Lehrstuhl für Regelungstechnik

Technische Universität München

Model Order Reduction by Krylov Subspace Methods with Global Error Bounds and Automatic Choice of Parameters

Heiko K. F. Panzer

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr.-Ing. Michael W. Gee
Prüfer der Dissertation:	1.	Univ.-Prof. Dr.-Ing. habil. Boris Lohmann
	2.	Univ.-Prof. Athanasios C. Antoulas, Ph. D. Universität Bremen

Die Dissertation wurde am 06.05.2014 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 15.09.2014 angenommen.

Abstract

This thesis presents rigorous global error bounds and automatic shift selection strategies in model order reduction of linear time-invariant systems by Krylov subspace methods.

The spatial discretization of partial differential equations, which can describe dynamic systems in various engineering domains, often leads to very large systems of ordinary differential equations, whose number increases with the demands on the accuracy of the model. To perform tasks like simulation, control, and optimization while complying with given limitations of available storage and time, a simplification of the model is therefore frequently inevitable. Numerous methods for this purpose have been described in the literature, which exhibit specific advantages and disadvantages. Krylov subspace methods, which are in the focus of this work, require comparably little numerical effort and are therefore practical for the reduction of very large models. However, they do not necessarily preserve stability of the model, nor do they provide information on the approximation quality, and they require the judicious choice of certain parameters, the so-called expansion points (or shifts), and of the order of the reduced model.

Starting from a novel formulation of the approximation error that results from the reduction, new approaches to these problems are presented. A cumulative reduction procedure, during which the reduced model is set up iteratively instead of all at a sudden, enables the adaptive choice of the reduced order. The shift selection is accomplished by optimization and leads to a descent method which yields optimal expansion points after a very small number of steps. Also, global error bounds for a class of state space models are introduced; their overestimation is faced by suitable modifications of the mentioned optimization problem. Finally, it is shown how the proposed methods can be applied efficiently to many second order systems.

Case studies including models from structural mechanics, electrothermics, and acoustics show the effectiveness of the presented methods.

Zusammenfassung

Die vorliegende Arbeit stellt rigorose Fehlerschranken und Verfahren zur automatischen Entwicklungspunktwahl bei der Modellordnungsreduktion linearer, zeitinvarianter Systeme mittels Krylow-Unterraum-Methoden vor.

Die örtliche Diskretisierung partieller Differentialgleichungen, welche zur Beschreibung dynamischer Systeme in diversen ingenieurwissenschaftlichen Bereichen zum Einsatz kommen, führt meist zu sehr großen Systemen gewöhnlicher Differentialgleichungen, deren Anzahl mit steigenden Ansprüchen an die Modellgenauigkeit zunimmt. Zur Erfüllung von Simulations-, Regelungs- oder Optimierungsaufgaben ist eine Vereinfachung des Modells daher oft unumgänglich; hierzu wurden zahlreiche Methoden mit spezifischen Vor- und Nachteilen beschrieben. Krylow-Unterraum-Methoden, die im Zentrum dieser Arbeit stehen, erfordern verhältnismäßig geringen numerischen Aufwand und sind daher zur Reduktion auch sehr großer Modelle geeignet. Allerdings erhalten sie nicht zwangsläufig die Stabilität des Modells, bieten keine Information über die Reduktionsgüte und erfordern die günstige Wahl gewisser Parameter, der sogenannten Entwicklungspunkte ("Shifts") sowie der Ordnung des reduzierten Modells.

Ausgehend von einer neuen Formulierung des Fehlersystems werden neue Zugänge zu diesen Problemstellungen aufgezeigt. Ein kumulatives Reduktionsvorgehen, währenddessen das reduzierte Modell iterativ aufgebaut wird, ermöglicht die adaptive Wahl der reduzierten Ordnung und der Entwicklungspunkte. Letztere erfolgt mittels Optimierung in einem Abstiegsverfahren, das oft nur wenige Schritte benötigt. Schließlich werden globale Fehlerschranken für eine Klasse von Zustandsraummodellen eingeführt; der verursachten Überschätzung wird durch Umformulierung des Optimierungsproblems begegnet.

Die vorgestellten Methoden können z. B. effizient auf viele Systeme zweiter Ordnung angewandt werden. Fallstudien anhand von Modellen aus der Strukturmechanik, Elektrothermik, Akustik u. a. belegen ihre Effektivität.

Danksagung / Acknowledgment

An erster Stelle gilt mein Dank meinem Doktorvater Herrn Prof. Boris Lohmann, der mir die wissenschaftliche Tätigkeit an seinem Lehrstuhl ermöglichte und dessen Türe mir in dieser Zeit mit meinen vielfältigen Anliegen stets offen stand. Seine jahrelange Erfahrung im Bereich der Modellreduktion und sein fundiertes systemtheoretisches Wissen lenkten diese Arbeit von Beginn an in geeignete Bahnen und legten den Grundstein für die späteren Ergebnisse.

I am also much obliged to Prof. Athanasios C. Antoulas for his commitment as second referee. Herrn Prof. Michael W. Gee gilt mein Dank für die Übernahme des Vorsitzes der Prüfungskommission.

Auch danke ich Herrn Prof. Timo Reis, der mich im Rahmen des Mentorats der TUM Graduate School unterstützte und stets ein offenes Ohr für meine fachlichen und sonstigen Fragen hatte.

Mein tiefer Dank gilt auch meinem ehemaligen Kollegen Dr. Rudy Eid, der mich im Rahmen meiner Diplomarbeit in das Feld der Modellreduktion eingeführt und noch während meiner Zeit als Doktorand geduldig betreut und begleitet hat. Sein Einsatz für mich und andere am Lehrstuhl ist mir ebenso unvergessen wie die einmalige Atmosphäre in unserem “MORLAB”.

Unermesslich bleibt auch der Beitrag, den mein Kollege Thomas Wolf zu dieser Arbeit geleistet hat. Seit den ersten Monaten der Promotion war er mir am Lehrstuhl ein treuer Begleiter, hat in stundenlangen fachlichen Diskussionen die Ergebnisse dieser Arbeit maßgeblich mitgestaltet und mich durch seine Gewissenhaftigkeit vor manchem vorschnellen Fehlschluss bewahrt.

Auch Frau Dr. Rosa Castañé Selga bin ich zu Dank verpflichtet, denn durch ihre Arbeit kam ich erstmals mit dem Konzept der Dissipativität in Berührung. Herrn

Alessandro Castagnotto und Frau Maria Cruz Varona danke ich vielmals für das sorgfältige und kritische Korrekturlesen dieser Arbeit.

Ebenso hat das restliche Kollegium am Lehrstuhl für Regelungstechnik auf vielfältige Weise zum Gelingen dieser Arbeit beigetragen; mein Dank gilt den vielen ehemaligen und verbliebenen Kollegen, die mich während der letzten Jahre unterstützt oder sich für die Gruppe insgesamt eingesetzt haben.

I also thank Dr. Jens Saak (and the whole MORWiki Group), Prof. Jeong Sam Han, Dr. Jan Lienemann, Prof. Evgenii Rudnyi, Tobias Hummel, and Stefan Jaensch for providing and/or helping me with benchmark models. I am also grateful to Dr. Ortwin Farle who helped me with the implementation of existing error bounds techniques, and to Prof. Zhaojun Bai for his support with the literature on error estimation.

Für die finanzielle Unterstützung im Rahmen der Promotionsförderung sowie für die vielen bereichernden Erfahrungen und Begegnungen, die mit diesem Privileg verbunden waren, bedanke ich mich ganz herzlich beim Cusanuswerk und seinen Mitarbeiter(inne)n sowie meinen ehemaligen Konstipendiat(inn)en.

Die Leistung der OpenSource-Gemeinde rund um LaTeX und verwandte Tools soll an dieser Stelle nicht unerwähnt bleiben.

Mein besonderer Dank gilt den Angehörigen meiner “alten” wie “neuen” Familie, die mich in den vergangenen Jahren vielfältig unterstützt und auf meinem Weg bestärkt haben, vor allem aber meiner Mutter Christiane Panzer – in deren liebevoller Pflege mir während längerer Krankheit zwei der wichtigsten Ideen dieser Arbeit in den Sinn kamen – und meiner wundervollen Frau Tina, die mich mit so viel Geduld und Nachsicht durch die zehrenden vergangenen Wochen und Monate getragen hat.

Contents

List of Figures

List of Sources

Glossary

Abbreviations

FEM	Finite element method
HFM	High fidelity model, i. e. a model of high accuracy and complexity
HSV	Hankel singular value
LSE	Linear system of equations
LTI	Linear time invariant
MOR	Model order reduction
ODE	Ordinary differential equation
PDE	Partial differential equation
ROM	Reduced order model
SVD	Singular value decomposition

General Notation

Scalars are printed italic	n, N
Vectors are printed bold lowercase	$\mathbf{x}$
Matrices are printed bold uppercase	$\mathbf{A}$

Set Symbols

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^+$	Set of real positive numbers
$\mathbb{C}$	Set of complex numbers
$\mathbb{N}$	Set of natural numbers

Matrix Operations

$(\cdot)^T$	Matrix transposition
$(\cdot)^H$	Matrix transposition and complex conjugation (HERMITE)
$(\cdot)^{-T}$	Matrix transposition and inversion
$\mathrm{diag}_i(\mathbf{A})$	Diagonal elements of square matrix $\mathbf{A}$
$\mathrm{tr}(\mathbf{A})$	Trace of square matrix, $\mathrm{tr}\,\mathbf{A} = \sum \mathrm{diag}_i(\mathbf{A})$
$\mathrm{sym}(\mathbf{A})$	Hermitian part of square matrix, $\mathrm{sym}\,\mathbf{A} = (\mathbf{A} + \mathbf{A}^H)/2$
$\mathrm{nnz}(\mathbf{A})$	Number of nonzero entries in matrix $\mathbf{A}$
$\|\mathbf{x}\|_2$	Euclidian vector norm, $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^H\mathbf{x}}$
$\|\mathbf{A}\|_2$	Spectral matrix norm, $\|\mathbf{A}\|_2 = \max_i \sqrt{\lambda_i(\mathbf{A}^H\mathbf{A})}$
$\|\cdot\|_\mathrm{F}$	FROBENIUS norm, $\|\mathbf{A}\|_\mathrm{F} = \mathrm{tr}(\mathbf{A}^H\mathbf{A})$
$\lambda_i(\mathbf{A})$	Eigenvalues (spectrum) of matrix $\mathbf{A}$
$\lambda_i(\mathbf{A}, \mathbf{E})$	Generalized eigenvalues; $\lambda_i(\mathbf{A}, \mathbf{E}) = \lambda_i(\mathbf{E}^{-1}\mathbf{A}) = \lambda_i(\mathbf{A}\mathbf{E}^{-1})$
$\sigma_i(\mathbf{A})$	Singular values of matrix $\mathbf{A}$; $\sigma_i(\mathbf{A}) = \sqrt{\lambda_i(\mathbf{A}^H\mathbf{A})}$, $i \leq j \Rightarrow \sigma_i \geq \sigma_j$

Other Symbols

$\mathbf{e}_i$	i-th unit vector
$\mathbf{G}(s)$	Laplace transfer function or corresponding dynamic LTI system
$\mathbf{H}(t)$	Impulse response
$\mathbf{I}_n$	Identity matrix of dimension n
$\mathbf{0}_{n\times m}$	Zero matrix with n rows and m columns
$\delta(t)$	DIRAC impulse function
σ	Shift, expansion point
$\sigma(t)$	HEAVISIDE step function

1. Introduction

"Das bewußte Reduzieren, das Weglassen, das
Vereinfachen hat eine tiefe ethische Grundlage:
Nie kann etwas zuwider sein, was einfach ist."

— Egon Eiermann

1.1. A Short Motivation of Model Reduction

Mathematical models of technical or physical systems have become an indispensable tool in countless applications and domains. The vast majority of modern control techniques, for instance, is in one way or the other based on mathematical models of the underlying system. Even more obviously, accurate models are part and parcel of computer simulations, and therefore constitute an important part of industrial development processes, optimization, the analysis or prediction of complex systems, and the exploration of novel technologies.

Yet for all mentioned applications, the accuracy and reliability of the model plays an important role. The better the model describes reality, the better the expectable results from simulation, and the more likely predictions apply, etc. Increasing demands on the accuracy, however, typically bring about higher complexity of the model which may complicate or even inhibit the fulfillment of the given task due to limitations of memory and/or computational capacity. Feedback controllers with state observers running on microcontrollers in real time are one example of such limitations; weather forecast simulations or computational fluid dynamics (CFD) occupying supercomputers or clusters for days and weeks are another representative of the trade-off between accuracy and manageability. [8]

Model simplification or, synonymously, model reduction techniques can be a remedy in such situations. Their general goal is to replace an existing high fidelity model (HFM) by another model which is just as well suited for the engineering task but of lower complexity. Thereby, efficiency can be dramatically increased, as comparable results can be produced in far less time. The difficulty is to identify and extract the parts of the HFM that are relevant for the specified task while discarding the superfluous components of the model. As mathematical models may take very different forms, depending on the kind of system they describe, every type of model requires customized techniques for its simplification.

One of the most important fields of application of model simplification arises in the context of partial differential equations (PDE). This class of models is well-suited for the description of a wide variety of physical and artificial systems, for instance in structural mechanics, diffusion and heat conduction, acoustics, and micro-electromechanics (MEMS). Typically, the systems possess a limited number m of inputs where they are influenced by actors or the environment. The goal in many applications is then to describe the dynamic behavior of the system at a certain number p of outputs, which are quantities of interest, possibly measured by sensors in the real-world system. p varies strongly with the given task; for simulation purposes, one is rather interested in a high number of outputs to obtain a complete picture of the system, while in control applications, p reflects the number of relevant process variables.

Either way, we consider models that are characterized by the way they map m input signals $u_i(t)$ to p output signals $y_j(t)$; see Figure 1.1. Please note that in this classical system theoretic view it is assumed that the outputs are determined by the inputs, but do not influence those in return—in contrast to the behavioral approach of describing system dynamics due to Willems [129].

But as the model is described *locally* by the PDE and may live on a complex geometry, analytic solutions for the global behavior are typically not attainable. Instead, the domain is spatially discretized in order to approximate the infinite-dimensional solution of the boundary value problem with a finite number of degrees of freedom, for example with the help of finite elements and a Galerkin method. The PDE is thereby replaced with a coupled system of ordinary differential equations (ODE) whose number N depends on the fineness of the discretization grid.

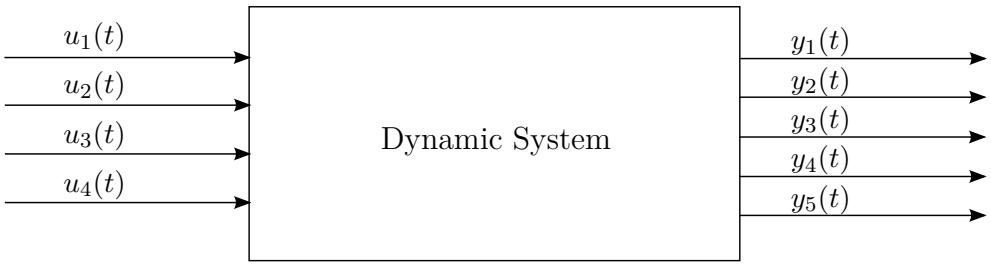


Figure 1.1.: Dynamic System as Operator from Input to Output Signals

Obviously, this method is another example of the goal conflict described above. On the one hand, finer spatial discretization leads to a better approximation of the true (infinite-dimensional) solution. On the other hand, it increases the number of ODEs, which in modern applications may even amount to millions of equations. However, due to their local and generic nature, the basis functions are often far from optimal, such that a small number of their linear combinations may suffice to obtain a similar approximation of the true model to that defined by the high-dimensional (high fidelity) ODE system. In this particular case, one usually speaks of model order reduction (MOR), because the number N of equations is also called the order of the model. Please note that the number of inputs and outputs remains unchanged during this procedure; only the state space in which the dynamics happen is replaced such that similar transfer behavior is mimicked with far less internal variables.

In fact, fine meshing and subsequent reduction of the model typically leads to much better results than applying a coarse grid from the beginning, which makes MOR a highly important tool whenever spatial discretization is to be applied.

1.2. Dissertation Goals and Overview

This section will provide a rough overview of what is in the scope of this thesis, how it is to be used, and who contributed to the presented results.

1.2.1. Objectives and Classification

This thesis is exclusively dedicated to MOR of linear time-invariant (LTI) systems in generalized state space realization (cf. Section 2.1.1)

$$\mathbf{E}\,\dot{\mathbf{x}}(t) = \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,\mathbf{u}(t), \tag{1.1}$$

$$\mathbf{y}(t) = \mathbf{C}\,\mathbf{x}(t) + \mathbf{D}\,\mathbf{u}(t). \tag{1.2}$$

Although parts of the presented methodology extend to certain "benign" DAE systems, we restrict ourselves to regular matrices $\mathbf{E}$ excluding algebraic states. The reasoning for the focus on LTI systems is discussed in Section 1.2.2.

Also, we only consider models in continuous time, i. e. systems of ordinary differential equations, but no discrete-time models consisting of *difference equations* as they would result from discretization in time. Again, modification of the presented results should be feasible, but is not carried out in this thesis, as all considered benchmark models are continuous in time. Note also, that exact discretization in time can be easily performed after a ROM has been found in continuous time, because the matrix exponential is then available.

Furthermore, as explained in the introduction, it is stressed that the philosophy behind MOR (as it is understood in this work) is to approximate the *transfer behavior* of a dynamical system, but *not necessarily* its internal behavior. This means, for instance, that given an input signal $\mathbf{u}(t)$, our goal is only to use a reduced order model (ROM) to approximate the output $\mathbf{y}(t)$ defined by (1.2), but not the whole high-dimensional state trajectory $\mathbf{x}(t)$.

This is an important point with the explanations from Section 1.1 in mind, because the basic idea of MOR is discarding the superfluous parts of the model—but of course the output equation is of vital importance to the question *which* parts of the model are dispensable. Accordingly, it is in general not an objective of MOR to provide any kind of physical interpretability of the state variables in the ROM.[1] Instead, the available degrees of freedom are exploited to obtain optimal approximation of the transfer behavior—even though this only gives an abstract description of the "inner life" of the HFM.

[1] Interestingly, the projective MOR framework as introduced in Section 2.3.1 is nevertheless based on the assumption that the high-dimensional state vector is also well approximated by the ROM.

This philosophy well suits typical control applications, where all quantities of interest are part of the output vector $\mathbf{y}(t)$. In simulation, however, one should treat the choice of the output matrix $\mathbf{C}$ with care: only output variables are regarded in the model reduction process, but the higher their number, the higher the complexity of the reduction process and the ROM.

Furthermore, please note that there are two basically different scenarios in which MOR is applied, depending on the size of the HFM. Either, starting from a medium-scale model, one may want to find an approximant with optimal accuracy-dimension ratio, for example to use it for a controller in an embedded system. Or the HFM is so large and complex that one is satisfied with finding any accurate ROM at all (together with error information), regardless of optimality. In this work, we focus on the second task, as the first problem has been solved quite comprehensively during the last decades. Also, having found a ROM of manageable size in a first step, one can always attach a second reduction step which aims to find a ROM of maximal compactness. Hence, we assume our given HFM to be of very high dimension.

Finally, it is stressed that the use of MOR in practical applications involves various challenges. First, of course, the physical system has to be modeled in some suitable software. Then, a HFM has to be extracted or made available, which typically requires some kind of toolchain. Next, MOR is performed to obtain a ROM in a preferably automatic procedure which requires no input from the user. Finally, the ROM is used to solve the given problem (simulation, controller and observer design, etc.) which may include postprocessing. [54, 101]

Accordingly, when model reduction is to be used in an industrial process, one must keep in mind that the actual model reduction process is only *one* step among many others, which are mainly disregarded in this thesis.

1.2.2. Why Linear Time Invariant Models?!

Linear time invariant systems and their reduction have been investigated for decades by now, and model reduction of time-variant, parameter-dependent, and nonlinear systems recently attracts more and more attention. So dedicating a whole thesis to LTI model reduction probably needs justification.

In fact, Antoulas gives five good reasons for the importance of linear techniques [8], among them the facts that many physical laws are indeed linear in "large ranges of the operating conditions" and that "all systems are locally linear". Two more motives are possibly worth mentioning: Firstly, MOR of LTI systems can still not be considered entirely solved (for open problems please refer to Sections 2.4 and 3.7). And secondly, many reduction approaches for other system classes trace the problem back to standard linear MOR. For instance, established reduction techniques for parametric models

$$\begin{cases} \mathbf{E}(\mathbf{p})\,\dot{\mathbf{x}}(t) &= \mathbf{A}(\mathbf{p})\,\mathbf{x}(t) &+ \mathbf{B}(\mathbf{p})\,\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}(\mathbf{p})\,\mathbf{x}(t) &+ \mathbf{D}(\mathbf{p})\,\mathbf{u}(t), \end{cases} \tag{1.3}$$

including constant dependencies on a parameter vector $\mathbf{p}$ evaluate the model locally for some reference values of $\mathbf{p}$. Then, the resulting LTI models are reduced independently by standard techniques ("offline stage"), before the local reduced models are used for some kind of interpolation ("online stage"). Examples of such methods are, for instance, described in [5, 17, 106, 126]; for a recent survey, see [28].

Even techniques for the reduction of nonlinear state space models

$$\begin{cases} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \\ \mathbf{y}(t) &= \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t), \end{cases} \tag{1.4}$$

like the Trajectory Piecewise Linearization (TPWL) [130, 131] use linear MOR theory.

So extending and improving existing linear methodology can also bring about further development of more general reduction techniques.

1.2.3. Outline

The contents of this thesis are structured as follows. Chapter 2 recalls relevant preliminaries on LTI systems theory and introduces basic concepts including Petrov-Galerkin projections. Chapter 3 gives an overview of MOR based on Sylvester equation, including rational Krylov subspace methods, which are in the focus of this work. The subsequent two chapters present new ideas to circumvent the problems related to rational Krylov methods, exploiting a factorized formulation of the error model. In Chapter 4, a cumulative reduction framework is introduced which enables the adaptive choice of parameters (expansion points and reduced order). Chapter 5 presents efficient upper bounds on the absolute and relative $\mathcal{H}_2$ and $\mathcal{H}_\infty$ error resulting from Sylvester-based model

reduction under the assumption that the HFM is available in strictly dissipative realization. Second order systems, which can often be formulated in such a way, are the topic of Chapter 6. Numerical examples and demonstrations are provided in Chapter 7, followed by conclusions in Chapter 8.

1.2.4. Acknowledgment of Foreign Scientific Contributions

My colleague Thomas Wolf was co-inventor of the error decomposition shown in Sections 3.3 and 4.2; he therefore had an important stake in the fundamental concepts from which the results presented in this thesis evolved. He is also the originator of the PORK algorithm in Section 3.6.5, which constitutes an important jigsaw piece of this work, and discovered reference [81] in the literature, on which the $\mathcal{H}_2$ error bound in Section 5.3 is based. I also owe him my understanding of Sylvester equations; but far beyond that he commented and helped improving many parts of this thesis, which would certainly not be the same without his influence.

Prof. Dr. Timo Reis corrected me in my understanding of algebraic constraints and DAE systems, which improved the respective results presented in [123].

Dr. Jens Saak recognized that the transformation towards a strictly dissipative realization (cf. Section 6.2) basically led to a change of the considered inner product; this connection is presented in Section 2.2.2.

Below are relevant contributions of students I supervised:

Stefan Jaensch within his diploma thesis [84] significantly contributed to Section 4.4, cf. [122]; in particular, he proposed the general idea of using a model function to speed up optimization, which formed the basis of Section 4.4.4.

Benjamin Kleinherne in his term paper [89] dealt with generalizations of the strictly dissipative realization of second order systems and helped extending existing proves; the results have been jointly published in [123].

Anna Kohl further developed the idea of error-controlled model reduction based on optimization in her master's thesis [91] and thus contributed to Section 5.6.

