the system is released from the state x_0 is the following quantity:

$$\|y(t)\|_2^2 = x_0^T \underbrace{\left[\int_0^\infty e^{A^T t} C^T C e^{At} dt\right]}_{Q} x_0 = x_0^T Q x_0. \tag{2.9}$$

The symmetric positive-semidefinite matrix Q is called an *observability gramian.*

[3] In general, if we have a linear space of operators, which act in a normed spaces, we can always define an *induced* norm in the set of operators: if $f : \mathcal{V}_1 \to \mathcal{V}_2$, then $\|f\|_i = sup_{v \in \mathcal{V}_1} \frac{\|fv\|}{\|v\|}$.

[4] Singular values of matrix A are the eigenvalues of the matrix AA^*. 2-norm of a matrix (denoted as $\|A\|_2$) equals to the maximal singular value of A. More on matrix and system norms in [18].

From our analysis it follows that if one picks x_0 being one of the eigenvectors of Q, the energy in the output will be exactly the corresponding eigenvalue of Q. The largest eigenvalue will correspond to the state which produces the largest response. However, as we see below, a simple change of basis can completely change both eigenvectors and eigenvalues of Q. In fact, Q is a matrix of *quadratic form*[5], which means that it transforms with the change of basis according to the following rule:

$$\tilde{x} = Tx \quad \Rightarrow \quad \|y(t)\|_2^2 = x_0^T Q x_0 = \tilde{x}_0^T \underbrace{T^{-T} Q T^{-1}}_{\tilde{Q}} \tilde{x}_0 \tag{2.10}$$

This means that the dominant observable states (states on the unit sphere having largest observability measure) are completely dependent on the choice of basis. A simple diagonal scaling can completely change dominant eigenvectors of Q.

The observability gramian is the solution to the following Lyapunov equation [18]:

$$A^T Q + QA + C^T C = 0. \tag{2.11}$$

Controllability

Now let's calculate how much energy in the input we need to provide in order to drive the system from zero initial condition at $t = -\infty$ to some state x_0 at $t = 0$. Note that for certain systems this is not always possible (such states are being referred as *uncontrollable*). If the state can be reached, there is an infinite set of input signals, which can achieve this goal. We need to find the signal with the smallest L_2 norm.

Assuming $x(-\infty) = 0$, the zero-state response of (1.7) is

$$x(t) = \int_{-\infty}^{t} e^{A(t-\tau)} B u(\tau) d\tau, \tag{2.12}$$

therefore we have the following linear least-squares problem for the unknown $u(t)$:

$$\text{minimize } \|u(t)\|_2^2, \quad \text{subject to } \int_{-\infty}^{0} e^{-A\tau} B u(\tau) d\tau = x_0$$

[5] For definition and basic properties of quadratic forms reader is referred to [26]

The solution to this problem is the following ([18], p. 27):

$$u(t) = B^T e^{-A^T t} \Big(\underbrace{\int_{-\infty}^{0} e^{-A\tau} B B^T e^{-A^T \tau} d\tau \Big)}_{P}^{-1} x_0$$

Therefore, the minimal energy needed to reach the state x_0 is

$$\|u(t)\|_2^2 = x_0^T \Big(\int_{-\infty}^{0} e^{-A\tau} B B^T e^{-A^T \tau} d\tau \Big)^{-1} x_0 = x_0 P^{-1} x_0 \tag{2.13}$$

The matrix P is a symmetric positive-semidefinite matrix called *controllability gramian.* It is a solution of the following Lyapunov equation [18]:

$$AP + PA^T + BB^T = 0. \tag{2.14}$$

Since P^{-1} is the matrix of a quadratic form, the controllability gramian changes with the change of coordinates according to different rules than the observability gramian:

$$\tilde{x} = Tx \quad \Rightarrow \quad \|u(t)\|_2^2 = x_o^T P^{-1} x_0 = \tilde{x}_0^T \underbrace{T^{-T} P^{-1} T^{-1}}_{\tilde{P}^{-1}} \tilde{x}_0, \quad \Rightarrow \quad \tilde{P} = TPT^T. \tag{2.15}$$

Again, the eigenvectors (as well as eigenvalues) of P are completely dependent on the choice of basis. Therefore, one can speak of dominant controllable states only relative to certain basis.

Hankel Singular Values, Hankel operator and Hankel norm of an LTI system

Let's consider how the product of the two gramians behaves with the change of the coordinates (2.15, 2.10):

$$\tilde{x} = Tx \quad \Rightarrow \quad \tilde{P}\tilde{Q} = TPT^T T^{-T} Q T^{-1} = TPQT^{-1},$$

therefore the eigenvalues of PQ are independent of the particular state-space realization of a given transfer function. In fact, being a product of two symmetric positive-semidefinite matrices, PQ has a real nonnegative spectrum.

Square roots of the eigenvalues of PQ, ordered nonincreasingly, are called *Hankel Singular Values*:

$$\sigma_i \triangleq \sqrt{\lambda_i(PQ)}, \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \tag{2.16}$$

The Hankel singular values are also the singular values of the (infinite-dimensional, but finite rank) *Hankel operator*, which maps past inputs to future outputs of the system (Section 2.3 in [28]):

Definition 2.2.2. *Hankel operator is a (linear) mapping $\Gamma_G: \; L^2(0,\infty) \to L^2(0,\infty)$ where*

$$(\Gamma_G v)(t) = \int_{-\infty}^{0} Ce^{A(t+\tau)}Bv(\tau)d\tau$$

Notice that if the input signal is $u(t) = v(-t)$ for $t < 0$ then the system's output for $t > 0$ will be $y(t) = (\Gamma_G v)$(t), assuming zero input for $t \geq 0$ (cf. Figure 2-1).

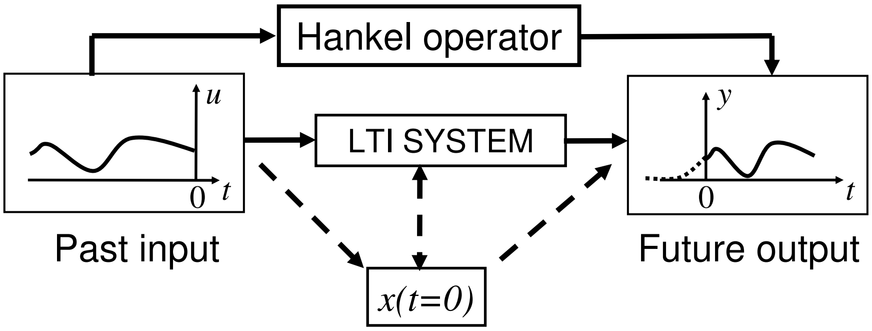

Figure 2-1: The Hankel operator maps the past inputs of the system to future outputs. The information about past inputs is stored in the system's state.

Definition 2.2.3. *Hankel norm of the system $(A, B, C, 0)$ is the largest of the Hankel singular values:*

$$\|G(s)\|_H \triangleq \sigma_1.$$

Since Hankel singular values of the system are the singular values of the Hankel operator, the Hankel norm is an L_2-induced norm of the Hankel operator:

$$\|G(s)\|_H = \sup_{u \in L^2(-\infty,0)} \frac{\|y(t)\big|_{t>0}\|_2}{\|u(t)\|_2},$$

where $y(t)\big|_{t>0}$ is the system's output after $t = 0$:

$$y(t)\big|_{t>0} = \begin{cases} 0, & t \leq 0 \\ y(t), & t > 0 \end{cases}.$$

this way, the Hankel norm gives the maximal L_2-gain from past inputs to future outputs of the system. Obviously, adding static gain to the system doesn't change neither the system's Hankel operator or Hankel singular values.

The Balanced Truncation reduction algorithm

The idea behind TBR is to perform a change of variables of the original system such that in the new coordinates both gramians P and Q are diagonal and equal to each other. Such change of variables is called *balancing transformation*. From the previous derivations it follows that in such representation $P = Q = \operatorname{diag}(\sigma_1, \ldots, \sigma_N)$, where σ_i denotes i^{th} largest Hankel singular value of the system[6].

Balancing transformation for asymptotically stable system $(A, B, C, 0)$ is guaranteed to exist if the system gramians P and Q are nonsingular (or equivalently the system is completely controllable and completely observable). Indeed, since Q is strictly positive-definite, we can perform Cholesky factorization of Q:

$$Q = RR^T,$$

and the product RPR^T is a symmetric positive-definite matrix, therefore we can

[6] In fact, for any two symmetric strictly positive-definite matrices A and B there exist such matrix T that both TBT^T and $T^{-T}AT^{-1}$ are diagonal. By scaling the columns of T we can always make both diagonals equal to each other. We are effectively showing this here.

perform an eigenvalue decomposition:

$$RPR^T = W\Sigma^2 W^T, \quad W^T W = I_{N\times N}.$$

Let's change system variables $\tilde{x} = Tx$, where

$$T = \Sigma^{-1/2} W^T R.$$

In the transformed coordinates [28]:

$$\begin{aligned}
\tilde{P} &= TPT^T = \Sigma^{-1/2} W^T RPR^T W \Sigma^{-1/2} = \Sigma, \\
\tilde{Q} &= T^{-T} Q T^{-1} = \Sigma^{1/2} W^T R^{-T} R^T R R^{-1} W \Sigma^{1/2} = \Sigma,
\end{aligned}$$

this way, matrix T is the balancing transformation.

The balancing transformation is not unique; in the case of all distinct Hankel singular values it is defined up to any state transformation $T = \text{diag}\{\pm 1, \cdots \pm 1\}$ [61].

From our previous derivations it should be clear that the states having small observability and controllability measures should not contribute much to the system's response. With this in mind, states corresponding to small Hankel singular values can be truncated.

Balanced truncation reduction algorithm effectively truncates all the states of the balancing transformation, which correspond to smallest $(N - q)$ Hankel singular values. One should keep in mind that in order to obtain the reduced system we need to calculate only the column spans of the first q rows of $W^T R$ and the first q columns of $R^{-1} W$, which are exactly the dominant eigenspaces of QP and PQ, respectively:

$$PQ = PR^T R = R^{-1}(RPR^T)R = (R^{-1}W)\Sigma^2 (R^{-1}W)^{-1}$$

The Balanced Truncation algorithm is outlined as Algorithm 1. It is important to note that this algorithm does not produce the system in the balanced form as the canonical TBR does (see Algorithm 3 in [44]). The scaling of columns of U and V in the Algorithm 1 is arbitrary, as long as $V^T U = I$, but this does not affect the transfer function of the reduced system. The Algorithm 1 produces a state-space equivalent of the reduced balanced model.

```
**Algorithm 1:** Balanced-truncation reduction algorithm (TBR)
**Input:** Initial system $(A, B, C, D)$, desired reduced order $q$
**Output:** Reduced-order system $(A^r, B^r, C^r, D)$
(1)   Find observability gramian $P$ from (2.14)
(2)   Find controllability gramian $Q$ from (2.11)
(3)   Compute $q$ left and right dominant eigenvectors of $PQ$:
      $(PQ)U = U\Sigma^2$, where $\Sigma^2 = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_n^2)$
      $V^T(PQ) = \Sigma^2 V^T$ and scale columns of $V$ such that
      $V^T U = I_{q \times q}$
(4)   Use $V$ and $U$ as projection matrices in (2.3):
      $A^r = V^T A U, \quad B^r = V^T B, \quad C^r = CU$
(5)   **return** $(A^r, B^r, C^r)$
```

The reduced system obtained by TBR algorithm is guaranteed to be stable, because the reduced gramian $\mathrm{diag}(\sigma_1, \ldots, \sigma_q)$ satisfies both controllability and observability Lyapunov equations of the reduced system ([28], Theorem 3.3). It is guaranteed to be asymptotically stable if $\sigma_q \neq \sigma_{q+1}$ [61]. It also satisfies the following H_∞ error bound [28, 22]:

$$\|G(s) - G^r(s)\|_\infty \equiv \sup_\omega \|G(j\omega) - G^r(j\omega)\|_2 \leq 2 \sum_{q+1}^{n} \sigma_i, \tag{2.17}$$

where $\sigma_i = \sqrt{\lambda(PQ)}$ are Hankel singular values of the original system, $G(s)$ and $G^r(s)$ refers to the transfer functions of the original and reduced system, respectively. Using this a-priori error bound, one can select the needed reduced order based on the target accuracy.

Below we summarize the benefits of TBR.

1. Guaranteed stability

2. Frequency domain a-priori error bound.

3. Order of the reduced system does not directly depend on the number of inputs

and outputs.

4. Is generally believed to be close to optimal in the H_∞ error metric.

5. Generalizations exists, which preserve various passivity notions.

The main drawbacks of the TBR are the following.

1. Not suitable for reduction of large models: requires storage of at least two dense matrices and $O(n^3)$ operations to solve the Lyapunov equations (2.14) and (2.11) and for eigenvalue decomposition of PQ.

2. Cannot take advantage of sparsity of A, since P and Q are generally dense.

Another feature of TBR which is worth mentioning is the fact that throughout our derivations we have assumed that all inputs (if the system has multiple inputs) are scaled similarly relative to one another. If some input signal is typically much less in magnitude than other input signals, this may render our controllability gramian inadequate. The same applies to the output signals. It is easy to take into account this feature by scaling rows of C and columns of B accordingly to the expected magnitudes of the input and output signals.

Cross-Gramian

If the system under consideration has the same number of inputs as the number of outputs $m = p$, it is possible to define the following matrix X called the *cross-gramian* as a solution of the following Sylvester equation [23]:

$$AX + XA + BC = 0, \tag{2.18}$$

which can also be expressed as

$$X = \int_0^\infty e^{At} B C e^{At} dt.$$

If, in addition, the transfer function of the system is *symmetric*, that is,

$$G(s) = G^T(s),\ \forall s,$$

then the following relation holds [76]:

$$X^2 = PQ,$$

which means that one can use only one gramian X, instead of P and Q in the Algorithm 1 for computation of the left and right projection matrices. The idea of approximating cross-gramian for purposes of large-scale model reduction was proposed in [76].

2.2.4 TBR algorithm for systems in the descriptor form.

Let's consider a system in the descriptor form (E, A, B, C, D) as in (1.8). Below we assume that the pair (E, A) is *regular*, that is, matrices E and A do not have a common kernel vector.

Balanced truncation model reduction has been generalized for systems in the descriptor form in [78, 79]. The major distinction between the descriptions in the form (A, B, C, D) and (E, A, B, C, D) is that the latter may not correspond to a proper transfer function (for example, the output signal may be a time-derivative of the input signal). In other words, such system may have poles at $s = \infty$, which can happen only if matrix E is singular.

In order to possess a finite H_∞ error bound (which is an essential property of TBR), the polynomial terms in (2.2) of the transfer function of the original system should be exactly preserved by the reduced system. The proper rational term $G^{p.r}(s)$, on the other hand, can be reduced by using TBR. The proper and polynomial parts of the transfer function can be separated by partitioning of the state space using projection of the system onto deflating subspaces of the pair (E, A) corresponding to infinite and finite eigenvalues, respectively (for details see [78], p. 18). As a result, the problem boils down to reducing the system (E, A, B, C, D), where the matrix E is nonsingular. With this in mind, we will assume below that the matrix E is nonsingular.

The observability and controllability gramians P and Q of a descriptor system with no poles at infinity are solutions to the following two *generalized Lyapunov equations* [78]:

$$APE^T + EPA^T + BB^T = 0, \tag{2.19}$$

and

$$A^TQE + E^TQA + C^TC = 0. \tag{2.20}$$

It is important to note that the controllability gramian P obtained in this way still has the same energy meaning of the quadratic form associated with the minimal energy required to reach a given state:

$$\|u(t)\|_2^2 = x_0^T P^{-1} x_0,$$

where the input $u(t)$ drives the system from $x = 0$ at $t = -\infty$ to $x = x_0$ at time $t = 0$ and has the minimal 2-norm among all such signals.

Note that the observability gramian Q for descriptor system does *not* have the meaning of a quadratic form associated with the output energy when the system is released from a given state. Instead, the matrix E^TQE has such a property[7]:

$$\|y(t)\|_2^2 = x_0^T E^T Q E x_0,$$

where the output signal $y(t)$ is observed after the system has been released from the state x_0, assuming zero input.

The Hankel singular values of the (proper) descriptor system are defined as eigenvalues of PE^TQE, and the reduced system can be found via projection using dominant eigenvectors of $QEPE^T$ and PE^TQE as the left and right projection matrices,

[7] In fact, MATLAB defines the observability gramian for descriptor systems exactly as E^TQE, where Q is solution of (2.20). This is a minor notation ambiguity, which is worth mentioning.

respectively.

The frequency error bound (2.17) holds for such generalization of the TBR. In fact, this reduction is mathematically equivalent to performing TBR on the system $(E^{-1}A, E^{-1}B, C, D)$, however the computation via generalized Lyapunov equations is better conditioned numerically.

2.2.5 Approximations to balanced truncation.

Almost all approximations to TBR employ substitution of low-rank approximations to P and Q in the Algorithm 1 [44, 45, 8].

There are many algorithms proposed for obtaining low-rank approximations to P and Q, for example by iteratively solving projected Lyapunov equations (Approximate Power Iteration) as in [35], or using ADI/Smith methods described in [60, 44, 31], or performing frequency-domain POD [40], or performing an approximate integration in (2.13) and (2.9).

As it was mentioned before, however, the dominant eigenspaces of P and Q completely depend on the choice of basis (i.e. choice of system representation); the states which are "mostly observable" in one basis may be "least observable" even after simple scaling of the variables. Therefore, such methods work reliably only for cases where observable states are controllable, for example for symmetric systems [44].

There exist several algorithms which try to fix the above mentioned problem. In [7], the low-rank approximation to the cross-gramian X is proposed. However, such method is directly applicable only to the systems with square symmetric transfer functions.

Another algorithm, called AISIAD, which attempts to tackle this problem is described in [90]. Since one of the contributions of this work is based on AISIAD method, it is described in detail in Chapter 5.

2.3 Non-projection based MOR methods

Non-projection methods do not employ construction of any projection matrices. The following are several most commonly used methods of this kind:

1. Hankel optimal model reduction,
2. Singular Perturbation approximation
3. Transfer function fitting methods

2.3.1 Hankel optimal model reduction

One of the most useful measures of how well one system approximates another system is an H_∞ norm of the error, i.e. the H_∞ norm of the error system on the figure 1-2:

$$\|G(s) - G^r(s)\|_\infty \equiv \sup_\omega \|G(j\omega) - G^r(j\omega)\|_2. \tag{2.21}$$

As it was mentioned, TBR method has a guaranteed bound for H_∞ error (2.17). However, the result of TBR reduction is not generally optimal in this error metric.

Unfortunately, there is no known polynomial-time algorithm which solves the problem of finding a reduced state-space model which strictly minimizes H_∞ norm of the error.

However, there exist an optimal algorithm, which exactly minimizes *Hankel norm* of the error system:

$$\|G(s) - G^r(s)\|_H \equiv \sigma_{\max}(G(s) - G^r(s)), \tag{2.22}$$

where $\sigma_{\max}$ denotes a maximal Hankel singular value of the system.

Strictly speaking, the Hankel norm is not a norm in the space of all finite-dimensional stable state-space systems of a given order, since it violates the following necessary property:

$$\|G(s)\| = 0 \;\Leftrightarrow\; G(s) \equiv 0,$$

which does not hold for the Hankel norm, because the Hankel norm of a static gain is zero. However, if we assume the feed-through term D of the system to be zero, then the Hankel norm will define a valid norm in this subspace. The following inequality holds for any stable transfer function $G(s)$ of order n with zero feed-through term [28]:

$$\|G(s)\|_H \leq \|G(s)\|_\infty \leq n\|G(s)\|_H,$$

which essentially indicates that these norms are *equivalent*. In fact, the equivalence of all norms in a finite-dimensional space is a widely known result.

All Hankel-optimal reduced models were characterized in [28] by providing an explicit computational algorithm. The a-priori H_∞ error bound for this reduction method is half the right-hand side of (2.17), i.e. half the bound on the TBR error. Approach for computing the Hankel optimal model uses balancing transformations. The overall complexity, therefore, is $O(n^3)$, as for the TBR. Comparison of this method to TBR (as well as some other methods) on practical examples are given in [4].

2.3.2 Singular Perturbation as alternative to projection

As it will be shown below, projection-based MOR methods can almost always be interpreted as performing a coordinate transformation of the original system's state space, followed by "truncation" of the system's states, effectively setting the last $(n-q)$ states to zero. As an alternative, one can instead set the derivatives of the states to be discarded to zero. This procedure is called *state residualization*, which is the same as a *singular perturbation approximation*.

Let's assume that we have constructed projection matrices U and V, by using any projection-based method. For simplicity, let's assume that the original system is in the form (A, B, C, D). If the product $V^T U$ is full rank (which is usually true for all projection MOR methods mentioned in this thesis), then we can always find such nonsingular matrix T that

$$\mathrm{colsp}(T_{1\ldots n,1\ldots q}) = \mathrm{colsp}(U), \quad \mathrm{colsp}((T^{-T})_{1\ldots n,1\ldots q}) = \mathrm{colsp}(V).$$

The reduced system (V^TAU, V^TB, CU, D) is equivalent to truncating the transformed system $(T^{-1}AT, T^{-1}B, CT, D)$, effectively setting $\tilde{x}_{q+1} \equiv 0, \dots \tilde{x}_n \equiv 0$ in the new coordinates. This implies that the original and reduced transfer functions are matched at the infinite frequency, $G^r(\infty) = G(\infty)$.

The singular perturbation approximation eliminates the states in a different way. Instead of setting the values of the variables to zero, it sets their derivatives to zero:

$$\dot{\tilde{x}}_{q+1} \equiv 0, \dots \dot{\tilde{x}}_n \equiv 0.$$

We have:

$$\begin{cases} \frac{d}{dt}\begin{bmatrix}\tilde{x}_1(t)\\ \tilde{x}_2(t)\end{bmatrix} = \underbrace{\begin{bmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\end{bmatrix}}_{T^{-1}AT}\begin{bmatrix}\tilde{x}_1(t)\\ \tilde{x}_2(t)\end{bmatrix} + \underbrace{\begin{bmatrix}B_1\\ B_2\end{bmatrix}}_{T^{-1}B}u(t) \\ y(t) = \underbrace{\begin{bmatrix}C_1 & C_2\end{bmatrix}}_{CT}\begin{bmatrix}\tilde{x}_1(t)\\ \tilde{x}_2(t)\end{bmatrix} + Du(t)\end{cases}, \qquad (2.23)$$

where we have partitioned the transformed system such that the states to be kept are in the vector $\tilde{x}_1 \in \mathbf{R}^q$. By setting $\dot{\tilde{x}}_2 = 0$ we have:

$$A_{21}\tilde{x}_1(t) + A_{22}\tilde{x}_2 + B_2u(t) \equiv 0,$$

and substituting x_2 in the equation (2.23) above, we obtain:

$$\begin{cases} \dot{\tilde{x}}_1(t) = \underbrace{(A_{11} - A_{12}A_{22}^{-1}A_{21})}_{A^r}\tilde{x}_1(t) + \underbrace{(B_1 - A_{12}A_{22}^{-1}B_2)}_{B^r}u(t) \\ y^r(t) = \underbrace{(C_1 - C_2A_{22}^{-1}A_{21})}_{C^r}\tilde{x}_1(t) + \underbrace{(D - C_2A_{22}^{-1}B_2)}_{D^r}u(t)\end{cases}.$$

Such obtained reduced system (A^r, B^r, C^r, D^r) no longer necessarily matches the frequency response of the original system at the infinite frequency ($D \neq D^r$). However, it matches it at $s = 0$:

$$G(0) = CA^{-1}B + D = C^r(A^r)^{-1}B^r + D^r = G^r(0),$$

and as a result the approximation near zero frequency should be better if singular perturbation is employed [8].

What should we expect if instead of truncating the states in the TBR algorithm (Algorithm 1) we would employ residualization? Basic properties of singular perturbation approximation for balanced systems were derived in [49]. In particular, it is shown that the resulting reduced system is stable and has the same error bound (2.17) as balanced truncation.

2.3.3 Transfer function fitting methods

In the area of computer-aided modeling of high-frequency interconnects, linear devices are usually characterized in the frequency domain. Variations in the dielectric permittivity, skin effect and other phenomena are best described, and most easily measured, in terms of frequency-dependent variables. In order to make time-domain simulations feasible, one can construct a state-space model that approximates the sampled transfer function of the system[8].

Such methods can be treated as model reduction methods, since we can always obtain a series of snapshots of the original transfer function and then use any of the transfer function fitting methods to obtain reduced model.

Methods based on rational fitting [16], vector fitting [33] and method based on quasi-convex optimization [77] fall into this category.

Rational and *vector* fitting methods are based on iterative application of linear least squares. Unknowns are systems poles and residues (for the vector fitting algorithm) or polynomial coefficients of numerator and denominator (rational fitting algorithm). Though there is no convergence proof for these methods, they usually work well in practice.

Quasi-convex optimization method [77] uses more rigorous techniques to obtain guaranteed stable models. It can be used to obtain parameterized models, which preserve additional properties such as *passivity*.

[8] Another way to treat this problem is by employing a direct convolution or recursive convolution methods [47].

All of these algorithms are currently limited to linear dynamical systems with either single input (SIMO) or single output (MISO), or both(SISO). Approximation to multiple input-multiple output (MIMO) dynamical system can be done by fitting each column of the transfer matrix followed by concatenation of inputs. Such obtained system will likely be reducible, therefore subsequent linear reduction (for example, balanced truncation) can be applied.

Chapter 3

Nonlinear model reduction methods

The derivative of a drunk party is money from selling empty bottles.
A party is called *nontrivial* *if its second derivative is nonzero.*

- Russian math students folklore.

Currently the vast majority of publications and known MOR methods are targeting linear dynamical models [8]. The methods for nonlinear model reduction are much less developed and are by far more challenging to develop and analyze.

The problem of nonlinear model reduction deals with approximations of the systems in the form of a nonlinear ODE (1.3). The nonlinear model reduction goal, broadly speaking, is to reduce costs of simulation of such systems. This involves not only reducing the dimensionality of the state vector x, but also finding ways to efficiently calculate the right-hand and left-hand side functions f and h. In addition to calculating the function values, some widely used ODE solvers (which employ for example trapezoid rule or backward-Euler integration methods) need to compute Jacobians of f and h (derivatives with respect to all components of state vector).

With this in mind, the problem of nonlinear model reduction consists of the following two sub-problems:

1. Reducing the dimensionality of the state vector.

2. Finding representations of the reduced nonlinear functions such that the values and derivatives can be computed efficiently.

In fact, addressing either one of these issues leads to computational gains. However, algorithms which address both of the issues are usually much more beneficial.

Up until now, the only practical developed nonlinear dimensionality reduction methods (i.e. methods which solve the first problem above) are based on projections [66, 71, 72]. As we did for the linear case, projection methods employ bi-orthogonal projection matrices $U, V \in \mathbb{R}^{n \times q}$ which can be obtained either from any linear MOR method applied to linearization of (1.3) or from sequence of snapshots from nonlinear simulation of some *training trajectory* [39, 5, 36, 54]. Sometimes aggregation of both bases in the single basis works the best.

Let's assume that we have nonlinear system in the form (1.2). As we did for the linear systems, we assume that state vector approximately evolves within some linear subspace spanned by columns of matrix U, therefore $x \approx Uz,\ z \in \mathbb{R}^q$. Projecting residual in (1.2) onto the rowspan of matrix V yields:

$$\begin{cases} \dot{z} = \underbrace{V^T f(Uz, u)}_{f^r(z,u)} \\ y^r = g(Uz, u) \end{cases} \tag{3.1}$$

Unfortunately, although the dimensionality of the state has been reduced, simulation of such system directly (in general) is still costly, because in order to evaluate values and the Jacobian of $f^r(z, u)$ we need to perform high-dimensional computation.

Here the second mentioned challenge of the nonlinear MOR arises, namely the *representation* problem [71, 64].

There are two main known solutions for this problem:

1. Performing Taylor series expansion of f and projecting the terms [15, 13, 14, 2], or

2. Using TPWL methods (or more generally, regression models based on function's snapshots) [71, 64, 85, 68].

The first solution is applicable to either quadratic (or polynomial) or weakly nonlinear systems. The second kind of algorithms is applicable for highly nonlinear systems, however tends to be less accurate if being simulated with inputs sufficiently different from training input.

3.1 Nonlinear reduction based on Taylor series.

Very first practical approaches to nonlinear model reduction were based on using Taylor series expansions of function f in [15, 66, 13, 14, 2, 37]. Let's assume that we have performed a Taylor series expansion of function f in the state-space model (1.2) around some nominal state x_0 and input u_0:

$$\dot{x} \approx f(x_0, u_0) + \left\{\frac{\partial f}{\partial x}\right\}(x - x_0) + \left\{\frac{\partial f}{\partial u}\right\}(u - u_0) + \frac{1}{2}\Bigg(\left\{\frac{\partial^2 f}{\partial x^2}\right\}(x - x_0) \otimes (x - x_0) + \\ + \left\{\frac{\partial^2 f}{\partial x \partial u}\right\}(x - x_0) \otimes (u - u_0) + \left\{\frac{\partial^2 f}{\partial u^2}\right\}(u - u_0) \otimes (u - u_0)\Bigg) + \ldots, \quad (3.2)$$

where all derivatives of f are taken at the expansion point (x_0, u_0).

We can assume that the Taylor series, truncated up to certain order, can approximate the original state-space model with sufficient accuracy.

Now we can employ projection strategy as we have described above. We represent $x \approx Uz,\ z \in \mathbb{R}^q$ and project the residual in (3.2) onto the rowspan of matrix V, assuming it is biorthogonal to U:

$$\dot{z} = \underbrace{V^T f(Uz_0, u_0)}_{f^r(z_0,u_0)} + \underbrace{V^T\left\{\frac{\partial f}{\partial x}\right\}U}_{\frac{\partial f^r}{\partial z}}(z - z_0) + \underbrace{V^T\left\{\frac{\partial f}{\partial u}\right\}}_{\frac{\partial f^r}{\partial u}}(u - u_0) + \\ + \frac{1}{2}\underbrace{\left(V^T\left\{\frac{\partial^2 f}{\partial x^2}\right\}U \otimes U\right)}_{\frac{\partial^2 f^r}{\partial z^2}}(z - z_0) \otimes (z - z_0) + \ldots. \quad (3.3)$$

This expansion is equivalent to the Taylor expansion of the function

$$f^r(z,u) \equiv V^T f(Uz,u)$$

with respect to reduced state z and input u up to the same order as in (3.2).

Using tensor manipulations, in theory, it is straightforward to obtain reduced models of any differentiable original system up to any order. Unfortunately, however, memory and computation requirements increase exponentially with increasing of the order of the Taylor expansion, making practical only expansions of low order (usually up to second or third).

The method has the following advantages and limitations:

- Employing Taylor series limits the applicability of the reduction to only *weakly nonlinear* dynamical systems. It is directly applicable to quadratic (or more generally, polynomial) systems.
- Quite frequently original system's Jacobian and higher-order derivatives are sparse. Projections in (3.3) destroys the sparsity of tensors. As a result, memory and computational costs impose severe constraints on the reduced order of the system, making large reduced models not practical.
- There is no global guarantee of stability of the reduced system. No error bounds are guaranteed, in general. Local stability can be established based on the linearization around equilibrium.
- There is little known about global stability of quadratic, as well as higher order, systems[1].

3.2 TPWL and regression-based methods

Another approach to deal with the representation complexity of the projected nonlinear function $f^r(z,u)$ in (3.1) is to approximate this function using snapshots of the

[1] Here by the term *global stability* we mean absence of diverging system trajectories

values and Jacobians of the original nonlinear function [68, 69, 71, 85].

For simplicity, as it was done in the original paper [68], let's assume that $f(x,u) \equiv f(x)+Bu$. In our later development we generalize the TPWL algorithm to the general (nonseparable) case.

Let's assume that we have obtained a set of "important" state vectors x_i, $i = 1 \ldots l$, near which the trajectories of the original system are most likely to be found.

We can approximate the original nonlinear function $f(x)$ near these points as the following:

$$\hat{f}(x) \approx \sum_{i=1}^{l} w_i(x)\Big(f(x_i) + \underbrace{\Big\{\frac{\partial f}{\partial x}\Big|_{x_i}\Big\}}_{J_f(x_i)}(x - x_i)\Big), \tag{3.4}$$

where $J_f\Big|_{x_i}$ is a Jacobian matrix of f evaluated at the state vector x_i. The weights w_i satisfy the following properties:

$$\sum_{i=1}^{l} w_i(x) \equiv 1, \qquad \lim_{x \to x_i} w_i(x) \to 1,\ \forall i.$$

This way, the approximation (3.4) is a convex combination of the linearizations of f.

The reduced model can be obtained using similar approximation as in (3.4) using projected snapshots of function values and projected Jacobians:

$$\hat{f}^r(x) = V^T f(Uz) \approx \sum_{i=1}^{l} w_i^r(z)\Big(V^T f(x_i) - V^T J_f x_i + V^T J_f(x_i) U z\Big), \tag{3.5}$$

which can be evaluated in time proportional to lq^2, assuming that the projected Jacobians and snapshots of f are computed off-line, during reduction step[2]. The reduced weights $w_i^r(z)$ are usually sought in the form similar to $w_i(x)$, and satisfy similar properties as unreduced weights:

$$\sum_{i=1}^{l} w_i^r(z) \equiv 1, \qquad \lim_{z \to V^T x_i} w_i^r(z) \to 1,\ \forall i,$$

[2] Computation of weights w_i^r is considered to be cheaper than this estimate, which is a typical case [71].

this way, points $V^T x_i$ are treated as projected linearization points. Note that (3.5) is not a projection of (3.4), because weighting functions $w_i^r(\cdot)$ are generally not projections of $w_i(\cdot)$.

The above described idea first appeared in [68] and further developed in [69, 70, 85, 56, 12]. In [64] the TPWL method was interpreted as performing a kernel-based approximation of $f^r(x)$.

The main benefit of the (nominal) TPWL approximation (3.4) over Taylor series-based approach is its applicability to strongly nonlinear systems, where the Taylor expansion would require too many terms to achieve needed accuracy.

The weaknesses of the (nominal) TPWL approximation (3.4) are the following:

- The expansion points x_i are usually collected my means of simulating a *training input*. Far away from this trajectory the approximation becomes poor.

- In general, no stability guarantees exist, and no error bounds are known.

- The method performs poorly if components of the state vector are scaled very differently. If the values of f are much more sensitive to some subset of coordinates of x, the weightings w_i should normally account for this.

- It is impossible, in general, to know how many snapshots are needed in order to achieve decent accuracy. The problems where the solution is a wave are especially difficult, because snapshots capture only certain waveform at a particular time, which makes any other waveform not represented correctly by TPWL approximation.

The error of the reduced representation (3.5), therefore, consists of two components:

- Error associated with the projection of nonlinear system

- Error associated with the TPWL approximation of the projected system.

Chapter 4

TBR-based trajectory piecewise-linear model reduction for nonlinear systems

Heavier-than-air flying machines are impossible.

- Lord Kelvin, president, Royal Society, 1895.

In this Chapter we develop further the TPWL framework described earlier and discuss how the choice of the linear reduction method affects the TPWL framework. Surprisingly, such analysis provided some insights into the fundamental properties of TBR linear reduction method, as well.

We consider a dynamical system in the following form:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = Cx(t) \end{cases}, \tag{4.1}$$

which is a slight simplification of (1.2).

As we did before, we assume nonlinear function f being differentiable for all values of x and u:

$$f(x,u) = f(x_0,u_0) + A(x - x_0) + B(u - u_0) + h.o.t., \tag{4.2}$$

where matrices A and B (which are dependent on the linearization point (x_0, u_0))

contain derivatives of f with respect to the components of the state and input signals respectively.

The TPWL approximation for such system can be generalized as the following:

$$f(x,u) \approx \sum_{i=1}^{l} \tilde{w}_i(x,u)\left(f(x_i,u_i) + A_i(x - x_i) + B_i(u - u_i)\right), \tag{4.3}$$

where x_i's and u_i's $(i = 1, \dots, l)$ are selected linearization points (samples of state and input values), A_i and B_i are derivatives of f with respect to x and u, evaluated at (x_i, u_i), and finally $\tilde{w}_i(x,u)$'s are state-and-input-dependent weights which satisfy:

$$\sum_{i=1}^{l} \tilde{w}_i(x,u) = 1 \quad \forall (x,u), \qquad \tilde{w}_i(x,u) \to 1 \text{ as } (x,u) \to (x_i,u_i). \tag{4.4}$$

Projecting the piecewise-linear approximation in (4.3) using biorthogonal projection matrices V and U yields the following reduced-order nonlinear dynamical system:

$$\begin{cases} \dot{z} = \gamma \cdot w(z,u) + (\sum_{i=1}^{l} w_i(z,u)A_{ir})z + (\sum_{i=1}^{l} w_i(z,u)B_{ir})u \\ y = C_r z \end{cases}, \tag{4.5}$$

where $z(t) \in R^q$ is the q-dimensional vector of states:

$$\gamma = \left[V^T(f(x_1,u_1) - A_1x_1 - B_1u_1) \quad \dots \quad V^T(f(x_l) - A_lx_l - B_lu_l) \right].$$

Here, $w(z,u) = [w_1(z,u) \dots w_l(z,u)]^T$ is a vector of weights, $A_{ir} = V^T A_i U$, $B_{ir} = V^T B_i$, and $C_r = CU$. One should note that $\sum_{i=1}^{l} w_i(z,u) = 1$ for all (z,u), $w_i \to 1$ as $(z,u) \to (V^T x_i, u)$, and that the evaluation of the right hand side of equation (4.5) requires at most $O(lq^2)$ operations, where l is the number of linearization points.

Linearization points (x_i, u_i) used in system (4.5) are usually selected from a 'training trajectory' of the initial nonlinear system, corresponding to some appropriately determined 'training input'. The choice of the training input is an important aspect of the reduction procedure, since this choice directly influences accuracy. As the general rule, the training signal should be as close as possible to the signals for which the

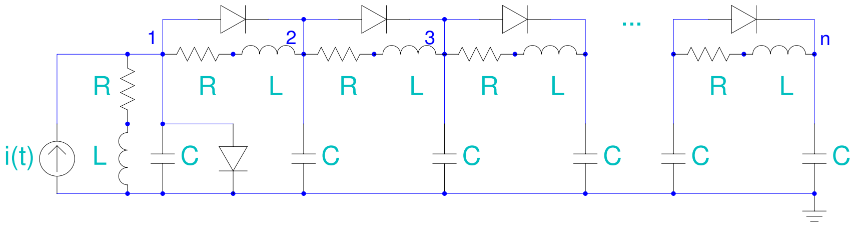

Figure 4-1: An example of a nonlinear transmission line RLC circuit model.

reduced system will be used. Additionally, this input signal should be rich enough to collect all “important” states in the set of linearization points (x_i, u_i) [81].

In the original papers [70] Krylov-subspace linear methods were solely used for TPWL reduced models. TBR reduction can be more accurate than Krylov-subspace reduction as it possesses a uniform frequency error bound [28], and TBR preserves the stability of the linearized model. This superior performance for the linear cases suggests that TPWL approximation models obtained using TBR are more likely to be stable and accurate. This is not necessarily the case, as will be shown below.

4.1 Examples of nonlinear systems

In this Section we consider three examples of nonlinear systems which arise in the modeling of MEMS devices that have nonlinear dynamical behaviors, which make good test cases for reduction algorithms.

4.1.1 Nonlinear transmission lines

The first two examples (the first one was also examined in [9] and [14]) refer to a nonlinear transmission line circuit model shown in Figure 4-1. The first circuit consists of resistors, capacitors, and diodes with a constitutive equation $i_d(v) = \exp(40v) - 1$. For simplicity we assume that all the resistors and capacitors have unit resistance and capacitance, respectively ($R = 1$, $C = 1$) (In this case we assume that $L = 0$). The input is the current source entering node 1: $u(t) = i(t)$ and the (single) output is chosen to be the voltage at node 1: $y(t) = v_1(t)$. Consequently, if the state vector is

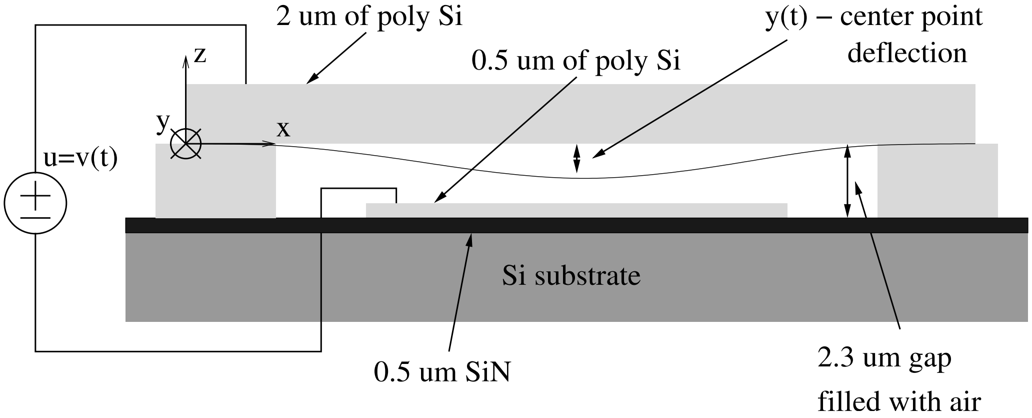

Figure 4-2: Microswitch example (following Hung et al. [36]).

taken as $x = [v_1, \ldots, v_N]$, where v_i is the voltage at node i, the system has symmetric Jacobians at any linearization point, and $B = C$. In this example we considered the number of nodes $N = 400$ and $N = 1500$. In the second example (cf. Figure 4-1) we also consider inductors (with inductance $L = 10$), connected in series with the resistors. We apply the RL formulation in order to obtain a dynamical system in form (4.1) with voltages and currents at subsequent nodes (or branches) of the circuit as state variables. In this case the Jacobians of f become nonsymmetric. The governing nonlinear system of equations which is being described by the form (4.1) is:

$$\begin{cases} \dot{v}_1 = (i + i_1 - i_2 - (e^{40v_1} - 1) - (e^{40(v_1 - v_2)} - 1))\frac{1}{C} \\ \dot{v}_2 = (i_2 - i_3 + (e^{40(v_1 - v_2)} - 1) - (e^{40(v_2 - v_3)} - 1))\frac{1}{C} \\ \ldots \\ \dot{v}_N = (i_n - (e^{40(v_{N-1} - v_N)} - 1))\frac{1}{C} \\ \dot{i}_1 = (-v_1 - i_1 R)\frac{1}{L} \\ \dot{i}_2 = (v_1 - v_2 - i_2 R)\frac{1}{L} \\ \ldots \\ \dot{i}_N = (v_{N-1} - v_N - i_N R)\frac{1}{L} \end{cases}$$

4.1.2 Micromachined switch

The third example is a fixed-fixed beam structure, which might be used as part of a microswitch or valve, shown in Figure 4-2. Following Hung et al. [36], the dynamical behavior of this coupled electro-mechanical-fluid system can be modeled with a 1D Euler's beam equation and the 2D Reynolds' squeeze film damping equation [36]:

$$\begin{cases} \hat{E}I\frac{\partial^4 w}{\partial x^4} - S\frac{\partial^2 w}{\partial x^2} = F_{elec} + \int_0^d (p - p_0)dy - \rho\frac{\partial^2 w}{\partial t^2} \\ \nabla \cdot ((1+6K)w^3 p \nabla p) = 12\mu\frac{\partial(pw)}{\partial t}. \end{cases} \tag{4.6}$$

Here, the axes x, y and z are as shown on figure 4-2, $\hat{E}$ is a Young's modulus, I is the moment of inertia of the beam, S is the stress coefficient, K is the Knudsen number, d is the width of the beam in the y direction, $w = w(x,t)$ is the height of the beam above the substrate, and $p(x,y,t)$ is the pressure distribution in the fluid below the beam. The electrostatic force is approximated assuming nearly parallel plates and is given by $F_{elec} = \frac{\epsilon_0 d v^2}{2w^2}$, where v is the applied voltage.

Spatial discretization of (4.6) described in detail in [71] uses a standard finite-difference scheme and leads to a nonlinear dynamical system in form of (4.1), with $N = 880$ states. After discretization, the state vector, x, consists of the concatenation of: heights of the beam above the substrate w, values of $\partial(w^3)/\partial t$, and values of the pressure below the beam. For the considered example, the output $y(t)$ was selected to be the deflection of the center of the beam from the equilibrium point (cf. Figure 4-2).

The remarkable feature of this example is that the system is strongly nonlinear, and no feasible Taylor expansion made at the initial state can correctly represent the nonlinear function f, especially in the so called *pull-in* region[1][36]. In addition, this example is illustrative in that it combines electrical actuation with the structural dynamics and is coupled to fluid compression. We expect model reduction methods that are effective for this example problem to be extendable to realistic micropumps and MEMS.

[1] If the beam is deflected by more than ≈1/3 of the initial gap, the beam will be pulled-in to the substrate.

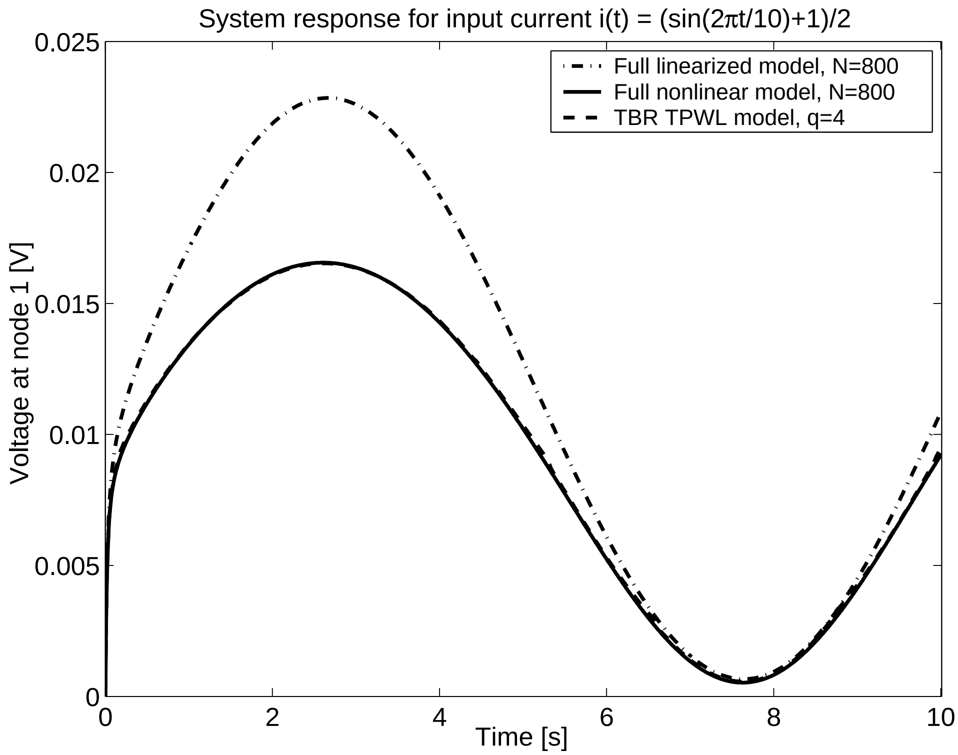

Figure 4-3: Comparison of system response (nonlinear transmission line RLC circuit) computed with nonlinear and linear full-order models, as well as TBR TPWL reduced order model (20 models of order $q = 4$) for the input current $i(t) = (\sin(2\pi t/10)+1)/2$. The TPWL model was generated using a unit step input current.

4.2 Computational results

In this Section results are presented for the models of transmission line and micromachined switch. The most challenging example was the micromachined switch.