Yvonne Stürz worked on the generalization of existing SISO methods to multivariable systems during her master's thesis [149] and thus assisted me with the topics treated in Sections 3.2 and 4.5.

1.2.5. MATLAB Source Code Listings

To illustrate implementational aspects and to avoid misconception of algorithms presented in this work, several ready-to-run functions and code snippets have been included in this thesis.

All listings are developed for MATLAB (The MathWorks, Inc.) and have been tested on the 64-bit version R2013b. The reason why MATLAB code is enclosed instead of pseudocode (or other informal ways of describing high-level programs) is that MATLAB has established in the scientific community and is very widely used, so it is the author's belief that the advantages of ready-to-run code (in which implementational details are resolved) outweigh the downsides.

For convenience the source code can be directly copied from the digital pdf version of this document. Please create new `.m`-files and copy the respective contents of the source listings into them. Store all files in one directory and include it in your MATLAB path.

The listings are not intended for immediate use in industrial settings (in particular, they come with no warranty; see license below) and are not optimized with respect to computational performance nor robustness (exception handling). However, much effort has been made to provide functional and modular code with a reasonable readability-performance ratio. The main goals are to facilitate the development of powerful source code for newcomers, to enable the reader to reproduce numerical results presented in this work, and to simplify the adaptation and usage of the algorithms for technical applications and further research.

The code is mainly stand-alone, but sometimes uses functions from the Control System Toolbox. As this toolbox, however, implements LTI systems as `ss`-objects which disregard sparsity of high-dimensional matrices, its usage is limited to medium-scale systems. To draw Bode plots of high-dimensional models, for instance, one has to replace the respective functions by manual implementations.

All source code listings enclosed are published under the BSD 3-Clause License as stated below.

The numerical experiments presented in this thesis, by the way, have been carried out on a standard PC architecture equipped with 6 GB RAM and an Intel Core i7-2630QM CPU (4 cores, 8 threads) running at 2 GHz.

1.2.6. Benchmark Examples

Various benchmark models have been used in this thesis for demonstrating purposes. In the following, a short description of the respective models is given, together with their origin.

SLICOT Benchmark Collection

The SLICOT Library [26] includes a couple of very widely used benchmark examples for MOR of LTI systems. A description of the models is given online and in [38]. The following table summarizes the models that were used in this thesis.

Model name	Order N	nnz($\mathbf{A}$)	Inputs m	Outputs p
CD Player	120	240	2	2
Clamped Beam	348	60726	1	1
Continuous Heat Equation	200	598	1	1
International Space Station (ISS)	270	405	3	3

Spiral Inductor

The spiral inductor was originally modeled by Kamon, Wang, and White in [87]; it is an integrated RF passive inductor, which can also be used as a proximity sensor. Li and Kamon in [102] provided a SISO state space model of order $N = 1434$, where $\text{nnz}(A) = 18228$, $\text{nnz}(E) > 1.1e6$. The goal is to find a ROM which can mimic the frequency-dependent resistance $R_p(\omega)$ and inductance $L_p(\omega)$ of the device, which are related to the transfer function via

$$R_p(\omega) + i\omega L_p(\omega) = Z_p(\omega) = \Big[G(i\omega)\Big]^{-1}. \tag{1.5}$$

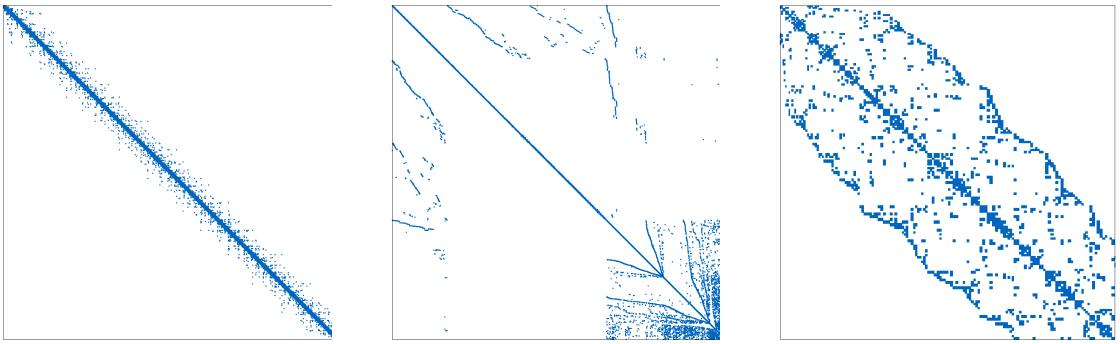
Steel Profile

This model describes the heat distribution in a steel profile during a cooling process. "The cooling process, which is realized by spraying cooling fluids on the surface, has to be controlled so that material properties, such as durability or porosity, achieve given quality standards" [31]. The model is therefore intended for the pre-calculation of different control laws in order to find an optimal cooling strategy.

It has six inputs which correspond to the activity of "phantom nozzles" [154] and seven outputs describing the temperature at certain points. The model has been created for four different meshes leading to 1357, 5177, 20209, or 79841 state variables, respectively.

Convective Thermal Flow Problems

This model was set up by Moosmann and Greiner and makes part of the Oberwolfach Benchmark Collection [96]. It describes "the heat exchange between a solid body and a fluid flow", has order $N = 9669$, one input and five outputs. The matrix $\mathbf{E}$ is diagonal; $\mathbf{A}$ has 67 391 nonzero entries, see Figure 1.2a).



a) Flow Meter (v=0) b) Power System c) Acoustic Field

Figure 1.2.: Sparsity Pattern of Matrix $\mathbf{A}$ in Various Benchmark Models

Power System

ROMMES provides a collection of models describing power systems from various types of studies on his website [132]; details are also given in the MOR Wiki [114]. In this work, the BIPS/1997 model (xingo_afonso_itaipu.mat) is used, which is a "planning model for the Brazilian Interconnected Power System" [133]. It has order $N = 13250$, but is a descriptor SISO model with singular diagonal $\mathbf{E}$ matrix having only 1664 entries on the diagonal, all 1. The sparsity pattern of $\mathbf{A}$ is depicted in Figure 1.2b); $\text{nnz}(\mathbf{A}) = 48735$.

Acoustic Field in Gas Turbine Combustor

Thermoacoustic effects in gas turbines and combustion units with high power density carry the danger of damaging or even destroying the combustion chamber due to high pressure oscillation amplitudes, which may be caused by a feedback coupling between heat release fluctuations of the flame and the acoustic field of the chamber. For a better understanding of the related phenomena, simplified burners are modeled to analyze such thermoacoustic instabilities.

The model at hand describes the acoustic field inside a burner and has been provided by TOBIAS HUMMEL from the Chair for Thermodynamics, Technische Universität München. It has one input (acoustic excitation) and one output (pressure at some location); the spatial discretization of the chamber led to an order of $N = 62665$. The number of nonzero entries in each of the matrices $\mathbf{A}$ and $\mathbf{E}$ amounts to $\text{nnz}(\mathbf{A}) = \text{nnz}(\mathbf{E}) \approx 3.18 \cdot 10^6$. A 500×500 section of the sparsity pattern is depicted in Figure 1.2c).

Parametric FEM Beam

This model of a cantilever TIMOSHENKO beam is available online in the form of an `.m`-file [125]. Many parameters including discretization, length, and YOUNG's modulus can be adapted. In this work, we used the preadjustment with 60 finite elements, leading to a $\widehat{N} = 300$.

Butterfly Gyroscope

"The Butterfly is a vibrating micro-mechanical gyro that has sufficient theoretical performance characteristics to make it a promising candidate for use in inertial navigation applications. [...] Repeated analyses of the sensor structure have to be conducted with respect to a number of important issues. Examples of such are sensitivity to shock, linear and angular vibration sensitivity, reaction to large rates and/or acceleration, different types of excitation load cases and the effect of force-feedback." [104]

The model is available online [34] as a second order system (6.1) of dimension $\widehat{N} = 17361$ with symmetric positive definite $\mathbf{M}, \mathbf{K}$, and $\mathbf{D} = 10^{-6} \cdot \mathbf{K}$. It has a single input and twelve outputs which describe the displacements of the four detection electrodes.

Wineglass

This model was provided by JEONG SAM HAN and is described in [78, 169]. It is a second order model of dimension $\widehat{N} = 368424$, with $\mathrm{nnz}(\mathbf{M}) \approx 5.87e6$, $\mathrm{nnz}(\mathbf{D}) \approx \mathrm{nnz}(\mathbf{K}) \approx 9.29e6$, one input and six outputs. The model is not reduced in this work but merely used for demonstration purposes in the context of second order systems.

2. Preliminaries

"Finally, we make some remarks on why linear systems are so important. The answer is simple: because we can solve them!"
— Richard Feynman [57]

2.1. Fundamentals from LTI System Theory

In the following, important preliminaries on LTI systems are summarized. For a more circumstantial introduction, please refer to [8, 46, 86], or other standard works.

2.1.1. State Space Models

A generalized state space model of a linear time-invariant (LTI) system is given by

$$\begin{cases} \mathbf{E}\,\dot{\mathbf{x}}(t) &= \mathbf{A}\,\mathbf{x}(t) &+ \mathbf{B}\,\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\,\mathbf{x}(t) &+ \mathbf{D}\,\mathbf{u}(t), \end{cases} \tag{2.1}$$

with $\mathbf{E}, \mathbf{A} \in \mathbb{R}^{N\times N}, \mathbf{B} \in \mathbb{R}^{N\times m}, \mathbf{C} \in \mathbb{R}^{p\times N}$, and $\mathbf{D} \in \mathbb{R}^{p\times m}$. $\mathbf{x}(t) \in \mathbb{R}^N$ is called the state vector; its dimension $N \in \mathbb{N}$ denotes the *order* of the model. $\mathbf{u}(t)$ and $\mathbf{y}(t)$ contain the input and output signals of the system, respectively. Systems with $m, p > 1$ are referred to as multi-input multi-output (MIMO) systems; the special case $m = p = 1$ is called single-input single-output (SISO) system.

State space models describe the way how an input signal $\mathbf{u}(t) \in \mathbb{R}^m$ is mapped to an output signal $\mathbf{y}(t) \in \mathbb{R}^p$ ("transfer behavior", "input/output-behavior"). At the same time, they can also give insight into the physics of the underlying technical system, if the entries of $\mathbf{x}(t)$ are related to "real-world" quantities—like, for instance, displacements of nodes in finite element methods for structural mechanics. In the sense of the introduction, however, we assume in the following that the purpose of the model is not to describe the

internal state of a physical system, but only its transfer behavior. Accordingly, in this work the term "system" refers to the operator which maps input signals to output signals rather than to a physical/technical object.

Common short notations of a generalized state space model (2.1) are

$$(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}) \quad \text{and} \quad \left[\begin{array}{c|c} \mathbf{E}, \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right]. \tag{2.2}$$

If $\mathbf{E} = \mathbf{I}_N$ is the identity matrix, we call (2.1) a *standard* state space model, otherwise a *generalized* state space model. Systems with $\det \mathbf{E} = 0$ are referred to as differential-algebraic equations (DAE) or descriptor systems[1], but play a minor role in this thesis: we will assume $\mathbf{E}$ to be regular, unless specified otherwise.

We denote the generalized eigenvalues $\lambda_i(\mathbf{A}, \mathbf{E}) = \lambda_i(\mathbf{E}^{-1}\mathbf{A}) \in \mathbb{C}$ of the system shortly by λ_i; the largest occurring real part is named spectral abscissa and denoted by $\alpha := \alpha(\mathbf{A}, \mathbf{E}) := \max_i \operatorname{Re} \lambda_i$. A system with strictly negative spectral abscissa is asymptotically stable, which means that for any initial state it converges to the origin as time tends to infinity, if no input signal is applied. In this thesis we focus on asymptotically stable models.

2.1.2. Transfer Function and Impulse Response

The transfer behavior of an LTI system can also be described with the help of its impulse response matrix $\mathbf{H}(t) \in \mathbb{R}^{p \times m}$, whose (i, j)-th entry describes the output signal $y_i(t)$ for the particular DIRAC input signal $\mathbf{u}(t) = \mathbf{e}_j \cdot \delta(t)$. Given an input signal $\mathbf{u}(t)$, the respective output follows by convolution:

$$\mathbf{y}(t) = (\mathbf{H} * \mathbf{u})(t) = \int_{-\infty}^{+\infty} \mathbf{H}(t - \tau) \cdot \mathbf{u}(\tau) d\tau. \tag{2.3}$$

The LAPLACE transform of the impulse response is called the transfer function,

$$\mathbf{G}(s) = \mathcal{L}\{\mathbf{H}(t)\}. \tag{2.4}$$

Assuming distinct poles p_i, $\mathbf{G}(s)$ can be written in pole-residual formulation:

$$\mathbf{G}(s) = \sum_{i=1}^{N} \frac{\mathbf{\Phi}_i}{s - p_i}. \tag{2.5}$$

[1] Note that in the literature, "descriptor system" sometimes means $\mathbf{E} \neq \mathbf{I}$ instead.

Another important formulation of the transfer function is given by the TAYLOR expansion about a complex shift σ,

$$\mathbf{G}(s) = \sum_{i=0}^{\infty} \boldsymbol{\eta}_i^{\sigma} \cdot (s-\sigma)^i, \tag{2.6}$$

where $\boldsymbol{\eta}_i^{\sigma}$ are known as the *moments* of $\mathbf{G}(s)$ about σ and given by

$$\begin{aligned} \boldsymbol{\eta}_0^{\sigma} &= \mathbf{D} - \mathbf{C}\left(\mathbf{A} - \sigma\mathbf{E}\right)^{-1}\mathbf{B}, \\ \boldsymbol{\eta}_i^{\sigma} &= -\mathbf{C}\left[\left(\mathbf{A} - \sigma\mathbf{E}\right)^{-1}\mathbf{E}\right]^{i-1}\left(\mathbf{A} - \sigma\mathbf{E}\right)^{-1}\mathbf{B} \qquad \text{for} \quad i \geq 1. \end{aligned} \tag{2.7}$$

Given a generalized state space model $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E})$, its transfer behavior is described by the LAPLACE transfer function

$$\mathbf{G}(s) = \mathbf{C}\left(s\mathbf{E} - \mathbf{A}\right)^{-1}\mathbf{B} + \mathbf{D} \tag{2.8}$$

and for $\det \mathbf{E} \neq 0$ the corresponding impulse response reads

$$\mathbf{H}(t) = \mathbf{C}e^{\mathbf{E}^{-1}\mathbf{A}t}\mathbf{E}^{-1}\mathbf{B} \cdot \sigma(t) + \mathbf{D} \cdot \delta(t). \tag{2.9}$$

While the transfer function $\mathbf{G}(s)$ of an input/output-system is unique, there are obviously infinitely many different state space models, which can, for instance, result from one another by state transformation.

If equation (2.8) holds for some state space model $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E})$ and a given transfer function $\mathbf{G}(s)$, we call the state space model a *realization* of the transfer function or the associated input/output-system; equivalently, one says that $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E})$ *realizes* $\mathbf{G}(s)$.

Due to the one-to-one relationship of an LTI input/output-system and its transfer function, the symbol $\mathbf{G}(s)$ is typically used for both of them. In fact, throughout this work $\mathbf{G}(s)$ may even denote the realization of the system, unless the context is unclear.

2.1.3. Controllability, Observability, and Minimal Realizations

Definition 2.1 (Controllability)**.** *The pair* $(\mathbf{A}, \mathbf{B})$ *with dimensions as in* (2.1) *is called controllable, if the matrix* $\left[\mathbf{B},\ \mathbf{AB},\ \ldots\ \mathbf{A}^{N-1}\mathbf{B}\right]$ *has full row rank.*

Definition 2.2 (Observability)**.** *The pair* $(\mathbf{C}, \mathbf{A})$ *with dimensions as in* (2.1) *is called observable, if the matrix* $\left[\mathbf{C}^T,\ \mathbf{A}^T\mathbf{C}^T,\ \ldots\ (\mathbf{A}^T)^{N-1}\mathbf{C}^T\right]^T$ *has full column rank.*

These matrix-based definitions provide necessary and sufficient criteria for the respective properties of a generalized state space model (2.1), which is completely controllable

if $(\mathbf{E}^{-1}\mathbf{A}, \mathbf{E}^{-1}\mathbf{B})$ is controllable, and completely observable if $(\mathbf{C}, \mathbf{E}^{-1}\mathbf{A})$ is observable. A completely controllable and observable state space model is called *least order* or *minimal* realization of a transfer function; its order is known as McMillan degree [134]. It can be shown that no other realization of the same transfer function can have smaller order.

2.1.4. Invariant Zeros

Definition 2.3 ([157]). *A complex number η is called an invariant zero of a state space model (2.1), if it satisfies*

$$\operatorname{rank}\left[\begin{array}{c|c} \eta\mathbf{E}-\mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right] < \text{normal rank}\left[\begin{array}{c|c} s\mathbf{E}-\mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right] := \max_{s\in\mathbb{C}} \operatorname{rank}\left[\begin{array}{c|c} s\mathbf{E}-\mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right].$$

If an invariant zero coincides with an eigenvalue of the realization, compensation may occur, such that the eigenvalue may be uncontrollable and/or unobservable and does not contribute to the transfer behavior as a pole. An invariant zero which is not compensated is called transmission zero. [109]

2.1.5. System Norms and Gramian Matrices

System norms play an important role in the analysis of LTI systems, as they quantify certain properties of the model. In this thesis, we will concentrate on the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms. Other norms like the Hankel norm can be found in the literature, cf. [8].

Definition 2.4 ([8]). *The $\mathcal{H}_\infty$ norm of an LTI system is defined as*

$$\|\mathbf{G}\|_{\mathcal{H}_\infty} := \sup_{\omega\in\mathbb{R}} \sigma_{max}\Big(\mathbf{G}(i\omega)\Big) = \sup_{\omega\in\mathbb{R}} \Big\|\mathbf{G}(i\omega)\Big\|_2. \tag{2.10}$$

Definition 2.5 ([8]). *The $\mathcal{H}_2$ norm of an LTI system is defined as*

$$\|\mathbf{G}\|_{\mathcal{H}_2} := \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} \operatorname{tr}\Big[\mathbf{G}^H(i\omega)\mathbf{G}(i\omega)\Big]d\omega} = \sqrt{\int_0^\infty \operatorname{tr}\Big[\mathbf{H}^T(t)\mathbf{H}(t)\Big]dt}. \tag{2.11}$$

Given a realization of $\mathbf{G}$, the $\mathcal{H}_2$ norm can be found algebraically as

$$\|\mathbf{G}\|_{\mathcal{H}_2} = \sqrt{\operatorname{tr}\left(\mathbf{B}^T\mathbf{Q}\mathbf{B}\right)} = \sqrt{\operatorname{tr}\left(\mathbf{C}\mathbf{P}\mathbf{C}^T\right)}, \tag{2.12}$$

where $\mathbf{P}$, $\mathbf{Q}$ are the solutions of the two dual generalized Lyapunov equations

$$\mathbf{A}\mathbf{P}\mathbf{E}^T + \mathbf{E}\mathbf{P}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T = \mathbf{0}, \quad \text{and} \tag{2.13}$$

$$\mathbf{A}^\mathbf{T}\mathbf{Q}\mathbf{E} + \mathbf{E}^\mathbf{T}\mathbf{Q}\mathbf{A} + \mathbf{C}^T\mathbf{C} = \mathbf{0}. \tag{2.14}$$

They are closely related to the so-called Gramian matrices [30, 82]:

Definition 2.6. *The Controllability and Observability Gramian of a realization (2.1) are defined as* $\mathbf{W}_C := \mathbf{P}$ *and* $\mathbf{W}_O = \mathbf{E}^T\mathbf{Q}\mathbf{E}$*, respectively.*

Yet as a matter of fact, for what follows the solution $\mathbf{Q}$ of the LYAPUNOV equation is often more convenient than the actual Observability Gramian $\mathbf{W}_O$. Despite the conflictive definition in [82], we will therefore sometimes refer to $\mathbf{P}$ and $\mathbf{Q}$ shortly as "Gramians" instead of "solutions of the LYAPUNOV equations".

The $\mathcal{H}_2$ norm is induced from an inner product.

Definition 2.7 ([76, 164])**.** *Given two LTI systems* $\mathbf{G}_1(s)$ *and* $\mathbf{G}_2(s)$ *with equal number of inputs and outputs* ($m_1 = m_2$, $p_1 = p_2$)*, their* $\mathcal{H}_2$ *inner product is defined as*

$$\left\langle \mathbf{G}_1, \mathbf{G}_2 \right\rangle_{\mathcal{H}_2} := \frac{1}{2\pi} \int_{-\infty}^{\infty} \operatorname{tr}\left[\mathbf{G}_1^H(i\omega)\mathbf{G}_2(i\omega)\right] d\omega = \int_0^{\infty} \operatorname{tr}\left[\mathbf{H}_1^T(t)\mathbf{H}_2(t)\right] dt. \tag{2.15}$$

Given two realizations

$$\mathbf{G}_1(s) = \left[\begin{array}{c|c} \mathbf{E}_1, \mathbf{A}_1 & \mathbf{B}_1 \\ \hline \mathbf{C}_1 & \mathbf{0} \end{array}\right], \quad \mathbf{G}_2(s) = \left[\begin{array}{c|c} \mathbf{E}_2, \mathbf{A}_2 & \mathbf{B}_2 \\ \hline \mathbf{C}_2 & \mathbf{0} \end{array}\right],$$

the inner product can be expressed with the help of algebraic equations

$$\left\langle \mathbf{G}_1, \mathbf{G}_2 \right\rangle_{\mathcal{H}_2} = \operatorname{tr}\left[\mathbf{C}_1\, \mathbf{X}\, \mathbf{C}_2^T\right] = \operatorname{tr}\left[\mathbf{B}_1^T\, \mathbf{Y}\, \mathbf{B}_2\right], \tag{2.16}$$

where $\mathbf{X}$ and $\mathbf{Y}$ solve the SYLVESTER equations

$$\mathbf{A}_1\mathbf{X}\mathbf{E}_2^T + \mathbf{E}_1\mathbf{X}\mathbf{A}_2^T + \mathbf{B}_1\mathbf{B}_2^T = \mathbf{0}, \tag{2.17}$$

$$\mathbf{A}_1^T\mathbf{Y}\mathbf{E}_2 + \mathbf{E}_1^T\mathbf{Y}\mathbf{A}_2 + \mathbf{C}_1^T\mathbf{C}_2 = \mathbf{0}. \tag{2.18}$$

Note that LYAPUNOV equations (2.13),(2.14) defining the Gramians are special cases of (2.17), (2.18) that follow for $\mathbf{G}_1 = \mathbf{G}_2$ in identical realization.

2.1.6. All-pass Systems

Definition 2.8 ([171, p. 176f])**.** *A real LTI system with* $p = m$ *is called all-pass, if*

$$\mathbf{G}^T(-s)\mathbf{G}(s) = \mathbf{G}(s)\mathbf{G}^T(-s) = k \cdot \mathbf{I}_m \qquad \forall s \in \mathbb{C} \tag{2.19}$$

for some $k \in \mathbb{R}$*, or, equivalently, if the product of its controllability and observability Gramian is the identity matrix multiplied by* k*:* $\mathbf{W}_C\mathbf{W}_O = \mathbf{P}\mathbf{E}^T\mathbf{Q}\mathbf{E} = k \cdot \mathbf{I}_N$.

We will refer to the particular case $k = 1$ as *unity* all-pass.

2.2. (Strict) Dissipativity and the Matrix Measure

In the following, we recall the concept of matrix dissipativity and the logarithmic norm.