4.2.1 Nonlinear transmission line models

First, we considered nonlinear transmission line RLC circuit model. The initial problem size n was equal to 800. The TBR reduced order $q = 4$ (using the linearized system at the initial state $x_0 = 0$). Figure 4-3 compares a transient computed with the obtained reduced order model (denoted as TBR TPWL model) with the transients obtained with full order nonlinear and linear models. One may note that TBR-based TPWL reduced model provides an excellent approximation of the transient for the initial system. It is also apparent that the model is substantially more accurate than a full order linear model of the transmission line.

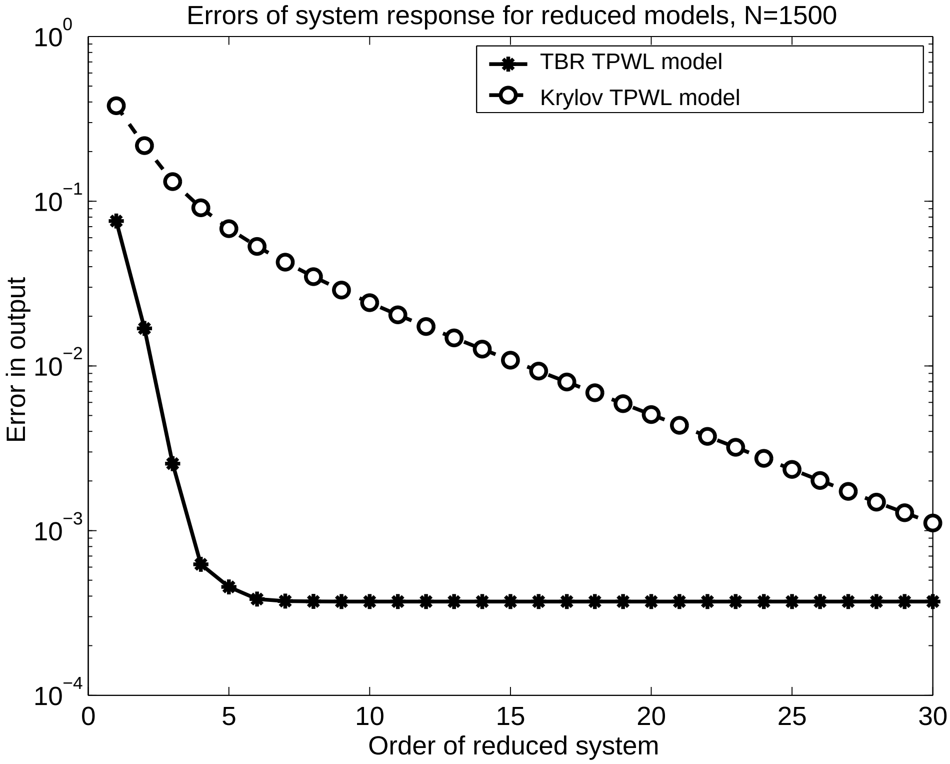

Figure 4-4: Errors in output computed with Krylov and TBR TPWL reduced order models (nonlinear transmission line RC circuit). Both training and testing inputs were unit step voltages. Initial order of system $n = 1500$. Note: solid and dashed lines almost overlap.

Similar results were obtained for the nonlinear RC circuit example. Figure 4-4 shows the error in the output signal $\|y_r - y\|_2$, where y_r is the output signal computed with TBR TPWL reduced order model, and y is computed with full order nonlinear model, for different orders q of the reduced model (in this example $\|y\|_2 = 0.44$). Analogous errors were also computed for reduced order TPWL models obtained with pure Krylov-based reduction. The results on the graph show that TBR TPWL models are significantly more accurate than the Krylov TPWL models of the same size. Also (limited by the quality of TPWL approximation to f) the TBR TPWL model achieves its best accuracy at a much lower order than the TPWL model based on Krylov subspace reduction.

It follows from Fig. 4-4 that the total error of TPWL reduced order approximation of a full nonlinear model consists of two components: the error due to projection procedure and the error associated with piecewise-linear approximation of nonlinear function f. The first component is dominant when the order of the reduced model is small. One may note that for TBR approach this error component becomes negligible as soon as the order of the reduced model is greater than 4. Further considerations on error estimation in TPWL models may be found in [70].

4.2.2 Micromachined switch example

The TBR TPWL model order reduction strategy was applied to generate macromodels for the micromachined switch example described in Section 4.1. The projection matrices were obtained using the linearized model of system (4.1) only at the initial state, and the initial state was included in the projections V and U.

Surprisingly, unlike in the nonlinear circuit examples, the output error did not decrease monotonically as the order q of the reduced system grew. Instead, macromodels with odd orders behaved very differently than macromodels with even orders. Models of even orders were substantially more accurate than models of the same order generated by Krylov reduction – cf. Figure 4-5. However, if q was odd, inaccurate and unstable reduced order models were obtained. This phenomenon is reflected in the error plot shown in Figure 4-5. Figure 4-6 illustrates that a fourth-order (even) reduced model accurately reproduces transient behavior.

This 'even-odd' phenomenon was observed in [55] and explained in the very general sense in [83]. The main result of [83] is described in Section 4.3. However, there is also an insightful but less general way of looking at this effect.

The 'even-odd' phenomenon can be viewed by examining eigenvalues of the reduced order Jacobians from different linearization points. For the switch example, the initial nonlinear system is stable and Jacobians of f at all linearization points are also stable. Nevertheless, in this example the generated reduced order basis corresponds to the balancing transformation only for the linearized system from the initial state x_0. Therefore, only the reduced Jacobian from x_0 is guaranteed to be stable. Other Jacobians, reduced with the same projection matrices, may develop eigenvalues with

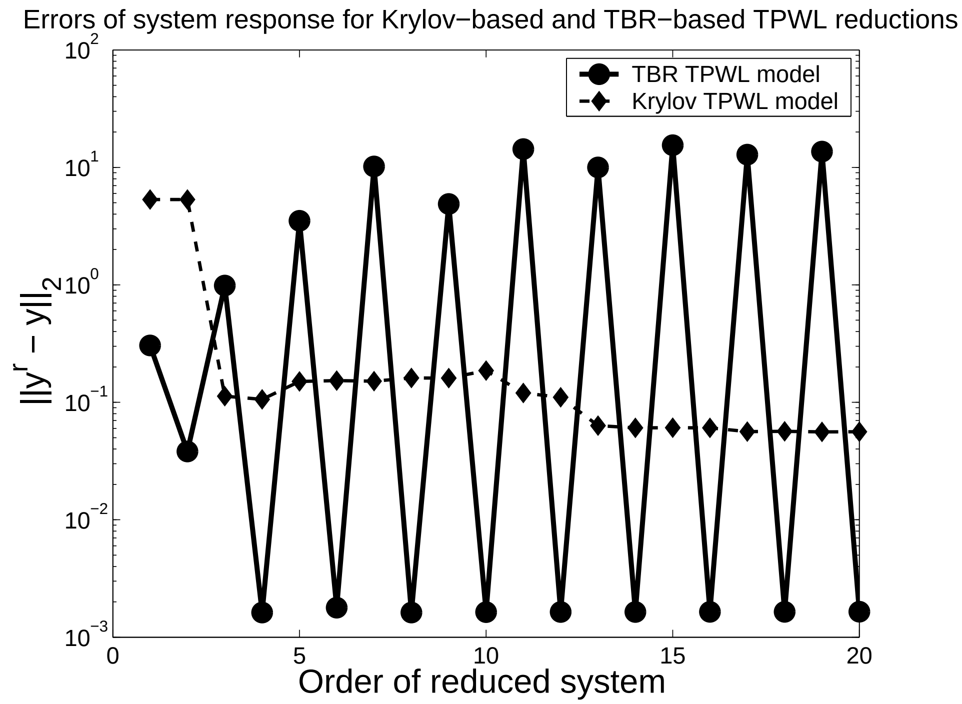

Figure 4-5: Errors in output computed by TPWL models generated with different MOR procedures (micromachined switch example); $n = 880$; 5.5-volt step testing and training input voltage.

positive real parts.

Figure 4-7 shows spectra of the reduced order Jacobians for models of order $q = 7$ and $q = 8$. One may note that, for $q = 8$, the spectra of the Jacobians from a few first linearization points are very similar. They also follow the same pattern: two of the eigenvalues are real, and the rest form complex-conjugate pairs. Increasing or decreasing the order of the model by 2 creates or eliminates a complex-conjugate pair of stable eigenvalues from the spectra of the Jacobians. If the order of the model is increased or decreased by 1 (cf. Figure 4-7 (left)), the situation is very different. A complex-conjugate pair will be broken, and a real eigenvalue will form. At the first linearization point this eigenvalue is a relatively small negative number. At the next linearization point, the corresponding eigenvalue shifts significantly to the right half-plane to form an unstable mode of the system. An obvious workaround for this problem in the considered example is to generate models of even order. Nevertheless, a true solution to this problem would involve investigating how perturbations in the

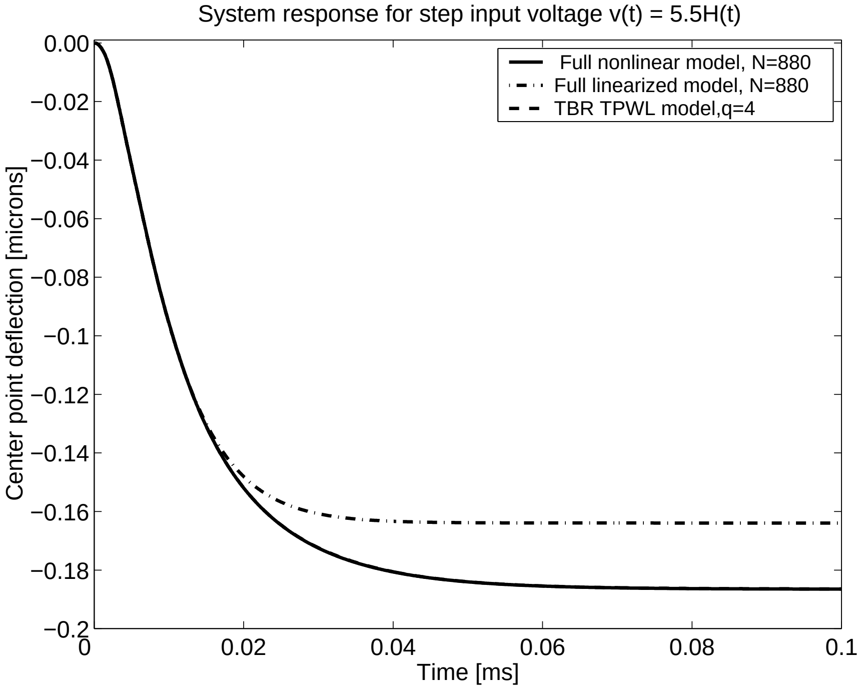

Figure 4-6: Comparison of system response (micromachined switch example) computed with both nonlinear and linear full-order models, and TBR TPWL reduced order model (7 models of order $q = 4$); 5.5-volt step testing and training input voltage. Note: solid and dashed lines almost fully overlap.

model affect the balanced reduction, and this is examined in Section 4.3.

4.3 Perturbation analysis of TBR reduction algorithm

For the micromachined switch example, the even-odd behavior exhibited by the TBR-TPWL model reduction can be investigated using perturbation analysis. Assume the projection bases V and U are computed using TBR reduction from a single linearization point. The key issue is whether or not the TBR basis obtained at one linearization point is still suitable for reducing piecewise-linear models further along the trajectory. To understand this issue, consider two linearizations of the nonlinear system (4.1) (A_0, B, C) (initial) and (A, B, C) (perturbed). Suppose TBR reduction is performed for both of these models, resulting in projection bases V, U and $\tilde{V}, \tilde{U}$

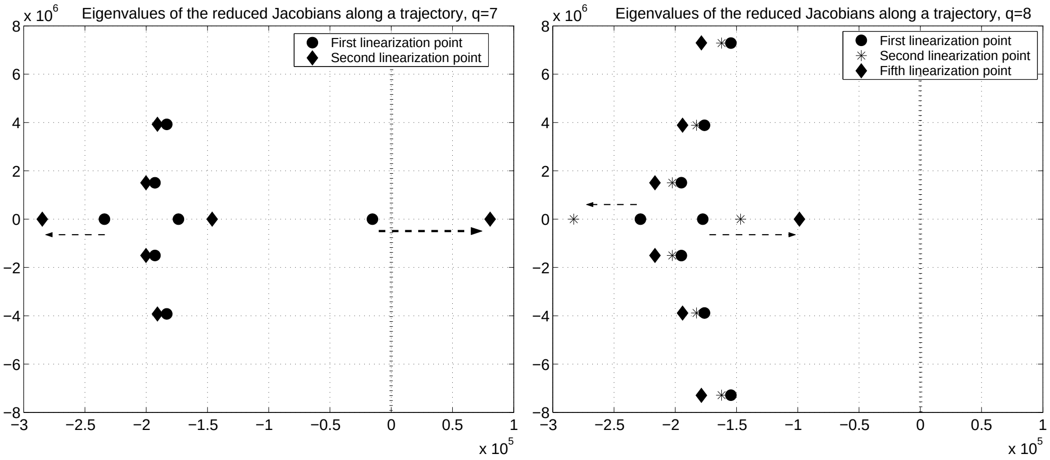

Figure 4-7: Eigenvalues of the Jacobians from the first few linearization points (micromachined switch example, Krylov-TBR TPWL reduction). Order of the reduced system $q = 7$ (left), $q = 8$ (right).

respectively. If these two bases are not significantly different, then perhaps V and U can be used to reduce the perturbed system, as is done for TPWL macromodels. This is true given some care, as will be made clear below.

4.3.1 Effect of Perturbation on Gramians

Consider the case for the controllability gramian P only, the results are valid for Q as well. Let $A = A_0 + A_\delta$, $P = P_0 + P_\delta$, where P_0 is an unperturbed gramian corresponding to unperturbed matrix A_0, and A_δ is relatively small so that P_δ is also small.

Using the perturbed values of A and P in the Lyapunov equation and neglecting $P_\delta A_\delta$ yields

$$A_0 P_\delta + P_\delta A_0^T + (A_\delta P_0 + P_0 (A_\delta)^T) = 0. \tag{4.7}$$

Note that (4.7) is a Lyapunov equation with the same matrix A_0 as for unperturbed system. This equation has a unique solution, assuming that the initial system is stable. The solution to (4.7) can be expressed using the following integral formula:

$$P_\delta = \int_0^\infty e^{A_0^T t} (A_\delta P_0 + P_0 (A_\delta)^T) e^{A_0 t} dt. \tag{4.8}$$

Assuming A is diagonalizable, P_δ can be bounded as

$$||P_\delta|| \leq 2(cond(T))^2||A_\delta||||P_0|| \int_0^\infty e^{2\mathfrak{Re}(\lambda_{\max}(A_0))t}dt, \tag{4.9}$$

where T is the matrix which diagonalizes A.

Since A is stable, the integral in (4.9) exists and yields an upper bound on infinitesimal perturbations of the gramian:

$$||P_\delta|| \leq \frac{1}{|\mathfrak{Re}(\lambda_{max}(A_0))|}(cond(T))^2||P_0||||A_\delta||. \tag{4.10}$$

Equation (4.10) shows that the bound on the norm of δP increases as the maximal eigenvalue of A_0 approaches the imaginary axis. In addition, note that perturbations in A will result in small perturbations in the gramian P as long as the system remains "stable enough", i.e. its eigenvalues are bounded away from the imaginary axis.

4.3.2 Effect of perturbations on the balancing transformation

As we have described in the Section 2.2.3, the balancing transformation in the TBR algorithm can be viewed essentially as a symmetric eigenvalue problem [28]:

$$RPR^T = W \underbrace{\text{diag}(\sigma_1^2, \ldots, \sigma_N^2)}_{\Sigma^2} W^T, \qquad T = \Sigma^{-1/2}W^TR, \tag{4.11}$$

where $R^TR = Q$ (R is a Cholesky factor of Q) and T is the coordinate transformation which diagonalizes both gramians. In the algorithm 1, the left projection matrix V consists of the first q columns of T^T, and the right projection matrix U consists of the first q columns of T^{-1}.

Applying the same perturbation analysis to the Cholesky factors, it can be shown that the perturbations in the Cholesky factors due to the perturbations in the original gramian are also small, provided that the system remains "observable enough", that is the eigenvalues of Q are bounded away from zero. Therefore we can state that the perturbation properties of the TBR algorithm are dictated by the symmetric eigenvalue problem $RPR^T = W\Sigma^2 W^T$.

The perturbation theory for the eigenvalue problem has been developed quite thoroughly [43], and one of the first observations is that small perturbations of a symmetric matrix can lead to large changes in the eigenvectors, if there are subsets of eigenvalues in the initial matrix which are very near to each other.

Below we summarize a perturbation theory for a symmetric eigenvalue problem with a nondegenerate spectrum.

Consider a symmetric matrix $M = M_0 + \delta M$, where M_0 is the unperturbed matrix with known eigenvalues and eigenvectors, and no repeated eigenvalues. Eigenvectors of M can be represented as a linear combination of eigenvectors of M_0:

$$x_k = \sum_{i=1}^{N} c_i^k x_i^0,$$

where x_k is the k-th eigenvector of the perturbed matrix M and x_i^0 is the i-th eigenvector of the unperturbed matrix. Coefficients c_i^k show how the eigenvectors of matrix M_0 are intermixed due to the perturbation δM, as in

$$(M_0 + \delta M)\sum_{i=1}^{N} c_i^k x_i^0 = \lambda_k \sum_{i=1}^{N} c_i^k x_i^0 \quad \Rightarrow \quad \sum_{i=1}^{N} c_i^k \delta M_{ji} = (\lambda_k - \lambda_j^0) c_j^k$$

where λ_k and λ_k^0 are the k-th eigenvalues of M and M_0 respectively and $\delta M_{ij} = (x_i^0)^T \delta M x_j^0$ is a matrix element of the perturbation in the basis of the unperturbed eigenvectors.

Now assume small perturbations and represent $\lambda_k = \lambda_k^0 + \lambda_k^{(1)} + \lambda_k^{(2)} + ...$ and $c_k^n = \delta_{kn} + c_k^{n(1)} + c_k^{n(2)} ...$ where each subsequent term represents smaller orders in magnitude. The first-order terms are:

$$\lambda_k^{(1)} - \lambda_k^0 = \delta M_{jj} \tag{4.12}$$

and

$$c_k^n = \frac{\delta M_{kn}}{\lambda_n^0 - \lambda_k^0}, k \neq n. \tag{4.13}$$

Equation (4.13) implies that the greater the separation between eigenmodes, the less they tend to intermix due to small perturbations. If a pair of modes have eigenvalues which are close, they change rapidly with perturbation. The following recipe for choosing an order of projection basis exploits this observation.

4.3.3 Recipe for using TBR with TPWL

Pick a reduced order to ensure that the remaining Hankel singular values are small enough and the last kept and first removed Hankel singular values are well separated.

The above recipe yields a revised TBR-based TPWL algorithm:

TBR-based TPWL with the linearization at the initial state

1. Perform the TBR linear reduction at the initial state x_0. Add x_0 to the projection matrices U and V by using biorthogonalization.

2. Choose the reduced order q such that the truncated Hankel singular values are:

   - Small enough to provide sufficient accuracy

   - separated enough from the Hankel singular values that are kept

3. Simulate the training trajectory and collect linearizations

4. Reduce linearizations using the projection matrices obtained in step 1.

4.3.4 Even-odd behavior explained

The perturbation analysis suggests that the sensitivity of TBR projection basis is strongly dependent on the separation of the corresponding Hankel singular values. The Hankel singular values for the linearization point of the micromachined switch example are shown in Figure 4-8.

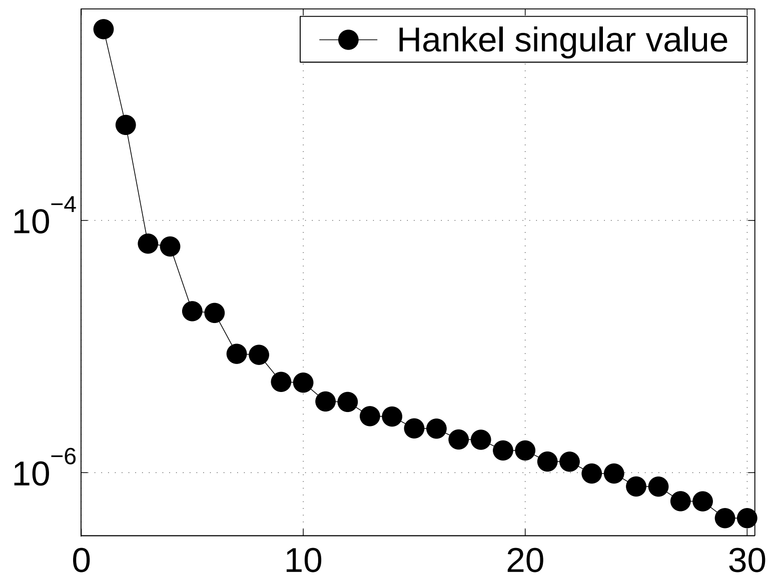

Figure 4-8: Hankel singular values of the balancing transformation at the initial state, Micromachined switch example.

As one can clearly see, the Hankel singular values for the microswitch example are arranged in pairs of values, and evidently, even-order models violates the recipe for choice of reduction basis.

Chapter 5

Modified AISIAD model reduction for LTI systems

Different groups of MOR researchers cannot understand each other without fighting

- Alex Megretski, 6.242 lecture, MIT 2004

In this Chapter we develop a linear model reduction algorithm for systems in the form (E, A, B, C, D), where matrix E is nonsingular. It is an approximation to TBR reduction.

5.1 Background and prior work

As we have already mentioned in Chapter 2, the majority of approximations to TBR, as well as Krylov-subspace methods, effectively approximate dominant eigenvectors of the controllability gramian P and/or observability gramian Q [44, 3, 31]. However, for the TBR we ultimately need approximations of the eigenvectors of products PQ and QP. The question arises as to whether a good approximations to P and Q leads to good approximations of the dominant eigenvectors of products PQ and QP. As we will show below, the answer to this question is "not necessarily".

If the dominant eigenspaces of system gramians P and Q are the same (for example, if gramians are equal), the reduction algorithms based on separate gramian approximations will provide a good approximation to TBR models. However, when eigenspaces of P and Q are different, approximation of dominant eigenspaces of PQ and QP in the Algorithm 1 can be poor when using low-rank approximations of P and Q. This issue was raised in [90, 76].

The work [90] was the first successful attempt of approximating the dominant eigenspaces of the products of gramians for generic MIMO systems. This work is the basis of the proposed algorithm.

5.1.1 Motivating example: simple RLC line

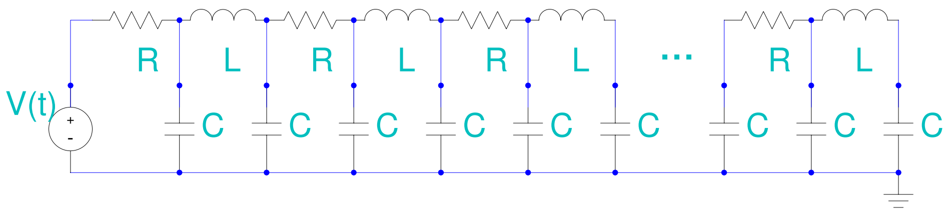

Figure 5-1: RLC transmission line model as a motivating example.

The fact that utilizing low-rank approximations of P and Q in the Algorithm 1 may not be sufficient to approximate the TBR reduction becomes apparent if one performs a modified nodal analysis (MNA) of the simple RLC transmission line depicted on figure 5-1 and then considers a very lightly damped case. In the MNA formulation, the state space consists of the voltages on the capacitors and currents through inductors of a circuit. Let the input to the system be the voltage applied to the first node, and the output be the current through the first resistor in the chain. The MNA analysis results in the descriptor system (E, A, B, C) of order $n = 2N$ with the positive semidefinite matrices E and $(-A)$ and $C = B^T$. We used a lightly damped line, with the parameters $R = 0.05, L = 10^{-10}, C = 10^{-15}$, the number of inductors $N = 100$. If we convert this system to the state-space model (A, B, C), the first N dominant eigenvectors of P and first N dominant eigenvectors of Q span almost completely orthogonal subspaces! This gives an approximation of PQ being almost zero.

This means that in order to get a good approximation of a product PQ one needs to get a low-rank approximations of P and Q essentially greater than N, and is consequently not applicable in a large-scale setting.

This example illustrates a fundamental problem: capturing dominant controllable and dominant observable modes separately is not sufficient to get a good approximation to TBR, and can lead to arbitrarily large errors in the frequency domain. Even the PRIMA algorithm [58], which guarantees passivity of the reduced model[1] produces quite poor approximations in the lightly damped cases in the H_∞ norm, due to the fact that it approximates only dominant controllable states[2] (see Section 5.7 for numerical results).

The method below is different in the sense that it directly approximates the product of PQ and therefore takes into account the fact that the separately determined most controllable and most observable states may be different than the states with the highest "controllability times observability" measure.

5.1.2 Original AISIAD algorithm

It is known that for the projection-based methods the transfer function of the reduced system depends only on the column spans of the projection matrices V and U (see [44], p. 23), therefore for the approximation of the TBR we need to approximate the dominant eigenspaces of PQ and QP.

The AISIAD algorithm approximates the dominant eigenspaces of the products PQ and QP using a power method, and then constructs projection matrices using these approximations.

The AISIAD algorithm was originally proposed in [90], and we present it here as Algorithm 2. It does not use low-rank approximations of gramians at all, however as we show below, it is highly desirable to use low-rank approximations of P and Q in order to produce accurate reduced models.

[1] For considerations on passivity enforcement read further sections

[2] Here we refer to the PRIMA algorithm which incorporates controllability approximation. There exist flavors of PRIMA which incorporate observability. However, this does not change our point that both of approximations are being accounted for independently of each other.

```
Algorithm 2: Original AISIAD algorithm
Input: System matrices $(A, B, C)$, reduced order $q$, initial orthogonal basis $V \in \mathbb{R}^{n\times q}$
Output: Order-$q$ reduced model $(A^r, B^r, C^r)$.
(1)  repeat
(2)      Approximate $X_i \approx PV_i$ by solving
         $AX_i + X_iH_i^T + \hat{M}_i = 0$, where
         $H_i = V_i^TAV_i, \quad \hat{M}_i = BB^TV_i$
(3)      Obtain orthogonal basis which spans the same subspace as
         $X_i$: $[U_i, S_i] = qr(X_i, 0)$
(4)      Approximate $Y_i \approx QU_i$ by solving
         $A^TY_i + Y_iF_i + \hat{N}_i = 0$, where
         $F_i = U_i^TAU_i, \quad \hat{N}_i = C^TCU_i$
(5)      Obtain orthogonal basis for the approximation of $QU_i$ and
         make it the next approximation of $V$: $[V_{i+1}, R_{i+1}] = qr(Y_i, 0)$
(6)  until convergence
(7)  Biorthogonalize the matrices $V_{i+1}$ and $U_i$:

         $$V_L \leftarrow V_{i+1}, \quad U_R \leftarrow U_i$$
         $$\begin{bmatrix} U_l & \Sigma & V_l \end{bmatrix} = svd(V_L^TU_R)$$
         $$V = V_LU_l\Sigma^{-1/2}, \quad U = U_RV_l\Sigma^{-1/2}$$

     Project the initial system using $V$ and $U$:

         $$A^r = V^TAU, \quad B^r = V^TB, \quad C^r = CU$$

(8)  return $(A^r, B^r, C^r)$
```

Consider the steps 2 and 4 of the Algorithm 2 in more detail. We present derivations for approximation of PV_i here, the derivations for QU_i are similar. From Lyapunov equation for P:

$$AP + PA^T + BB^T = 0 \tag{5.1}$$

Multiplying from the right-hand side by V_i, we get the following equation:

$$A\underbrace{PV_i}_{X_i} + \underbrace{PV_i}_{X_i}\underbrace{V_i^TAV_i}_{H_i} + \underbrace{P(I - V_iV_i^T)A^TV_i + BB^TV_i}_{M_i} = 0 \tag{5.2}$$

As we see, in the original AISIAD algorithm the term $P(I - V_iV_i^T)A^TV_i$ is neglected. We suggest that neglecting this term can result in a poor approximation quality, and this term can be instead approximated, using a low-rank approximant for the gramian.

5.1.3 Solution of a specialized Sylvester equation

The most important routine in the algorithm 2 is obtaining a solution of the Sylvester equation

$$AX + XH + M = 0, \quad A \in \mathbb{R}^{n\times n}, H \in \mathbb{R}^{q\times q}, M \in \mathbb{R}^{n\times q} \tag{5.3}$$

where $q \ll n$ (hence the name "specialized").

Original solver of Sylvester equation

Consider the following matrix:

$$S = \begin{bmatrix} A & M \\ 0 & -H \end{bmatrix} \tag{5.4}$$

and assume that we have found the matrices $V_1 \in \mathbb{R}^{n\times q}, Z \in \mathbb{R}^{q\times q}$ and nonsingular $V_2 \in \mathbb{R}^{q\times q}$ such that

$$\begin{bmatrix} A & M \\ 0 & -H \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} Z \tag{5.5}$$

Then one can clearly see that the matrix $V_1 V_2^{-1}$ satisfies (5.3). Moreover, for the purposes of the algorithm 2 it is sufficient to find only V_1, since we make use only of the column span of X.

Evidently, $\Lambda(S) = \Lambda(A) \cup \Lambda(-H)$, and since $-HV_2 = V_2 Z$, matrix V_2 is nonsingular if and only if $\Lambda(Z) = \Lambda(-H)$. Note that the matrices A and $(-H)$ should not have common eigenvalues, otherwise (5.3) does not have a unique solution.

In the original AISIAD the use of Implicitly Restarted Arnoldi (IRA) method [75] is proposed as the means of solving (5.5). This way, one can obtain a partial Schur decomposition of S with upper-triangular matrix Z in (5.5). In order to impose the condition $\Lambda(Z) = \Lambda(-H)$ using IRA, authors [75] needed to restrict H to be a Hurwitz matrix. Therefore, original algorithm is applicable only to the cases where

H is Hurwitz (i.e. the initial matrix A satisfies the condition of $V_i^T A V_i$ being Hurwitz for all choices of orthogonal basis V_i). This imposes a severe constraint on matrix A and narrows the class of applicable systems for the whole original method.

5.2 Algorithm for specialized Sylvester equation

It is possible to solve equation (5.3) in the following way.

Let's consider a complex Schur decomposition of matrix $H = WSW^*$, where S is upper-triangular, and W is unitary. Since matrix H is small $q \times q$, this Schur decomposition is inexpensive. Multiplying (5.3) from right by W yields:

$$A(XW) + (XW)S + MW = 0, \tag{5.6}$$

Backsolving for each column of the matrix ($\tilde{X} = XW$) starting from the first one:

$$(A + s_{jj} I_n)\tilde{x}_j = (MW)_j - \sum_{i=1}^{j-1} s_{ij}\tilde{x}_i, \tag{5.7}$$

Here $\tilde{x}_j$ denotes j^{th} column of the matrix XW, and s_{ij} denotes (i, j)-th element of the matrix S. The speed of these q solutions of linear system of equations determines the overall speed of the proposed algorithm. We can employ a sparse solver if matrix A is sparse. Alternatively, if fast matrix-vector products can be computed, one can employ an iterative Krylov-subspace solver such as GMRES in order to solve (5.7).

After the matrix $\tilde{X} = XW$ has been computed, the solution X can be recovered using multiplication by W^* from the right.

We summarize our algorithm for solving (5.3) in algorithm 3.

It is evident that for single-input single-output (SISO) system the proposed approximation is equivalent to rational Krylov method [30], for the interpolation points being $(-\Lambda(H))$ (assuming H being diagonalizable). If one performs projection of the initial system onto dominant eigenspaces of these approximations of P and Q, such obtained reduced model will match the initial model at $s_{1\ldots q} = -\Lambda(V_i^T A V_i)$. This im-

```
Algorithm 3: Solving generalized sylvester equation
Input: Matrices $A, H$ and $M$
Output: Solution $X$
Perform a complex Schur decomposition of $H$:
$H = WSW^*$
$\tilde{M} \leftarrow MW$
for j=1 to q
    Solve for $\tilde{x}_j$:
    $(A + s_{jj}I_n)\tilde{x}_j = \tilde{M}_j - \sum_{i=1}^{j-1} s_{ij}\tilde{x}_i$
    Assign $j^{th}$ column of $\tilde{X}$ being $\tilde{x}_j$.
return $X = \tilde{X}W^*$
```

portant fact unifying Krylov-subspace model reduction and low-rank approximation of gramians was first noted in [25].

5.2.1 Comparison of the two Sylvester solvers

For Hurwitz H both methods are equivalent assuming exact arithmetic. The method described in the section 5.1.3 ensures that matrix $\begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$ contains orthonormal set of vectors. However, this fact does not impose any constraint on the conditioning of V_1. Matrix V_1 can have a very small condition number, whereas columns of V_1 may be almost linearly dependent. On contrary, the proposed method employs only orthogonal transformations, therefore it is more numerically favorable.

In addition, the proposed method eliminates the above mentioned important applicability constraint. It can be applied to any solvable Sylvester equation, broadening the set of applicable problems.

5.3 Employing low-rank gramian approximations

As another important modification, we do not discard terms $P(I - V_iV_i^T)A^TV_i$ in (5.2). We can use any well-developed method to obtain low-rank approximations of P and Q, such as Low-rank ADI [44] or LR-Smith algorithms [3], or projection-based methods [76]. In our code we use simple projection-based algorithm outlined in Algorithm 4 for the example of controllability gramian approximation.

Algorithm 4: Low-rank approximation of gramians
Input: Matrices A and B, desired order of approximation k
Output: Low-rank approximation of $P \approx V_p S_p V_p^T$

```
Compute orthogonal basis for the Krylov subspace as an initial
guess:
colspan($V_0$) = $\mathcal{K}_k(A^{-1}, B)$
repeat
    Approximate $X_i \approx PV_i$ by solving
    $AX_i + X_i H_i^T + \hat{M}_i = 0$, where
    $H_i = V_i^T A V_i, \quad \hat{M}_i = BB^T V_i$
    Perform SVD of $X$:
    $[V_{i+1}, S_{i+1}, G_{i+1}] = svd(X, 0)$
until convergence
return $V_{i+1}, S_{i+1}$
```

5.4 The modified AISIAD algorithm

Combining two improvements outlined above, we now describe the modified AISIAD method which we propose as a replacement for the algorithm 2. We outline it as the Algorithm 5

We would like to note that if full exact gramians are known, the modified AISIAD algorithm becomes the power method for the matrices PQ and QP and therefore is guaranteed to converge to the exact TBR solution.

5.5 Modified AISIAD algorithm for descriptor systems

We have generalized the modified AISIAD (Algorithm 5) for the systems in the descriptor form with nonsingular matrix E. As we have discussed in Section 2.2.4, the treatment of the case of singular matrix E boils down to the case of nonsingular descriptor matrix.

From our discussion in Section 2.2.4, the projection matrices V and U in TBR for descriptor systems span the dominant eigenspaces of $QEPE^T$ and PE^TQE respectively, where P and Q are solutions of the generalized Lyapunov equations (2.19,

Algorithm 5: Proposed algorithm for approximation of TBR.
Input: System matrices (A, B, C), desired reduced order q, initial projection matrix $V \in \mathbb{R}^{n\times q}$
Output: Order-q reduced model (A^r, B^r, C^r).

```
(1) Get low-rank approximations of gramians
    $\tilde{P} \approx P$ and $\tilde{Q} \approx Q$ using any applicable algorithm.
(2) repeat
(3)     Solve using algorithm 3
        $AX_i + X_i H_i^T + \hat{M}_i = 0$, where
        $H_i = V_i^T A V_i$
        $\hat{M}_i = BB^T V_i + \tilde{P}(I - V_i V_i^T) A^T V_i$
(4)     $[U_i, S_i] = qr(X_i, 0)$
(5)     Solve using algorithm 3
        $A^T Y_i + Y_i F_i + \hat{N}_i = 0$, where
        $F_i = U_i^T A U_i$
        $\hat{N}_i = C^T C U_i + \tilde{Q}(I - U_i U_i^T) A U_i$
(6)     $[V_{i+1}, R_{i+1}] = qr(Y_i, 0)$
(7) until convergence
(8) Biorthogonalize matrices $V_{i+1}$ and $U_i$:
        $V_L \leftarrow V_{i+1}, \quad U_R \leftarrow U_i$
        $\begin{bmatrix} U_l & \Sigma & V_l \end{bmatrix} = svd(V_L^T U_R)$
        $V = V_L U_l \Sigma^{-1/2}, \quad U = U_R V_l \Sigma^{-1/2}$
    Project the initial system using $V$ and $U$:
        $A^r = V^T A U, \quad B^r = V^T B, \quad C^r = CU$
(9) return $(A^r, B^r, C^r)$
```

2.20). The reduced system is $(V^T EU, V^T AU, V^T B, CU)$.

In the modified AISIAD algorithm for descriptor systems, we use the approximated power iterations in order to obtain the dominant eigenspaces of $QEPE^T$ and PE^TQE by approximating PE^TV_i and QEU_i. For the approximation of the first product, multiply the generalized Lyapunov equation for P from right by V_i:

$$A\underbrace{PE^TV_i}_{X} + E\underbrace{PE^TV_i}_{X}\underbrace{V_i^TEA^TV_i}_{H} + \underbrace{EP(I - E^TV_iV_i^T)A^TV_i + BB^TV_i}_{M} = 0 \qquad (5.8)$$

As before, we can compute a low-rank approximation for the gramian $\hat{P} \approx P$ using methods, for example, described in [79], and therefore obtain approximation of the term $\hat{M} \approx M$.

The equation (5.8) leads to the following matrix equation:

$$AX + EX\hat{H} + \hat{M} = 0, \tag{5.9}$$

We can solve (5.9) analogously to solving (5.3) by performing a Schur decomposition of $H = WSW^*$ and then solving for the columns of matrix XW. In this case instead of (5.7) we will have to solve the following system of equations:

$$(A + s_{jj}E)\tilde{x}_j = (MW)_j - \sum_{i=1}^{j-1} s_{ij}\tilde{x}_i \tag{5.10}$$

Again, this system can be solved fast if matrices A and E are sparse, or if the fast solver is available.

The calculations for approximation of QEU_i are analogous.

We outlined the resulting algorithm as Algorithm 6.

5.6 Advantages and limitations of the proposed algorithm

The proposed algorithm is applicable to any stable linear system in a state-space form. We have extended it for descriptor systems with nonsingular descriptor matrix E.

Advantages of the proposed method with respect to the original AISIAD is its extended applicability to a broader range of systems (original AISIAD is applicable only to the cases where $A > 0$) and its improved accuracy. It comes at extra cost, however - the cost usually comparable to gramian approximation.

The major factor, which governs the accuracy of the proposed method is the accuracy of low-rank approximations of P and Q.

In addition, there is no benefit of applying AISIAD to the symmetric systems

Algorithm 6: Modified AISIAD algorithm for descriptor systems with nonsingular E.

```
**Input:** System matrices $(E, A, B, C)$, desired reduced order $q$, initial
projection matrix $V \in \mathbb{R}^{n \times q}$
**Output:** Order-$q$ reduced model $(E^r, A^r, B^r, C^r)$.
Get a low-rank approximations of proper gramians
  $\tilde{P} \approx P$ and $\tilde{Q} \approx Q$
**repeat**
    Solve $AX_i + EX_iH_i + \hat{M}_i = 0$, where
      $H_i = V_i^T A^T V_i$
      $\hat{M}_i = BB^T V_i + E\tilde{P}(I - E^T V_i V_i^T) A^T V_i$
    $[U_i, S_i] = qr(X_i, 0)$
    Solve $A^T Y_i + E^T Y_i F_i + \hat{N}_i = 0$, where
      $F_i = U_i^T A U_i$,
      $\hat{N}_i = C^T C U_i + E^T \tilde{Q}(I - EU_iU_i^T) AU_i$
    $[V_{i+1}, R_{i+1}] = qr(Y_i, 0)$
**until** convergence
Set $V \leftarrow V_{i+1}$ and $U \leftarrow U_i$,
Project the initial system using $V$ and $U$:

  $E^r = V^T E U, \quad A^r = V^T A U, \quad B^r = V^T B, \quad C^r = CU$

**return** $(E^r, A^r, B^r, C^r)$
```

$(A = A^T, B = C^T)$, since for such systems $P = Q$, and AISIAD cannot do better than dominant gramian eigenspace method (DGE).

5.6.1 Complexity of the modified AISIAD algorithm

The computational cost of the modified AISIAD algorithm is directly proportional to the cost of solving q linear systems of equations in (5.10). If we assume that the matrices A and E are sparse enough to compute the solution in order-n time, this will correspond to linear complexity of the whole algorithm with respect to scaling by n. Our numerical experiments on the RLC circuit example (described in the next section) fully support this statement: for RLC example the time taken to reduce the system scales linearly with n. The largest model we tried so far had the order $n = 500,000$.

One can employ iterative solvers for the solution of (5.10) if the matrices are dense.

If the sparse solver is employed, the cost of the algorithm with available low-rank approximations to P and Q is approximately

$$2N_{it}q(C_{factor} + C_{bksolve}),$$

where N_{it} is a number of modified AISIAD iterations, C_{factor} is a cost of a matrix factorization of $A + s_{jj}E$, and $C_{bksolve}$ is a cost of backward-solving for the solution.

An interesting feature of the proposed algorithm is that it uses one backward solve per one matrix factorization, therefore for each iteration $2q$ matrix factorizations and $2q$ backward solves need to be performed. The linear systems in (5.10) are essentially the same as in the multiple-point Padé approximation via Krylov-subspaces [62]. However, modified AISIAD algorithm uses q different shift parameters, whereas PVL method generally uses less than q, therefore for PVL the number of backward solves per one matrix factorization is usually more than one. The Arnoldi algorithm requires only one matrix factorization and q backward solves. Therefore, both PVL and Arnoldi are faster than the modified AISIAD algorithm by a constant factor.

5.6.2 Passivity preservation

The modified AISIAD method does not impose any assumptions on the physical nature of the input and output signals. In other words, this method is *generic.* However, it is very important for many model reduction problems to preserve some properties of the transfer function, like positive-realness (in case where input signals are port voltages and output signals are port currents) or bounded-realness (in case of S-parameter modeling).

So far, the only method which is applicable for large-scale model reduction and which preserves passivity[3] is the PRIMA algorithm [58]. This method is based on Krylov-subspace projections, which can be viewed as approximating dominant controllable states [25]. As it was mentioned before, this can sometimes lead to large errors in the frequency domain, which do not necessarily decrease with increasing of

[3] with assumption A being positive semidefinite and $B = C^T$

the reduced order. This is fully consistent with the experimental results which we present in the next section. The same can be said about variants of PRIMA, which uses dominant eigenspaces of P and Q for the projection bases.

As a practical (and widely used) solution, we can obtain a passive model by post-processing. Since modified AISIAD produces a very accurate models in the frequency domain, we can, for example, use the poles of the reduced model, and re-fit the reduced transfer function using any convex optimization algorithms which ensure passivity [17, 34, 77]. We have tested this approach on the RLC line example and present our results in the next section.

5.7 Computational results

For the test cases we used four benchmark systems, which we describe below. For each of these systems we compared the original AISIAD, modified AISIAD, dominant gramian eigenspaces (DGE), low-rank square root (LRSQRT), Arnoldi [30, 58] and Padé via Lanczos (PVL) [62] reduction algorithms. As an error metric, we used the H_∞ norm of the difference between sufficiently accurate reduced model [4] (in the examples it was the TBR model of order 100-150) and all above mentioned approximations. Note that our error metric is essentially the maximum of the difference between the original and reduced system's transfer functions over the entire $j\omega$ axis. We assumed an error to be infinity if the reduced model was unstable (these cases correspond to discontinuities of the lines on our error plots).

Our results showed that the modified AISIAD always outperforms all of the above mentioned methods, with the exception of LRSQRT. For example of the rail cooling and some RLC circuits, modified AISIAD performed much better than LRSQRT. However, for other cases it showed almost identical performance. For several RLC examples modified AISIAD turned out to be slightly inferior to LRSQRT method.

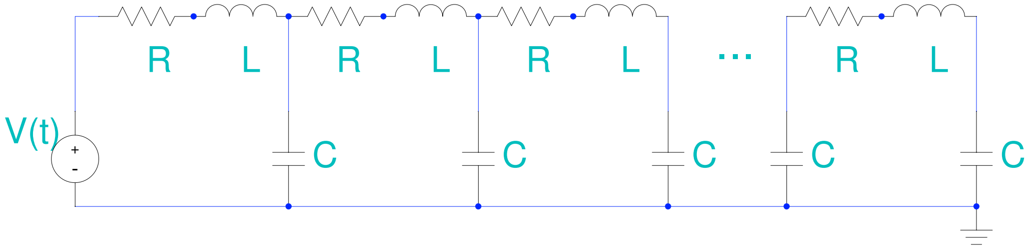

Figure 5-2: RLC transmission line example

5.7.1 RLC transmission line

The first system is an RLC transmission line depicted on figure 5-2, with varying values for R, L and C. Input signal $u(t)$ is the voltage at the first node. The outputs are the voltage at the last node and current flowing through the first inductor. The state vector consists of node voltages and inductor currents, and nodal analysis equations result in a system (A, B, C) with non-symmetric, indefinite matrix A. We varied the size of this system from several hundreds to hundreds of thousands, for different values of R, L and C and different choices of output signals. The maximal order of the system was 500,000.

Our results showed that modified AISIAD method always produces more accurate results than any above mentioned reduction methods in the H_∞ error metric.

On the figure 5-3 the H_∞ errors of the reductions for this RLC line are plotted versus the reduced order q. The initial order of the system was 1000. As the figure suggests, the errors for the DGE method (as well as all other methods!) is much bigger than the errors for the modified AISIAD algorithm. We'd like to stress that here we used exact low-rank approximant for DGE method, whereas for the modified AISIAD we used approximated gramians (the ones provided by algorithm 4). This way, the curve for DGE is a universal upper bound for all family of methods that approximate P and Q separately. Evidently, AISIAD is the best method for this case, significantly outperforming the original AISIAD method.

[4] Using non-reduced model for computing H_∞ norm is very expensive

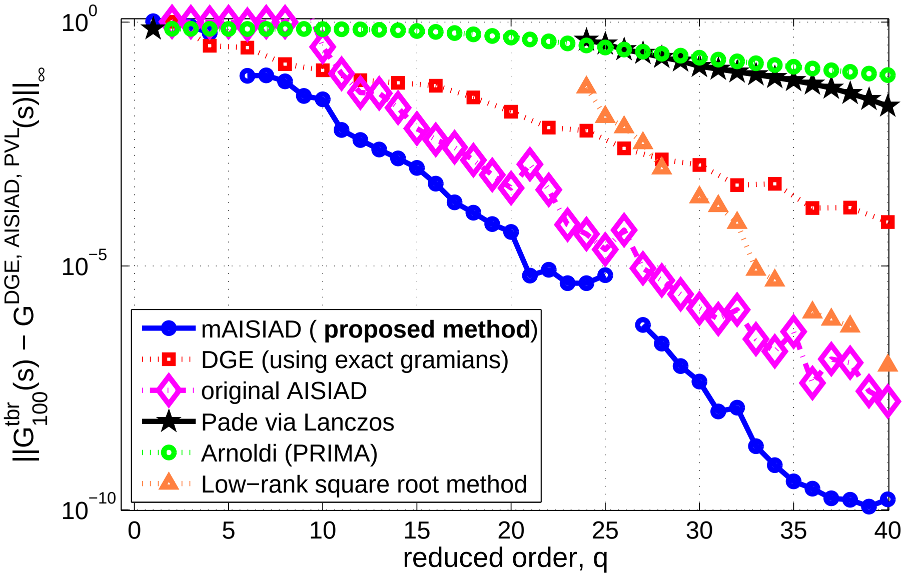

Figure 5-3: Reduction errors for the RLC transmission line of order $n = 1000$, $R = 1, L = 2, C = 3$

RLC line - MNA formulation

We have used modified nodal analysis (MNA) formulation for the transmission line depicted on figure 5-4. The inputs were the voltage sources either at a single end or both ends, and the outputs were either currents through the end resistors or (in the case of a single input) voltage at the other end of a line.