2.2.1. Basic Results

Definition 2.9 ([42, 107]). *The logarithmic 2-norm (also called numerical abscissa or matrix measure) of a matrix* $\mathbf{A} \in \mathbb{C}^{N\times N}$ *is defined as and given by*

$$\mu := \mu_2(\mathbf{A}) := \lim_{h\to 0+} \frac{\|\mathbf{I} + h\mathbf{A}\|_2 - 1}{h} = \max_i \lambda_i \left(\frac{\mathbf{A} + \mathbf{A}^H}{2}\right). \tag{2.20}$$

If $\mu \leq 0$*, we call* $\mathbf{A}$ *dissipative, i.e. its* Hermite *part is negative semidefinite,* $\mathbf{A}+\mathbf{A}^H \leq \mathbf{0}$*. If* $\mu < 0$*, which is equivalent to* $\mathbf{A} + \mathbf{A}^H < \mathbf{0}$*, we call* $\mathbf{A}$ *strictly dissipative.*

Please note that this definition must not be confused with "dissipativity" in the sense of Willems [161]—which is a property of a transfer function—nor as a property of the imaginary part of a matrix as in [52, 151].

Dahlquist [42] and Lozinskii [107] independently found the following fundamental result:

Theorem 2.1 ([141]). *The numerical abscissa fulfills* $\|e^{\mathbf{A}t}\|_2 \leq e^{\mu_2(\mathbf{A})\cdot t} \quad \forall t \geq 0.$

This has two very important consequences in the context of LTI systems. Consider the special case of a standard state space model $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$ of an autonomous system ($\mathbf{u}(t) \equiv \mathbf{0}$). Firstly, for an arbitrary initial state $\mathbf{x}(0)$, the 2-norm of the state trajectory fulfills

$$\|\mathbf{x}(t)\|_2 = \left\|e^{\mathbf{A}t}\cdot\mathbf{x}(0)\right\|_2 \leq \left\|e^{\mathbf{A}t}\right\|_2 \cdot \|\mathbf{x}(0)\|_2 \leq e^{\mu_2(\mathbf{A})\cdot t}\cdot\|\mathbf{x}(0)\|_2 \qquad \text{for } t \geq 0. \tag{2.21}$$

Secondly, dissipativity is a criterion for stability:

Lemma 2.1 ([41, 45, 141]). *The spectral abscissa* α *is less or equal to the numerical abscissa* μ*, i. e. the real part of all eigenvalues of* $\mathbf{A}$ *is less or equal to* μ*.*

In particular, a standard state space model whose $\mathbf{A}$-matrix is strictly dissipative ($\mu < 0$) is asymptotically stable, because all its eigenvalues must have strictly negative real part. In fact, this also follows from Theorem 2.1: If μ is negative, the scalar exponential function decays and one obtains a convergent envelope for the norm of the transition matrix $e^{\mathbf{A}t}$, so for $\mathbf{u}(t) \equiv \mathbf{0}$ any initial state decays towards the origin, which is the definition of asymptotic stability.

In fact, the case $\mu_2(\mathbf{A}) < 0$, or equivalently $\mathbf{A} + \mathbf{A}^H < \mathbf{0}$, is of particular interest in this dissertation. Not only does it allow for stability preservation in projective MOR (cf. Lemma 2.4), but also for global error bounds presented in Chapter 5.

2.2.2. Generalization towards Symmetric Positive Definite E

In order to exploit the dissipativity property in model reduction, we need to get control of the effects of the matrix $\mathbf{E}$. Sure enough, in a generalized state space model, $e^{\mathbf{A}t}$ is not actually the quantity of interest. Rather, the solution of the autonomous ODE with $\mathbf{E} \neq \mathbf{I}_N$ and initial state $\mathbf{x}(0)$ is given by

$$\mathbf{x}(t) = e^{\mathbf{E}^{-1}\mathbf{A}t}\mathbf{x}(0). \tag{2.22}$$

In fact, a common way to trace back the general case to standard state space methodology is to pre-multiply the ODE in (2.1) by $\mathbf{E}^{-1}$ in order to obtain the standard realization $(\mathbf{E}^{-1}\mathbf{A}, \mathbf{E}^{-1}\mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{I})$. However, this transformation can have a massive impact on the dissipativity of $\mathbf{A}$, i. e. the numerical abscissa of $\mathbf{A}$ is in general not at all related to that of $\mathbf{E}^{-1}\mathbf{A}$, so $\mu_2(\mathbf{A}) < 0$ does not imply $\mu_2(\mathbf{E}^{-1}\mathbf{A}) < 0$.

The logarithmic norm for general matrix pencils $(\mathbf{E}, \mathbf{A})$ has therefore been discussed in [79, 80]. In the following, however, we will concentrate on the special case that $\mathbf{E}$ is symmetric and positive definite. Here, it is possible to take use of the results for standard state space realizations in a quite straightforward way. We will now consider two different ways of doing so; both utilize the Cholesky decomposition $\mathbf{L}^T\mathbf{L} = \mathbf{E}$.

The first idea is to define an inner product and a norm [44]:

Definition 2.10 ($\mathbf{E}$-inner product and elliptic $\mathbf{E}$ vector norm)**.**

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{E}} := \mathbf{x}^H\mathbf{E}\mathbf{y}, \tag{2.23}$$

$$\|\mathbf{x}\|_{\mathbf{E}} := \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_{\mathbf{E}}} = \sqrt{\mathbf{x}^H\mathbf{E}\mathbf{x}} = \sqrt{\mathbf{x}^H\mathbf{L}^T\mathbf{L}\mathbf{x}} = \left\|\mathbf{L}\mathbf{x}\right\|_2. \tag{2.24}$$

The induced matrix norm is given by

$$\left\|\mathbf{A}\right\|_{\mathbf{E}} := \max_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|_{\mathbf{E}}}{\|\mathbf{x}\|_{\mathbf{E}}} = \max_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{L}\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{L}\mathbf{x}\|_2} = \max_{\mathbf{y}\neq\mathbf{0}} \frac{\|\mathbf{L}\mathbf{A}\mathbf{L}^{-1}\mathbf{y}\|_2}{\|\mathbf{y}\|_2} = \left\|\mathbf{L}\mathbf{A}\mathbf{L}^{-1}\right\|_2.$$

The key idea is to apply this norm to (2.22) and measure $\mathbf{x}(t)$ in the elliptic $\mathbf{E}$-norm

instead of the Euclidian norm [44, 88]:

$$\begin{aligned} \|\mathbf{x}(t)\|_{\mathbf{E}} &\leq \|e^{\mathbf{E}^{-1}\mathbf{A}t}\|_{\mathbf{E}} \cdot \|\mathbf{x}(0)\|_{\mathbf{E}} = \|\mathbf{L}e^{\mathbf{E}^{-1}\mathbf{A}t}\mathbf{L}^{-1}\|_2 \cdot \|\mathbf{x}(0)\|_{\mathbf{E}} \\ &= \|e^{\mathbf{L}^{-T}\mathbf{A}\mathbf{L}^{-1}\cdot t}\|_2 \cdot \|\mathbf{x}(0)\|_{\mathbf{E}} \\ &\leq e^{\mu_2\left(\mathbf{L}^{-T}\mathbf{A}\mathbf{L}^{-1}\right)\cdot t} \cdot \|\mathbf{x}(0)\|_{\mathbf{E}} \end{aligned} \tag{2.25}$$

The important observation is that $\mu_2(\mathbf{A}) < 0$ implies $\mu_2(\mathbf{L}^{-T}\mathbf{A}\mathbf{L}^{-1}) < 0$, so a strictly dissipative matrix $\mathbf{A}$ is still guaranteed to yield a convergent bound for arbitrary positive definite $\mathbf{E} \neq \mathbf{I}$, $\mathbf{E} = \mathbf{E}^T > \mathbf{0}$.

For that reason, we generalize the definition of the spectral abscissa:

Definition 2.11. *Given symmetric positive definite* $\mathbf{E}$ *and its* Cholesky *factorization* $\mathbf{L}^T\mathbf{L} = \mathbf{E}$*, the generalized numerical abscissa of a matrix* $\mathbf{A}$ *is defined as*

$$\mu := \mu_{\mathbf{E}}(\mathbf{A}) := \mu_2\left(\mathbf{L}^{-T}\mathbf{A}\mathbf{L}^{-1}\right) = \max_i \lambda_i\left(\frac{\mathbf{A}+\mathbf{A}^H}{2},\ \mathbf{E}\right).$$

Corollary 2.1 ([44])**.** *The generalized numerical abscissa fulfills* $\|e^{\mathbf{E}^{-1}\mathbf{A}\cdot t}\|_{\mathbf{E}} \leq e^{\mu_{\mathbf{E}}(\mathbf{A})\cdot t}$ *for* $t \geq 0$*. Also, if the (Euclidian) numerical abscissa is negative, so is the generalized one:*

$$\mu_2(\mathbf{A}) < 0 \quad \Leftrightarrow \quad \mu_{\mathbf{E}}(\mathbf{A}) < 0. \tag{2.26}$$

Also, Lemma 2.1 extends to the generalized spectral abscissa: $\alpha \leq \mu_{\mathbf{E}}(\mathbf{A})$.

For more details, the excellent introduction in [44, Section 1.4] is recommended.

As mentioned above, there is a second way to deal with the generalized state space model. The idea is to perform the following transformation of realization (2.1):

$$\begin{aligned} \overbrace{\mathbf{L}^{-T}\mathbf{E}\mathbf{L}^{-1}}^{\mathbf{I}}\ \dot{\hat{\mathbf{x}}}(t) &= \overbrace{\mathbf{L}^{-T}\mathbf{A}\mathbf{L}^{-1}}^{\widehat{\mathbf{A}}}\ \hat{\mathbf{x}}(t) + \overbrace{\mathbf{L}^{-T}\mathbf{B}}^{\widehat{\mathbf{B}}}\ \mathbf{u}(t), \\ \mathbf{y}(t) &= \underbrace{\mathbf{C}\mathbf{L}^{-T}}_{\widehat{\mathbf{C}}}\ \hat{\mathbf{x}}(t)\ . \end{aligned} \tag{2.27}$$

The numerical abscissa of $\widehat{\mathbf{A}}$ is then given by

$$\mu_2\left(\widehat{\mathbf{A}}\right) = \tfrac{1}{2}\max_i \lambda_i\left(\mathbf{L}^{-1}\mathbf{A}\mathbf{L}^{-T} + \left(\mathbf{L}^{-1}\mathbf{A}\mathbf{L}^{-T}\right)^T\right) = \max_i \lambda_i\left(\tfrac{\mathbf{A}+\mathbf{A}^T}{2}, \mathbf{E}\right) = \mu_{\mathbf{E}}(\mathbf{A}), \tag{2.28}$$

so this procedure has the very same effect as changing the norm of interest from the Euclidian one to the $\mathbf{E}$-norm. Of course, the product $\mathbf{L}^{-1}\mathbf{A}\mathbf{L}^{-T}$ is never computed explicitly, but still the second approach seems less elegant than the first one.

To conclude: We have seen that a strictly dissipative matrix $\mathbf{A}$ in combination with a symmetric positive definite matrix $\mathbf{E}$ defines a particular realization with useful properties which are related to the logarithmic norm. One can establish this connection by performing a transformation including the CHOLESKY factor of $\mathbf{E}$, or by regarding the norm induced by $\mathbf{E}$ rather than the Euclidian norm.

Definition 2.12. *Realizations with* $\mathbf{E} = \mathbf{E}^T > \mathbf{0}$ *and* $\mathbf{A} + \mathbf{A}^H < \mathbf{0}$ *are called strictly dissipative; they fulfill* $\mu_{\mathbf{E}}(\mathbf{A}) < 0$.

Corollary 2.2. *Asymptotically stable realizations with symmetric positive definite* $\mathbf{E} = \mathbf{E}^T > \mathbf{0}$ *and symmetric* $\mathbf{A} = \mathbf{A}^T$ *are strictly dissipative.*

Proof. $\mathbf{A}$ is negative definite in this case, $\mathbf{A} = \mathbf{A}^T < \mathbf{0}$. □

2.2.3. Properties and Retrieval of Strictly Dissipative Realizations

First of all, we recall that the numerical abscissa is not a property of the transfer function, but—in the first place—of the realization; to be precise: of $\mathbf{A}$ and $\mathbf{E}$. Accordingly, it can be affected by state transformations or by pre-multiplication of the ODE with a regular matrix $\mathbf{T}$.

The question is: Given the above mentioned amenities of a strictly dissipative realization, can we always retrieve such a system from an arbitrary realization? Suppose we want to preserve the state variables, so instead of a state transformation we choose the second option and equivalently modify the ODE in (2.1) towards

$$\underbrace{\mathbf{TE}}_{\widetilde{\mathbf{E}}}\dot{\mathbf{x}}(t) = \underbrace{\mathbf{TA}}_{\widetilde{\mathbf{A}}}\mathbf{x}(t) + \underbrace{\mathbf{TB}}_{\widetilde{\mathbf{B}}}\mathbf{u}(t). \tag{2.29}$$

Lemma 2.2 ([88, 123])**.** *Define* $\mathbf{T} \in \mathbb{R}^{N\times N}$ *as* $\mathbf{T} := \mathbf{E}^T\mathbf{P}$, *where* $\mathbf{P} = \mathbf{P}^T > \mathbf{0} \in \mathbb{R}^{N\times N}$ *solves the generalized* LYAPUNOV *inequality*

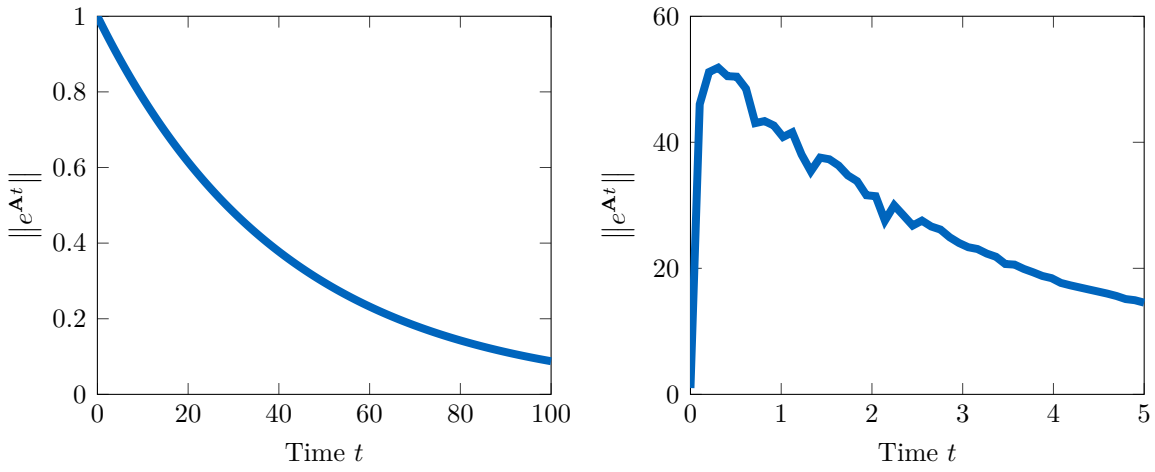
$$\mathbf{E}^T\mathbf{PA} + \mathbf{A}^T\mathbf{PE} < \mathbf{0}. \tag{2.30}$$

Then the transformed realization given by (2.29) *is strictly dissipative.*

Proof. The transformed matrix $\widetilde{\mathbf{E}} = \mathbf{TE} = \mathbf{E}^T\mathbf{PE}$ is clearly positive definite, so the first condition is fulfilled. Secondly, strict dissipativity requires

$$\widetilde{\mathbf{A}} + \widetilde{\mathbf{A}}^T < \mathbf{0} \quad \Leftrightarrow \quad \mathbf{E}^T\mathbf{PA} + \left(\mathbf{E}^T\mathbf{PA}\right)^T < \mathbf{0}.$$ □

It is clear that the computational effort to find such a transformation matrix can in general be tremendous for high order N, even if algorithms like in [40] may find a solution efficiently. In general, one will probably not be able to exploit the dissipativity-based features unless μ happens to be negative by itself. This is, for instance, the case for symmetric systems (cf. Corollary 2.2) as they arise in the context of diffusion and heat transfer (benchmark examples are the Spiral Inductor, the Flow Meter, or the Steel Profile). But we will also see in Section 6.2 that for typical second order systems it is possible to find a suitable transformation at very low computational cost.



a) CD Player b) ISS Station

Figure 2.1.: The "Hump" in Norm of Matrix Exponential

Finally, it is mentioned that the matrix measure fulfills the differential inequality [41]

$$\frac{d}{dt}\left\|\mathbf{x}(t)\right\|_{\mathbf{E}} \leq \mu_{\mathbf{E}}(\mathbf{A}) \cdot \left\|\mathbf{x}(t)\right\|_{\mathbf{E}} \tag{2.31}$$

and can therefore be interpreted as a worst-case speed of contraction; it describes the slowest relative decay of the $\mathbf{E}$-norm of $\mathbf{x}(t)$ over time. Accordingly, if $\left\|e^{\mathbf{E}^{-1}\mathbf{A}t}\right\|_{\mathbf{E}}$ is not a strictly monotonic function, there are states $\mathbf{x}$ starting from which the system expands; a worst-case estimation can then not lead to a convergent envelope.

In fact, there is a strong connection between dissipativity and the concept of non-normality, as already noted in [116]: "When A is normal, [...] the 'hump' phenomenon does not exist. These observations lead us to conclude that the $e^{\mathbf{A}}$ problem is 'well conditioned' when A is normal." The "hump" refers to the norm of the matrix exponential

over time, as it can be seen in Figure 2.1. The first example exhibits strictly monotonic behavior, its matrix measure μ is negative and $\mathbf{A}$ is indeed normal: $\mathbf{A}\mathbf{A}^T = \mathbf{A}^T\mathbf{A}$. The bound $e^{\mu t}$ would be hardly distinguishable from the real function. Figure 2.1b) shows a counterexample: Here, $\mu \approx +1.9 \cdot 10^{+3}$, the matrix is highly non-normal, its exponential features a hump, and the upper bound $e^{\mu(t)}$ diverges so steeply it would leave Figure 2.1b) after 0.002 seconds—although all eigenvalues of $\mathbf{A}$ have negative real part.

Details on this connection and further references can also be found in [115, 152].

2.2.4. Computing the Matrix Measure

We conclude this section with remarks on the numerical computation of the generalized matrix measure. A MATLAB implementation can be seen in Source 2.1. It tries to perform a Cholesky decomposition of $-\mathbf{A} - \mathbf{A}^H$. If this breaks with an error, the matrix is not positive definite, accordingly $\mathbf{A}$ is not strictly dissipative and $\mu_{\mathbf{E}}(\mathbf{A})$ is not negative.

Otherwise, one needs to find the extremal solution of a generalized symmetric eigenvalue problem of dimension N. Due to the test for positive definiteness of $-\mathbf{A} - \mathbf{A}^H$, we know for sure that all generalized eigenvalues $\lambda_i(\mathrm{sym}\,\widetilde{\mathbf{A}}, \widetilde{\mathbf{E}})$ are negative, so the largest of them is the one closest to zero. Accordingly, instead of finding the eigenvalue with largest real part, we can look for the eigenvalue of smallest magnitude. In fact, despite the additional effort of performing a Cholesky decomposition, this works out much faster and also seems to be more robust than using the option "Largest Algebraic" (`'LA'`) of MATLAB's `eigs` routine.

Table 2.1.: Simulation Results for Computation of Matrix Measure

Model	Dimension	nnz($\mathbf{A}$)	nnz($\mathbf{E}$)	$\mu_{\mathbf{E}}(\mathbf{A})$	Time [s]
CD Player	120	240	120	$-0.024\,344$	0.020
FlowMeter (v=0)	9669	67 391	9669	-1.3993	0.27
Steel Profile	20 209	139 233	139 473	$-1.7969 \cdot 10^{-5}$	0.46
Gyroscope[2]	34 722	4 084 636	2 042 452	-49.9994	8.2

[2]In strictly dissipative realization as presented in Section 6.2 with $\gamma = 50$.

In Source 2.1, the `eigs` command is configured to exploit symmetry, and the start vector is determined to avoid random influences (see [150]). The tolerance and LANCZOS options are chosen according to [123]. The settings in Source 2.1 worked well for all considered examples. Results can be seen in Table 2.1.

Source 2.1: Computation of Generalized Spectral Abscissa $\mu_{\widetilde{\mathbf{E}}}(\widetilde{\mathbf{A}})$

```
function [mu, L_S, P_S] = SpectralAbscissa(A,E)
% Compute generalized spectral abscissa
%   Input:  A, E:     HFM matrices
%   Output: mu:       generalized spectral abscissa mu_E(A)
%           L_S, P_S: Cholesky decomposition of S=-A-A'
%

p    = 20;      % number of Lanczos vectors
tol  = 1e-10;   % convergence tolerance
opts = struct('issym',true, 'p',p, 'tol',tol, 'v0',sum(E,2));

[L_S,e,P_S] = chol(sparse(-A-A')); % L_S'*L_S = P_S'*(-A-A')*P_S
if e, error('A is not strictly dissipative.');
else
    symA = @(x) P_S*(L_S\(L_S'\(P_S'*x*2)));
    mu   = -eigs(symA, size(A,1), E, 1, 0, opts);
end
```

2.3. Projective MOR

Having summarized the system theoretic concepts that are most important for our purposes, we turn toward basics in model reduction.

2.3.1. Petrov-Galerkin Approximation

Projective MOR is based on the assumption that typically the state trajectory $\mathbf{x}(t)$ does not transit all regions of the state space equally often, but mainly constrains to stay within a subspace $\mathcal{V}$ of lower dimension. Given a basis $\mathbf{V} \in \mathbb{R}^{N\times n}$ of $\mathcal{V}$, one can therefore approximate the exact trajectory with a reduced state vector $\mathbf{x}_r(t)$ of dimension $n \ll N$:

$$\mathbf{x}(t) \approx \mathbf{V}\mathbf{x}_r(t). \tag{2.32}$$

The difference is usually denoted by the error

$$\mathbf{e}(t) := \mathbf{x}(t) - \mathbf{V}\mathbf{x}_r(t). \tag{2.33}$$

Replacing $\mathbf{x}(t)$ with its approximation turns the state equation from (2.1) into an overdetermined ODE and introduces a remainder $\boldsymbol{\epsilon}(t)$,

$$\mathbf{EV}\,\dot{\mathbf{x}}_r(t) = \mathbf{AV}\,\mathbf{x}_r(t) + \mathbf{B}\,\mathbf{u}(t) + \boldsymbol{\epsilon}(t). \tag{2.34}$$

As $\mathbf{EV}$, $\mathbf{AV}$, and $\mathbf{B}$ span different subspaces of $\mathbb{R}^N$, in general no trajectory $\mathbf{x}_r(t)$ can achieve exact equality. This is remedied by projecting (2.34) onto the n-dimensional subspace $\mathbf{EV}$ in which the left hand side of the equation lives. To this end, a projector matrix

$$\boldsymbol{\Pi} := \mathbf{EV}\left(\mathbf{W}^T\mathbf{EV}\right)^{-1}\mathbf{W}^T \quad \in \mathbb{R}^{N\times N} \tag{2.35}$$

is designed with the help of a matrix $\mathbf{W} \in \mathbb{R}^{N\times n}$, chosen such that $\det\left(\mathbf{W}^T\mathbf{EV}\right) \neq 0$. The projector $\boldsymbol{\Pi}$ then maps any vector $\mathbf{x}$ onto the subspace span $\mathbf{EV}$ along straight lines that are orthogonal to the subspace $\mathcal{W} := \operatorname{span}\mathbf{W}$; it fulfills $\boldsymbol{\Pi}^2 = \boldsymbol{\Pi}$. Obviously, when (2.34) is multiplied from the left by $\boldsymbol{\Pi}$, all resulting vectors lie in span $\mathbf{EV}$. The preceding term $\mathbf{EV}\left(\mathbf{W}^T\mathbf{EV}\right)^{-1}$ can therefore be omitted in all summands of the equation:

$$\mathbf{W}^T\mathbf{EV}\,\dot{\mathbf{x}}_r(t) = \mathbf{W}^T\mathbf{AV}\,\mathbf{x}_r(t) + \mathbf{W}^T\mathbf{B}\,\mathbf{u}(t) + \mathbf{W}^T\boldsymbol{\epsilon}(t). \tag{2.36}$$

As this regular reduced ODE system can be solved exactly by $\mathbf{x}_r(t)$ for given $\mathbf{u}_r(t)$, the residual term $\mathbf{W}^T\boldsymbol{\epsilon}(t)$ identically amounts to zero; this is known as Petrov-Galerkin condition.[3]

The final reduced order model (ROM) including the output equation is given by

$$\mathbf{G}_r : \begin{cases} \overbrace{\mathbf{W}^T\mathbf{EV}}^{\mathbf{E}_r}\,\dot{\mathbf{x}}_r(t) = \overbrace{\mathbf{W}^T\mathbf{AV}}^{\mathbf{A}_r}\mathbf{x}_r(t) + \overbrace{\mathbf{W}^T\mathbf{B}}^{\mathbf{B}_r}\,\mathbf{u}(t), \\ \mathbf{y}_r(t) = \underbrace{\mathbf{CV}}_{\mathbf{C}_r}\,\mathbf{x}_r(t) + \underbrace{\mathbf{D}}_{\mathbf{D}_r}\,\mathbf{u}(t). \end{cases} \tag{2.37}$$

Please note that the feedthrough matrix $\mathbf{D}$ of the original system is inherited by the ROM in this framework. It is also possible to choose $\mathbf{D}_r \neq \mathbf{D}$ and exploit the additional degrees of freedom [60], but then the $\mathcal{H}_2$ error (see below) is infinite. For that reason we restrict ourselves to the trivial choice $\mathbf{D}_r := \mathbf{D}$ in this thesis. This is also a more natural approach given that feedthrough is a static component of the transfer behavior and does not contribute to the dynamics one aims to approximate. For the ease of presentation, we will mostly assume $\mathbf{D} = \mathbf{0}$ in the following.