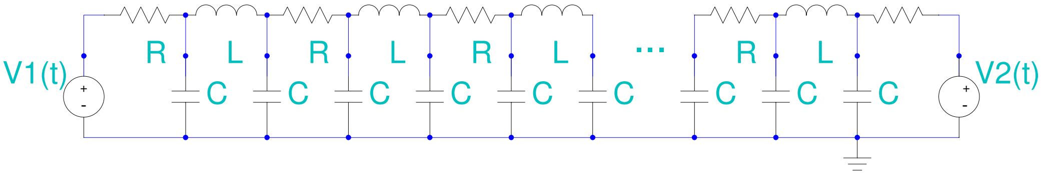

Figure 5-4: RLC transmission line two-port model.

The MNA formulation for this line results in a dynamical system in the form (E, A, B, C). We have observed that the modified AISIAD method always works better than PVL, DGE and Arnoldi (which is the PRIMA algorithm [58])[5]. However, low-rank square root method sometimes gives comparable results as modified AISIAD, and for two-port impedance model in some cases it even produces inferior results with respect to LRSQRT. However, the two-port impedance model is almost irreducible, it's Hankel singular values are quite high.

Passive post-processing

We have used the RLC transmission line model with input being the voltage at the first node and the output being current through the first resistor of the line (cf. Figure 5-1). The passivity constraint implies the transfer function being *positive-real*, that is, in addition to being stable, it satisfies the following condition:

$$\mathfrak{Re}(H(j\omega)) > 0, \quad \forall\omega \tag{5.11}$$

The parameters of RLC line were $N = 1000$, $R = 0.1, L = 2, C = 15$. For this model, the modified AISIAD reduced model of order $q = 30$ is not passive, with the H_∞ norm of error being 0.70%. We have used this model for the passive fitting algorithm from [77] and obtained a positive-real reduced model of order $q = 20$, with an H_∞ error 0.96%. The PRIMA algorithm for this order has a tremendously higher H_∞ error, which is 88.2%. Figure 5-5 shows the real parts of the above mentioned transfer functions.

RC line

In order to test the modified AISIAD algorithm on a symmetric system ($A = A^T, B = C^T$), we used a simple RC line (figure 5-2 with $L = 0$) with input being the voltage at the first node and output being the current through the first resistor. The state vector was the vector of node voltages. For this system $P = Q$ and dominant eigenspaces of PQ and QP will be the same as the ones of P and Q separately, therefore modified AISIAD should achieve exactly the same accuracy as DGE method. Our numerical

[5] PRIMA algorithm has it's own advantages though, because it preserves passivity of a reduced model. However, we are concerned here only with H_∞ norm as an error measure.

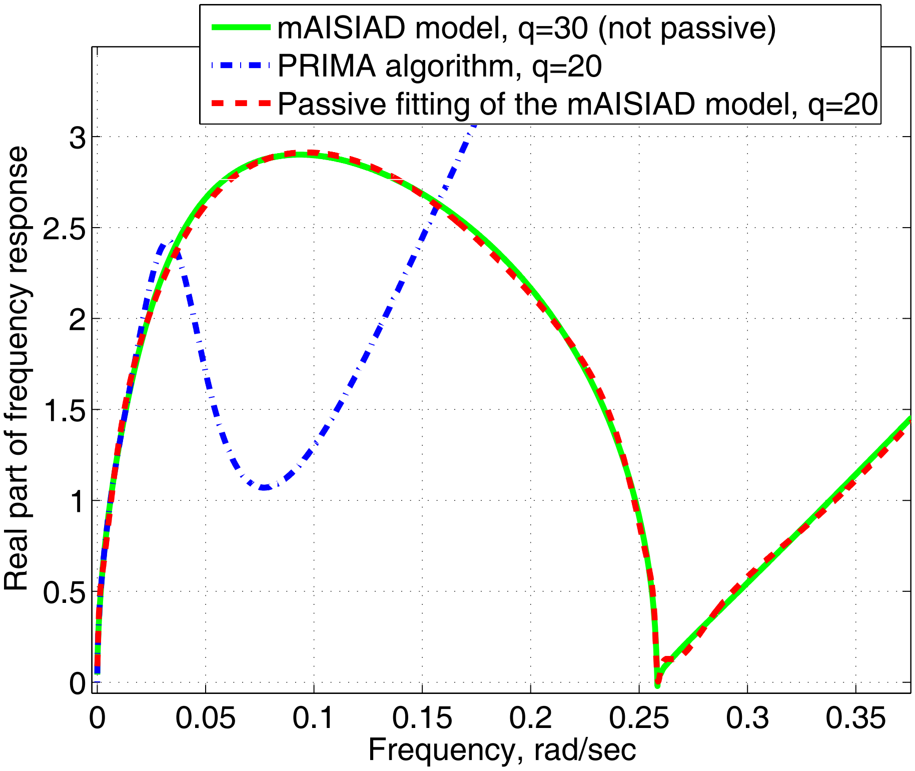

Figure 5-5: Real parts of the transfer functions for the modified AISIAD reduced model (which has been used for the post-processing, solid line), PRIMA reduced model (dash-dotted line) and the model obtained after post-processing of modified AISIAD model (dashed line). One can note that PRIMA algorithm poorly approximates the original transfer function away from the expansion point (which is at zero frequency). The non-reduced transfer function is almost indistinguishable from the modified AISIAD model.

experiments fully support this statement: errors for DGE and modified AISIAD are the same for this test case.

5.7.2 Linearization of micromachined pump

The second example was the linearization of the micromachined pump (fixed-fixed beam), which has been discussed in the previous Chapter. The linearization of this model around equilibrium leads to the nonsymmetric system (A, B, C) with indefinite system matrix A. The order $n = 880$.

On the figure 5-6 the errors for the MEMS test case are presented. Here still

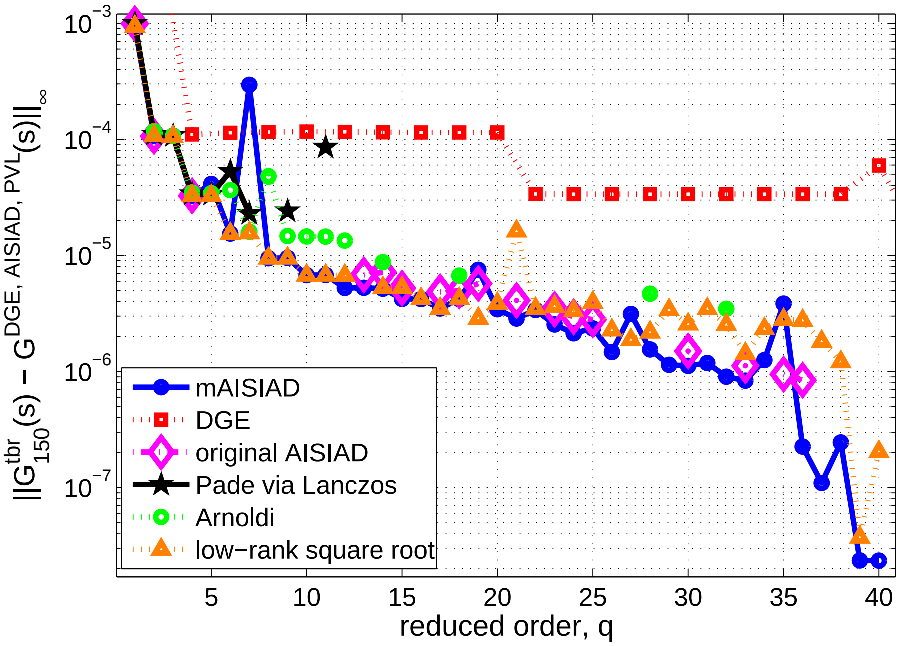

Figure 5-6: Errors for the MEMS linearization, $N = 880$.

modified AISIAD method performs better than any other method, but the difference is not as dramatic as for other examples. The LRSQRT method showed the results similar to the modified AISIAD.

5.7.3 Cooling profile of steel rail

This test case was obtained from the Oberwolfach Model Reduction Benchmark Collection web site [1]. The reader is referred to the description of *Heat transfer problem for cooling of steel profiles* benchmark on the mentioned web site for descriptions. This is the model in a descriptor form (E, A, B, C) with $n = 1357$, having 7 inputs and 6 outputs.

For this example the modified AISIAD showed superior performance with respect to any other approximations, including LRSQRT method.

On the figure 5-7 we present the error plot for this example. Here, again, AISIAD greatly outperforms any other approximations to TBR, as well as Krylov-subspace

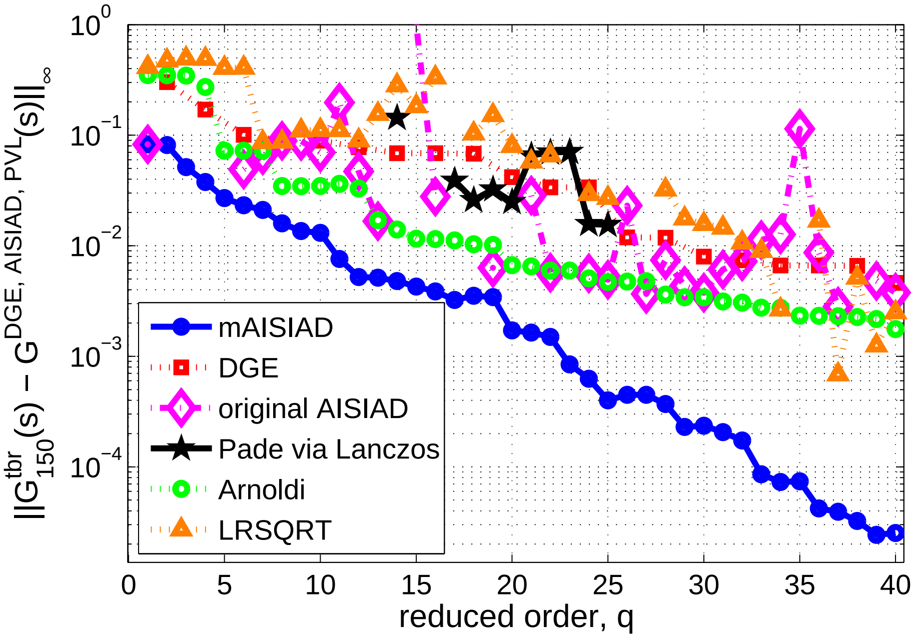

Figure 5-7: Reduction errors for the rail example, $n = 1357$, 7 inputs, 6 outputs

based reductions. The reduced models of order $q = 2, 3, 4$ are unstable, but it's even smaller than the number of inputs. As expected, modified AISIAD outperforms original AISIAD algorithm.

5.7.4 Optical filter

This test case was obtained from the Oberwolfach Model Reduction Benchmark Collection web site [1]. The reader is referred to the description of *Tunable Optical Filter* benchmark on the mentioned web site for descriptions. This is the model in a descriptor form (E, A, B, C) with $n = 1668$, having 1 input and 5 outputs. The corresponding errors are presented on figure 5-8. Here the dominant gramian eigenspace projection was computed using the same approximate gramians which were used for the modified AISIAD method. The LRSQRT method showed very similar errors as the modified AISIAD.

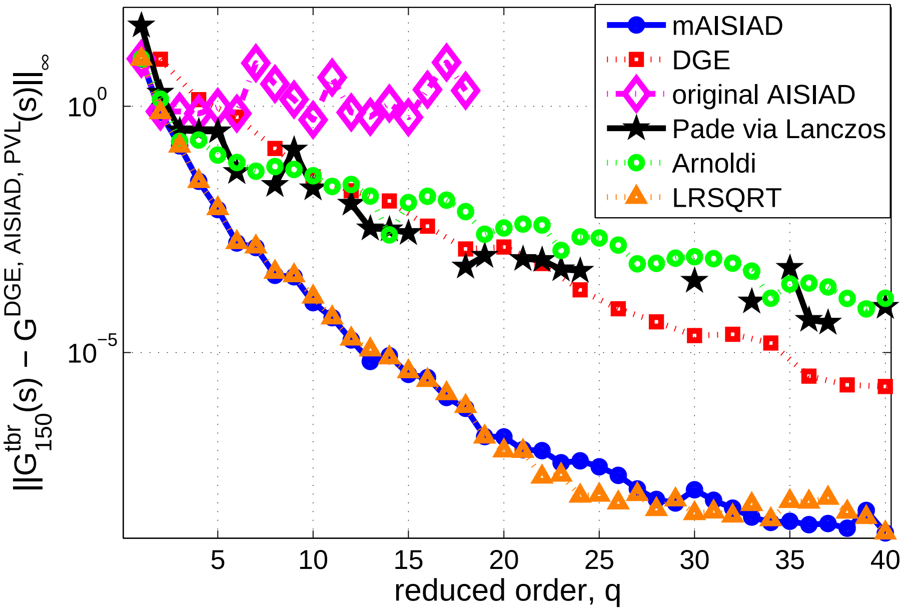

Figure 5-8: Reduction errors for the optical filter example, $n = 1668$, 1 inputs, 5 outputs

Chapter 6

A graph-based model reduction for parameterized RC networks

That's OK. I do it all the time

- Irina Missiuro (after falling down the stairs)

The work in this Chapter has been done in collaboration with Dr. Joel Phillips and Dr. Zhenhai Zhu during an internship at Cadence Research Labs, Berkeley, CA. This work has been published in [84].

In this Chapter we present a graph-based reduction algorithm for RC networks.

6.1 Problem setup, notation and prior work

The problem of reduction of RC networks typically appears within integrated circuit simulation software. For example, every piece of interconnect can be represented as an RC network. External connections are made to *port nodes* of such RC network. The goal of the reduction is to approximate the network's response at the port nodes. Depending on the formulation, the inputs can be port currents and outputs can be port voltages (Z-formulation) or the other way around (Y-formulation). The output model can be another smaller RC circuit or just a general state-space model.

The *nodal analysis* applied to such network leads to the description in the form (E, A, B, C, D) as in (1.8). In the case where inputs are considered to be currents (Z-formulation), system states are node voltages, matrix A is negative of the *conductance matrix*, matrix E is a *capacitance matrix* and matrix $B = C^T$ maps port numbers to the port nodes. In order to emphasize the physical meaning of the variables, in the derivations of this Chapter we will denote state vector (consisting of node voltages) by $\mathbf{v}$, conductance matrix by G and capacitance matrix by C. External currents into the port nodes will be denoted by $\mathbf{J}$. The ground node will be one of the port nodes. Other notation remains the same.

For deterministic and non-parameter varying RC networks, model order reduction is a mature area with three main classes of well-established algorithms:

1. Methods based on TBR;

2. Moment-matching methods such as PVL [62] and PRIMA [57];

3. Graph-based reduction methods [52, 21, 82, 74, 6], among which the TICER algorithm [74, 6] is most widely known.

These classes of methods complement each other: while methods from the first group are generally slow, but provide more reduction, methods from the last group are much faster but less accurate.

More recently, several approaches have been proposed for reduction that considers process variability effects [67, 48, 19, 86, 46, 51] in which the parametric models are considered. Algorithms in [48, 19, 86, 46] are moment matching type algorithms, or related to moment matching type algorithms.

Interestingly, to our best knowledge, there is very little work on the graph-based algorithms for the parameterized model order reduction. In this Chapter, we propose such an algorithm.

There are three independent works which are the predecessors to the reduction algorithm outlined in Section 6.2. The first work is described in the Ph.D. thesis of McCormick [52]. This work presents a general method (Moment Polynomial Nodal Analysis) that computes the voltage response time moments of a linear circuit up to any prescribed order. It is based on standard Gaussian elimination and the single-variable Taylor series expansion. Another two related works are [82] and [74]. Basic node elimination rules were derived in [82], and simple node selection rules based on node time constants were introduced in the TICER algorithm [74].

The node elimination rules presented in Section 6.2 can be treated as a special case of the general method in [52] where the truncation is up to the quadratic polynomials of s, the complex frequency in Laplace domain. We have extended the single-variable Taylor expansion in [52, 21, 74] to the multi-variable one and used it as the theoretical foundation for the parameterized model order reduction. This reduction algorithm automatically adjusts to the parameter variation range. Smaller range results in smaller reduced model and vice versa.

6.2 Graph-Based Nominal Reduction

The graph-based reduction algorithms for RC networks are quite different than the methods we were dealing with so far in this work. From one hand, the applicability of the graph-based methods is very well defined, therefore we can use our insight from circuits theory. From another hand, this method operates differently than all previously described MOR methods. Though, as we show below, our graph-based method can be interpreted as performing certain projections, it does not make sense to implement it as a series of projections. Instead, this algorithm is implemented in a way similar to a symmetric sparse matrix solver where the circuit elements are stored as a graph and nodes of the graph are eliminated in such way as to minimize the fill-ins. This way, a single reduction step corresponds to elimination of a single node of a network, or equivalently reducing the dynamical system's order by one.

Another important feature of the presented method is that it preserves the sparsity structure of the original RC network. For example, if the original circuit has a tree-like structure, so will be the reduced circuit.

In this Section we derive and describe the proposed graph-based reduction method for nominal (non-parameterized) RC circuits. We highlight the differences between the original TICER algorithm and the proposed method.

6.2.1 Basic Formulation

The frequency-domain circuit description of an RC network with N nodes can be written as:

$$(sC+G)\mathbf{v}=\mathbf{J}, \tag{6.1}$$

where

$$C_{ij}=\begin{cases} -c_{ij}, & i\neq j \\ \sum_{m=1}^{N} c_{mi}, & i=j \end{cases}, \quad G_{ij}=\begin{cases} -g_{ij}, & i\neq j \\ \sum_{m=1}^{N} g_{mi}, & i=j \end{cases}, \tag{6.2}$$

g_{ij} and c_{ij} are respectively the conductance and the capacitance between nodes i and j, $\mathbf{v}$ is a vector of node voltages, $\mathbf{J}$ is a vector of external currents into the circuit, and s is the complex frequency. Without loss of generality, let the N-th node be the internal node we want to eliminate. Then the matrices in (6.1) can be partitioned as following:

$$\begin{bmatrix} s\tilde{C}+\tilde{G} & -(s\mathbf{c}_N+\mathbf{g}_N) \\ -(s\mathbf{c}_N+\mathbf{g}_N)^T & sC_{NN}+G_{NN} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{v}} \\ v_N \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{J}} \\ J_N \end{bmatrix} \tag{6.3}$$

where matrices $\tilde{C}$ and $\tilde{G}$ are the capacitance and conductance matrices for the subcircuit excluding node N, $\mathbf{g}_N$ and $\mathbf{c}_N$ are the vectors of size $(N-1)$ with conductances g_{iN} and capacitances c_{iN} between node N and other nodes, $\tilde{\mathbf{v}}$ contains all node voltages except for v_N, and $\tilde{\mathbf{J}}$ contains all external current sources except for J_N. Since node N is an internal node, by definition, $J_N=0$. We can solve the second equation in (6.3) for v_N and obtain

$$v_N=\frac{s\mathbf{c}_N^T+\mathbf{g}_N^T}{sC_{NN}+G_{NN}}\tilde{\mathbf{v}}. \tag{6.4}$$

Substituting v_N into the first equation in (6.3), we obtain

$$(s\tilde{C}+\tilde{G}-E)\tilde{\mathbf{v}}=\tilde{\mathbf{J}}, \tag{6.5}$$

where

$$E=\frac{(s\mathbf{c}_N+\mathbf{g}_N)(s\mathbf{c}_N^T+\mathbf{g}_N^T)}{sC_{NN}+G_{NN}}. \tag{6.6}$$

It should be noted that the procedure in (6.4)-(6.6) is nothing but one step in the standard Gaussian elimination for symmetric matrix $(sC + G)$.

Matrix E is not an affine function of frequency s, hence the circuit described by (6.5) can not be realized by an RC circuit with $(N - 1)$ nodes. In other words, equation (6.5) can not be cast into the form like that in (6.1). Hence the procedure in (6.4)-(6.6) does not land itself as a recursive node-elimination algorithm that we can use to perform reduction. It is proposed in [52, 21, 74] that one can use the truncated Taylor series to approximate matrix E such that the reduced network is realizable. We briefly summarize this procedure in the following. Re-write (6.6) as

$$\begin{aligned} E &= \frac{(s\mathbf{c}_N + \mathbf{g}_N)(s\mathbf{c}_N^T + \mathbf{g}_N^T)}{G_{NN}}(1 + \frac{sC_{NN}}{G_{NN}})^{-1} \\ &\simeq \frac{(s\mathbf{c}_N + \mathbf{g}_N)(s\mathbf{c}_N^T + \mathbf{g}_N^T)}{G_{NN}}(1 - s\frac{C_{NN}}{G_{NN}}) \\ &\simeq \frac{\mathbf{g}_N\mathbf{g}_N^T}{G_{NN}} + s\Big(\frac{\mathbf{c}_N\mathbf{g}_N^T + \mathbf{g}_N\mathbf{c}_N^T}{G_{NN}} - \frac{C_{NN}}{G_{NN}}\frac{\mathbf{g}_N\mathbf{g}_N^T}{G_{NN}}\Big) \end{aligned} \tag{6.7}$$

where the first approximate equal sign is due to the truncated Taylor series expansion at $s = 0$ and the second approximate equal sign is due to the truncated polynomial of s. The leading truncation term after the first approximate equal sign is $s^2\frac{C_{NN}^2}{G_{NN}^2}$. This suggests the following truncation criterion for this step:

$$\left|s_{max}\frac{C_{NN}}{G_{NN}}\right| < \epsilon_1, \tag{6.8}$$

where $s_{\max} = j\omega_{\max}$ is the maximal complex frequency of interest and ϵ_1 is a user-defined small constant. The truncated term after the second approximate equal sign is $s^2\frac{\mathbf{c}_N\mathbf{c}_N^T}{G_{NN}}$. Since this is essentially an entry-by-entry perturbation $s^2\frac{c_{iN}c_{jN}}{G_{NN}}$ to a symmetric matrix, we have chosen to enforce that these errors be small with respect to the corresponding diagonal elements. Therefore, a reasonable truncation criterion for this step is[1]

$$\begin{cases} \left|s^2 \frac{c_{iN}c_{jN}}{G_{NN}}\right| < \epsilon_2 \left|sC_{ii} + G_{ii}\right| \\ \left|s^2 \frac{c_{iN}c_{jN}}{G_{NN}}\right| < \epsilon_2 \left|sC_{jj} + G_{jj}\right| \end{cases} \qquad \forall i, j \neq N; s = s_{\max}, \tag{6.9}$$

where ϵ_2 is another user-defined small constant. One should note that the tolerances ϵ_1 in (6.8) and ϵ_2 in (6.9) should be different because they have very different origin. In addition, the truncation criteria in (6.8) and in (6.9) are equally important. However, in the TICER algorithm [74], the criteria in (6.9) are not enforced and the negative term $-\frac{C_{NN}}{G_{NN}} \frac{g_{iN}g_{jN}^T}{G_{NN}}$ in (6.11) is dropped.

Substituting (6.7) into (6.5), we obtain

$$(s\hat{C} + \hat{G})\hat{\mathbf{v}} = \tilde{\mathbf{J}}, \tag{6.10}$$

where

$$\hat{C}_{ij} = \tilde{C}_{ij} - \Delta C_{ij}, \;\; \Delta C_{ij} = \frac{c_{iN}g_{jN} + g_{iN}c_{jN}}{G_{NN}} - \frac{C_{NN}}{G_{NN}} \frac{g_{iN}g_{jN}}{G_{NN}}, \tag{6.11}$$

and

$$\hat{G}_{ij} = \tilde{G}_{ij} - \Delta G_{ij}, \;\; \Delta G_{ij} = \frac{g_{iN}g_{jN}}{G_{NN}}. \tag{6.12}$$

The vector $\hat{\mathbf{v}}$ approximates the original vector of voltages $\tilde{\mathbf{v}}$, due to the approximation made in (6.5) for the term E. The terms ΔC_{ij} in (6.11) and ΔG_{ij} in (6.12) can be viewed respectively as a capacitance update and a conductance update to the sub-circuit excluding the node to be eliminated.