[3] Please note that the residual $\boldsymbol{\epsilon}(t)$ must not be confused with the error $\mathbf{e}(t)$. Clearly, $\mathbf{W}^T\mathbf{e}(t) \neq 0$!

Further details on the projective MOR framework can be found in many works, e. g. [8, 43, 73, 77] and references therein, to mention just a few of them.

Definition 2.13. *Projective MOR with* $\operatorname{span}\mathbf{V} = \operatorname{span}\mathbf{W}$*, or in particular* $\mathbf{V} = \mathbf{W}$*, is referred to as one-sided reduction.*

2.3.2. Error Model and Error Norms

Every ROM is associated with a corresponding error system:

Definition 2.14. *Given an original system* $\mathbf{G}(s)$ *and some reduced order model* $\mathbf{G}_r(s)$*, we define the associated error model* $\mathbf{G}_e(s)$ *as* $\mathbf{G}_e(s) := \mathbf{G}(s) - \mathbf{G}_r(s)$*.*

It is of high importance for the analysis of the approximation quality, which is typically judged by means of $\|\mathbf{G}_e\|$, where $\|\cdot\|$ denotes a system norm of interest.

We therefore define the absolute and relative error norms.

Definition 2.15. *The absolute* $\mathcal{H}_2$ *and* $\mathcal{H}_\infty$ *error norm is defined as*

$$\epsilon_{\mathcal{H}_2} := \|\mathbf{G}_e\|_{\mathcal{H}_2} \qquad \textit{and} \qquad \epsilon_{\mathcal{H}_\infty} := \|\mathbf{G}_e\|_{\mathcal{H}_\infty}, \tag{2.38}$$

respectively. The relative error norms are defined as

$$\epsilon_{\mathcal{H}_2,rel} := \frac{\|\mathbf{G}_e\|_{\mathcal{H}_2}}{\|\mathbf{G}\|_{\mathcal{H}_2}} \qquad \textit{and} \qquad \epsilon_{\mathcal{H}_\infty,rel} := \frac{\|\mathbf{G}_e\|_{\mathcal{H}_\infty}}{\|\mathbf{G}\|_{\mathcal{H}_\infty}}. \tag{2.39}$$

2.3.3. Invariance Properties

Before taking on the question of how to actually choose the projection matrices $\mathbf{V}$ and $\mathbf{W}$ in the above section, we recall an important property of projective MOR.

Lemma 2.3 ([139])**.** *Replacing the projection matrices* $\mathbf{V}, \mathbf{W}$ *with matrices* $\hat{\mathbf{V}}, \hat{\mathbf{W}} \in \mathbb{R}^{N\times n}$ *of the same dimension and spanning the same respective subspace,*

$$\operatorname{span}\mathbf{V} = \operatorname{span}\hat{\mathbf{V}} \quad \textit{and} \quad \operatorname{span}\mathbf{W} = \operatorname{span}\hat{\mathbf{W}},$$

results in another realization of the same ROM:

$$\hat{\mathbf{G}}_r(s) := \left[\begin{array}{c|c} \hat{\mathbf{W}}^T\mathbf{E}\hat{\mathbf{V}}, \hat{\mathbf{W}}^T\mathbf{A}\hat{\mathbf{V}} & \hat{\mathbf{W}}^T\mathbf{B} \\ \hline \mathbf{C}\hat{\mathbf{V}} & \mathbf{D} \end{array}\right] = \left[\begin{array}{c|c} \mathbf{W}^T\mathbf{E}\mathbf{V}, \mathbf{W}^T\mathbf{A}\mathbf{V} & \mathbf{W}^T\mathbf{B} \\ \hline \mathbf{C}\mathbf{V} & \mathbf{D} \end{array}\right] = \mathbf{G}_r(s).$$

Proof. The proof is repeated here because it helps understanding the effect of a change of basis in projective MOR. From the assumption follows the existence of regular matrices

$\mathbf{T}, \mathbf{M} \in \mathbb{R}^{n\times n}$ with $\hat{\mathbf{V}} = \mathbf{VT}$ and $\hat{\mathbf{W}} = \mathbf{WM}^T$. Accordingly,

$$\begin{aligned}\hat{\mathbf{G}}_r(s) &= \mathbf{C}\hat{\mathbf{V}}\left(s\hat{\mathbf{W}}^T\mathbf{E}\hat{\mathbf{V}} - \hat{\mathbf{W}}^T\mathbf{A}\hat{\mathbf{V}}\right)^{-1}\hat{\mathbf{W}}^T\mathbf{B} + \mathbf{D} = \\ &= \mathbf{CVT}\left(s\mathbf{MW}^T\mathbf{EVT} - \mathbf{MW}^T\mathbf{AVT}\right)^{-1}\mathbf{MW}^T\mathbf{B} + \mathbf{D} = \\ &= \mathbf{CV}\left(s\mathbf{W}^T\mathbf{EV} - \mathbf{W}^T\mathbf{AV}\right)^{-1}\mathbf{W}^T\mathbf{B} + \mathbf{D} = \mathbf{G}_r(s). \qquad \square\end{aligned}$$

The above lemma makes clear that in theory the input/output behavior of the ROM is invariant under change of basis of the projection subsets $\mathcal{V}$ and $\mathcal{W}$. The choice of the subsets themselves matters; different bases, however, only lead to different realizations, either in form of a state transformation $\mathbf{T}$ or pre-multiplication of the ODE system by $\mathbf{M}$, which does not even change the solution $\mathbf{x}_r(t)$.

On the other hand, the influence of numerical effects (roundoff errors) is highly dependent on the actual bases $\mathbf{V}$ and $\mathbf{W}$. It is therefore often inevitable to exploit the invariance property for judicious implementation in inexact arithmetics.

2.4. State of the Art and Problem Formulation

2.4.1. General Objectives and Challenges

We are now ready to express the main goals of MOR in more detail. They are threefold [8]:

- We want $\mathbf{G}_r(s)$ to approximate the original model well, which means the associated error shall be small with respect to a given norm. Also, information on the absolute or relative induced error is generally desirable.
- Properties of the HFM like stability, passivity, structure etc. shall be preserved.
- The reduction procedure must be numerically efficient (fast and robust) and automatable, i. e. it should not require interference from the user.

2.4.2. Some Selected Model Reduction Techniques

It is not in the scope of this thesis to provide extensive details on LTI model reduction methods that exist in the literature, as a wholehearted attempt to do so would alone fill dozens of pages. Indeed, an excellent comprehensive survey has been provided very recently by Baur, Benner and Feng [18], so for a detailed overview the reader is referred

to this contribution and the numerous references included, or to previous surveys like, for instance, [12, 22, 29, 75, 140], or the MOR standard work [8].

However, to classify the results presented in this thesis it is helpful to point out the practical problems that are related to the goals listed in Section 2.4.1, and to mention some strengths and weaknesses of the various MOR methods.

Remember the first objective: information on the approximation quality. Methods providing the exact $\mathcal{H}_2$ or $\mathcal{H}_\infty$ error or provable bounds on the respective norm require one or both system Gramians; examples are the Truncated Balanced Realization (TBR) [117] and its derivatives, or the Optimal Hankel Approximation [70]. Both, by the way, preserve stability of the HFM. Solving high-dimensional LYAPUNOV equations by direct methods, however, requires high numerical effort even for sparse matrices, and is therefore not in line with the third objective: numerical efficiency. In fact, in really large-scale settings, it is not practicable at all to compute a full-rank solution of a LYAPUNOV equation. Indeed, many methods exist to find approximate solutions in the form of low-rank factors—for instance, the Alternating Directions Implicit (ADI) iteration [103] or Krylov Plus Inverted Krylov (K-PIK), also known as Extended Krylov Subspace Method (EKSM), and Rational Krylov Subspace Method (RKSM) [48, 144]—, but then the rigorous error bounds do no longer hold and the ROM may even be unstable in theory.

Available error estimation techniques for situations in which no Gramian is available are covered in Section 5.1. In brief one can say that they are either not rigorous (i. e. not provably greater or equal to the true error), not global (only a narrow frequency band is considered), or quite restrictive in their assumptions ($\mathbf{A}$ and $\mathbf{E}$ symmetric; lossless system), cf. [18].

As to preservation of stability, one way to obtain stable ROMs without Gramians at hand is the explicit placement of the reduced poles in the left half complex plane. Modal truncation, for instance, features this intrinsically, and is in this regard superior.

In the range of interpolation-based reduction techniques, a combination of pole placement and moment matching was presented in [9] and can be used to preserve stability. The question *where* to place the poles, however, remains unanswered for the time being.

There is another approach of interest:

Lemma 2.4 ([142, 143])**.** *One-sided reduction of a dissipative model delivers a stable ROM.*

Proof. The proof is based on the fact that dissipativity is preserved by one-sided reduction. To show this, we must consider the respective conditions on $\mathbf{E}_r$ and $\mathbf{A}_r$. Symmetry of $\mathbf{E}_r$ is clear. Positive definiteness of $\mathbf{E}$ implies positive definiteness of $\mathbf{E}_r = \mathbf{V}^T\mathbf{E}\mathbf{V}$ because

$$\mathbf{x}^T\mathbf{E}\mathbf{x} > 0 \quad \forall \mathbf{x} \in \mathbb{R}^N \qquad \Rightarrow \qquad \mathbf{x}_r^T\mathbf{E}_r\mathbf{x}_r = \mathbf{x}_r^T\mathbf{V}^T\mathbf{E}\mathbf{V}\mathbf{x}_r > 0 \quad \forall \mathbf{x}_r \in \mathbb{R}^n.$$

Similarly, dissipativity of $\mathbf{A}$ implies

$$\begin{aligned} &\mathbf{x}^T(\mathbf{A} + \mathbf{A}^T)\mathbf{x}_r \leq 0 \quad \forall \mathbf{x} \in \mathbb{R}^N \\ \Rightarrow \quad &\mathbf{x}_r^T(\mathbf{A}_r + \mathbf{A}_r^T)\mathbf{x}_r = \mathbf{x}_r^T\mathbf{V}^T(\mathbf{A} + \mathbf{A}^T)\mathbf{V}\mathbf{x}_r < 0 \quad \forall \mathbf{x}_r \in \mathbb{R}^n. \end{aligned}$$

□

Remark 2.1. *Though* Silveira et al. *presented this result only for the special case of* $\mathbf{V}$ *resulting from the* Arnoldi *algorithm, the proof comprised general one-sided reduction. Still, it seems to have been re-worked several times since 1996.*

Odabasioglu et al., by the way, extended this finding to passivity in [119, 120].

The remarkable property of Lemma 2.4 is its generality. The statement holds, no matter whether $\mathbf{V}$ stems from Krylov subspace methods, Proper Orthogonal Decomposition (POD) or other subspace identification techniques. Two drawbacks, however, must be pointed out: Firstly, we have seen in Section 2.2.3 that finding a dissipative realization is hardly possible unless the property is given *a priori*. Secondly, the result of one-sided reduction is dependent on the realization of the HFM, so the procedure involves a certain randomness.

To sum up: preservation of stability and computation of reliable error information is closely related to the calculation of system Gramians, which, however, is time consuming or not even feasible for large-scale HFMs.

The goal of this thesis is therefore to develop Gramian-free methods which can satisfactorily fulfill the requirements of MOR. To this end, model reduction using Sylvester equations, which is based on *local* approximation of the HFM, is introduced in the next chapter. The subsequent Chapters 4 and 5 are then dedicated to the most important related issues: choice of order, stability preservation and rigorous error estimation.

3. Model Reduction based on Sylvester Equations

> *"These methods [model reduction techniques preserving meaningful parameters of the full order model] provide a (more or less) large and handy toolbox, instead of a single 'finished product'."*
>
> — C. De Villemagne and R. Skelton [43]

3.1. Historical Remark: Multipoint-Padé, Rational Krylov, and Sylvester Equations

The idea of creating reduced models that interpolate the HFM at certain frequency points of interest in the complex Laplace domain has a very rich history, see e. g. [73, 157]. Numerous methods to perform this kind of model reduction have evolved over the 20th century and are known in the literature as Partial Realization, (standard, shifted, or multipoint) Padé approximation, rational interpolation, asymptotic waveform evaluation (AWE) and Padé via Lanczos (PVL) [55], to mention just the most famous of them.

"A large amount of existing work was repeated", as Grimme remarks [73], which is plausible from a system theoretic point of view: in the SISO case, for instance, a ROM is uniquely determined by $2n$ well-posed conditions, so the various reduction methods can only differ from each other in the formulation (i. e., the particular state space realization or—in earlier days—rational transfer function) they deliver, and in the numerical properties of the procedure, but *not* in the reduced model itself. In fact, explicit moment matching algorithms like the AWE are mathematically equivalent to PVL, but numerically

ill-conditioned and unstable [65]—even some time-domain based reduction approaches like Laguerre methods [90] have been shown to be equivalent to PADÉ approximation [50].

Meanwhile, the projective MOR framework with rational KRYLOV subspace methods in combination with orthogonalization procedures has turned out to constitute a robust and general implementation of PADÉ approximation both for the SISO and the MIMO case. Some of the most important contributions towards this result are due to RUHE [135], DE VILLEMAGNE and SKELTON [43], GRIMME [73], BAI, FELDMANN and FREUND [14, 55, 65], and GALLIVAN, VANDENDORPE and VAN DOOREN [66, 67, 157].

In addition, GALLIVAN, VANDENDORPE and VAN DOOREN have presented the tight connection of rational KRYLOV subspaces and SYLVESTER equations. As a matter of fact, given the dual SYLVESTER equations

$$\mathbf{A}\mathbf{V}\mathbf{R}_1 + \mathbf{E}\mathbf{V}\mathbf{R}_2 + \mathbf{B}\mathbf{Y} = \mathbf{0}, \tag{3.1}$$

$$\mathbf{L}_1\mathbf{W}^T\mathbf{A} + \mathbf{L}_2\mathbf{W}^T\mathbf{E} + \mathbf{X}\mathbf{C} = \mathbf{0}, \tag{3.2}$$

with $\mathbf{R}_1, \mathbf{R}_2, \mathbf{L}_1, \mathbf{L}_2 \in \mathbb{R}^{n\times n}$, $\mathbf{Y} \in \mathbb{R}^{m\times n}$, and $\mathbf{X} \in \mathbb{R}^{n\times p}$, the solutions $\mathbf{V} \in \mathbb{R}^{N\times n}$ and $\mathbf{W} \in \mathbb{R}^{N\times n}$ can be written as sums of bases of KRYLOV subspaces and eigenspaces [68, 69, 157]. Though we will not explicitly solve SYLVESTER equations by numerical methods as presented in [145], we will see below that they provide a beautiful unifying formulation and are of great theoretical importance, cf. [13].

Note that a holistic perspective on the *realization problem* has been provided by ANTOULAS and ANDERSON by means of the LÖWNER framework [7, 8], which is not only suitable in the context of model reduction by rational interpolation (including tangential interpolation and descriptor systems) [110], but also for identification purposes of linear or parameter-dependent systems [98, 99, 100].

A very recent survey on model reduction by rational interpolation is due to BEATTIE and GUGERCIN [22].

3.2. Projective MOR and Sylvester Equations

It is not in the scope of this dissertation to exhaustively study the links and connections between the various MOR procedures related to SYLVESTER equations. Neither does this chapter present substantial new results on that topic (except Section 3.3).

Yet the actual results of this thesis essentially base upon Sylvester equations. The error bounds presented in Chapter 5, for instance, only apply when a particular Sylvester equation is fulfilled and its solution is known. This is, however, indeed the case for common rational Krylov subspace methods (leading to "moment matching") as well as for modal truncation (preserving eigenvalues of interest).

This section is therefore intended to recall preliminary results on how the mentioned reduction techniques can be formalized with the help of Sylvester equations. In Section 3.3 we will then modify the deduced equations towards a MOR-related formulation that meets the requirement of the subsequent chapters.

3.2.1. Transformations of Sylvester Equations

Assume we want to use the solution $\mathbf{V}$ of the Sylvester equation (3.1) for projective MOR. Then, according to Lemma 2.3, the ROM does not change in its transfer behavior when we replace $\mathbf{V}$ with another basis $\hat{\mathbf{V}}$ of the same subset.

Let therefore $\mathbf{T}$ be a regular matrix of generalized eigenvectors of $(\mathbf{R}_2, \mathbf{R}_1)$, such that $\mathbf{R}_2\mathbf{T} = \mathbf{R}_1\mathbf{T}\mathbf{S}_V$ holds and $\mathbf{S}_V$ is a matrix in Jordan canonical form. Then, define $\hat{\mathbf{V}}$ such that $\mathbf{V} = \hat{\mathbf{V}}\mathbf{T}^{-1}\mathbf{R}_1^{-1}$. This changes the Sylvester equation (3.1) to

$$\mathbf{A}\hat{\mathbf{V}}\mathbf{T}^{-1} + \mathbf{E}\hat{\mathbf{V}}\mathbf{T}^{-1}\mathbf{R}_1^{-1}\mathbf{R}_2 + \mathbf{B}\mathbf{Y} = \mathbf{0}.$$

Multiplication from the right by $\mathbf{T}$ equivalently yields

$$\mathbf{A}\hat{\mathbf{V}} + \mathbf{E}\hat{\mathbf{V}}\underbrace{\mathbf{T}^{-1}\mathbf{R}_1^{-1}\mathbf{R}_2\mathbf{T}}_{\mathbf{S}_V} + \mathbf{B}\underbrace{\mathbf{Y}\,\mathbf{T}}_{\hat{\mathbf{Y}}} = \mathbf{0} \qquad \Leftrightarrow \qquad \mathbf{A}\hat{\mathbf{V}} + \mathbf{E}\hat{\mathbf{V}}\mathbf{S}_V + \mathbf{B}\hat{\mathbf{Y}} = \mathbf{0}.$$

Accordingly, if $\mathbf{R}_1$ is regular, we can always find a related formulation of Sylvester equation (3.1) with $\hat{\mathbf{R}}_1 = \mathbf{I}_n$ and a particular $\hat{\mathbf{R}}_2 = \mathbf{S}_V$ such that the column span of the solution—and therefore the resulting ROM in projective reduction—remains unchanged. Note that the generalized spectrum is invariant under such transformation: $\lambda_i(\mathbf{R}_2, \mathbf{R}_1) = \lambda_i(\hat{\mathbf{R}}_2, \hat{\mathbf{R}}_1) = \lambda_i(\mathbf{S}_V)$.

In practice, one will therefore usually not use the solution $\hat{\mathbf{V}}$ of this normalized formulation (which may easily be ill-conditioned) but conduct an orthogonalization of its columns instead, for instance using a (modified) Gram-Schmidt process [136]. This again does not affect the transfer function of the ROM, but enables a numerically stable procedure.

Still, the normalized formulation is of great theoretical value. In the next subsections we will briefly discuss various MOR procedures and specify the particular SYLVESTER equation they implicitly solve. We will assume the following form and notation[1]:

$$\mathbf{AV} + \mathbf{EV}(-\mathbf{S}_V) + \mathbf{B}(-\widetilde{\mathbf{C}}_r) = \mathbf{0}. \tag{3.3}$$

Please note that all results carry over to the dual SYLVESTER equation (3.2)

$$\mathbf{W}^T\mathbf{A} + (-\mathbf{S}_W)\mathbf{W}^T\mathbf{E} + (-\widetilde{\mathbf{B}}_r)\mathbf{C} = \mathbf{0}. \tag{3.4}$$

3.2.2. Particular Sylvester Equations

We will see in the following that KRYLOV subspaces and invariant subspaces fulfill particular SYLVESTER equations.

Rational Krylov Subspaces

Definition 3.1. *Consider a SISO LTI system with* $\mathbf{B} = \mathbf{b} \in \mathbb{R}^{N\times 1}$, $\mathbf{C} = \mathbf{c} \in \mathbb{R}^{1\times N}$. *For a given shift* $\sigma \in \mathbb{C}$ *and multiplicity* $n \in \mathbb{N}$, *the corresponding rational input* KRYLOV *subspace is defined as the column space of*

$$\mathbf{V} := \left[\mathbf{A}_\sigma^{-1}\mathbf{b}, \quad \mathbf{A}_\sigma^{-1}\mathbf{E}\mathbf{A}_\sigma^{-1}\mathbf{b}, \quad \dots \quad \left(\mathbf{A}_\sigma^{-1}\mathbf{E}\right)^{n-1}\mathbf{A}_\sigma^{-1}\mathbf{b}\right] \in \mathbb{C}^{N\times n}, \tag{3.5}$$

where $\mathbf{A}_\sigma := \mathbf{A} - \sigma\mathbf{E}$, *while the rational output* KRYLOV *subspace is the column space of*

$$\mathbf{W} := \left[\mathbf{A}_\sigma^{-T}\mathbf{c}^T, \quad \mathbf{A}_\sigma^{-T}\mathbf{E}^T\mathbf{A}_\sigma^{-T}\mathbf{c}^T, \quad \dots \quad \left(\mathbf{A}_\sigma^{-T}\mathbf{E}^T\right)^{n-1}\mathbf{A}_\sigma^{-T}\mathbf{c}^T\right] \in \mathbb{C}^{N\times n}. \tag{3.6}$$

For the basis $\mathbf{V}$ in (3.5), the SYLVESTER equation (3.3) holds with

$$\mathbf{S}_V = \begin{bmatrix} \sigma & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \sigma \end{bmatrix} \in \mathbb{C}^{n\times n} \qquad \text{and} \qquad \widetilde{\mathbf{C}}_r = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{1\times n},$$

and $\mathbf{W}$ fulfills the dual equation(3.4) with $\mathbf{S}_W = \mathbf{S}_V^T$ and $\widetilde{\mathbf{B}}_r = \widetilde{\mathbf{C}}_r^T$. In fact, an output KRYLOV subspace can always be understood as the dual counterpart of an input KRYLOV subspace which results from $\mathbf{A} \to \mathbf{A}^T$, $\mathbf{E} \to \mathbf{E}^T$, $\mathbf{B} \to \mathbf{C}^T$, so we will only consider the input case in the following.

[1] It will become more clear in Chapter 4 why we name the matrices $\widetilde{\mathbf{C}}_r$ instead of $\mathbf{Y}$.

Note that the columns of $\mathbf{V}$ in (3.5) are given recursively by $\mathbf{v}_i = \mathbf{A}_{\sigma_i}^{-1}\mathbf{E}\mathbf{v}_{i-1}$ for $i \geq 2$. This is the reason why this basis can not only be orthogonalized *a posteriori* by means of a GRAM-SCHMIDT procedure, but one can use a *modified* GRAM-SCHMIDT algorithm, which means that every new column is orthogonalized and normalized, before the next column is computed (cf. Source 3.1) [136]. This does not change the subspace, but is numerically much more stable.

Remark 3.1. *Please note that in the literature, "shifts" and "expansion points" are sometimes defined differently, namely with converse sign. In this thesis, both expressions mean the same thing.*

Multipoint Rational Krylov Subspaces

A true generalization is given by means of a multipoint rational KRYLOV subspace, which can, however, take two different forms. Let some shifts be given by σ_i, $i = 1 \dots n$. Then one can replace $\mathbf{A}_\sigma$ in (3.5) by $\mathbf{A}_{\sigma_i}$, which yields

$$\mathbf{V} := \begin{bmatrix} \mathbf{A}_{\sigma_1}^{-1}\mathbf{b}, & \mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}, & \dots & \mathbf{A}_{\sigma_n}^{-1}\mathbf{E}\cdots\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} \end{bmatrix} \in \mathbb{C}^{N\times n}. \tag{3.7}$$

This matrix fulfills (3.3) with

$$\mathbf{S}_V = \begin{bmatrix} \sigma_1 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \sigma_n \end{bmatrix} \in \mathbb{C}^{n\times n} \qquad \text{and} \qquad \widetilde{\mathbf{C}}_r = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{1\times n}. \tag{3.8}$$

More widely spread, however, is the usage of the basis

$$\mathbf{V} := \begin{bmatrix} \mathbf{A}_{\sigma_1}^{-1}\mathbf{b}, & \mathbf{A}_{\sigma_2}^{-1}\mathbf{b}, & \dots & \mathbf{A}_{\sigma_n}^{-1}\mathbf{b} \end{bmatrix} \in \mathbb{C}^{N\times n}, \tag{3.9}$$

which spans the same subspace, but leads to diagonal $\mathbf{S}_V$ in (3.3); more precisely,

$$\mathbf{S}_V = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \in \mathbb{C}^{n\times n} \qquad \text{and} \qquad \widetilde{\mathbf{C}}_r = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} \in \mathbb{R}^{1\times n}. \tag{3.10}$$

Note, however, that the latter basis is at risk of being ill-conditioned if $\sigma_i \approx \sigma_j$ for some $i \neq j$, because orthogonalization can only be performed *a posteriori*, while the first formulation (3.7) can again be combined with a *modified* GRAM-SCHMIDT procedure [136] and incorporates the special case that all (or some!) σ_i are equal. On the other hand, parallelization is only possible with the second formulation of the SYLVESTER equation,

because for the computation of the columns of $\mathbf{V}$ in (3.7) one requires the preceding column, which forces serial implementation.

Tangential Krylov Subspaces

Given a MIMO LTI system, one can choose a tangential vector $\mathbf{t} \in \mathbb{C}^{m\times 1}$ and treat $\mathbf{Bt} \in \mathbb{C}^{N\times 1}$ like the input vector of a SISO system. The KRYLOV subspaces defined above in (3.5), (3.7), (3.9) then fulfill SYLVESTER equation (3.3) with respective $\mathbf{S}_V$ like before; $\widetilde{\mathbf{C}}_r$, however, becomes a matrix (it had only one row for SISO) and changes accordingly to

$$\widetilde{\mathbf{C}}_r = \begin{bmatrix} \mathbf{t} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix} \in \mathbb{C}^{m\times n} \qquad \text{or} \qquad \widetilde{\mathbf{C}}_r = \begin{bmatrix} \mathbf{t} & \mathbf{t} & \dots & \mathbf{t} \end{bmatrix} \in \mathbb{C}^{m\times n}.$$

It is, however, also possible to choose one tangential direction $\mathbf{t}_i$ per shift σ_i. Note that then the "cascaded" basis (3.7) cannot be used anymore, but rather the following form:

$$\mathbf{V} := \begin{bmatrix} \mathbf{A}_{\sigma_1}^{-1}\mathbf{B}\mathbf{t}_1, & \mathbf{A}_{\sigma_2}^{-1}\mathbf{B}\mathbf{t}_2, & \dots & \mathbf{A}_{\sigma_n}^{-1}\mathbf{B}\mathbf{t}_n \end{bmatrix} \in \mathbb{C}^{N\times n} \tag{3.11}$$

with

$$\mathbf{S}_V = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \in \mathbb{C}^{n\times n} \quad \text{and} \quad \widetilde{\mathbf{C}}_r = \begin{bmatrix} \mathbf{t}_1 & \dots & \mathbf{t}_n \end{bmatrix} \in \mathbb{C}^{m\times n}. \tag{3.12}$$

For that reason, it is not easily possible to use a modified GRAM-SCHMIDT algorithm here, so $\mathbf{V}$ cannot be orthogonalized in a straightforward way like before, but requires an *a posteriori* GRAM-SCHMIDT procedure. An implementation can be seen in Source 3.3.

Block Krylov Subspaces

Alternatively, one can replace $\mathbf{b}$ in (3.5), (3.7), (3.9) by the whole matrix $\mathbf{B}$ in the MIMO case, which increases the number of columns of $\mathbf{V}$ by m per shift. $\mathbf{V}$ is then called a block input KRYLOV subspace and the respective matrices in (3.3) read

$$\mathbf{S}_V = \begin{bmatrix} \sigma_1\mathbf{I}_m & \mathbf{I}_m & & \\ & \ddots & \ddots & \\ & & \ddots & \mathbf{I}_m \\ & & & \sigma_n\mathbf{I}_m \end{bmatrix} \in \mathbb{C}^{mn\times mn} \quad \text{and} \quad \widetilde{\mathbf{C}}_r = \begin{bmatrix} \mathbf{I}_m & \mathbf{0}_m & \dots & \mathbf{0}_m \end{bmatrix} \in \mathbb{C}^{m\times mn}$$

or, respectively,

$$\mathbf{S}_V = \begin{bmatrix} \sigma_1 \mathbf{I}_m & & \\ & \ddots & \\ & & \sigma_n \mathbf{I}_m \end{bmatrix} \in \mathbb{C}^{mn \times mn} \quad \text{and} \quad \widetilde{\mathbf{C}}_r = \begin{bmatrix} \mathbf{I}_m & \mathbf{I}_m & \dots & \mathbf{I}_m \end{bmatrix} \in \mathbb{C}^{m \times mn}.$$

Note that both for tangential and block Krylov subspaces, the columns of the matrix $\mathbf{V}$ may become linearly dependent. In this case, deflating techniques must be applied to find a projection matrix of full column rank [136].

Invariant Subspaces

If $\mathbf{V}$ spans an invariant subspace or generalized eigenspace, it fulfills

$$\mathbf{AV} = \mathbf{EV}\boldsymbol{\Lambda}, \tag{3.13}$$

where $\boldsymbol{\Lambda} \in \mathbb{C}^{n \times n}$ is a diagonal matrix containing n generalized eigenvalues of $(\mathbf{A}, \mathbf{E})$, or possibly a Jordan matrix for defective eigenvalues of higher multiplicity. Anyway, this equation is the special case of the Sylvester equation (3.3) for $\mathbf{S}_V = \boldsymbol{\Lambda}$ and $\widetilde{\mathbf{C}}_r = \mathbf{0}_{m \times n}$.

3.2.3. Related Model Reduction Techniques

In this section, we will recall the meaning of the Sylvester equations from above on a ROM when the respective solutions are used as projection matrices according to (2.36). For details, please refer to [69].

Rational Interpolation / Moment Matching / Padé approximation

Consider the SISO case first. It is well known that projective MOR with a right hand side matrix $\mathbf{V}$ as in (3.5) will—independently of the choice of $\mathbf{W}$, as long as all inverses exist—deliver a ROM which satisfies the interpolation conditions

$$\mathbf{G}^{(i)}(\sigma) = \mathbf{G}_r^{(i)}(\sigma) \quad \Leftrightarrow \quad \boldsymbol{\eta}_i^{\sigma} = \boldsymbol{\eta}_{r,i}^{\sigma} \qquad \forall\, i = 0,\, 1, \dots (n-1), \tag{3.14}$$

or, in other words, whose first n moments $\boldsymbol{\eta}_{r,i}^{\sigma}$ about σ equal those of the HFM; this is known as (singlepoint) Padé approximation.

If $\mathbf{V}$ spans a rational input Krylov subspace as (3.7), (3.9), then the ROM interpolates the transfer function of the HFM at the various frequencies σ_i in the Laplace complex plane, which is referred to as multipoint Padé approximation.

More generally, one also speaks of "moment matching" or "rational interpolation", because the reduced (rational) transfer function $\mathbf{G}_r(s)$ interpolates the much more complicated transfer function $\mathbf{G}(s)$ of the original systems.

The results extend to the MIMO case if block KRYLOV subspaces are employed. Here, $\mathbf{G}^{(i)}(\sigma)$ and $\mathbf{G}_r^{(i)}(\sigma)$ in (3.14) are indeed complex matrices of dimension $p \times m$.

Also, note that rational interpolation works the same way if $\mathbf{W}$ spans an output KRYLOV subspace, so all results carry over to this dual case. For a summary see [139].

Tangential Interpolation

Using a tangential KRYLOV subspace (3.11) for $\mathbf{V}$ leads to so-called tangential interpolation, i. e.

$$\mathbf{G}^{(i)}(\sigma)\mathbf{t} = \mathbf{G}_r^{(i)}(\sigma)\mathbf{t} \quad \Leftrightarrow \quad \boldsymbol{\eta}_i^{\sigma}\mathbf{t} = \boldsymbol{\eta}_{r,i}^{\sigma}\mathbf{t} \qquad \forall\ i = 0,\ 1, \ldots (n-1) \tag{3.15}$$

for singlepoint PADÉ, or for general multipoint PADÉ:

$$\mathbf{G}(\sigma_i)\mathbf{t}_i = \mathbf{G}_r(\sigma_i)\mathbf{t}_i \quad \Leftrightarrow \quad \boldsymbol{\eta}_0^{\sigma_i}\mathbf{t}_i = \boldsymbol{\eta}_{r,0}^{\sigma_i}\mathbf{t}_i \qquad \forall\ i = 1,\ 2, \ldots n. \tag{3.16}$$

This means that the respective moments (matrices in $\mathbb{C}^{p\times m}$) of the ROM and the HFM are not equal, but the linear combinations defined by $\mathbf{t}_i \in \mathbb{C}^{m\times 1}$ of their columns are. [67]

Using a tangential output KRYLOV subspace with some tangential vector $\mathbf{t}_{W,i} \in \mathbb{C}^{1\times p}$ leads to dual interpolation behavior: $\mathbf{t}_{W,i}\mathbf{G}^{(i)}(\sigma) = \mathbf{t}_{W,i}\mathbf{G}_r^{(i)}(\sigma)$ etc..

Tangential interpolation is a true generalization of block moment matching. Performing tangential interpolation with all n unit vectors $\mathbf{e}_i$ instead of only one vector $\mathbf{t}$ is the same as building a block KRYLOV subspace by using the whole matrix $\mathbf{B}$ at a time.

Two-Sided Rational Krylov / Hermite Interpolation

If both $\mathbf{V}$ and $\mathbf{W}$ span input and output KRYLOV subspaces, respectively, then the single implications on the interpolation behavior "sum up", i. e. the ROM matches moments at the eigenvalues both of $\mathbf{S}_V$ and $\mathbf{S}_W$. If eigenvalues coincide, then higher moments are matched at the respective frequency—at least in the SISO case; in the MIMO case, the situation depends on the tangential vectors.

Two-sided rational interpolation for MIMO systems is generally a comparably inconvenient task, because one has to find matrices $\mathbf{V}$ and $\mathbf{W}$ with the same number of columns.

Input and output block KRYLOV subspaces, however, have $n \cdot m$ and $n \cdot p$ columns, respectively, so in general one cannot use the same number of moments in $\mathbf{V}$ and $\mathbf{W}$. Tangential interpolation circumvents this problem, but may still suffer from loss of rank in general. This can be counteracted by deflation, but then the linearly dependent columns must be replaced *ad hoc* by additional information, if necessary.

The special SISO case that the same expansion points σ_i are used for $\mathbf{V}$ and for $\mathbf{W}$ is called HERMITE interpolation. Here, the ROM fulfills

$$\mathbf{G}(\sigma_i) = \mathbf{G}_r(\sigma_i) \qquad \text{and} \qquad \tfrac{\mathrm{d}}{\mathrm{d}s}\mathbf{G}(\sigma_i) = \tfrac{\mathrm{d}}{\mathrm{d}s}\mathbf{G}_r(\sigma_i) \qquad \forall\ i = 1,\, 2, \ldots n. \tag{3.17}$$

Modal Truncation

If the basis of an invariant subspace is used for projective model reduction, the eigenvalues of the HFM contained in $\mathbf{\Lambda}$ are preserved by the ROM. If both $\mathbf{V}$ and $\mathbf{W}$ span right and left handed invariant subspaces, respectively, then the respective eigenvalues are preserved together with their residuals. Obviously, this kind of model reduction offers far less degrees of freedom than KRYLOV subspace methods, because one can only choose among the eigenvalues of the HFM. Although modal reduction methods provide insight in the physical meaning of the reduced state variables, this goal is typically subordinate to a precise approximation of the transmission behavior—in particular for models whose modes, unlike lightly damped mechanical structures, do not have a clear physical interpretation.

Modal truncation is therefore not in the focus of this thesis, but rather mentioned here to show that the results of subsequent chapters apply to this model reduction technique as well as to KRYLOV subspace methods. For details, see [33, 105, 118, 146, 158] and references therein.

3.2.4. Important Properties

In general, solutions of SYLVESTER equations can be used to match moments at certain frequencies and preserve given eigenvalues at the same time. To this end, the respective bases are simply concatenated and the various implications for the ROM "sum up". In fact, most of the results presented in the following chapters apply to MOR by arbi-

trary Sylvester equation. Sometimes, however, it is important to distinguish Krylov subspaces from eigenspaces.

Lemma 3.1 ([157]). *If* $\mathbf{V}$ *spans an input* Krylov *subspace and* $\mathbf{S}_V, \widetilde{\mathbf{C}}_r$ *fulfill* Sylvester *equation* (3.3)*, then the pair* $(\widetilde{\mathbf{C}}_r, \mathbf{S}_V)$ *is observable.*

If $\mathbf{W}$ *spans an output* Krylov *subspace and* $\mathbf{S}_W, \widetilde{\mathbf{B}}_r$ *fulfill* Sylvester *equation* (3.4)*, then the pair* $(\mathbf{S}_W, \widetilde{\mathbf{B}}_r)$ *is controllable.*

This can best be seen in a formulation of the Sylvester equation in which $\mathbf{S}_V$ or $\mathbf{S}_W$, respectively, is diagonal. If right or left eigenvectors are contained in the subspace spanned by $\mathbf{V}$ or $\mathbf{W}$, respectively, then the corresponding rows in $\widetilde{\mathbf{C}}_r$ or columns in $\widetilde{\mathbf{B}}_r$ are zero.

Also, we recall some additional invariance properties extending the results from Section 2.3.3.

Lemma 3.2. *Let an ODE system* (2.1) *be pre-multiplied by a regular matrix* $\mathbf{T}$ *from the left; this yields the equivalent realization* $\widetilde{\mathbf{E}} = \mathbf{TE}$, $\widetilde{\mathbf{A}} = \mathbf{TA}$, $\widetilde{\mathbf{B}} = \mathbf{TB}$, $\widetilde{\mathbf{C}} = \mathbf{C}$. *Then:*

- *The solution of an input type* Sylvester *equation* (3.3) *does not change,* $\widetilde{\mathbf{V}} = \mathbf{V}$.
- *The solution of an output type* Sylvester *equation* (3.4) *fulfills* $\widetilde{\mathbf{W}} = \mathbf{T}^{-T}\mathbf{W}$.

Lemma 3.3 ([139]). *In two-sided* Sylvester*-based MOR, the ROM is independent of the realization of the HFM.*

3.2.5. Judicious Implementation

Although only the matrices $\mathbf{V}$ and $\mathbf{W}$ are needed to perform projective MOR, some of the techniques presented in this thesis require the accompanying matrices $\mathbf{S}_V$, $\widetilde{\mathbf{C}}_r$ and $\mathbf{S}_W$, $\widetilde{\mathbf{B}}_r$. These, however, can be set up during the calculation of $\mathbf{V}$ and $\mathbf{W}$, respectively, without costly additional computations.

A possible MATLAB implementation to calculate orthogonal bases of input and output Krylov subspaces is given in Source 3.1. It uses a modified Gram-Schmidt algorithm Source 3.2 for the orthogonalization, during which $\mathbf{S}_V$, $\widetilde{\mathbf{C}}_r$, $\mathbf{S}_W$ and $\widetilde{\mathbf{B}}_r$ are properly adapted. An implementation for tangential multipoint Padé approximation can be seen in Source 3.3, which includes the respective SISO case, but requires $\sigma_i \neq \sigma_j$ for $i \neq j$. Of course these listings can be modified to include the other cases presented above.

Source 3.1: Rational Krylov Subspace

```
function [V,S_V,Crt,W,S_W,Brt] = RationalKrylov(A,B,C,E,s0,n)
% Input and Output Krylov subspace
%   Input:  A,B,C,E:   HFM matrices;
%           s0; n:     (real) shift; dimension of ROM
%   Output: A*V - E*V*S_V - B*Crt = 0,  W.'*A - S_W*W.'*E - Brt*C = 0
%

% initialization and preallocation
N=size(A,1); V=zeros(N,n); S_V=zeros(n,n); Crt=eye(1,n); tempV = B;
             W=zeros(N,n); S_W=zeros(n,n); Brt=eye(n,1); tempW = C.';
[L,U,P,Q] = lu(sparse(A-s0*E)); % ==> P*A*Q = L*U

for i=1:n
    % compute new basis vector
    tempV  = Q*(U\(L\(P*tempV)));    tempW = (tempW.'*Q/U/L*P).';
    V(:,i) = tempV; S_V(i,i) = s0;   W(:,i)  = tempW; S_W(i,i) = s0;
    if i>1, S_V(i-1,i)=1; S_W(i,i-1)=1; end
    % orthogonalize new column
    [V,S_V,Crt] = GramSchmidt(V,S_V,Crt,[i i]);
    [W,S_W,Brt] = GramSchmidt(W,S_W.',Brt.',[i i]); S_W=S_W.'; Brt=Brt.';
    tempV = E*V(:,i);                tempW = E'*W(:,i);
end
```

Source 3.2: Gram Schmidt Orthogonalization

```
function [X,Y,Z] = GramSchmidt(X,Y,Z,cols)
% Gram-Schmidt orthonormalization
%   Input:  X,[Y,[Z]]:  matrices in Sylvester eq.: V,S_V,Crt or W.',S_W.',Brt.'
%           cols:       2-dim. vector: number of first and last column to be treated
%   Output: X,[Y,[Z]]:  solution of Sylvester eq. with X.'*X = I
%

if nargin<4, cols=[1 size(X,2)]; end
for k=cols(1):cols(2)
    for j=1:(k-1)                       % orthogonalization
        T = eye(size(X,2)); T(j,k)=-X(:,k)'*X(:,j);
        X = X*T;
        if nargout>=2, Y=T\Y*T; end
        if nargout>=3, Z=Z*T; end
    end
    h = norm(X(:,k));  X(:,k)=X(:,k)/h; % normalization
    if nargout>=2, Y(:,k) = Y(:,k)/h; Y(k,:) = Y(k,:)*h; end
    if nargout==3, Z(:,k) = Z(:,k)/h; end
end
```

Source 3.3: Multipoint Padé via Rational Krylov

```
function [V,S_V,Crt,W,S_W,Brt] = TangentialKrylov(A,B,C,E,s0,t_B,t_C)
% Tangential Input and Output Rational Krylov Subspaces
%   Input:  A,B,C,E: HFM matrices;
%           s0:      vector of shifts;
%           t_B,t_C: matrices of tangential directions as column/row vectors
%   Output: A*V - E*V*S_V - B*Crt = 0,  W.'*A - S_W*W.'*E - Brt*C = 0
%

% initialization and preallocation
N=size(A,1); n=length(s0); m=size(B,2); p=size(C,1); i=1;
V=zeros(N,n); S_V=zeros(n,n); Crt=zeros(m,n); W=V; S_W=S_V; Brt=zeros(n,p);

while i<=n
    s = s0(i);
    % compute new basis vectors
    [L,U,P,Q,R] = lu(sparse(A-s*E)); % ==> P*(R\A)*Q = L*U
    tempV = Q*(U\(L\(P*(R\(B*t_B(:,i))))));
    tempW = (t_C(i,:)*C*Q/U/L*P/R).';
    if ~isreal(s)   % complex conjugated pair of shifts -> two columns
        V(:,i:(i+1))    = [real(tempV), imag(tempV)];
        Crt(:,i:(i+1))  = [real(t_B(:,i)), imag(t_B(:,i))];
        S_V(i:(i+1),i:(i+1)) = [real(s), imag(s); -imag(s), real(s)];
        W(:,i:(i+1))    = [real(tempW), imag(tempW)];
        Brt(i:(i+1),:)  = [real(t_C(i,:)); imag(t_C(i,:))];
        S_W(i:(i+1),i:(i+1)) = [real(s), -imag(s); imag(s), real(s)];
        i = i+2;
    else            % real shift -> one column
        V(:,i)  = real(tempV);  Crt(:,i) = real(t_B(:,i)); S_V(i,i) = s;
        W(:,i)  = real(tempW);  Brt(i,:) = real(t_C(i,:)); S_W(i,i) = s;
        i = i+1;
    end
end
% orthogonalization
[V,S_V,Crt] = GramSchmidt(V,S_V,Crt,[1 n]);
[W,S_W,Brt] = GramSchmidt(W,S_W.',Brt.',[1 n]); S_W=S_W.'; Brt=Brt.';
```

In order to verify if the respective SYLVESTER equation is fulfilled, by the way, one can evaluate its residual:

```
norm(A*V   - E*V*S_V   - B*Crt) / norm(A*V)
norm(W.'*A - S_W*W.'*E - Brt*C) / norm(W.'*A)
```

Accordingly, the accompanying small-scale matrices can be assumed to be available in general—at almost zero additional cost. If, as a matter of fact, this should not be the case, one can also compute them *a posteriori*, as will be shown at the end of the following section.

3.3. A Novel Formulation of the Sylvester Equation

We have seen that many projective MOR methods employ matrices $\mathbf{V}$, $\mathbf{W}$ that solve generalized SYLVESTER equations (3.3),(3.4). This is a crucial property for the results derived in the rest of this thesis. However, for our purposes we will require the SYLVESTER equations to take a slightly different form. This novel formulation was presented in [166] for the special case of KRYLOV subspaces, but holds in general.