6.2.2 Nominal Reduction Algorithm

The procedure in (6.1)-(6.10) suggests a recursive node-elimination algorithm for model order reduction. The accuracy of this algorithm can be controlled by tolerances in the elimination criteria (6.8) and (6.9). Smaller values of tolerances $\epsilon_{1,2}$ will lead to less reduction but better accuracy. Since the essence of this algorithm is the

[1] The inequalities in (6.8,6.9) should hold for any $s \in [0, s_{\max}]$, however obviously the worst-case condition is when $s = s_{\max}$.

truncated Gaussian elimination for symmetric sparse matrix, the most efficient implementation is based on a graph representation of the RC network [27] and hence the name graph-based reduction. The same as in standard symmetric sparse matrix solver, the minimum degree ordering is used to minimize the number of fill-ins. The algorithm starts with the nodes with the fewest neighboring nodes (the degree) and stops when the degree of each node is above a user-specified value. The proposed algorithm is summarized as Algorithm 7.

```
Algorithm 7: Graph Based Nominal Reduction
Input: $C, G$; truncation tolerance $\epsilon_1$ and $\epsilon_2$; $d_m$: maximal degree allowed for a node to be considered for elimination
Output: $\hat{C}, \hat{G}$
(1)  Set up graph $\mathcal{G}$ for $C, G$
(2)  Find all internal nodes with degree less than $d_m$ and put them
     into a set $\Omega$. Order the nodes in $\Omega$ by the number of neighbors.
(3)  foreach $node_N \in \Omega$
(4)    if $node_N$ satisfies (6.8) and (6.9)
(5)      Eliminate $node_N$ and all edges (resistors and capacitors)
         connected to it from graph $\mathcal{G}$
(6)      foreach node pair $(i, j)$ that had been connected to $node_N$
         by either a resistor or a capacitor
(7)        add conductance $\Delta G_{ij}$ in (6.12) between nodes $i$ and $j$
           in graph $\mathcal{G}$
(8)        add capacitance $\Delta C_{ij}$ in (6.11) between nodes $i$ and $j$
           in graph $\mathcal{G}$
(9)      Update neighbor counts of the nodes in $\Omega$ and eliminate $N$
         from $\Omega$.
(10) Go to step 2 and iterate until no node from $\Omega$ satisfies (6.8) and
     (6.9).
```

6.2.3 Passivity Preservation

As shown above, the RC circuit description in (6.10) generated from each node elimination step in Algorithm 7 is potentially realizable. However, it is possible that the added capacitance ΔC_{ij} in (6.11) is negative. Consequently, the final circuit may have some negative capacitances. It is for this reason that the negative term $-\frac{C_{NN}}{G_{NN}}\frac{g_{iN}g_{jN}^T}{G_{NN}}$ in (6.11) is dropped in TICER algorithm [74]. As the following Theorem states, even if the final circuit does have negative capacitances, such circuit is always stable and passive.

Theorem 6.2.1. *The resulting system (6.10) obtained after each node elimination in Algorithm 7 is stable and passive.*

A complete proof of this statement is presented in the Appendix A.

This is an important result because it allows us to use the full series-based elimination rules with rigorous error control (6.8, 6.9), which we generalize for parameterized circuits in the next Section. As our results will show, ad-hoc deletions of components results in a reduction approach that is less reliable than when more rigorous rules are used. All such sources of "scatter" can potentially cause problems for timing convergence. While at one time the existence of negative capacitors introduced potential problems in a timing analysis flow, more modern timing and signal integrity analysis engines make heavier use of advanced modeling engines. These engines can often accept general state-space or pole-residue type macromodels. Most SPICE-type circuit simulators can also accept these general macromodels. Due to this change, it is now practical in many contexts to consider using the full rigorous rules, which as we will show provide more accurate and reliable results.

6.3 Graph-Based Parameterized Reduction

In this Section we describe a generalization of the nominal graph-based reduction method for circuits where the values depend on a set of parameters. A remarkable feature of the nominal reduction method is that it considers the frequency band of interest and will produce more compact models for smaller bandwidth. Likewise, the parameter-dependent generalization utilizes the limits of parameter's variability.

6.3.1 Formulation

Similar to [67, 48, 19, 86, 46], we assume that the conductance and capacitance are affine function of parameters $\lambda_1 \dots \lambda_\nu$ as

$$C(\bar{\lambda}) = C^0 + \sum_{k=1}^{\nu} \lambda_k C^k, \;\; G(\bar{\lambda}) = G^0 + \sum_{k=1}^{\nu} \lambda_k G^k \tag{6.13}$$

where C^0 and G^0 are respectively nominal capacitance and conductance matrix, C^k and G^k are respectively capacitance and conductance sensitivity with respect to parameter λ_k. The circuit equation for the RC network is identical to (6.1) except that C and G are replaced by $C(\bar{\lambda})$ and $G(\bar{\lambda})$.

Assuming node N is to be eliminated and following the same procedure as in (6.3)-(6.5), we obtain

$$(s\tilde{C}(\bar{\lambda}) + \tilde{G}(\bar{\lambda}) - E(\bar{\lambda}, s))\tilde{\mathbf{v}} = \tilde{\mathbf{J}} \tag{6.14}$$

where

$$E(\bar{\lambda}, s) = \frac{\alpha(\bar{\lambda}, s)\alpha^T(\bar{\lambda}, s)}{sC_{NN}(\bar{\lambda}) + G_{NN}(\bar{\lambda})} \tag{6.15}$$

$$\alpha(\bar{\lambda}, s) = s\mathbf{c}_N(\bar{\lambda}) + \mathbf{g}_N(\bar{\lambda}). \tag{6.16}$$

Following the similar truncation procedure in (6.7), we obtain

$$\begin{aligned} E(\bar{\lambda}, s) &= \frac{\alpha(\bar{\lambda}, s)\alpha^T(\bar{\lambda}, s)}{G^0_{NN}}\Big(1 + \frac{sC_{NN}(\bar{\lambda}) + \sum_{k=1}^{\nu}\lambda_k G^k_{NN}}{G^0_{NN}}\Big)^{-1} \\ &\simeq \frac{\alpha(\bar{\lambda}, s)\alpha^T(\bar{\lambda}, s)}{G^0_{NN}}\Big(1 - \frac{sC_{NN}(\bar{\lambda}) + \sum_{k=1}^{\nu}\lambda_k G^k_{NN}}{G^0_{NN}} + \\ &\quad +2sC^0_{NN}\frac{\sum_{k=1}^{\nu}\lambda_k G^k_{NN}}{(G^0_{NN})^2}\Big) \\ &\simeq s(\Delta C^0 + \sum_{k=1}^{\nu}\lambda_k \Delta C^k) + (\Delta G^0 + \sum_{k=1}^{\nu}\lambda_k \Delta G^k) \end{aligned} \tag{6.17}$$

where

$$\Delta G^0_{ij} = \frac{g^0_{iN} g^0_{jN}}{G^0_{NN}} \tag{6.18}$$

$$\Delta G^k_{ij} = \frac{g^0_{iN} g^k_{jN} + g^k_{iN} g^0_{jN}}{G^0_{NN}} - G^k_{NN}\frac{g^0_{iN} g^0_{jN}}{(G^0_{NN})^2} \tag{6.19}$$

$$\Delta C_{ij}^0 = \frac{g_{iN}^0 c_{jN}^0 + c_{iN}^0 g_{jN}^0}{G_{NN}^0} - C_{NN}^0 \frac{g_{iN}^0 g_{jN}^0}{(G_{NN}^0)^2} \tag{6.20}$$

$$\begin{aligned}
\Delta C_{ij}^k &= \frac{g_{iN}^k c_{jN}^0 + c_{iN}^0 g_{jN}^k + c_{iN}^k g_{jN}^0 + g_{iN}^0 c_{jN}^k}{G_{NN}^0} \\
&- G_{NN}^k \frac{c_{iN}^0 g_{jN}^0 + g_{iN}^0 c_{jN}^0}{(G_{NN}^0)^2} - C_{NN}^0 \frac{g_{iN}^k g_{jN}^0 + g_{iN}^0 g_{jN}^k}{(G_{NN}^0)^2} \\
&- C_{NN}^k \frac{g_{iN}^0 g_{jN}^0}{(G_{NN}^0)^2} + 2C_{NN}^0 G_{NN}^k \frac{g_{iN}^0 g_{jN}^0}{(G_{NN}^0)^3}.
\end{aligned} \tag{6.21}$$

The first approximate equal sign in (6.17) is due to the truncated Taylor series expansion. We use the following condition to ensure the leading truncation term is small

$$\left| \frac{s_{max} C_{NN}(\bar{\lambda}) + \sum_{k=1}^{\nu} \lambda_k G_{NN}^k}{G_{NN}^0} \right| < \epsilon_1 \tag{6.22}$$

where ϵ_1 is a user-specified small constant. The second approximate equal sign in (6.17) is due to the truncated polynomial of s and $\bar{\lambda}$. To ensure its leading truncation term is small, we require

$$\begin{aligned}
\frac{1}{G_{NN}^0} \Big| \delta g_{iN}(\bar{\lambda}) \delta g_{jN}(\bar{\lambda}) + s(\delta c_{iN}(\bar{\lambda}) \delta g_{jN}(\bar{\lambda}) + \delta g_{iN}(\bar{\lambda}) \delta c_{jN}(\bar{\lambda})) \\
+ s^2 c_{iN}(\bar{\lambda}) c_{jN}(\bar{\lambda}) \Big| \ll \min_{k=i,j} \Big| s C_{kk}(\bar{\lambda}) + G_{kk}(\bar{\lambda}) \Big|, \\
\forall i \neq N, j \neq N
\end{aligned} \tag{6.23}$$

where

$$\delta g_{iN}(\bar{\lambda}) = \sum_{k=1}^{\nu} \lambda_k g_{iN}^k, \quad \delta c_{iN}(\bar{\lambda}) = s \sum_{k=1}^{\nu} \lambda_k c_{iN}^k. \tag{6.24}$$

However, it is no longer sufficient to enforce (6.23) at the maximal frequency of interest to ensure that the inequality (6.23) holds at all frequencies and parameter values! To avoid a search for the worst-case corner over all frequencies, we suggest using a slightly more conservative but more convenient set of conditions:

$$\begin{cases} \left|\frac{\delta g_{iN}^{\max}\delta g_{jN}^{\max}}{G_{NN}^{0}}\right| < \epsilon_2 \min_{k=i,j} G_{kk}^{\min} \\ \left|s\frac{\delta g_{iN}^{\max}\delta c_{jN}^{\max}}{G_{NN}^{0}}\right| < \epsilon_2 \min_{k=i,j} \left|sC_{kk}^{\min} + G_{kk}^{\min}\right| \\ \left|s^2\frac{c_{iN}^{\max}c_{jN}^{\max}}{G_{NN}^{0}}\right| < \epsilon_2 \min_{k=i,j} \left|sC_{kk}^{\min} + G_{kk}^{\min}\right| \end{cases} , \quad \forall i \neq N, j \neq N, \tag{6.25}$$

where ϵ_2 is another user-specified small constant and

$$\begin{aligned} \delta c_{iN}^{\max} &= \max_{\lambda_1,\ldots\lambda_\nu} \left|\delta c_{iN}(\bar{\lambda})\right|, \delta g_{iN}^{\max} = \max_{\lambda_1,\ldots\lambda_\nu} \left|\delta g_{iN}(\bar{\lambda})\right|, \\ C_{kk}^{\min} &= \min_{\lambda_1,\ldots\lambda_\nu} \left|C_{kk}(\bar{\lambda})\right|, \;\; G_{kk}^{\min} = \min_{\lambda_1,\ldots\lambda_\nu} \left|G_{kk}(\bar{\lambda})\right|, \\ c_{iN}^{\max} &= \max_{\lambda_1,\ldots\lambda_\nu} \left|c_{iN}(\bar{\lambda})\right|, \;\; g_{iN}^{\max} = \max_{\lambda_1,\ldots\lambda_\nu} \left|g_{iN}(\bar{\lambda})\right|. \end{aligned}$$

It is easy to verify that enforcing the last two conditions in (6.25) at maximal frequency of interest $s_{\max}$ is sufficient to ensure that they are satisfied at all frequencies within the bandwidth of interest.

Substituting (6.17) into (6.14) and in view of (6.13), we obtain

$$\left[s(\hat{C}^0 + \sum_{k=1}^{\nu} \lambda_k \hat{C}^k) + (\hat{G}^0 + \sum_{k=1}^{\nu} \lambda_k \hat{G}^k)\right]\tilde{\mathbf{v}} = \tilde{\mathbf{J}}, \tag{6.26}$$

where

$$\hat{C}_{ij}^0 = \tilde{C}_{ij}^0 - \Delta C_{ij}^0, \;\; \hat{C}_{ij}^k = \tilde{C}_{ij}^k - \Delta C_{ij}^k \tag{6.27}$$

$$\hat{G}_{ij}^0 = \tilde{G}_{ij}^0 - \Delta G_{ij}^0, \;\; \hat{G}_{ij}^k = \tilde{G}_{ij}^k - \Delta G_{ij}^k. \tag{6.28}$$

Similar to the nominal reduction, the terms in (6.18)-(6.21) can be viewed as updates to the original circuit excluding the node to be eliminated.

6.3.2 Parameterized Reduction Algorithm

We summarize the procedure in (6.13)-(6.28) in the following algorithm

The Algorithm 8 automatically incorporates parameter variation ranges. Smaller variation range means that more nodes might satisfy conditions in (6.22) and (6.25) and hence are to be eliminated. This directly leads to a smaller reduced model.

Algorithm 8: Graph Based Parameterized Reduction
Input: C^0, C^k, G^0, G^k; truncation tolerance ϵ_1 and ϵ_2; d_m: maximal degree allowed for a node to be considered for elimination
Output: $\hat{C}^0, \hat{C}^k, \hat{G}^0, \hat{G}^k$

```
(1)  Set up graph $\mathcal{G}$ for $C^0, C^k, G^0, G^k$
(2)  Find all internal nodes with degree less than $d_m$ and put them
     into a set $\Omega$. Order the nodes in $\Omega$ by the number of neighbors.
(3)  **foreach** $node_N \in \Omega$
(4)     **if** $node_N$ satisfies (6.22) and (6.25)
(5)        Eliminate $node_N$ along with all edges connected to it from
           graph $\mathcal{G}$
(6)        **foreach** node pair $(i, j)$ that had been connected to node
           $N$ by either a resistor or a capacitor
(7)           Add the nominal conductance update in (6.18) between
              nodes $i$ and $j$
(8)           Add the nominal capacitance update in (6.20) between
              nodes $i$ and $j$
(9)           **foreach** sensitivity $k$ affecting elements between the
              node $N$ and node $i$ and $j$
(10)             Add the conductance sensitivity in (6.19) between
                 nodes $i$ and $j$
(11)             Add the capacitance sensitivity in (6.21) between
                 nodes $i$ and $j$
(12)       Update neighbor counts of the nodes in $\Omega$ and eliminate
           node $N$ from $\Omega$.
(13) Go to step 2 and iterate until no node from $\Omega$ satisfies (6.22)
     and (6.25).
```

It should be noted that the transformation in (6.17) does not appear to have a projection interpretation like the one described in the proof of Lemma A.0.2 in Appendix A. Therefore, the question of preserving the stability and passivity in the reduced model is still open.

6.4 Numerical results

In this Section, we first compare the accuracy of Algorithm 7 in Section 6.2.2 to that of the TICER algorithm in [74]. We then show the accuracy of the parameterized reduction (Algorithm 8 in Section 6.3.2). All examples used here are practical industry examples.

In our analysis the following relative error measure was used:

$$E = \max_{i,j} \max_{s \in [s_{\min}, s_{\max}]} \left| \frac{h_{ij}(s) - h_{ij}^r(s)}{h_{ij}(s)} \right| \tag{6.29}$$

where $h_{ij}(s)$ and $h_{ij}^r(s)$ denote the (i,j)-th element of the transfer functions of the original and reduced circuits, respectively.

We use the so-called compression ratio to measure the effectiveness of the graph-based reduction algorithms. It is defined as the ratio between the number of nodes in the reduced model and the original model. Therefore, smaller ratio means more effective reduction.

Unless stated otherwise, the parameters in (6.8) and (6.9) as well as in (6.22) and (6.25) are $s_{max} = 2\pi j \times 10^{11}, \epsilon_1 = 0.1$ and $\epsilon_2 = 10^{-4}$. The maximum degree threshold in both Algorithm 7 and 8 is set to be $d_m = 3$.

6.4.1 Accuracy of the nominal reduction algorithm

We use two collections of non-parameterized RC circuits in this Section, denoted as *collection A* and *collection B*. The *collection A* contains 24792 RC circuits, most of which are small and hence almost irreducible. The *collection B* contains 35900 RC circuits, most of which are large and hence reducible.

In order to clearly see the impact of keeping the negative capacitance term, we re-implemented our code in such way that the criterion (6.8) was checked only for the nodes of the initial circuit. This way, we have fixed the nodes to be eliminated and the elimination order, regardless of the node elimination rules. The reduction errors of the elimination rules of the Algorithm 7 and the original TICER are shown as a CDF (actually 1-CDF) in Figure 6-1. We show the “1-CDF” plot because it most clearly exposes the data of interest in assessing a reduction algorithm. Typically we are interested in how many cases fail to meet a given accuracy metric, typically a few percent or fractions of percent. The number of failures is usually small so we display 1-CDF instead of the CDF. It is clear from Figure 6-1 that keeping the negative capacitance term indeed produces more accurate results.

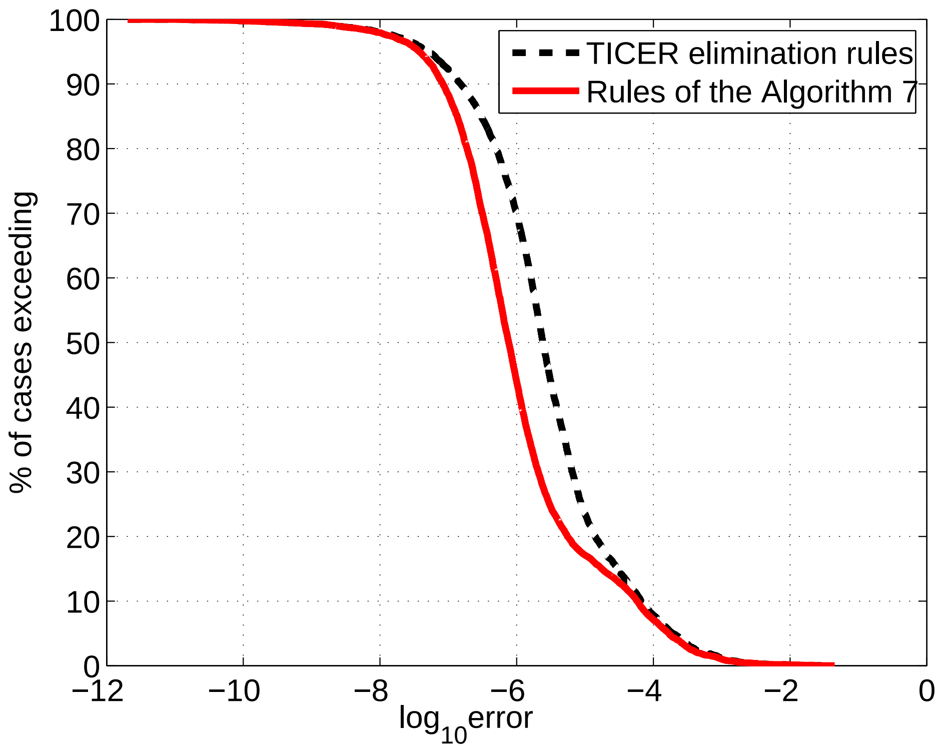

Figure 6-1: Number of cases for which the reduction error exceeds a given value for different update rules. Solid line: using the update based on correct Taylor series as in (6.11). Dashed line: using the original TICER update, without the last negative term in (6.11). In both cases the same nodes were eliminated, in the same order.

We then re-ran the Algorithm 7 with both conditions (6.8) and (6.9). The reduction errors of Algorithm 7 and the original TICER are compared in Table 6.1 where $\mathcal{E}$ refers to the errors for 98% of the reduced circuits. From Table 6.1 it is clear that the severity of outlier cases (cases with large error) are *significantly* reduced when both conditions (6.8) and (6.9) are used for node selection. Since the outlier cases often influence global parameter settings and therefore overall performance of the reduction algorithm, elimination of them can have a measurable impact on the overall reduction strategy. This is the most important benefit of using the full rigorous rules. It should be noted that the compression ratio by both algorithms is the same in this experiment (given similar parameter settings).

Table 6.1: Error spreads for different elimination conditions

	collection A	collection B
TICER	$2.1e-5 < \mathcal{E} < 3.3e-2$	$9.9e-5 < \mathcal{E} < 2.3e-1$
Algorithm 7	$1.8e-5 < \mathcal{E} < 2.2e-3$	$5.8e-6 < \mathcal{E} < 6.5e-3$

6.4.2 Accuracy of the parameterized reduction algorithm

We have run Algorithm 8 on a collection of 501 RC circuits with 8 parameters. The circuit size varies from 11 to 192. In order to assess the accuracy of the method, we have measured the error in (6.29) for 3000 random drawings in the parameter space. The error histogram is shown in Figure 6-2 where only 1% of cases have an error greater than 1.71e-4 and not a single case has error bigger than 1e-3. The compression ratio is plotted in Figure 6-3, where the mean compression is 0.23.

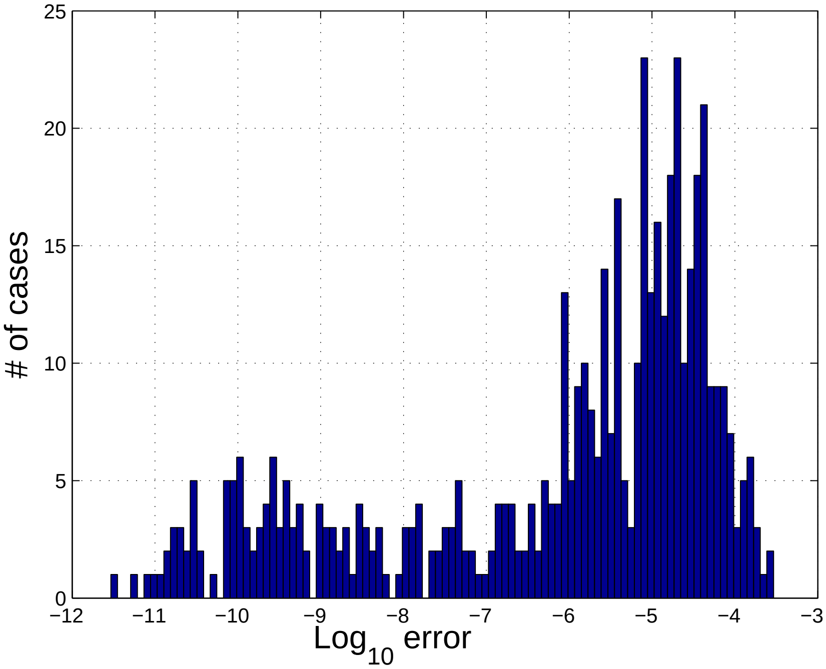

Figure 6-2: Error histogram for Algorithm 8

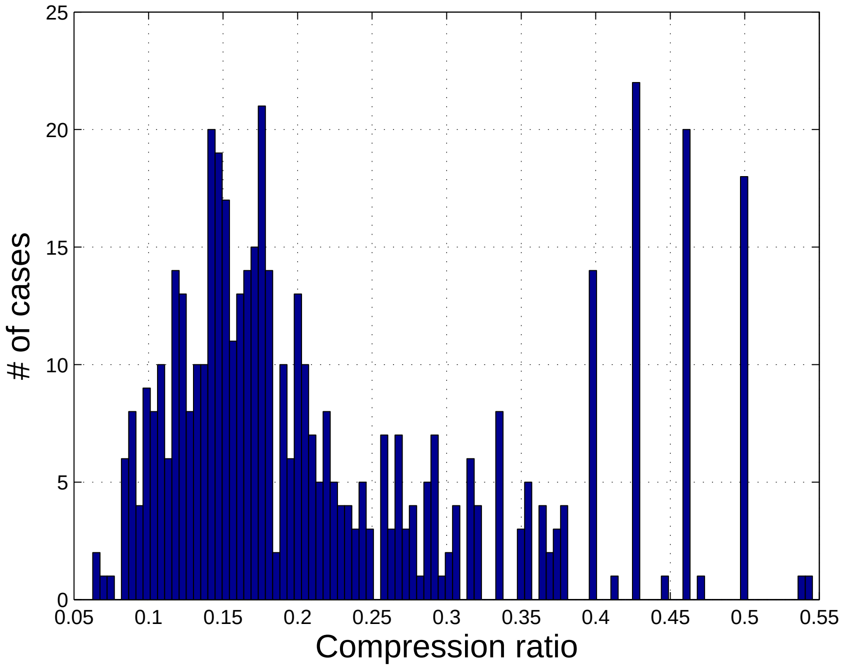

Figure 6-3: Compression ratio for Algorithm 8

Chapter 7

Case study: linear and nonlinear models of microfluidic channel

In this Chapter we illustrate how some of the described reduction methods work on a particular example.