Let $\mathbf{V}$ fulfill (3.3) for some known $\mathbf{S}_V, \widetilde{\mathbf{C}}_r$ and let $\mathbf{W}$ be an arbitrary matrix such that $\det \mathbf{W}^T\mathbf{E}\mathbf{V} \neq 0$. Then a projector $\mathbf{\Pi}$ is defined according to (2.35). Now multiply (3.3) from the left with the complementary projector $(\mathbf{I}_N - \mathbf{\Pi})$:

$$(\mathbf{I}_N - \mathbf{\Pi})\,\mathbf{A}\mathbf{V} - (\mathbf{I}_N - \mathbf{\Pi})\,\mathbf{E}\mathbf{V}\mathbf{S}_V - \underbrace{(\mathbf{I}_N - \mathbf{\Pi})\,\mathbf{B}}_{\mathbf{B}_\perp}\widetilde{\mathbf{C}}_r = \mathbf{0}. \tag{3.18}$$

Due to the properties of the projector ($\mathbf{E}\mathbf{V} = \mathbf{\Pi}\mathbf{E}\mathbf{V}$, cf. Section 2.3.1), the second summand is zero. It therefore follows that

$$\mathbf{A}\mathbf{V} - \mathbf{E}\mathbf{V}\mathbf{E}_r^{-1}\mathbf{A}_r - \mathbf{B}_\perp\widetilde{\mathbf{C}}_r = \mathbf{0}. \tag{3.19}$$

Note that $\mathbf{B}_\perp$ can be computed very easily:

$$\mathbf{B}_\perp := (\mathbf{I}_N - \mathbf{\Pi})\,\mathbf{B} = \mathbf{B} - \mathbf{E}\mathbf{V}\left(\mathbf{W}^T\mathbf{E}\mathbf{V}\right)^{-1}\mathbf{W}^T\mathbf{B}_r = \mathbf{B} - \mathbf{E}\mathbf{V}\mathbf{E}_r^{-1}\mathbf{B}_r \tag{3.20}$$

The dual equation for $\mathbf{W}$ solving (3.4) and admissible $\mathbf{V}$ follows from multiplication from the right by $(\mathbf{I}_N - \mathbf{\Pi}_W)$, where $\mathbf{\Pi}_W := \mathbf{V}\left(\mathbf{W}^T\mathbf{E}\mathbf{V}\right)^{-1}\mathbf{W}^T\mathbf{E}$. It reads

$$\mathbf{W}^T\mathbf{A} - \mathbf{A}_r\mathbf{E}_r^{-1}\mathbf{W}^T\mathbf{E} - \widetilde{\mathbf{B}}_r\mathbf{C}_\perp = \mathbf{0}. \tag{3.21}$$

Note that (3.19) and (3.21) present a new type of SYLVESTER equation and also contain different information. Consider, for instance, the input-type equation (3.19) in comparison to its origin (3.3). $\mathbf{E}\mathbf{V}$ is now multiplied by $\mathbf{E}_r^{-1}\mathbf{A}_r$, whose eigenvalues present the spectrum of the ROM, instead of $\mathbf{S}_V$, whose eigenvalues were given by the shifts employed for the computation of $\mathbf{V}$.

One immediate application of the novel formulation is that it allows for the *a posteriori* computation of $\mathbf{S}_V$ and $\widetilde{\mathbf{C}}_r$ if these matrices are unknown. One can then project (3.3) towards (3.19) using an arbitrary matrix $\mathbf{W}$ with $\det \mathbf{W}^T\mathbf{E}\mathbf{V} \neq 0$. Assuming $\mathbf{B}_\perp$ to have full column rank, it follows:

$$\widetilde{\mathbf{C}}_r = \left(\mathbf{B}_\perp^T\mathbf{B}_\perp\right)^{-1}\mathbf{B}_\perp^T\left(\mathbf{A}\mathbf{V} - \mathbf{E}\mathbf{V}\mathbf{E}_r^{-1}\mathbf{A}_r\right). \tag{3.22}$$

Pre-multiplication of (3.3) from the left by $\mathbf{W}^T$ yields

$$\mathbf{W}^T\mathbf{A}\mathbf{V} - \mathbf{W}^T\mathbf{E}\mathbf{V}\mathbf{S}_V - \mathbf{W}^T\mathbf{B}\widetilde{\mathbf{C}}_r = \mathbf{0} \quad \Leftrightarrow \quad \mathbf{S}_V = \mathbf{E}_r^{-1}\left(\mathbf{A}_r - \mathbf{B}_r\widetilde{\mathbf{C}}_r\right). \tag{3.23}$$

The true significance of the novel formulation of the SYLVESTER equation will become apparent in Section 4.2.

3.4. Excursus: Solving Linear Systems of Equations

The main numerical effort associated with rational KRYLOV subspace methods falls upon the solution of large, sparse, linear systems of equations (LSE)

$$\mathbf{v} = (\mathbf{A} - \sigma\mathbf{E})^{-1}\,\mathbf{b} =: \mathbf{A}_\sigma^{-1}\mathbf{b} \quad \Leftrightarrow \quad \mathbf{A}_\sigma\mathbf{v} = \mathbf{b}. \tag{3.24}$$

Although this numerical problem is clearly not in the scope of this thesis, it constitutes *the* main limiting factor from a practical point of view. This section is therefore intended to provide a brief summary of the implications this has on KRYLOV-based model reduction as it is considered in this work. Details on the numerical aspects can be found in [71].

The traditional way to solve (3.24) is Gaußian elimination including an LU-decomposition of the shifted matrix $\mathbf{A}_\sigma = \mathbf{L}\mathbf{U}$. Once the triangular matrices $\mathbf{L}$ and $\mathbf{U}$ are available, the task of solving (3.24) simplifies to two forward/backward substitutions, which work out comparably fast. Such a procedure is referred to as a *direct method*. Depending on the structure of $\mathbf{A}$, pivoting may be mandatory because otherwise the LU-decomposition would be numerically ill-posed or not exist at all [71]. Depending on the sparsity pattern of $\mathbf{A}_\sigma$, however, its LU-factors may become dense even though $\mathbf{A}_\sigma$ is sparse, so even for medium-size models their storage requirements can easily exceed the available RAM.

In such situations, one must resort to iterative solvers which attempt to find $\mathbf{v}$ in (3.24) by generating a converging sequence of approximate solutions $\hat{\mathbf{v}}_k$ and essentially involve the matrix $\mathbf{A}_\sigma$ only in the context of matrix-vector multiplication [71]. This is typically done by iteratively reducing the residual $\mathbf{A}\hat{\mathbf{v}} - \mathbf{b}$. Examples of such methods are the Generalized Minimal Residual method (GMRES), Preconditioned Conjugate Gradient method (PCG), Biconjugate Gradient method (BiCG) and others, many of which are also directly available in MATLAB. Their convergence behavior depends strongly on preconditioning measures, for instance by an incomplete LU-decomposition.

During rational interpolation, when multiple moments shall be matched about some shift σ—and the projection matrices $\mathbf{V}$ and $\mathbf{W}$ take the forms (3.5) and (3.6), respectively—the use of direct methods is particularly advantageous, because the LU-decomposition only needs to be performed once, followed by fast substitutions and orthogonalization. In iterative methods, however, the effort is only slightly reduced if multiple solves with the same matrix $\mathbf{A}_\sigma$ are required, because only the preconditioning can be recycled for the subsequent solves.

The solution of linear systems of equations is therefore not an independent problem, but of significant importance for the optimal strategy during model reduction. For instance, the ratio of the solution time to the effort of the preliminary measures (LU-decomposition or preconditioning) is actually essential for the decision how often some expansion point at hand should be used (to match higher order moments) until a new shift is selected. Still, these aspects will play a minor role in this thesis for the sake of generality, but are rather mentioned here to sensitize the reader to their general relevance.

Also, the influence of numerical round-off errors and their propagation is out of the scope of this thesis. For more information on the error introduced by inexact solves and other related topics, like recycling techniques, please refer to [3, 4, 19, 168].

3.5. How to Choose the Expansion Points?

Grimme in his PhD thesis was the first to analyze the effect of the shift position on the approximation. He discovered the rule of thumb that imaginary shifts lead to precise, but very local approximation in the respective frequency range of the amplitude response; real shifts, on the other hand, lead to a broader approximation, yet in general without exact appraisal of the amplitude response for any imaginary frequency. [73]

It is indeed an interesting observation that by matching moments about some real shift σ, one can in practice often achieve reasonable approximation in the area $s \approx \sigma \cdot i$ in the complex Laplace plane. Eid exploited this fact in his algorithm ICOP [50, 51], which has also been provided in Source 3.4. It is the simplest automatic shift selection scheme known to the author, which still fulfills some kind of optimality.

Of course one can also choose shifts σ_i manually, but in an *ad hoc* process the user is

Source 3.4: Matlab Implementation of ICOP

```
function [V,S_V,Crt,W,S_W,Brt,k,aopt_] = ICOP(A,B,C,E,n,tol,mx)
% Iterative Computation of Optimal Point [Eid 2009]
%   Input:  A,B,C,E:   HFM matrices;
%           n:         dimension of ROM;
%           [tol; mx]: convergence tolerance; maximum number of iterations
%   Output: A*V - E*V*S_V - B*Crt = 0,  W.'*A - S_W*W.'*E - Brt*C = 0
%           aopt:      optimal shift (real)
%

if size(B,2)>1 || size(C,1)>1, error('ICOP works for SISO only.'), end
if (nargin<6), tol=1e-4; end    % standard tolerance
if (nargin<7), mx=10; end       % maximal number of iterations
aopt = 0; t=tic;
for k=1:mx
    [V,S_V,Crt,W,S_W,Brt] = RationalKrylov(A,B,C,E,aopt,n);
    Ar = W.'*A*V; Er = W.'*E*V; Br = W.'*B; Cr = C*V;
    P = lyap(Ar,Br*Br.',[],Er);  P = Er*P*Er'; Y = lyap(Ar,(P+P')/2,[],Er);
    Cr_ = Cr/Er;  aopt_ = sqrt(abs((Cr_*Ar*Y*Ar'*Cr_')/(Cr*Y*Cr')));
    if norm((aopt_-aopt)/aopt) < tol, break, end
    aopt = aopt_;
end
disp(['ICOP required ' num2str(k) ' LUs and ' num2str(toc(t),'%.1f') 'seconds.'])
```

not likely to obtain good approximation by such a random shift selection.

In fact, for the SISO case it is easy to see that *any* ROM may result from rational interpolation—no matter how badly it approximates the HFM. The obvious reason is that the error model has $N + n$ zeros (possibly at infinity), each of which corresponds to a matched moment. Accordingly, *any* ROM interpolates the original model at the same number $N + n$ of frequencies (possibly at infinity, which corresponds to Markov matching). Gallivan, Vandendorpe and Van Dooren were surprised to find that in fact any reduction technique (including the truncated balanced realization) can be instantiated by rational interpolation [66, 157].[2]

So one must note that moment matching is not a value *per se*, because any (SISO) ROM interpolates the HFM at the same number of frequencies. The question is: where should one force it to do so? This problem has attracted considerable attention and is covered in the following section and in Chapter 4.

[2]In the MIMO case, the question whether a given ROM can be obtained by projection at all has been considered in [77] and is not so easy to answer.

3.6. $\mathcal{H}_2$ model reduction

3.6.1. A Short Survey

One common goal in MOR is finding a ROM which minimizes the error with respect to a given system norm. Optimal Hankel norm approximation, for instance, was treated in [70]; $\mathcal{H}_\infty$ model reduction by rational interpolation has recently been discussed in [60]. Yet in what follows, we will concentrate on the minimization of the $\mathcal{H}_2$ norm.

Necessary conditions for $\mathcal{H}_2$ extrema were known since 1967 due to Meier and Luenberger [111], Wilson derived necessary optimality conditions for MIMO systems in state space in 1970 [162], and various contributions followed over time (e. g. [83]). However, no practical algorithm to actually compute an optimum efficiently was available until 2006, when Gugercin, Beattie, and Antoulas first presented the Iterative Rational Krylov Algorithm (IRKA) [74, 76], which is still considered as "gold standard" [21] and will be revised below.

Still, other techniques have been proposed. For $n = 1$ and $n = 2$, [2] and [1], respectively, reformulate $\mathcal{H}_2$ MOR as the search for roots of polynomials.

Beattie and Gugercin also investigated $\mathcal{H}_2$ MOR by means of descent algorithms and optimization methods in [20, 21]. These works were the starting point for the idea presented in [122] (cf. Section 4.4) and will be introduced in Section 3.6.4.

3.6.2. Definition of local $\mathcal{H}_2$ Optimality and Pseudo-Optimality

Unfortunately, the computation of a global $\mathcal{H}_2$ minimum, i. e. that (stable) ROM $\mathbf{G}_r(s)$ of given order n for which $\|\mathbf{G} - \mathbf{G}_r\|_{\mathcal{H}_2}$ is minimal, is a hard task [76]. Instead, one concentrates on finding a *local* minimum, in whose vicinity there is no other ROM with smaller $\mathcal{H}_2$ error. For the SISO case, Gugercin et al. used the following result, known as Meier-Luenberger conditions.

Theorem 3.1 ([76])**.** *Let a ROM $\mathbf{G}_r(s)$ of order n have simple poles at $\lambda_{r,i}$ and be a local minimizer. Then, it interpolates both $\mathbf{G}(s)$ and its first derivative at $-\lambda_{r,i}$:*

$$\mathbf{G}(-\lambda_{r,i}) = \mathbf{G}_r(-\lambda_{r,i}) \quad \textit{and} \quad \tfrac{\mathrm{d}}{\mathrm{d}s}\mathbf{G}(-\lambda_{r,i}) = \tfrac{\mathrm{d}}{\mathrm{d}s}\mathbf{G}_r(-\lambda_{r,i}) \qquad \forall\ i = 1, \dots n. \tag{3.25}$$

The theorem is derived from structured optimality conditions. One key observation is

that among all ROMs sharing the same set of poles $\lambda_{r,i}$, that whose moments at $-\lambda_{r,i}$ match the moments of the HFM is the global optimum with respect to the $\mathcal{H}_2$ error. This necessary condition for $\mathcal{H}_2$ optimality is in fact equivalent to the $\mathcal{H}_2$ scalar product between $\mathbf{G}_r(s)$ and $\mathbf{G}_e(s)$ being zero. As this is a remarkable property, we make the following definition.

Definition 3.2. *A ROM $\mathbf{G}_r(s)$ is called an $\mathcal{H}_2$ pseudo-optimal approximant of $\mathbf{G}(s)$, if*

$$\langle \mathbf{G}_r, \mathbf{G}_e \rangle_{\mathcal{H}_2} = \langle \mathbf{G}_r, \mathbf{G} - \mathbf{G}_r \rangle_{\mathcal{H}_2} = 0, \tag{3.26}$$

or, equivalently, $\langle \mathbf{G}_r, \mathbf{G} \rangle_{\mathcal{H}_2} = \langle \mathbf{G}_r, \mathbf{G}_r \rangle_{\mathcal{H}_2}$.

An important consequence of such a configuration is that the ROM, the HFM, and the error model span a kind of Thales' circle (cf. Figure 3.1), because

$$\|\mathbf{G}_e\|_{\mathcal{H}_2}^2 = \|\mathbf{G}\|_{\mathcal{H}_2}^2 - 2\langle \mathbf{G}, \mathbf{G}_r \rangle_{\mathcal{H}_2} + \|\mathbf{G}_r\|_{\mathcal{H}_2}^2 = \|\mathbf{G}\|_{\mathcal{H}_2}^2 - \|\mathbf{G}_r\|_{\mathcal{H}_2}^2. \tag{3.27}$$

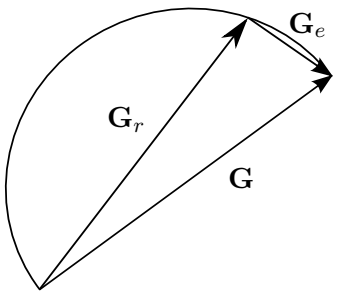


Figure 3.1.: Thales Circle in $\mathcal{H}_2$ pseudo-optimal MOR

Accordingly, the three error norms are tightly related. In fact, two implications are immediate. The relative error norm is less than one; and the larger the norm of the ROM, the smaller the corresponding error norm. These findings about $\mathcal{H}_2$ pseudo-optimality will be fundamental for several results presented in the subsequent chapters of this thesis, because $\mathcal{H}_2$ pseudo-optimality is necessary for the first-order $\mathcal{H}_2$ optimality conditions.

The conditions (3.25) can be generalized to the MIMO case, but read slightly different then.

Theorem 3.2 ([76])**.** *Let a ROM be given in diagonal form with distinct eigenvalues $\lambda_{r,i}$, such that $\mathbf{A}_r$ is a diagonal matrix and $\mathbf{E}_r$ is identity. Denote by $\mathbf{b}_i$ the n rows of the*

corresponding matrix $\mathbf{B}_{r,i}$ *and by* $\mathbf{c}_{r,i}$ *the* n *columns of* $\mathbf{C}_r$*. Then, first-order necessary conditions for the ROM to be a local* $\mathcal{H}_2$ *optimum read:*

$$\begin{aligned} i) \quad & \mathbf{G}_r(-\lambda_{r,i})\,\mathbf{b}_{r,i}^T &&= \mathbf{G}(-\lambda_{r,i})\,\mathbf{b}_{r,i}^T \\ ii) \quad & \mathbf{c}_{r,i}^T\,\mathbf{G}_r(-\lambda_{r,i}) &&= \mathbf{c}_{r,i}^T\,\mathbf{G}(-\lambda_{r,i}) \\ iii) \quad & \mathbf{c}_{r,i}^T\,\tfrac{\mathrm{d}}{\mathrm{d}s}\mathbf{G}_r(-\lambda_{r,i})\,\mathbf{b}_{r,i}^T &&= \mathbf{c}_{r,i}^T\,\tfrac{\mathrm{d}}{\mathrm{d}s}\mathbf{G}(-\lambda_{r,i})\,\mathbf{b}_{r,i}^T. \end{aligned} \tag{3.28}$$

Please note that Definition 3.2 directly carries over to the multivariable case.

3.6.3. An Iterative Rational Krylov Algorithm (IRKA)

Let us now briefly consider the algorithm that Gugercin et al. developed based on the above findings in [76]; assume a SISO model first.

The clue insight was that given an $\mathcal{H}_2$ optimal ROM, Hermite interpolation about its mirrored eigenvalues reproduces the ROM. Accordingly, a fixed-point iteration is defined by the following: Starting with an arbitrary set of expansion points, Hermite interpolation (two-sided Rational Krylov) is performed. The poles of the resulting ROM are mirrored with respect to the imaginary axis and used as the expansion points of the next iteration, until convergence occurs.

The extension to multivariable systems by tangential interpolation was sketched in [76], and further developed independently in [155, 156] and [37]. The MIMO versions basically work like the SISO version, but one uses tangential interpolation where the tangential directions follow from the eigendecomposition of the ROM.

The stunning appeal of IRKA lies in its powerfulness despite the algorithmic simplicity. An implementation is given in Source 3.5; the actual iteration requires only a couple of lines.

IRKA suffers, however, from some severe drawbacks: It is not guaranteed to converge to a local minimum[3] and the $\mathcal{H}_2$ error norm does not decrease monotonically during the iteration [20]. Besides, it requires a high number of LSE solves in comparison to manual shift selection or ICOP, which can compromise performance. The choice of the initial values has a strong influence on the process; possible ways of choosing them is computing eigenvalues of the HFM and mirroring them at the imaginary axis, or generating the

[3]Local convergence for symmetric systems has been shown in [59].

Source 3.5: Matlab Implementation of IRKA [76]

```
function [V,S_V,Crt,W,S_W,Brt,k] = IRKA(A,B,C,E,n,tol,mxi)
% Iterative Rational Krylov Algorithm [Gugercin et al. 2006]
%   Input:  A,B,C,E:   HFM matrices;
%           n:         dimension of ROM
%           tol; mxi:  convergence tolerance; maximum number of iterations
%   Output: Krylov subspaces for locally H2 optimal Hermite interpolation
%             A*V - E*V*S_V - B*Crt = 0,  W.'*A - S_W*W.'*E - Brt*C = 0
%

t=tic; % time measurement
[V,~,~,W] = RationalKrylov(A,sum(B,2),sum(C,1),E,0,n);
[v,s0] = eig(full(W'*A*V), full(W'*E*V));
s0 = cplxpair(diag(-s0));  t_B = (v.'*(W'*B)).'; t_C = (C*V*v).';

if (nargin<6), tol=1e-4; end    % standard tolerance
if (nargin<7), mxi=20;  end    % standard max. number of iterations
for k=1:mxi
    [V,S_V,Crt,W,S_W,Brt] = TangentialKrylov(A,B,C,E,s0,t_B,t_C);
    % compute new shifts from eigendecomposition of ROM
    [v,D] = eig(full(W.'*A*V), full(W.'*E*V));
    if norm((cplxpair(s0)-cplxpair(-diag(D))))/norm(s0) < tol, break, end
    s0 = -diag(D); s0=s0.*sign(real(s0)); % new shifts (mirror if necessary)
    t_B = (v\((W.'*E*V)\(W.'*B))).';      % input  tangential directions
    t_C = (C*V*v).';                      % output tangential directions
end
if k==mxi, warning('IRKA stopped prematurely.'); end
disp(['IRKA required ' num2str(k*n) ' LUs and ' num2str(toc(t),'%.1f') 'seconds.'])
```

zeroth ROM as some more or less heuristic Padé approximant. Finally, unstable ROMs may occur during the iteration; in this case, however, one can simply mirror the respective eigenvalues along the imaginary axis to obtain a set of shifts which is comprised in the right half complex plane (see Source 3.5).

3.6.4. Descent Algorithms

Inspired by that, Beattie and Gugercin proposed ideas towards $\mathcal{H}_2$ model reduction by means of descent optimization methods that would feature monotonic decay of the error norm. In [21], the n complex shifts of a two-sided Rational Krylov reduction (Hermite interpolation) are optimized by a Newton method. In [20], a trust region algorithm is used to optimize poles and residues of the reduced transfer function. The two methods may speed-up IRKA, but as was noted in [122], none of them can yet be considered mature for the following reasons:

- For HERMITE interpolation as used in [21], unstable models may result. As was shown in [122], starting from some initial values it may sometimes be impossible to reach a local minimum without trespassing a region of shift configurations leading to unstable reduced models.
- The number of optimization variables in [20] is doubled, as poles and residues are varied independently.
- Even if

$$\mathcal{J} := \|\mathbf{G}_e\|_{\mathcal{H}_2}^2 - \|\mathbf{G}\|_{\mathcal{H}_2}^2 = -2\,\langle \mathbf{G}, \mathbf{G}_r \rangle_{\mathcal{H}_2} + \|\mathbf{G}_r\|_{\mathcal{H}_2}^2 \tag{3.29}$$

  is used to avoid the necessity of computing $\|\mathbf{G}\|_{\mathcal{H}_2}$, the evaluation of the cost functional requires additional effort for the $\mathcal{H}_2$ scalar product.
- Analytic gradient and Hessian expressions are derived based on "standard" bases of KRYLOV subspaces like (3.7), which can be numerically ill-conditioned. Attaching orthogonalization, however, seems to severely complicate the procedure.
- Double poles are excluded.
- It is not guaranteed that the ROM is real-valued.
- The reduced order has to be chosen *ad hoc*.

A certain enhancement of these descent methods will therefore be presented in Section 4.4.

3.6.5. Pseudo-Optimal Rational Krylov (PORK)

In the remainder of this chapter, the Pseudo-Optimal Rational Krylov (PORK) algorithm is presented. It delivers the $\mathcal{H}_2$ pseudo-optimal ROM for a given shift configuration and constitutes an important element in the subsequent chapters of this dissertation. PORK was invented and firstly published by WOLF ET AL. in [163] for the SISO case; the generalization to MIMO is, however, quite straightforward, at least from an algorithmical point of view.