7.1 Model description

The presented example, first suggested in [89], is the injection of a (marker) fluid into a U-shaped three-dimensional microfluidic channel. The carrying (buffer) fluid is driven electrokinetically as depicted in Figure 7-1, and the channel has a rectangular cross-section of height d and width w. In this example, the electrokinetically driven flow of a buffer (carrier) fluid is considered to be steady, with the fluid velocity directly proportional to the electric field as in:

$$\vec{v}(\underbrace{x, y, z}_{\vec{r}}) = -\mu \nabla \Phi(\vec{r}),$$

where μ is an electroosmotic mobility of the fluid. The electric field can be determined from Laplace's equation

$$\nabla^2 \Phi(\vec{r}) = 0,$$

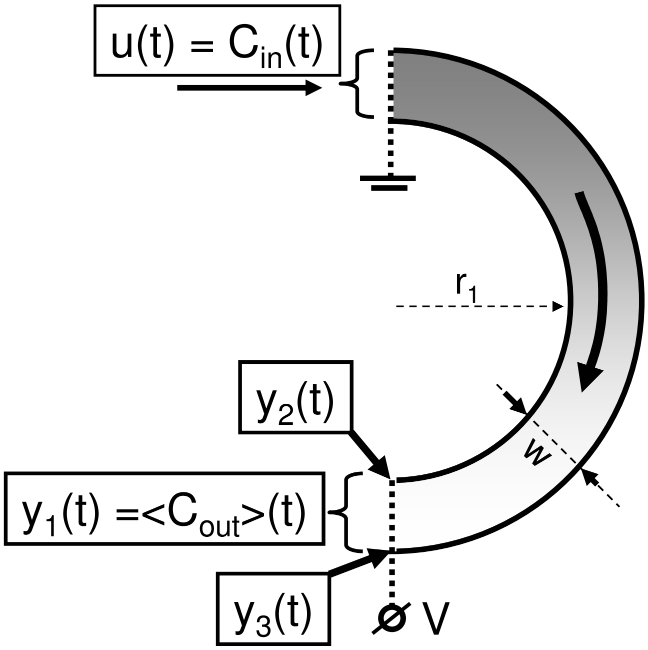

Figure 7-1: The microfluidic channel.

with Neumann boundary conditions on the channel walls [20]. If the concentration of the marker is not small, the electroosmotic mobility can become dependent on the concentration, i.e. $\mu \equiv \mu(C(\vec{r},t))$, where $C(\vec{r},t)$ is the concentration of a marker fluid. Finally, the marker can diffuse from the areas with the high concentration to the areas with low concentration. The total flux of the marker, therefore, is:

$$\vec{J} = \vec{v}C - D\nabla C, \tag{7.1}$$

where D is the diffusion coefficient of the marker. Again, as the concentration of the marker grows, the diffusion will be governed not only by the properties of the carrying fluid, but also by the properties of a marker fluid, therefore D can depend on concentration. Conservation applied to the flux equation (7.1) yields a convection-diffusion equation [42]:

$$\frac{\partial C}{\partial t} = -\nabla \cdot \vec{J} = \nabla\Phi \cdot (C\nabla\mu(C) + \mu(C)\nabla C) + \nabla D(C) \cdot \nabla C + D(C)\nabla^2 C. \tag{7.2}$$

The standard approach is to enforce zero normal flux at the channel wall boundaries, but since $\vec{v}$ has a zero normal component at the walls, zero normal flux is equivalent to enforcing zero normal derivative in C. The concentration at the inlet is determined by the input, and the normal derivative of C is assumed zero at the outlet.

Note that equation (7.2) is nonlinear with respect to marker concentration as long as either electroosmotic mobility or diffusion coefficient is concentration dependent.

A state-space system was generated from (7.2) by applying a second order three-dimensional coordinate-mapped finite-difference spatial discretization to (7.2) on the half-ring domain in Figure 7-1. The states were chosen to be concentrations of the marker fluid at the spatial locations inside the channel. The concentration of the marker at the inlet of the channel is the input signal, and there are three output signals: the first being the average concentration at the outlet, the second and third signals being the concentrations at the inner and outer radii of the outlet of the channel, respectively.

Figure 7-2 illustrates the way an impulse of concentration of the marker at the inlet propagates through the channel: diffusion spreads the pulse, and due to the curvature of the channel, the front of the impulse becomes tilted with respect to the channel's cross-section. That is, the marker first reaches the points at the inner radius (point 1).

7.2 Microchannel - linear model via modified AISIAD

First, in order to demonstrate the effectiveness of TBR linear reduction, we consider applying balanced-truncation algorithm to the linear microchannel model. This corresponds to the problem of a very diluted solution of a marker in the carrier liquid (a widely used approximation in the literature). The values used for the electroosmotic mobility and diffusion coefficients are from [89]: $\mu = 2.8 \times 10^{-8} m^2 V^{-1} s^{-1}$, $D = 5.5 \times 10^{-10} m^2 s^{-1}$. Physical dimensions of the channel were chosen to be $r_1 = 500\mu m$, $w = 300\mu m$, $d = 300\mu m$. Finite-difference discretization led to a linear time-invariant system (A, B, C) of order $N = 2842$ (49 discretization points by

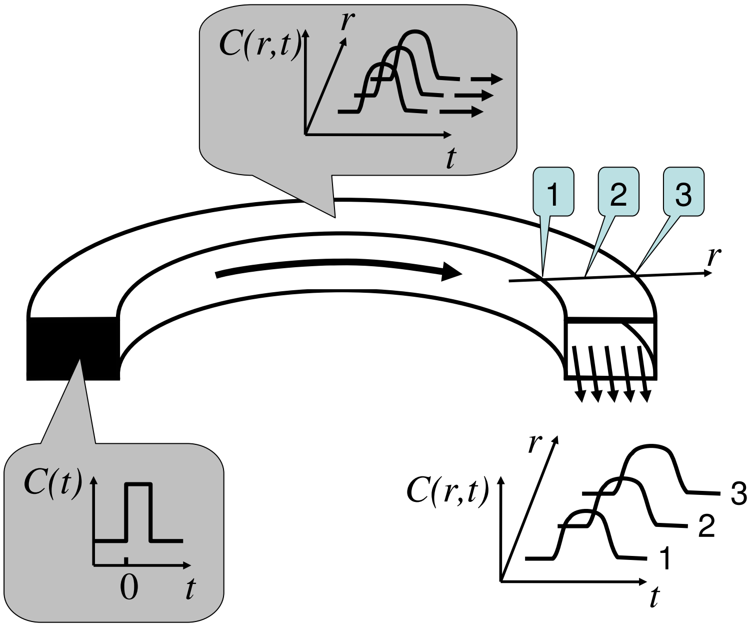

Figure 7-2: Propagation of the square impulse of concentration of the marker through the microfluidic channel. Due to the difference in lengths of the inner and outer arc, the marker reaches different points at the outlet with different delay.

angle, 29 by radius, and 2 by height). Since Algorithm 1 requires $O(n^3)$ computation, the discretized system is quite costly to reduce using original TBR algorithm. We have also used a fast-to-compute approximation to the TBR called modified AISIAD, which we present in Chapter 5.

As shown in Figure 7-3, applying TBR reduction to the spatial discretization of (7.2) with constant diffusion and mobility coefficients demonstrates excellent efficiency of the TBR reduction algorithm. The reduction error decreases exponentially with increasing reduced model order, both in frequency-domain and in time-domain measurements (see also Figure 7-4).

We have also compared the modified AISIAD method with Krylov subspace-based reduction (Arnoldi method [30], described in Section 2.2.2) and the original TBR method in both time and frequency domains. As shown in Figure 7-3, TBR and modified AISIAD are much more accurate than the Krylov method, and are nearly indistinguishable. Though, the modified AISIAD model is much faster to compute.

To demonstrate the time-domain accuracy of the reduced model, we first re-defined

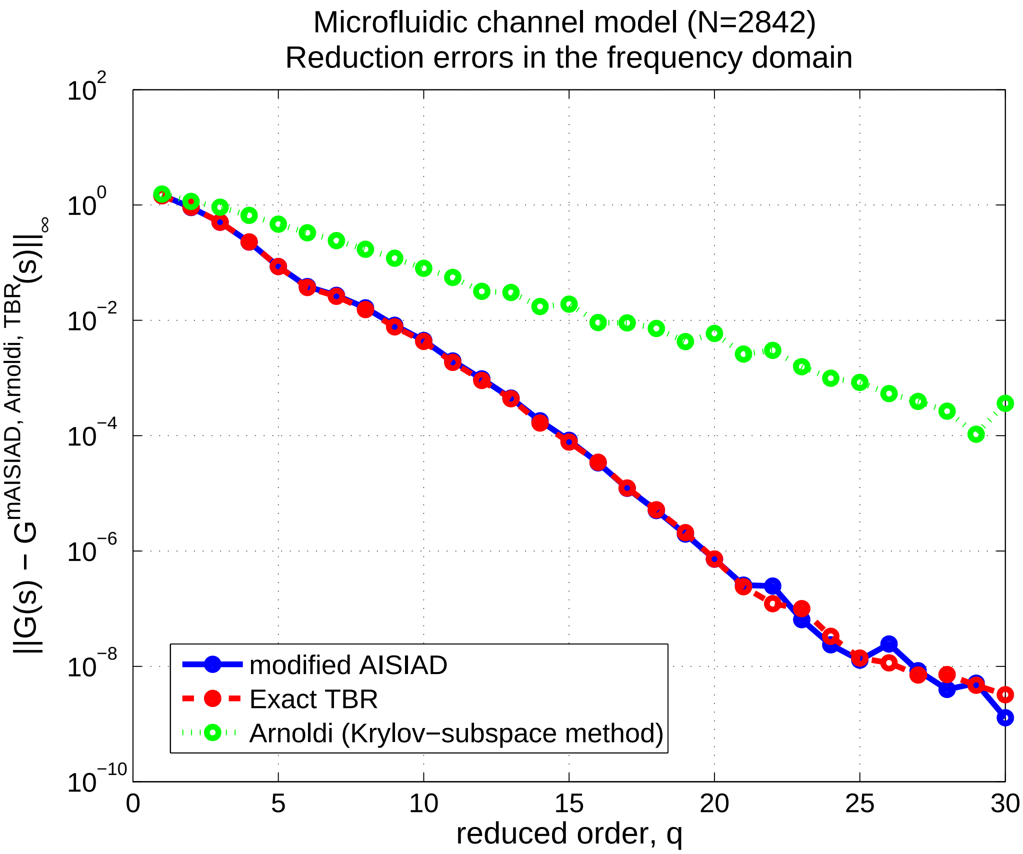

Figure 7-3: H-infinity errors (maximal discrepancy over all frequencies between transfer functions of original and reduced models) for the Krylov, TBR and modified AISIAD reduction algorithms.

the outputs of the model as concentrations at the points 1, 2 and 3 on Figure 7-2, and then performed approximate TBR reduction using the modified AISIAD method.

In Figure 7-4, the output produced by a 0.1 second unit pulse is shown. The results for the 2842 state model and modified AISIAD reduced model of order 13 are compared. One can clearly see that the reduced model nearly perfectly represents different delay values and the spread of the outputs. For example, in the time-domain simulations, the maximum error in the unit step response for the reduced model of order $q = 20$ (over a 100 times reduction) was lower than 10^{-6} for all three output signals.

The runtime comparison between the modified AISIAD approximation of order 30 and TBR model reduction is given in the Table 7.1. One should note that the

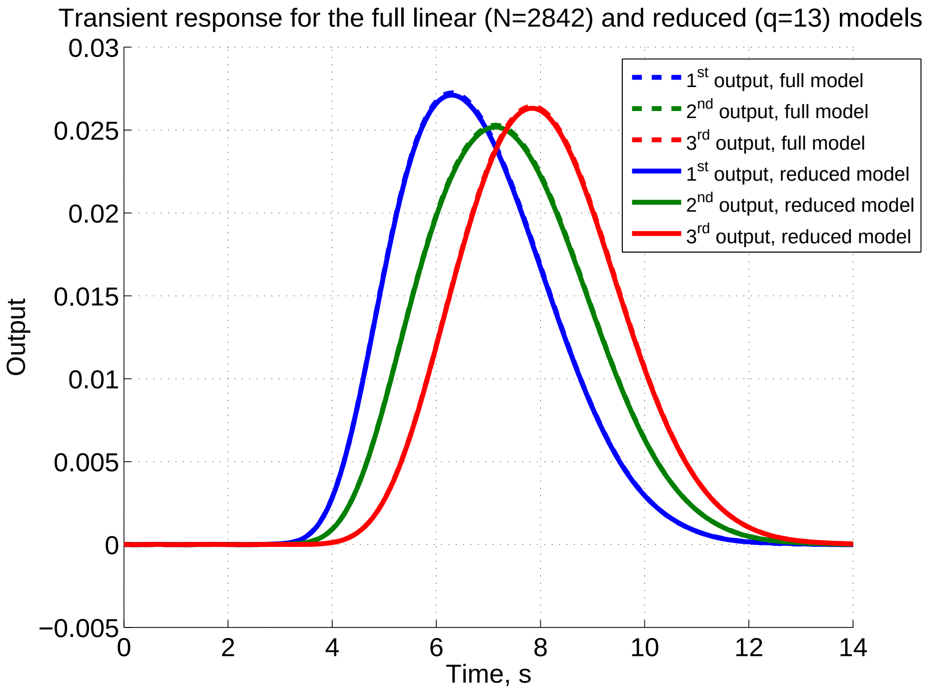

Figure 7-4: Transient response of the original linear (dashed lines) and the reduced by using modified AISIAD (solid lines) model (order $q = 13$). Input signal: unit pulse with duration 0.1 seconds. The maximum error between these transients is $\approx 1 \times 10^{-4}$, therefore the difference is barely visible. The different outputs correspond to the different locations along the channel's outlet (from left to right: innermost point, middle point, outermost point).

complexity of the TBR reduction is proportional to n^3, however the complexity of the modified AISIAD reduction is directly proportional to complexity of the sparse matrix factorization, and for this model it is close to linear (approximately $O(n^{1.2})$).

7.3 Nonlinear microfluidic example via TBR-based TPWL

Consider introducing a mild nonlinearity into the mobility and diffusion coefficients in (7.2):

$$\begin{aligned} \mu(C) &= (28 + C \cdot 5.6) \times 10^{-9} m^2 V^{-1} s^{-1}, \\ D(C) &= (5.5 + C \cdot 1.1) \times 10^{-10} m^2 s^{-1} \end{aligned} \tag{7.3}$$

Table 7.1: Execution times of the modified AISIAD (Chapter 5) of order 30 and TBR reduction algorithms for the linear microfluid model[2]

Model size (# of states)	TBR reduction runtime, s	modified AISIAD runtime, s
1296	212.5	27.9
1421	287.5	31.7
2871	2207.4	72.8

Our experiments showed that even such a small nonlinearity creates a challenging problem for the TPWL algorithm. For this problem, the choice of training input significantly affects the set of the inputs signals for which the reduced model produces accurate outputs. For the case of a pulsed marker, this example has, in effect, a traveling wave solution. Therefore, linearizing at different time-points implies linearizing different spatially local regions of the device, and many linearizations will be needed to cover the entire device.

Our experiments showed that a workable choice of projection matrices V and U for this example is an aggregation of the TBR basis and some of the linearization states x_i. Therefore, the projection used was a mix between TBR and snapshots-based projection [40]. For example, the reduced model whose transient step response is presented in Figure 7-5 was obtained using an aggregation of an order-15 TBR basis and 18 linearization states. The resulting system size was $q = 33$, and the number of linearization points was 23 (the initial model size was $N = 2842$). The linearization points were generated using the same step input for which the reduced simulation was performed. Although the results from the reduced model match when the input is the same as the training input, the errors become quite large if other inputs are used. For these nonlinear wave propagation problems, one needs to use a richer set of training inputs, which will result in a larger set of TPWL linearization points. In addition, instability in this simulation is still an issue, which makes the exact choice of projection basis an ad-hoc procedure.

Remarkably, we have found that the Taylor series-based reduction described in Section 3.1 works much more reliably for this example. We present this result in

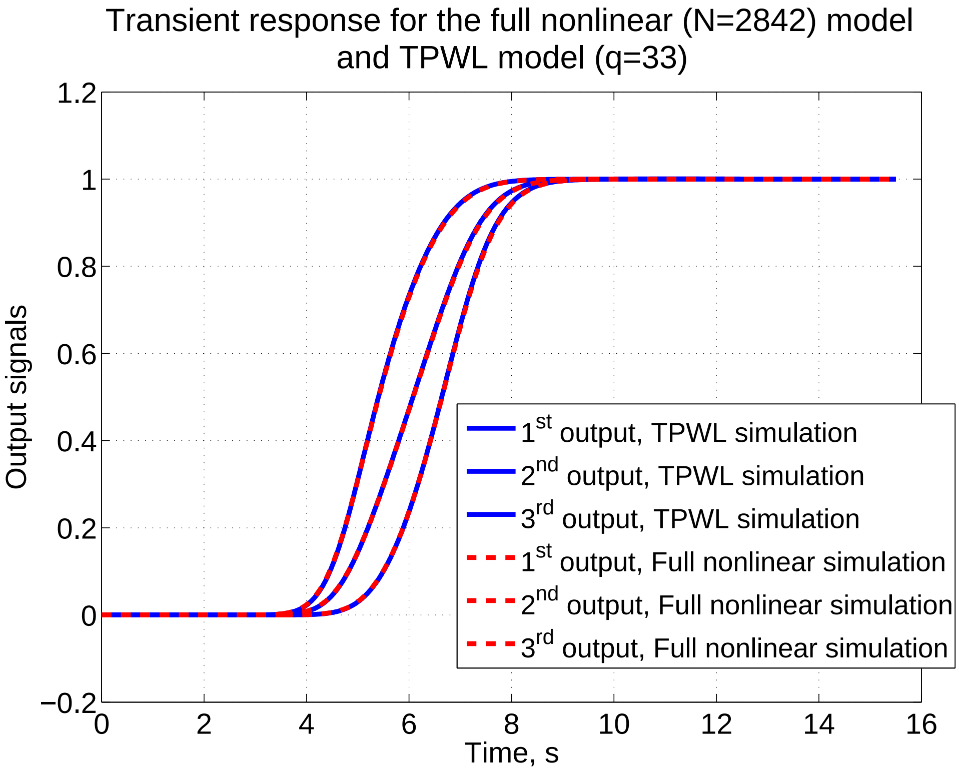

Figure 7-5: Step response of reduced and initial microfluidic model. Solid lines: order-33 TPWL reduced model obtained by using step training input. Dashed lines - full nonlinear model, N=2842. Note: solid and dashed lines almost fully overlap. The leftmost lines is the second input, which corresponds to the concentration closer to the center of the channel's curvature. The middle lines correspond to the first output signal (average concentration at the outlet). The rightmost lines correspond to the concentration at the outlet's points away from the center of curvature.

7.4 Quadratic model reduction as a better alternative for microchannel example

As it was mentioned before, microchannel example provides a challenging case for TPWL reduction algorithm. In fact, the linear models of diffusion and mobility coefficients (7.3) lead to quadratic dynamical model, and therefore we can apply the Taylor series-based reduction by projecting the system's Jacobian and Hessian as described in Section 3.1. An obvious benefit of such approach is that no training trajectories are required for model construction, therefore the model is expected to work for any input signal.

This approach appears to work more reliably for microchannel example. From our observations, using oblique projections leads to unstable models. Models obtained by using orthogonal projections, on the other hand, are almost always stable.

On Figure 7-6 the transient response is plotted for the quadratic reduction method. The projection basis was obtained at $x = 0$ using Arnoldi method ($\mathrm{colsp}(V) = \mathcal{K}_{60}(A^{-1}, A^{-1}B)$, where A denotes Jacobian matrix). As it can be seen, the reduced model approximates the response of the original system quite well; the response which is quite different from the linearized system (dash-dotted line on the graph).

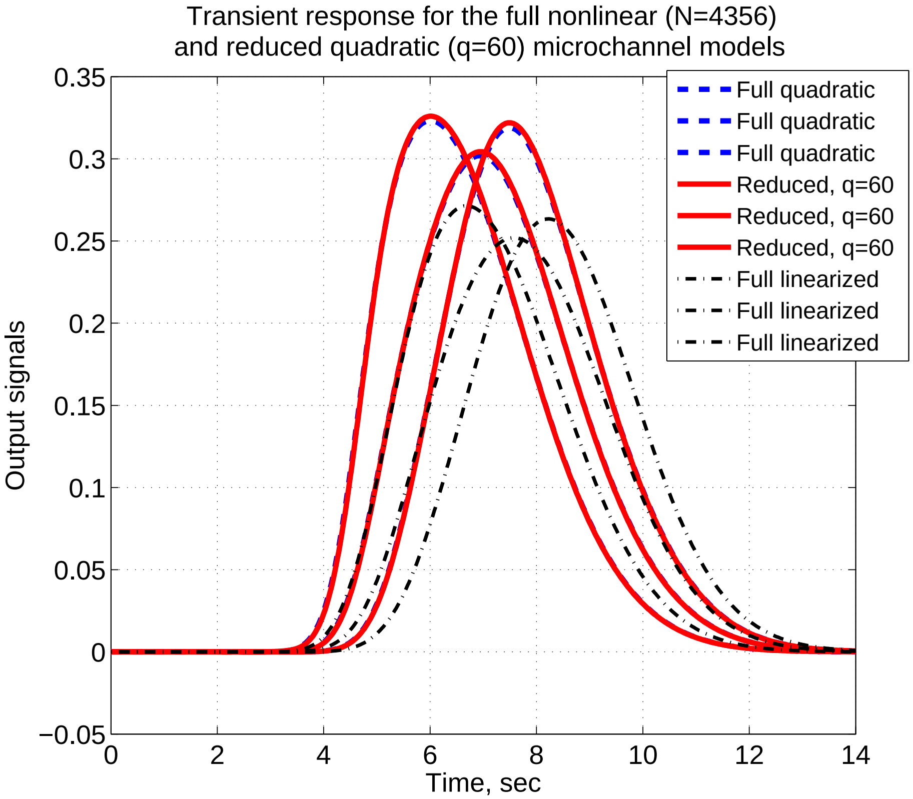

Figure 7-6: Transient response of the original quadratic (dashed lines) model of order $N = 4356$ and the reduced quadratic (solid lines) model (order 60). Input signal: unit pulse of duration 1 second. Projection basis was obtained by using Arnoldi method. The dash-dotted line is the response of the linearized model.

Chapter 8

Conclusions

In the presented work we have made several contributions to the field of model order reduction for linear and nonlinear systems. We have analyzed TBR projection for its applicability to nonlinear TPWL model reduction and we have found analysis based on perturbation theory providing important insight into this problem. We have shown that TBR-based TPWL models provide much more compact macromodels, however certain rules should be used in order to make the reduced models stable. We have found that for nonlinear convection-diffusion models TPWL is less robust than model reduction based on Taylor series. We have proposed and tested a new reduction method called modified AISIAD, which is an iterative approximation to TBR and is applicable to linear systems in descriptor form, for which controllability and observability gramians may not share common dominant eigenspace. In the work which was carried at Cadence Research Laboratories we have also improved TICER model reduction algorithm and showed that it belongs to projection-based model reduction family; in addition we have generalized this method to parameter-dependent RC circuits.

Appendix A

Passivity proof of the graph-based nominal reduction

Below we will show that the Algorithm 1 always produces a *passive* models given a legitimate RC circuit as the input. This means that the reduced model never generates energy, and using such reduced model as a part of a more complex interconnection can never lead to instabilities.

The outline of the proof requires some system-theoretic insight and will proceed in the following steps:

1. Cast the circuit description into an input-output state-space description.

2. Show that each step of the Algorithm 1 is equivalent to projection of the state-space model using some matrix M, which will be explicitly provided.

3. Use the fact that the projection does not change the definiteness of system matrices to establish passivity.

Let's assume that we are given a circuit description (6.1), with possibly negative capacitors. Without loss of generality, and in accordance with the notation in Eq. (6.1), we consider the input signals to be the currents of "current sources" that are connected to the external nodes of the circuit. With some abuse of terminology we call these "port" nodes. In reality the "port" nodes could be connected to any circuitry, depending on how the reduced circuit is used, and there could be very many of them. For example, we might need these nodes to represent capacitive couplings to other networks. We define the system's outputs as the voltages at the port nodes. This way, we can re-write the circuit's input-output relationships in the following *state space* form of ordinary differential equation:

$$\begin{cases} C\dot{v}(t) = -Gv(t) + Bi(t) \\ u(t) = B^T v(t) \end{cases}, \tag{A.1}$$

where $v(t)$ are node voltages, $i(t)$ are port currents, $u(t)$ are port voltages, and the matrix B of size (# of nodes × # of ports) maps port numbers to the circuit nodes:

$$b_{ij} = \begin{cases} 1, & \text{if current source } j \text{ is connected to node } i, \\ 0, & \text{otherwise} \end{cases}$$

In the description (A.1) we assume that the variable corresponding to the ground node has been removed, this way implicitly assigning zero potential to the ground node. In the following derivations we make an assumption that the matrices C and G do not have a common kernel vector, that is, there is no such vector v that $Cv = Gv = 0$. If C and G share at least one common kernel vector, this would mean that a given circuit consists of several completely separate subnetworks (not connected neither by resistors nor by capacitors). The analysis below is not directly applicable for this case, because any finite current into such a subnetwork may result in infinitely large node voltages; however, the passivity result holds even for such case, by applying the presented results for each subnetwork.

The *transfer function* $H(s)$ of the system (A.1) provides an algebraic relationship between input and output signals in the frequency domain, and is equal to the impedance of the circuit:

$$U(s) = H(s)I(s), \quad H(s) = B^T(sC + G)^{-1}B,$$

where $I(s)$ and $U(s)$ are Laplace transforms of the port currents and voltages, respectively. This way, $H(s)$ is a complex matrix-valued function of a complex frequency s.

It is known that the described system is passive if and only if the transfer function $H(s)$ satisfies the following three properties [87]:

1. $H(\overline{s}) = \overline{H(s)}$ for all s,

2. $H(s)$ is analytic on the open right half of the complex plane $\Re(s) > 0$,

3. $H(s)$ is positive-semidefinite matrix for $\Re(s) > 0$, that is,

$$z^*(H(s) + H^*(s))z \geq 0, \quad \forall s, \text{ such that } \Re(s) > 0, \, \forall z \in \mathbb{C}^p, \tag{A.2}$$

where the "*" is a complex-conjugate transpose operator, and p denotes the number of ports.

The following statement, proved for a broader class in [57], will be used to establish passivity:

Lemma A.0.1. *If in the description (A.1) the matrices C and G are real symmetric positive-semidefinite, do not share common kernel vectors and B is real, then such system is passive.*

Proof. Since matrices C, G and B are real, the condition 1) above is satisfied.

To show that the condition 2) holds, it suffices to show that the matrix $(sC + G)$ is always invertible for all s having positive real part. Let's consider otherwise; then there exist such s_0 with $\Re(s_0) > 0$ and vector v_0, such that

$$(s_0 C + G)v_0 = 0, \quad \Rightarrow \quad v_0^*(s_0 C + G)v_0 = 0$$

Taking a complex-conjugate transpose of the expression on the left, and multiplying from the right by v_0, we have:

$$v_0^*(s_0^* C + G)v_0 = 0.$$

Comparing two last equalities, we conclude that s_0 is real and v_0 is a real vector. We have:

$$v_0^T(s_0C + G)v_0 = s_0 \underbrace{v_0^T C v_0}_{\geq 0} + \underbrace{v_0^T G v_0}_{\geq 0} = 0.$$

Since both C and G are positive-semidefinite, the equality above cannot hold for positive s_0. This is a contradiction. Therefore, the condition 2) is satisfied.

Checking the condition 3) is straightforward:

$$\begin{aligned} z^*(H(s) + H^*(s))z &= (Bz)^*((sC + G)^{-1} + (s^*C + G)^{-1})(Bz) = \\ &= ((sC + G)^{-1}Bz)^*(s^*C + G + sC + G)((sC + G)^{-1}Bz)) \geq 0, \end{aligned}$$

provided $\Re(s) > 0$. This proves the condition 3). □

Let us now consider a single elimination step of the Algorithm 1. Again, for simplicity let the last node N be subjected to elimination.

Let the circuit description before the elimination be in the form (A.1), which is equivalent to (6.1), and the obtained state-space model after elimination becomes

$$\begin{cases} \hat{C}\dot{\hat{v}}(t) = -\hat{G}\hat{v}(t) + \hat{B}i(t) \\ \hat{u}(t) = \hat{B}^T\hat{v}(t) \end{cases}, \tag{A.3}$$

which correspond to the system in (6.10).

The following Lemma provides a connection between the Algorithm 1 and projection-based reduction methods, and is the key to establishing the passivity of the reduced system.

Lemma A.0.2. *Each node elimination step in Algorithm 7 is equivalent to imposing a projection on the system matrices C, G and B in (A.1):*

$$\hat{C} = M^TCM, \quad \hat{G} = M^TGM, \quad \hat{B} = M^TB, \tag{A.4}$$

Proof. Let us define the following matrix:

$$M = \begin{bmatrix} I_{(N-1)\times(N-1)} \\ \\ \mathbf{g}_N^T / G_{NN} \end{bmatrix}, \tag{A.5}$$

where $I_{(N-1)\times(N-1)}$ is an identity matrix of size $(N-1)\times(N-1)$. Using partitioned forms of matrices C and G from (6.3), we can easily verify that

$$\begin{aligned} M^T C M = \begin{bmatrix} I & \mathbf{g}_N / G_{NN} \end{bmatrix} \begin{bmatrix} \tilde{C} & -\mathbf{c}_N \\ -\mathbf{c}_N & sC_{NN} \end{bmatrix} \begin{bmatrix} I \\ \mathbf{g}_N^T / G_{NN} \end{bmatrix} = \\ = \tilde{C} - \frac{\mathbf{c}_N \mathbf{g}_N^T + \mathbf{g}_N \mathbf{c}_N^T}{G_{NN}} + \frac{C_{NN}}{G_{NN}^2} \mathbf{g}_N \mathbf{g}_N^T, \end{aligned}$$

which is the same expression as (6.11). The equivalence of $\hat{G} = M^T G M$ and (6.12) can be shown analogously.

Since the node N is not the port node, $\hat{B}$ is a sub-matrix of B with the last row excluded, wherefore $\hat{B} = M^T B$ holds as well. □

(Note that, after reduction finishes, we can take $\hat{B} = I$, i.e. every remaining node can be connected to external circuitry, as is usual for a reduced circuit, without affecting any of the theorems in this section.)

The main result directly follows.

Corollary A.0.1. *The system description resulting from Algorithm 1 is always passive.*

Proof. One can note that for the original (unreduced) circuit in (6.1) the conditions of the Lemma A.0.1 are satisfied, because matrices C and G are diagonally dominant, with positive diagonals. At each step, due to (A.4), the system matrices are projected as

$$\hat{G} = M^T G M, \quad \hat{C} = M^T C M,$$

which is a congruence transform, and therefore matrices $\hat{G}$ and $\hat{C}$ remain positive semidefinite. By induction, the system is passive after every elimination step of the Algorithm 1. $\square$

The passivity proof is now complete.

Bibliography

[1] Oberwolfach model reduction benchmark collection, http://www.imtek.uni-freiburg.de/simulation/benchmark/.

[2] J.A. De Abreu-Garcia A.A.Mohammad. A transformation aproach for model order reduction of nonlinear systems. In *Proceedings of the 16th Annual conference of IEEE Industrial electronics Society*, volume 1, pages 380–383, 1990.

[3] A.C.Antoulas, D.C.Sorensen, and S.Gugercin. A modibed low-rank smith method for large-scale lyapunov equations. Technical report, Rice University, 2001.

[4] A.C.Antoulas, D.C.Sorensen, and S.Gugercin. A survey of model reduction methods for large-scale systems. *Structured Matrices in Operator Theory, Numerical Analysis, Control, Signal and Image Processing, Contemporary Mathematics*, 280:193–219, 2001.

[5] R.L.; Emami-Naeini A.; Ebert J.L. Aling, H.; Kosut. Nonlinear model reduction with application to rapid thermal processing. In *Proceedings of the 35th IEEE Decision and Control, 1996*, volume 4, pages 4305–4310, 1996.

[6] Chirayu S. Amin, Masud H. Chowdhury, and Yehea I. Ismail. Realizable RLCK circuit crunching. pages 226–231, Anaheim, CA, 2003. ACM Press.

[7] A. C. Antoulas and Dan C Sorensen. The sylvester equation and approximate balanced reduction. *Linear Algebra and It's Applications, Fourth Special Issue on Linear Systems and Control*, pages pp. 351–352, 671–700, 2002.

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

[9] Z. Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Applied Numerical Mathematics*, 43(1-2):9 – 44, 2002.

[10] Moore B.C. Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Trans. Autom. Control*, AC-26(1):17–32, 1981.

[11] Gal Berkooz, Philip Holmes, and John L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Ann. Rev. Fluid Mech.*, 25:539–75, 1993.

[12] B. Bond and L. Daniel. Parameterized model order reduction of nonlinear dynamical systems. In *Proceedings on Computer-Aided Design, 2005. ICCAD-2005. IEEE/ACM International Conference*, pages 487–494, 2005.

[13] J. Chen and 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*, volume 2, pages 445–448, 2000.

[14] Y. Chen and J. White. A quadratic method for nonlinear model order reduction. In *Proceedings of the International Conference on Modeling and Simulation of Microsystems*, pages 477 – 480, 2000.

[15] Yong Chen. *Model order reduction for nonlinear systems.* PhD thesis, Massachusetts Institute of Technology, 1999.

[16] Carlos P. Coelho, Joel R. Phillips, and L. Miguel Silveira. Robust rational function approximation algorithm for model generation. In *DAC '99: Proceedings of the 36th ACM/IEEE conference on Design automation*, pages 207–212, New York, NY, USA, 1999. ACM Press.

[17] C.P. Coelho, Phillips J., and Silveira L.M. A convex programmin approach for generating guaranteed passive approximations to tabulated frequency-data. *IEEE Trans. CAD*, 23(2):293–301, 2004.

[18] Mohammed Dahleh, Munther Dahleh, and George Verghese. Lectures on dynamic systems and control. Department of Electrical Engineering and Computer Science, MIT.

[19] L. Daniel, Ong Chin Siong, L.S. Chay, Kwok Hong Lee, and J. White. A multiparameter moment-matching model-reduction approach for generating geometrically parameterized interconnect performance models. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 23(5):678– 693, 2004.

[20] Ning Dong and Jaijeet Roychowdhury. Piecewise polynomial nonlinear model reduction. In *DAC '03: Proceedings of the 40th conference on Design automation*, pages 484–489, New York, NY, USA, 2003. ACM.

[21] S. K. Griffiths E. B. Cummings and R. H. Nilson. Irrotationality of uniform electroosmosis. In *SPIE Conference on Microfluidic Devices and Systems II (Santa Clara, CA)*, 1999.

[22] P. J. H. Elias and N. P. van der Meijs. Including higher-order moments of RC interconnections in layout-to-circuit extraction. In *IEEE European Design and Test Conference, March 1996*, page 362, March 1996.

[23] Dale Enns. *Model Reduction for Control System Design.* PhD thesis, Stanford University, 1984.

[24] K. V. Fernando and H. Nicholson. On the cross-gramian for symmetric mimo systems. *IEEE Transactions on Circuits and Systems*, 32(5):487–489, 1985.

[25] Emad Gad and Michel Nakhla. Model order reduction of nonuniform transmission lines using integrated congruence transform. In *DAC '03: Proceedings of the 40th conference on Design automation*, pages 238–243, New York, NY, USA, 2003. ACM Press.

[26] K. Gallivan, A. Vandendorpe, and P. Van Dooren. Sylvester equations and projection-based model reduction. *J. Comput. Appl. Math.*, 162(1):213–229, 2004.

[27] Felix R. Gantmacher. *Matrix Theory Vol. 1.* American Mathematical Society, 1990.

[28] Alan George, John R. Gilbert, and Joseph W.H. Liu. *Graph Theory and Sparse Matrix Computation.* Springer, 1993.

[29] Keith Glover. All optimal hankel-norm approximations of linear multivariable systems and their l^{∞} -error bounds. *International Journal of Control*, 39(6):1115–1193, 1984.

[30] E. J. Grimme, D. C. Sorensen, and P. Van Dooren. Model reduction of state space systems via an implicitly restarted Lanczos method. *Numerical Algorithms*, 12(1–2):1–31, 1996.

[31] Eric Grimme. *Krylov Projection Methods for Model Reduction.* PhD thesis, Coordinated-Science Laboratory, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, 1997.

[32] S. Gugercin, D. C. Sorensen, and A. C. Antoulas. A modified low-rank smith method for large-scale lyapunov equations. Technical report, Rice University, 2002.

[33] Serkan Gugercin and Athanasios C. Antoulas. A survey of model reduction by balanced truncation and some new results. *International Journal of Control*, 77:748–766(19), May 20, 2004.

[34] B. Gustavsen and A. Semlyen. Rational approximation of frequency domain responses by vector fitting. *IEEE Trans. Power Delivery*, 14(3):1052–1061, 1999.

[35] B. Gustavsen and A. Semlyen. Enforcing passivity for admittance matrices approximated by rational functions. *IEEE Trans. power systems*, 16(1):97–104, 2001.

[36] A. Scottedward Hodel, Bruce Tenison, and Kameshwar Poolla. Numerical solution of large Lyapunov equations by Approximate Power Iteration. 236:205–230, 1996.

[37] E. Hung, Y.Yang, and S. Senturia. Low-order models for fast dynamical simulation of mems microstructures. In *Proc. of IEEE Int. Conf. on Solid State Sensors and Actuators*, 1997.

[38] P.; Donnay S.; Tilmans H.A.C.; Sansen W.; De Man H. Innocent, M.; Wambacq. An analytic volterra-series-based model for a mems variable capacitor. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 22(2):124–131, 2003.

[39] Luca Daniel Jung Hoon Lee and Jacob White. Formalization of the moment matching graph idea. Technical report, Research Laboratory of Electronics, MIT, 2004.

[40] J. Peraire K. C. Hall K. E. Willcox, J. D. Paduano. Low order aerodynamic models for aeroelastic control of turbomachines. *American Institute of Aeronautics and Astronautics Paper 99-1467*, page 1999.

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

[42] K.Willcox, J.Peraire, and J.White. An arnoldi approach for generation of reduced-order models for turbomachinery. *Computers & Fluids*, 31(3):369–89, 2002.

[43] L.D. Landau and E.M. Lifshitz. *Fluid Mechanics*, volume 6. Butterworth-Heinemann, 1977.

[44] L.D. Landau and E.M. Lifshitz. *Quantum Mechanics: Non-Relativistic Theory*, volume 3, chapter 36, pages 133–137. Butterworth-Heinemann, 1977.

[45] Jing-Rebecca Li. *Model Reduction of Large Linear Systems via Low Rank System Gramians*. PhD thesis, Massachusetts Institute of Technology, 2000.

[46] Jing-Rebecca Li, Frank Wang, and Jacob K. White. An efficient lyapunov equation-based approach for generating reduced-order models of interconnect. In *Design Automation Conference*, pages 1–6, 1999.

[47] X. Li, P. Li, and L. Pileggi. Parameterized interconnect order reduction with Explicit-and-Implicit multi-Parameter moment matching for Inter/Intra-Die variations. pages 806–812, San Jose, CA, November 2005.

[48] S. Lin and E. S. Kuh. Transient simulation of lossy interconnects based on the recursive convolution formulation. *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, 39(11):879–892, 1992.

[49] Y. Liu, L. T. Pileggi, and A. J. Strojwas. Model order reduction of RC(L) interconnect including variational analysis. pages 201–206, June 1999.

[50] Yi Liu and Brian D.O.Anderson. Singular perturbation approximation of balanced systems. *International Journal of Control*, 50(4):1379–1405, 1989.

[51] J. White M. Kamon, F. Wang. Generating nearly optimally compact models from krylov-subspace based reduced-order models. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 47(4):239–248, 2000.

[52] James D. Ma and Rob A. Rutenbar. Interval-valued reduced order statistical interconnect modeling. pages 460–467, San Jose, CA, November 2004.

[53] Steven P McCormick. *Modeling and simulation of VLSI interconnections with moments*. PhD thesis, Massachusetts Institute of Technology, 1989.

[54] Alexandre Megretski. Lecture notes on model order reduction, mit, 2004.

[55] M.E.Kowalski and J-M. Jin. Karhunen-loéve based model order reduction of nonlinear systems. In *Proceedings of the IEEE Custom Integrated Circuits Conference*, 1988.

[56] E.M. Abdel-Rahman M.I. Younis and Ali Nayfeh. A reduced-order model for electrically actuated microbeam-based mems. *Journal of MEMS*, 12(5):672–680, 2003.

[57] A. Odabasioglu, M. Celik, and L. T. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. 17(8):645–654, August 1998.

[58] Altan Odabasioglu, Mustafa Celik, and Lawrence T. Pileggi. Prima: passive reduced-order interconnect macromodeling algorithm. In *ICCAD*, pages 58–65, 1997.

[59] Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. *Signals & systems (2nd ed.).* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.

[60] Thilo Penzl. A cyclic low-rank smith method for large sparse lyapunov equations. *SIAM Journal on Scientific Computing*, 21(4):1401–1418, 1999.

[61] L. Pernebo and L. Silverman. Model reduction via balanced state space representations. *Automatic Control, IEEE Transactions on*, 27(2):382–387, 1982.

[62] P.Feldman and R. Freund. Efficient linear circuit analysis by padé approximation via lanczos process. *IEEE Transactions on Computer-Aided Design*, 14(5):639 – 649, 1995.

[63] J. Phillips, L. Daniel, and M. Silveira. Guaranteed passive balancing transformations for model order reduction. In *DAC '02: Proceedings of the 39th conference on Design automation*, 2002.

[64] Joel Phillips, ao Afonso Jo Arlindo Oliveira, and L. Miguel Silveira. Analog macromodeling using kernel methods. In *ICCAD '03: Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design*, page 446, Washington, DC, USA, 2003. IEEE Computer Society.

[65] Joel Phillips and L. Miguel Silveira. Poor man's tbr: A simple model reduction scheme. In *Proceedings of the conference on Design, automation and test in*

Europe, volume 02, page 20938, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

[66] Joel R. Phillips. Projection frameworks for model reduction of weakly nonlinear systems. In *DAC '00: Proceedings of the 37th conference on Design automation*, pages 184–189, New York, NY, USA, 2000. ACM Press.

[67] Joel R. Phillips. Variational interconnect analysis via PMTBR. pages 872–879, San Jose, CA, November 2004.

[68] M. Rewieński and 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*, pages 252–257, 2001.

[69] M. Rewieński and 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*, pages 128–131, 2002.

[70] M. Rewieński and 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*, 22(2):155–170, 2003.

[71] Michał Rewieński. *A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems.* PhD thesis, Massachusetts Institute of Technology, 2003.

[72] J. E. Marsden S. Lall and S. Glavaski. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International Journal of Robust and Nonlinear Control*, 12(6):519–535, 2002.

[73] Lawrence F. Shampine and Mark W. Reichelt. The matlab ode suite. *SIAM J. Sci. Comput.*, 18(1):1–22, 1997.

[74] Bernard N. Sheehan. TICER: realizable reduction of extracted RC circuits. pages 200–203, San Jose, CA, November 1999.

[75] D. C. Sorensen. Implicit application of polynomial filters in a k-step arnoldi method. *SIAM J. Matrix Anal. Appl.*, 13(1):357–385, 1992.

[76] D.C. Sorensen and A.C. Antoulas. Projection methods for balanced model reduction. Technical report, Rice University, 2001.

[77] Kin Cheong Sou, Alexandre Megretski, and Luca Daniel. A quasi-convex optimization approach to parameterized model order reduction. In *DAC '05: Proceedings of the 42nd annual conference on Design automation*, pages 933–938, New York, NY, USA, 2005. ACM Press.

[78] Tatjana Stykel. *Analysis and numerical solution of generalized Lyapunov equations*. PhD thesis, Technischen universität Berlin, 2002.

[79] Tatjana Stykel. Gramian-based model reduction for descriptor systems. *Mathematics of Control, Signals, and Systems (MCSS)*, 16(4):297–319, 2004.

[80] S. Thompson and L. F. Shampine. A friendly fortran dde solver. *Appl. Numer. Math.*, 56(3):503–516, 2006.

[81] S.K. Tiwary and R.A. Rutenbar. Scalable trajectory methods for on-demand analog macromodel extraction. In *Proceedings of the 42nd annual conference on Design automation*, pages 403–408, 2005.

[82] A.J. van Genderen and N.P. van der Meijs. Reduced RC models for IC interconnections with coupling capacitances. pages 132–136, Brussels, Belgium, March 1992.

[83] D. Vasilyev, M. Rewieński, and J. White. Perturbation analysis of tbr model reduction in application to trajectory-piecewise linear algorithm for mems structures. In *Proceedings of the 2004 NSTI Nanotechnology Conference*, volume 2, pages 434 – 437, 2004.

[84] D. Vasilyev, Zhenhai Zhu, and Joel Phillips. A graph-based parametrized model order reduction method. Technical report, Cadence Berkeley Labs, Berkeley, CA, 2006.

[85] Dmitry Vasilyev, Michał Rewieński, and Jacob 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 conference on Design automation*, pages 490–495. ACM Press, 2003.

[86] J. Wang, P. Ghanta, and S. Vrudhula. Stochastic analysis of interconnect performance in the presence of process variations. pages 880–886, San Jose, CA, November 2004.

[87] J. C. Willems. Dissipative dynamical systems. *Arch. Rational Mechanics and Analysis*, 45:321–393, 1972.

[88] John Wyatt. Lectures on dynamics of nonlinear systems, fall 2005.

[89] D. Djukic V. Modi A.C.West J.Yardley Z. Tang, S. Hong and R.M.Osgood. Electrokinetic flow control for composition modulation in a microchannel. *Journal of Micromechanics and Microengineering*, 12(6):870–877, 2002.

[90] Yunkai Zhou. *Numerical methods for large-scale matrix equations with applications in LTI System Model reduction.* PhD thesis, Rice University, 2002.