Theorem 3.3. *Let* $\mathbf{V} \in \mathbb{R}^{N\times n}$ *solve* SYLVESTER *equation* (3.3) *where all eigenvalues of* $\mathbf{S}_V$ *have positive real part, the pair* $(\widetilde{\mathbf{C}}_r, \mathbf{S}_V)$ *is observable, and* $[\mathbf{EV},\ \mathbf{B}]$ *is of full column rank. Let further* $\widetilde{\mathbf{Q}}_r = \widetilde{\mathbf{Q}}_r^T$ *solve the* $n \times n$ LYAPUNOV *equation*

$$(-\mathbf{S}_V^T)\,\widetilde{\mathbf{Q}}_\mathbf{r} + \widetilde{\mathbf{Q}}_\mathbf{r}\,(-\mathbf{S}_V) + \widetilde{\mathbf{C}}_r^T\widetilde{\mathbf{C}}_r = \mathbf{0}. \tag{3.30}$$

Define $\mathbf{W} := [\mathbf{EV},\ \mathbf{B}]\,\mathbf{K}^T$, *where*

$$\mathbf{K} := [\mathbf{0}_{n\times m}\ \mathbf{I}_n]\begin{bmatrix}\widetilde{\mathbf{Q}}_r^{-1}\widetilde{\mathbf{C}}_r^T & \mathbf{I}_n\\ \mathbf{I}_m & \mathbf{0}_{m\times n}\end{bmatrix}^{-1}\Big([\mathbf{EV},\ \mathbf{B}]^T\,[\mathbf{EV},\ \mathbf{B}]\Big)^{-1} \in \mathbb{R}^{n\times(n+m)}.$$

Then, it holds:

i) The resulting reduced order matrices read

$$\mathbf{B}_r = -\widetilde{\mathbf{Q}}_{\mathbf{r}}^{-1}\widetilde{\mathbf{C}}_r^T,\ \mathbf{A}_r = \mathbf{S}_V + \mathbf{B}_r\widetilde{\mathbf{C}}_r,\ \mathbf{C}_r = \mathbf{CV},\ \mathbf{D}_r = \mathbf{D},\ \textit{and}\ \ \mathbf{E}_r = \mathbf{I}_n. \tag{3.31}$$

ii) The spectra of $\mathbf{A}_r$ *and* $\mathbf{S}_V$ *are mirror images with respect to the imaginary axis,*

$$\lambda_i(\mathbf{A}_r) = -\lambda_i(\mathbf{S}_V). \tag{3.32}$$

iii) $\widetilde{\mathbf{Q}}_{\mathbf{r}}$ *is the inverse of the resulting Controllability Gramian* $\mathbf{P}_r$.

iv) The given ROM is an $\mathcal{H}_2$ *pseudo-optimal approximation of* $\mathbf{G}(s)$.

Proof. For the first part, we note that

$$\mathbf{AV} = [\mathbf{EV},\ \mathbf{B}]\begin{bmatrix}\mathbf{S}_V\\ \widetilde{\mathbf{C}}_r\end{bmatrix},\quad \mathbf{EV} = [\mathbf{EV},\ \mathbf{B}]\begin{bmatrix}\mathbf{I}\\ \mathbf{0}_{m\times n}\end{bmatrix},\quad \mathbf{B} = [\mathbf{EV},\ \mathbf{B}]\begin{bmatrix}\mathbf{0}_{n\times n}\\ \mathbf{I}_m\end{bmatrix}. \tag{3.33}$$

Also,

$$\mathbf{W}^T\,[\mathbf{EV},\ \mathbf{B}] = [\mathbf{0}_{n\times m}\ \mathbf{I}_n]\begin{bmatrix}\widetilde{\mathbf{Q}}_r^{-1}\widetilde{\mathbf{C}}_r^T & \mathbf{I}_n\\ \mathbf{I}_m & \mathbf{0}_{m\times n}\end{bmatrix}^{-1} = \begin{bmatrix}\mathbf{I}_n & -\widetilde{\mathbf{Q}}_r^{-1}\widetilde{\mathbf{C}}_r^T\end{bmatrix}.$$

Therefore,

$$\begin{aligned}
\mathbf{E}_r = \mathbf{W}^T\mathbf{EV} &= \begin{bmatrix}\mathbf{I}_n & -\widetilde{\mathbf{Q}}_r^{-1}\widetilde{\mathbf{C}}_r^T\end{bmatrix}\begin{bmatrix}\mathbf{I_n}\\ \mathbf{0}_{m\times n}\end{bmatrix} = \mathbf{I}_n\\
\mathbf{B}_r = \ \mathbf{W}^T\mathbf{B}\ &= \begin{bmatrix}\mathbf{I}_n & -\widetilde{\mathbf{Q}}_r^{-1}\widetilde{\mathbf{C}}_r^T\end{bmatrix}\begin{bmatrix}\mathbf{0}_{n\times n}\\ \mathbf{I}_m\end{bmatrix} = -\widetilde{\mathbf{Q}}_r^{-1}\widetilde{\mathbf{C}}_r^T\\
\mathbf{A}_r = \mathbf{W}^T\mathbf{AV} &= \begin{bmatrix}\mathbf{I}_n & -\widetilde{\mathbf{Q}}_r^{-1}\widetilde{\mathbf{C}}_r^T\end{bmatrix}\begin{bmatrix}\mathbf{S}_V\\ \widetilde{\mathbf{C}}_r\end{bmatrix} = \mathbf{S}_V - \widetilde{\mathbf{Q}}_r^{-1}\widetilde{\mathbf{C}}_r^T\widetilde{\mathbf{C}}_r = \mathbf{S}_V + \mathbf{B}\widetilde{\mathbf{C}}_r
\end{aligned}$$

The second part follows directly with (3.30):

$$\mathbf{A}_r = \mathbf{S}_V - \widetilde{\mathbf{Q}}_r^{-1}\widetilde{\mathbf{C}}_r^T\widetilde{\mathbf{C}}_r = -\widetilde{\mathbf{Q}}_r^{-1}\mathbf{S}_V^T\widetilde{\mathbf{Q}}_r. \tag{3.34}$$

So the eigenvalues of $\mathbf{A}_r$ are the negative eigenvalues of $\mathbf{S}_V^T$.

The reduced controllability Gramian is defined by

$$\begin{aligned} \mathbf{0} &= \mathbf{A}_r\mathbf{P}_r + \mathbf{P}_r\mathbf{A}_r^T + \mathbf{B}_r\mathbf{B}_r^T = -\widetilde{\mathbf{Q}}_r\mathbf{S}_V\widetilde{\mathbf{Q}}_r^{-1}\mathbf{P}_r - \mathbf{P}_r\widetilde{\mathbf{Q}}_r^{-1}\mathbf{S}_V^T\widetilde{\mathbf{Q}}_r + \widetilde{\mathbf{Q}}_r^{-1}\widetilde{\mathbf{C}}_r^T\widetilde{\mathbf{C}}_r\widetilde{\mathbf{Q}}_r^{-1} \\ \Leftrightarrow \quad & -\mathbf{S}_V^T\widetilde{\mathbf{Q}}_r\mathbf{P}_r\widetilde{\mathbf{Q}}_r - \widetilde{\mathbf{Q}}_r\mathbf{P}_r\widetilde{\mathbf{Q}}_r\mathbf{S}_V^T + \widetilde{\mathbf{C}}_r^T\widetilde{\mathbf{C}}_r = \mathbf{0}. \end{aligned} \tag{3.35}$$

Due to uniqueness,

$$\widetilde{\mathbf{Q}}_r\mathbf{P}_r\widetilde{\mathbf{Q}}_r = \widetilde{\mathbf{Q}}_r \quad \Leftrightarrow \quad \mathbf{P}_r\widetilde{\mathbf{Q}}_r = \mathbf{I} \quad \Leftrightarrow \quad \mathbf{P}_r = \widetilde{\mathbf{Q}}_r^{-1}. \tag{3.36}$$

For the last part, we must show that the $\mathcal{H}_2$ scalar product of the original and the reduced system as defined by (2.17) fulfills

$$\langle \mathbf{G}, \mathbf{G}_r \rangle_{\mathcal{H}_2} = \langle \mathbf{G}_r, \mathbf{G}_r \rangle_{\mathcal{H}_2} = \|\mathbf{G}_r\|_{\mathcal{H}_2}^2 .$$

To this end, we show that $\mathbf{V}\mathbf{P}_r$ solves (2.17):

$$\begin{aligned} & \mathbf{A}\mathbf{V}\mathbf{P}_r\mathbf{E}_r^T + \mathbf{E}\mathbf{V}\mathbf{P}_r\mathbf{A}_r^T + \mathbf{B}\mathbf{B}_r^T = \\ = \; & \left[\mathbf{E}\mathbf{V}\mathbf{S}_V + \mathbf{B}\widetilde{\mathbf{C}}_r\right]\mathbf{P}_r + \mathbf{E}\mathbf{V}\mathbf{P}_r\left[-\widetilde{\mathbf{Q}}_\mathbf{r}^{-1}\mathbf{S}_V^T\widetilde{\mathbf{Q}}_\mathbf{r}\right]^T + \mathbf{B}\mathbf{B}_r^T \\ = \; & \mathbf{E}\mathbf{V}\mathbf{S}_V\mathbf{P}_r - \mathbf{E}\mathbf{V}\mathbf{S}_V\mathbf{P}_r + \mathbf{B}\widetilde{\mathbf{C}}_r\mathbf{P}_r - \mathbf{B}\widetilde{\mathbf{C}}_r\mathbf{P}_r = \mathbf{0}. \end{aligned} \tag{3.37}$$

It therefore holds: $\langle \mathbf{G}, \mathbf{G}_r \rangle_{\mathcal{H}_2} = \mathbf{C}\mathbf{V}\mathbf{P}_r\mathbf{C}_r^T = \langle \mathbf{G}_r, \mathbf{G}_r \rangle_{\mathcal{H}_2}$ and $\langle \mathbf{G} - \mathbf{G}_r, \mathbf{G}_r \rangle_{\mathcal{H}_2} = 0$. □

Remark 3.2. *It is important to stress that the matrix* $\mathbf{W}$ *does never have to be computed explicitly, but the matrices of the ROM follow directly from* (3.31)*. Accordingly, knowing* $\mathbf{V}$, $\mathbf{S}_V$, *and* $\widetilde{\mathbf{C}}_r$, *the remaining numerical effort consists in matrix-vector products (*$\mathbf{C}_r = \mathbf{C}\mathbf{V}$*) and small-scale operations (mainly the reduced-order* Lyapunov *equation). Besides, the full column rank of* $[\mathbf{E}\mathbf{V},\ \mathbf{B}]$ *is not actually necessary but was supposed for simplicity of the proof because otherwise* $\mathbf{W}$ *cannot be written as above.*

The PORK algorithm implicitly places the poles of the ROM at the mirror images of the shifts contained in the rational Krylov subspace[4]. Different from existing pole placement techniques [9], however, this requires no additional large-scale computations and works for MIMO, as well. For shifts with strictly positive real part, PORK implicitly guarantees asymptotic stability. In addition, it features $\mathcal{H}_2$ pseudo-optimality, meaning that the ROM delivers the smallest $\mathcal{H}_2$ error among all others sharing its pole configuration. This was shown in [76] for the SISO case, in [167] for tangential interpolation, and in [164] for block Krylov subspaces. A summary and more details can be found in [23].

[4]Note that this is the reason why the pair $(\widetilde{\mathbf{C}}_r, \mathbf{S}_V)$ must be observable: according to Lemma 3.1, this characterizes a Krylov subspace which does not contain eigenvectors.

Source 3.6: Pseudo-Optimal Rational Krylov (PORK) [163]

```
function [Ar,Br,Cr,Er] = PORK_V(V,S_V,Crt,C)
% Pseudo-Optimal Rational (Input) Krylov PORK [Wolf et al. 2013]
%   Input:  V,S_V,Crt:      solution of  A*V - E*V*S_V - B*Crt = 0
%           C:              HFM output matrix
%   Output: Ar,Br,Cr,Er:    ROM matrices
%

Qr_c = lyapchol(-S_V', Crt');
Br = -Qr_c\(Qr_c'\Crt');
Ar = S_V+Br*Crt;
Cr = C*V;
Er = eye(size(Ar));
```

```
function [Ar,Br,Cr,Er] = PORK_W(W,S_W,Brt,B)
% Pseudo-Optimal Rational (Output) Krylov PORK [Wolf et al. 2013]
%   Input:  W,S_W,Brt:      solution of  W.'*A - S_W*W.'*E - Brt*C = 0
%           B:              HFM input matrix
%   Output: Ar,Br,Cr,Er:    ROM matrices
%

Pr_c = lyapchol(-S_W, Brt);
Cr = -Brt.'/Pr_c/Pr_c.';
Ar = S_W+Brt*Cr;
Br = W.'*B;
Er = eye(size(Ar));
```

Theorem 3.3 can also be derived in a dual way. The result is given without proof:

Corollary 3.1. *Let* $\mathbf{W} \in \mathbb{R}^{N\times n}$ *solve* Sylvester *equation* (3.4) *where all eigenvalues of* $\mathbf{S}_W$ *have positive real part,* $(\mathbf{S}_W, \widetilde{\mathbf{B}}_r)$ *is controllable, and* $\left[\mathbf{E}^T\mathbf{W},\ \mathbf{C}^T\right]$ *is of full column rank. Let further* $\widetilde{\mathbf{P}}_r = \widetilde{\mathbf{P}}_r^T$ *solve the* Lyapunov *equation*

$$(-\mathbf{S}_W)\widetilde{\mathbf{P}}_\mathbf{r} + \widetilde{\mathbf{P}}_\mathbf{r}(-\mathbf{S}_W^T) + \widetilde{\mathbf{B}}_r\widetilde{\mathbf{B}}_r = \mathbf{0}. \tag{3.38}$$

Then, the ROM defined by the matrices

$$\mathbf{B}_r = \mathbf{W}^T\mathbf{B},\ \mathbf{A}_r = \mathbf{S}_W + \widetilde{\mathbf{B}}_r\mathbf{C}_r,\ \mathbf{C}_r = -\widetilde{\mathbf{B}}_r^T\widetilde{\mathbf{P}}_r^{-1},\ \mathbf{D}_r = \mathbf{D},\ \textit{and}\ \ \mathbf{E}_r = \mathbf{I}_n \tag{3.39}$$

is an $\mathcal{H}_2$ *pseudo-optimal approximation of* $\mathbf{G}(s)$*, whose observability Gramian is* $\widetilde{\mathbf{P}}_r^{-1}$ *and whose eigenvalues are the mirror images of* $\mathbf{S}_W$*.*

Remark 3.3. *Like in a two-sided Rational* Krylov *method (cf. Lemma 3.3), the ROM resulting from PORK does not depend on the realization of the HFM.*

3.7. Conclusions and Open Problems

We have seen that the Sylvester-based projective MOR framework—in particular, rational Krylov subspace methods—provides a very flexible and powerful tool for the reduction of even very high-dimensional models. However, some drawbacks remain (and outline the way of the next chapters):

- Preservation of stability is not generally guaranteed, but at least in special cases or with the help of pole placement techniques like the $\mathcal{H}_2$ pseudo-optimal algorithm PORK.
- One has to find suitable expansion points. Although algorithms like IRKA can perform this task well, convergence is not generally guaranteed and does, in fact, not always occur (especially when stability issues emerge). Existing descent algorithms which aim at better convergence have not yet reached maturity.
- The order of the reduced system must still be determined *ad hoc*.
- This is particularly tricky when no information on the resulting error is available, not even *a posteriori*.

In a way, the quotation of De Villemagne and Skelton at the beginning of this section is therefore still true more than 25 years later.

4. CURE: A Cumulative Reduction Scheme

"Faust. *Nun kenn ich deine würd'gen Pflichten!*
Du kannst im Großen nichts vernichten
Und fängst es nun im Kleinen an."
— Johann Wolfgang von Goethe

This chapter deals with adaptive reduction techniques. Their goal is to automatically choose the reduced system order and all degrees of freedom (i. e. the expansion points) during the reduction process, thus circumventing the need of their *ad hoc* determination by the user. This is therefore a crucial step towards a solution "at the push of a button" as desired in industrial processes.

The general problem of adaptive methods was pointed out in [160]: "A good new vector is the one that is as different from the ones we already have as possible, and thus, cannot be well approximated by the currently available subspace. However, this unleashes the question: how can we determine if a candidate sample point will generate a block vector that adds rank to our set without computing it? Furthermore, how can we know if this new block vector will help to minimize the number of samples needed to obtain a good ROM? The answer is simple, we cannot." Existing techniques for the adaptive selection of expansion points in moment matching methods therefore aim to find new shifts with the help of approximate error expressions, typically based on efficient residual terms.

Section 4.1 contains a brief survey on adaptive shift selection strategies. A new approach is then presented in the remainder of the chapter. It is based on a factorization of the error model as derived in Section 4.2. By applying the factorization iteratively, an adaptive reduction procedure can be performed, during which an overall ROM is constructed by cumulative augmentation of small-scale ROMs. For their computation, a

descent algorithm including minimization of the true $\mathcal{H}_2$ error is suggested in Section 4.4 to avoid the convergence and stability issues related to the "gold standard" IRKA.

4.1. State of the Art

The literature contains various approaches to an adaptive choice of expansion points; in fact an iterative shift selection scheme was already proposed by Grimme [73].

Jaimoukha, Kasenally, and Frangos derived expressions similar to the error factorization presented in Section 4.2 under the name of *Arnoldi-* and *Lanczos-like equations* [64, 85], but from an algorithmic perspective and with focus on restart mechanisms in the Arnoldi and Lanczos processes. Nonetheless, they formed the basis for adaptive shift selection strategies [61, 62, 63], which iteratively choose imaginary expansion points based on various heuristic error estimates derived from residual expressions.

Druskin et al. in [49] exploit the skeleton approximation [153] to minimize a residual expression along the boundary of a polygonial set $\mathcal{S}_m$ which approximates the mirrored spectrum of $\mathbf{A}$; a generalization to MIMO systems is given in [47]. Although the procedure performed well in some numerical experiments, its use is restricted to systems in strictly dissipative realization, descent behavior is not guaranteed, and the cost functional is heuristic to some extent (in particular, the choice of $\mathcal{S}_m$).

Other shift selection strategies are due to Villena and Silveira, who presented the ARMS algorithms in [159, 160], to Feng and Benner, who exploited symmetry to obtain a good error estimate in [56], and to Zhao et al. who use a formulation of the error model similar to that presented in Section 4.2 as an "error monitor" for an adaptive selection strategy of shifts and their multiplicity [170].

Further, Bodendiek and Bollhöfer presented the adaptive greedy-type shift selection method AORA-RK for application to MOR of Maxwell's equation [35, 36], and Sommer, Farle, and Dyczij-Edlinger considered the reduction of models of phased antennas [147]. Köhler et al. described an adaptive multi-point moment matching algorithm in [92] which selects imaginary shifts based on an error indicator. The procedure includes sampling of the HFM along the imaginary axis and the judicious evaluation of a sensitivity measure for the selection of new shifts.

Fehr, Fischer, and Eberhard used the local error bound [94, 95] (cf. Section 5.1) for the greedy choice of imaginary expansion points in the context of (almost) lossless second order systems [53, 58].

To conclude, existing methods for the adaptive selection of expansion points are typically limited to the computation of *one* (mostly purely imaginary) new shift at a time. Most of them are based on residual expressions and therefore heuristically motivated algorithms (cf. Section 5.1) without monotonicity of the induced error, or restrictive in their assumptions.

The Cumulative Reduction ("CURE") framework which will be presented in the remainder of this chapter provides more flexibility, because the size of the increments, from which the overall ROM is constructed iteratively, can be chosen flexibly, as well as the way they are computed. Also, monotonic decay of the $\mathcal{H}_2$ error norm can be attained with the help of the SPARK algorithm derived in Section 4.4, which is based on efficient descent optimization of the $\mathcal{H}_2$ error norm.

4.2. A Factorized Formulation of the Error System

4.2.1. Motivation

The commonly used realization of the error model $\mathbf{G}_e(s)$ as defined above reads

$$\begin{aligned} \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_r \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{x}}_r(t) \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_r \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}_r(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{B}_r \end{bmatrix} \mathbf{u}(t), \\ \mathbf{y}_e(t) &= \begin{bmatrix} \mathbf{C} & -\mathbf{C}_r \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}_r(t) \end{bmatrix}. \end{aligned} \tag{4.1}$$

This formulation, however, suffers from a crucial drawback. Assume $\mathbf{G}_r(s)$ is a serviceable approximation of $\mathbf{G}(s)$, such that the output error $\mathbf{y}_e(t)$ is small. Then to some extent, the dominant dynamics of the HFM is contained *twice* in the decoupled ODE system, i. e. in the augmented, $(N+n)$-dimensional state vector. Only in the output equation are the two components subtracted.

This generally does not seem beneficial from a numerical point of view, but it is particularly unfavorable with regard to consecutive reduction steps. Imagine the quality of the

ROM does not suffice and shall be improved in an additional reduction step by searching a low-order approximation of the remaining error. Then as motivated in Section 2.3.1, one would aim to find a subspace of the augmented state space which contains the "important" dynamics in the error model (4.1). As a matter of fact, this procedure is likely to reproduce the subspace from the first reduction step, although the "important" subspace would be the one that captures the *difference* $\mathbf{x}(t) - \mathbf{V}\mathbf{x}_r(t)$. But as the impact of observability is not captured in this approach (and particularly in one-sided projection), it is probably doomed to failure.

For that reason, a novel formulation of the error system $\mathbf{G}_e(s)$ has been derived in [165, 166] and will be presented in the following.

4.2.2. Factorization Based on Sylvester Equation

Starting from (4.1), perform the state transformation

$$\begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}_r(t) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_N & \mathbf{V} \\ \mathbf{0} & \mathbf{I}_n \end{bmatrix} \begin{bmatrix} \mathbf{e}(t) \\ \mathbf{x}_r(t) \end{bmatrix}, \tag{4.2}$$

and multiply the ODE system from the left by

$$\mathbf{M} = \begin{bmatrix} \mathbf{I}_N & -\mathbf{E}\mathbf{V}\mathbf{E}_r^{-1} \\ \mathbf{0} & \mathbf{I}_n \end{bmatrix}. \tag{4.3}$$

The resulting realization then reads

$$\begin{aligned} \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_r \end{bmatrix} \begin{bmatrix} \dot{\mathbf{e}}(t) \\ \dot{\mathbf{x}}_r(t) \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & (\mathbf{I}-\mathbf{\Pi})\mathbf{A}\mathbf{V} \\ \mathbf{0} & \mathbf{A}_r \end{bmatrix} \begin{bmatrix} \mathbf{e}(t) \\ \mathbf{x}_r(t) \end{bmatrix} + \begin{bmatrix} (\mathbf{I}-\mathbf{\Pi})\mathbf{B} \\ \mathbf{B}_r \end{bmatrix} \mathbf{u}(t), \\ \mathbf{y}_e(t) &= \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{e}(t) \\ \mathbf{x}_r(t) \end{bmatrix}. \end{aligned} \tag{4.4}$$

Now let $\mathbf{V}$ fulfill the SYLVESTER equation (3.19). Then the upper right entry in the augmented system matrix $\mathbf{A}_e$ can be resolved to

$$(\mathbf{I}-\mathbf{\Pi})\mathbf{A}\mathbf{V} = \mathbf{A}\mathbf{V} - \mathbf{E}\mathbf{V}\mathbf{E}_r^{-1}\mathbf{A}_r = \mathbf{B}_\perp \widetilde{\mathbf{C}}_r.$$

Remember that according to (3.20), $\mathbf{B}_\perp = \mathbf{B} - \mathbf{E}\mathbf{V}\mathbf{E}_r^{-1}\mathbf{B}_r$ can be computed easily, and that $\widetilde{\mathbf{C}}_r$ is directly available from the computation of $\mathbf{V}$ (cf. Chapter 3) or otherwise given by (3.22).

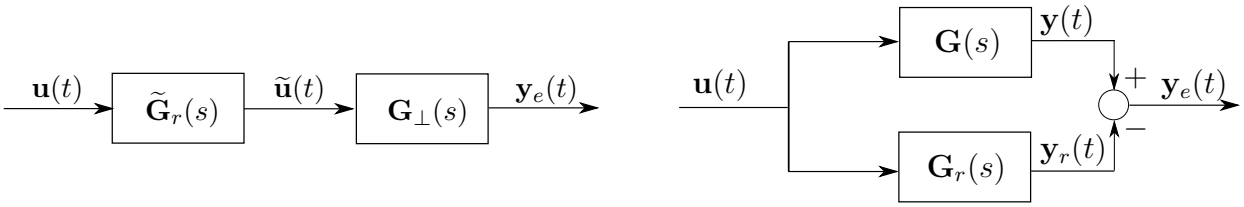
Anyway, one can see that the decoupled second line in the ODE system (4.4) influences the upper line via a term $\mathbf{B}_\perp \widetilde{\mathbf{C}}_r \mathbf{x}_r(t)$. As the regular input term $(\mathbf{I}-\mathbf{\Pi})\mathbf{B}\mathbf{u}(t) = \mathbf{B}_\perp \mathbf{u}(t)$

points in the same set of directions (columns of $\mathbf{B}_\perp$), we can withdraw the lower part of the ODE system and rewrite the error model using an auxiliary signal $\tilde{\mathbf{u}}(t) := \widetilde{\mathbf{C}}_r \mathbf{x}_r(t) + \mathbf{u}(t) \in \mathbb{R}^m$:

$$\begin{aligned} \mathbf{E}\ \dot{\mathbf{e}}(t) &= \mathbf{A}\ \mathbf{e}(t) + \mathbf{B}_\perp(t)\,\tilde{\mathbf{u}}(t), & \mathbf{y}_e(t) &= \mathbf{C}\,\mathbf{e}(t), \\ \mathbf{E}_r\,\dot{\mathbf{x}}_r(t) &= \mathbf{A}_r\,\mathbf{x}_r(t) + \mathbf{B}_r(t)\,\mathbf{u}(t), & \tilde{\mathbf{u}}(t) &= \widetilde{\mathbf{C}}_r \mathbf{x}_r(t) + \mathbf{u}(t). \end{aligned} \tag{4.5}$$

Obviously, the final error output $\mathbf{y}_e(t)$ results from the input $\mathbf{u}(t)$ by composition of two transmission lines. We can therefore rewrite the error model as a *product* of two systems, as depicted in Figure 4.1:

$$\mathbf{G}_e(s) = \underbrace{\left[\begin{array}{c|c} \mathbf{E}, \mathbf{A} & \mathbf{B}_\perp \\ \hline \mathbf{C} & \mathbf{0} \end{array}\right]}_{\mathbf{G}_\perp(s)} \cdot \underbrace{\left[\begin{array}{c|c} \mathbf{E}_r, \mathbf{A}_r & \mathbf{B}_r \\ \hline \widetilde{\mathbf{C}}_r & \mathbf{I}_n \end{array}\right]}_{\widetilde{\mathbf{G}}_r^R(s)}. \tag{4.6}$$



a) Factorized Formulation: Product b) Standard Realization: Difference

Figure 4.1.: Factorized vs. Standard Formulation of the Error Model

Remarkably, the first factor $\mathbf{G}_\perp(s)$ resembles the HFM except for its input matrix.[1] The second factor $\widetilde{\mathbf{G}}_r^R(s) \in \mathbb{C}^{m \times m}$ shares the ODE of the ROM (in particular its order n) and differs from it only in its output equation, which exhibits a unity feedthrough matrix.

All results hold true analogously if $\mathbf{W}$ fulfills the dual Sylvester equation (3.21). Then the error model can be factorized as follows, where $\widetilde{\mathbf{G}}_r^L(s) \in \mathbb{C}^{p \times p}$ and $\mathbf{C}_\perp = \mathbf{C} - \mathbf{C}\mathbf{V}\mathbf{E}_r^{-1}\mathbf{W}^T\mathbf{E}$:

$$\mathbf{G}_e(s) = \underbrace{\left[\begin{array}{c|c} \mathbf{E}_r, \mathbf{A}_r & \widetilde{\mathbf{B}}_r \\ \hline \mathbf{C}_r & \mathbf{I}_n \end{array}\right]}_{\widetilde{\mathbf{G}}_r^L(s)} \cdot \underbrace{\left[\begin{array}{c|c} \mathbf{E}, \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C}_\perp & \mathbf{0} \end{array}\right]}_{\mathbf{G}_\perp(s)}. \tag{4.7}$$

[1] Also, $\mathbf{G}_\perp(s)$ has no feedthrough, even if $\mathbf{D} \neq \mathbf{0}$ in $\mathbf{G}(s)$.

4.2.3. Properties, Special Cases, and Features

One immediate feature of the factorized formulation is the physical interpretation it provides. Let $\mathbf{u}(t)$ be some typical input signal and assume the error model corresponding to some ROM has been factorized according to (4.6). Then the output $\tilde{\mathbf{u}}(t)$ of $\widetilde{\mathbf{G}}_r^R(t)$ can be easily computed and forms the input to $\mathbf{G}_\perp(s)$ whose output gives the error signal $\mathbf{y}_e(t)$. As the input matrix $\mathbf{B}_\perp$ is known and the dynamics of $\mathbf{G}_\perp(s)$ is that of the HFM, this may give instructive insight into the approximation error from a physical point of view.

If, for instance, the model describes some heat transfer problem, then $\tilde{\mathbf{u}}(t)$ can be interpreted as a disturbance heat source, and $\|\mathbf{B}_\perp \tilde{\mathbf{u}}(\cdot)\|$ is the total energy fed in. As the effect of the disturbance on the system describes precisely the error resulting from the model reduction, an engineer might quickly estimate the impact of the model reduction error based on the amount of energy applied by the disturbance.

In the following, we will analyze some system theoretic properties of the decomposition.

Proposition 4.1. *If* $\mathbf{V}$ *fulfills* Sylvester *equation* (3.3) *and the error model is factorized as in* (4.6)*, then* $\widetilde{\mathbf{G}}_r^R(s)$ *has invariant zeros at the eigenvalues* σ_i *of* $\mathbf{S}_V$ *in* (3.3)*.*

Proof. For simplicity we assume that $\mathbf{S}_V$ has distinct eigenvalues σ_i. Let $\mathbf{z}_i$ be an eigenvector of $\mathbf{S}_V$ corresponding to σ_i; then $\mathbf{S}_V \mathbf{z}_i = \sigma_i \mathbf{z}_i$. Now multiply (3.3) from the left by $\mathbf{W}^T$ and from the right by $\mathbf{z}_i$. We obtain

$$\mathbf{A}_r \mathbf{z}_i - \mathbf{E}_r(\sigma_i \mathbf{z}_i) - \mathbf{B}_r \widetilde{\mathbf{C}}_r \mathbf{z}_i = \mathbf{0} \quad \Rightarrow \quad \begin{bmatrix} \mathbf{A}_r - \sigma_i \mathbf{E}_r & \mathbf{B}_r \\ \widetilde{\mathbf{C}}_r & \mathbf{I}_m \end{bmatrix} \cdot \begin{bmatrix} \mathbf{z}_i \\ -\widetilde{\mathbf{C}}_r \mathbf{z}_i \end{bmatrix} = \mathbf{0}.$$

Accordingly, the Rosenbrock matrix is rank deficient, which completes the proof. □

In fact, this result generalizes to the case of higher multiplicity, i. e when $\mathbf{S}_V$ is not diagonalizable but contains defective eigenvalues (Jordan blocks). For example, in single-point Padé approximation, when $\mathbf{V}$ takes the form (3.5), then $\widetilde{\mathbf{G}}_r^R(s)$ has a transfer zero of multiplicity n at σ. The proof, however, does not work out via the Rosenbrock matrix as easily as above.

In the dual factorization (4.7), the invariant zeros of $\widetilde{\mathbf{G}}_r^L(s)$ are given by the eigenvalues of $\mathbf{S}_W$ in (3.4).

In the case of rational KRYLOV subspaces, when the shifts σ_i have positive real part and the HFM is asymptotically stable, compensation cannot occur. The eigenvalues of $\mathbf{S}_V$ or $\mathbf{S}_W$ corresponding to rational KRYLOV subspaces therefore lead to transmission zeros of $\widetilde{\mathbf{G}}_r^R(s)$ or $\widetilde{\mathbf{G}}_r^L(s)$, respectively.

In modal truncation, however, the spectrum of $\mathbf{S}_V$ describes those eigenvalues of the HFM that are carried over to the ROM (cf. Section 3.2.3). For that reason, the corresponding invariant zero in $\widetilde{\mathbf{G}}_r^R(s)$ coincides with an eigenvalue of $\mathbf{G}_r(s)$ and compensation occurs, as the concerned eigenvalue is unobservable if $\widetilde{\mathbf{C}}_r\mathbf{z}_i = \mathbf{0}$. The above realization of $\widetilde{\mathbf{G}}_r^R(s)$ is therefore not minimal. In fact, if $\mathbf{V}$ consists only out of eigenvectors, then $\widetilde{\mathbf{C}}_r = \mathbf{0}$ and the transfer behavior is purely static: $\widetilde{\mathbf{G}}_r^R(s) \equiv \mathbf{I}_m$.

If, in addition, $\mathbf{W}$ contains the corresponding left handed eigenvectors, then the respective eigenvalues in $\mathbf{G}_\perp(s)$ become uncontrollable as their spectral component is removed in $\mathbf{B}_\perp$. Thinking of the error transfer function in pole-residue-formulation (2.5), the respective modal components of $\mathbf{G}$ and $\mathbf{G}_r$ cancel, so it makes sense that the mode does not appear in the transfer behavior of $\mathbf{G}_\perp(s)\widetilde{\mathbf{G}}_r^R(s)$.

Proposition 4.2. *In two-sided Rational* KRYLOV*, i. e. when both* $\mathbf{V}$ *and* $\mathbf{W}$ *solve respective* SYLVESTER *equations* (3.3) *and* (3.4)*,* $\mathbf{G}_\perp(s)$ *has invariant zeros of corresponding multiplicity at the eigenvalues of* $\mathbf{S}_W$ *in* $\mathbf{V}$*-based decomposition* (4.6) *or at the eigenvalues of* $\mathbf{S}_V$ *in* $\mathbf{W}$*-based decomposition* (4.7)*.*

Proof. For simplicity, we consider the SISO case only. It is obvious that the error model has transfer zeros of corresponding multiplicity at the eigenvalues both of $\mathbf{S_V}$ and $\mathbf{S_W}$. In $\mathbf{V}$-based decomposition, the spectrum of $\mathbf{S_V}$ leads to invariant zeros in $\widetilde{\mathbf{G}}_r^R(s)$, so the eigenvalues of $\mathbf{S_W}$ must induce invariant zeros in $\mathbf{G}_\perp(s)$. Analogous considerations hold true for $\mathbf{W}$-based decomposition. □

In the remainder of this subsection, we will concentrate on the interesting special case of $\mathcal{H}_2$ pseudo-optimal reduction. It is obvious (in the SISO case) that matching moments at σ_i and placing the reduced poles at $-\sigma_i$, as one does by means of PORK according to (3.32), delivers a model $\widetilde{\mathbf{G}}_r^R(s)$ or $\widetilde{\mathbf{G}}_r^L(s)$ with poles and zeros vis-à-vis relative to the imaginary axis: an all-pass system. In fact, this result also holds for MIMO systems:

Theorem 4.1. *If* $\mathbf{V}$ *is the basis of a rational* Krylov *subspace and* $\mathcal{H}_2$ *pseudo-optimal reduction is performed with the PORK algorithm, then* $\widetilde{\mathbf{G}}_r^R(s)$ *is a unity all-pass system. Moreover, the transfer functions of the error model* $\mathbf{G}_e(s)$ *and the large-scale system* $\mathbf{G}_\perp(s)$ *have the same* Frobenius *norm along the imaginary axis:*

$$\|\mathbf{G}_e(i\omega)\|_{\mathrm{F}} = \|\mathbf{G}_\perp(i\omega)\|_{\mathrm{F}} \qquad \forall \omega \in \mathbb{R}.$$

In the SISO case, this means that their amplitude responses are identical.

Proof. According to Theorem 3.3, $\mathbf{P}_r\widetilde{\mathbf{Q}}_r = \mathbf{I}$ holds where $\widetilde{\mathbf{Q}}_r$ solves (3.30). Therefore, we must show that $\widetilde{\mathbf{Q}}_r$ is in fact the observability Gramian of $\widetilde{\mathbf{G}}_r^R(s)$.

$$\begin{aligned}\mathbf{A}^T\widetilde{\mathbf{Q}}_\mathbf{r} + \widetilde{\mathbf{Q}}_\mathbf{r}\mathbf{A} + \tilde{\mathbf{C}}_r^T\tilde{\mathbf{C}}_r &= \left(\mathbf{S}_V - \widetilde{\mathbf{Q}}_r^{-1}\tilde{\mathbf{C}}_r^T\tilde{\mathbf{C}}_r\right)^T + \widetilde{\mathbf{Q}}_\mathbf{r}\left(\mathbf{S}_V - \widetilde{\mathbf{Q}}_r^{-1}\tilde{\mathbf{C}}_r^T\tilde{\mathbf{C}}_r\right) + \tilde{\mathbf{C}}_r^T\tilde{\mathbf{C}}_r \\ &= \mathbf{S}_V^T\widetilde{\mathbf{Q}}_\mathbf{r} + \widetilde{\mathbf{Q}}_\mathbf{r}\mathbf{S}_V - \tilde{\mathbf{C}}_r^T\tilde{\mathbf{C}}_r = \mathbf{0}.\end{aligned}$$

For the second part, observe that

$$\begin{aligned}\|\mathbf{G}_e(i\omega)\|_{\mathrm{F}} &= \left\|\mathbf{G}_\perp(i\omega)\cdot\widetilde{\mathbf{G}}_r^R(i\omega)\right\|_{\mathrm{F}} \\ &= \mathrm{tr}\left[\mathbf{G}_\perp(i\omega)\cdot\widetilde{\mathbf{G}}_r^R(i\omega)\left(\widetilde{\mathbf{G}}_r^R(i\omega)\right)^H\cdot\mathbf{G}_\perp^H(i\omega)\right] \\ &= \mathrm{tr}\left[\mathbf{G}_\perp(i\omega)\mathbf{G}_\perp^H(i\omega)\right] = \|\mathbf{G}_\perp(i\omega)\|_{\mathrm{F}}\,.\end{aligned}$$

□

Of course the dual version of this theorem for $\mathbf{W}$ being the basis of an output Krylov subspace holds true as well.

One important consequence is that the $\mathcal{H}_2$ norms of $\mathbf{G}_e(s)$ and $\mathbf{G}_\perp(s)$ equal, which leads to the following important statement:

Proposition 4.3. *In* $\mathcal{H}_2$ *pseudo-optimal MOR, when* $\widetilde{\mathbf{G}}_r^R(s)$ *in* (4.6) *or* $\widetilde{\mathbf{G}}_r^L(s)$ *in* (4.7)*, respectively, is a unity all-pass factor, it holds*

$$\|\mathbf{G}_\perp\|_{\mathcal{H}_2} < \|\mathbf{G}\|_{\mathcal{H}_2}$$

unless the ROM is a zero element.

Proof. This is a direct consequence of Theorem 4.1 and (3.27):

$$\|\mathbf{G}_\perp\|_{\mathcal{H}_2}^2 = \|\mathbf{G}_e\|_{\mathcal{H}_2}^2 = \|\mathbf{G}\|_{\mathcal{H}_2}^2 - \underbrace{\|\mathbf{G}_r\|_{\mathcal{H}_2}^2}_{>0} < \|\mathbf{G}\|_{\mathcal{H}_2}^2\,.$$

□

4.3. Adaptive Model Order Reduction

We will see in this section that due to the factorized formulations (4.6) and (4.7) of the error model, it is now perfectly possible to perform additional reduction steps.

4.3.1. Iterative Error Factorization

Imagine one performed SYLVESTER-based projection with $\mathbf{V}$ solving (3.19) and expressed the error model as a product according to (4.6). Then, it holds:

$$\mathbf{G}(s) = \mathbf{G}_r(s) + \mathbf{G}_e(s) = \mathbf{G}_r(s) + \mathbf{G}_\perp(s) \cdot \widetilde{\mathbf{G}}_r^R(s). \tag{4.8}$$

If $\mathbf{G}_r(s)$ does not offer a sufficient approximation, one can improve it by performing an additional reduction step, now of the error model $\mathbf{G}_\perp(s) \cdot \widetilde{\mathbf{G}}_r^R(s)$. We have seen before that—at least under certain conditions—the first factor is more crucial, as it contains the relevant dynamics that has not been captured by the reduced model so far. Besides, it is of high order and offers much potential to be approximated with another small-scale system.Therefore, we reduce $\mathbf{G}_\perp(s)$ in a second SYLVESTER-based projection as if it were the original model (in fact, $\mathbf{G}(s)$ and $\mathbf{G}_\perp(s)$ are very similar: only their input matrices $\mathbf{B}$ and $\mathbf{B}_\perp$ differ). We can then express $\mathbf{G}_\perp(s)$ as the sum of the second reduced model $\mathbf{G}_{r,2}(s)$ and the corresponding error $\mathbf{G}_{e,2}(s)$, which again can be factorized:

$$\mathbf{G}_\perp(s) = \mathbf{G}_{r,2}(s) + \mathbf{G}_{e,2}(s) = \mathbf{G}_{r,2}(s) + \mathbf{G}_{\perp,2}(s) \cdot \widetilde{\mathbf{G}}_{r,2}^R(s). \tag{4.9}$$

Inserting this into (4.8) and renaming $\mathbf{G}_r(s) \to \mathbf{G}_{r,1}(s)$, $\widetilde{\mathbf{G}}_r^R(s) \to \widetilde{\mathbf{G}}_{r,1}^R(s)$ yields

$$\begin{aligned} \mathbf{G}(s) &= \mathbf{G}_{r,1}(s) + \left[\mathbf{G}_{r,2}(s) + \mathbf{G}_{\perp,2}(s) \cdot \widetilde{\mathbf{G}}_{r,2}^R(s)\right] \cdot \widetilde{\mathbf{G}}_{r,1}^R(s) \\ &= \underbrace{\mathbf{G}_{r,1}(s) + \mathbf{G}_{r,2}(s) \cdot \widetilde{\mathbf{G}}_{r,1}^R(s)}_{\mathbf{G}_{r,2}^\Sigma(s)} + \mathbf{G}_{\perp,2}(s) \cdot \underbrace{\widetilde{\mathbf{G}}_{r,2}^R(s) \cdot \widetilde{\mathbf{G}}_{r,1}^R(s)}_{\widetilde{\mathbf{G}}_{r,2}^{\Sigma,R}(s)} \end{aligned} \tag{4.10}$$

A key observation is that $\mathbf{G}_{r,1}(s)$ and $\widetilde{\mathbf{G}}_{r,1}^R(s)$ share the same poles, so the order of system $\mathbf{G}_{r,2}^\Sigma(s)$ is the order n_1 of $\mathbf{G}_{r,1}(s)$ plus the order n_2 of $\mathbf{G}_{r,2}(s)$, i. e. the sum of the two reduced dimensions. In fact one can find a compact state space realization of the system (see below).

Obviously, the second reduction step did not change the structure of equation (4.8): The HFM $\mathbf{G}(s)$ is still expressed as a sum of a small-scale ROM and the product of a high-dimensional system and another small-scale model that exhibits feedthrough and has the same eigenvalues as the ROM. Accordingly, there is no obstacle to reduce $\mathbf{G}_{\perp,2}(s)$ again and iterate this procedure until the overall ROM is satisfactory. This line of action, which has first been published in [122] by means of the SPARK algorithm (cf. Section 4.4), offers two advantages:

- We do not have to fix the order of the ROM *a priori*, but can perform a kind of "salami technique", reducing the model slice by slice rather than off the reel.
- This gives us the opportunity to adapt and optimize the reduction in each step to the error model that actually remains at this point of the iteration.

Note that one can also perform reduction steps and error decompositions based on $\mathbf{W}$ solving Sylvester equation (3.4). Then, dually to (4.10), one obtains

$$\mathbf{G}(s) = \underbrace{\mathbf{G}_{r,1}(s) + \widetilde{\mathbf{G}}^{L}_{r,1}(s) \cdot \mathbf{G}_{r,2}(s)}_{\mathbf{G}^{\Sigma}_{r,2}(s)} + \underbrace{\widetilde{\mathbf{G}}^{L}_{r,1}(s) \cdot \widetilde{\mathbf{G}}^{L}_{r,2}(s)}_{\widetilde{\mathbf{G}}^{\Sigma,L}_{r,2}(s)} \cdot \mathbf{G}_{\perp,2}(s). \tag{4.11}$$

In fact, one can even alternate between the two decompositions. For instance, starting from (4.8) one can conduct a reduction step based on appropriate $\mathbf{W}$. This leads to

$$\begin{aligned} \mathbf{G}(s) &= \mathbf{G}_{r,1}(s) + \Big[\mathbf{G}_{r,2}(s) + \widetilde{\mathbf{G}}^{L}_{r,2}(s) \cdot \mathbf{G}_{\perp,2}(s)\Big] \cdot \widetilde{\mathbf{G}}^{R}_{r,1}(s) \\ &= \underbrace{\mathbf{G}_{r,1}(s) + \mathbf{G}_{r,2}(s) \cdot \widetilde{\mathbf{G}}^{R}_{r,1}(s)}_{\mathbf{G}^{\Sigma}_{r,2}(s)} + \underbrace{\widetilde{\mathbf{G}}^{L}_{r,2}(s)}_{\widetilde{\mathbf{G}}^{\Sigma,L}_{r,2}(s)} \cdot \mathbf{G}_{\perp,2}(s) \cdot \underbrace{\widetilde{\mathbf{G}}^{R}_{r,1}(s)}_{\widetilde{\mathbf{G}}^{\Sigma,R}_{r,2}(s)}. \end{aligned} \tag{4.12}$$

In this most general case, the high-order component $\mathbf{G}_{\perp,k}(s)$ of the error system is clamped between left- and right-hand sided small-scale models, so the overall structure is indeed more complicated than after a single reduction step.

However, we will see in the following, how the iterative reduction framework can be implemented very efficiently by recursion. Unfortunately, the mathematical formulation is much more complicated than the actual implementation and the notation becomes quite confusing, therefore the syntax is explained first:

- A lower index $(_r)$ denotes a matrix of reduced order.
- An additional upper index $(^{\Sigma})$ marks the "overall" ROMs that are set up by accumulation, i. e. $\mathbf{G}^{\Sigma}_{r,k}(s)$, $\widetilde{\mathbf{G}}^{\Sigma,L}_{r,k}(s)$, and $\widetilde{\mathbf{G}}^{\Sigma,R}_{r,k}(s)$.
- The lower index $(_k)$ or $(_{k-1})$ refers to the current or previous step, respectively.
- An upper index ^{R} or ^{L} marks matrices belonging exclusively to $\widetilde{\mathbf{G}}^{\Sigma,L}_{r,k+1}(s)$ and $\widetilde{\mathbf{G}}^{\Sigma,R}_{r,k+1}(s)$, respectively.

At the beginning of the iteration, $\mathbf{B}_{\perp,0} := \mathbf{B}$ and $\mathbf{C}_{\perp,0} := \mathbf{C}$; all other matrices are empty.

Theorem 4.2. *The following decomposition holds after each reduction step k,*

$$\mathbf{G}(s) = \mathbf{G}^{\Sigma}_{r,k}(s) \; + \; \widetilde{\mathbf{G}}^{\Sigma,L}_{r,k}(s) \cdot \; \mathbf{G}_{\perp,k}(s) \cdot \widetilde{\mathbf{G}}^{\Sigma,R}_{r,k}(s), \tag{4.13}$$

where realizations of the arising systems are given by

$$\mathbf{G}^{\Sigma}_{r,k}(s) = \left[\begin{array}{c|c} \mathbf{E}^{\Sigma}_{r,k}, \mathbf{A}^{\Sigma}_{r,k} & \mathbf{B}^{\Sigma}_{r,k} \\ \hline \mathbf{C}^{\Sigma}_{r,k} & \mathbf{0} \end{array}\right] \in \mathbb{C}^{p\times m}, \;\; \textit{order } n^{\Sigma} = \sum_{i=1}^{k} n_i, \tag{4.14}$$

$$\mathbf{G}_{\perp,k}(s) = \left[\begin{array}{c|c} \mathbf{E}, \mathbf{A} & \mathbf{B}_{\perp,k} \\ \hline \mathbf{C}_{\perp,k} & \mathbf{0} \end{array}\right] \in \mathbb{C}^{p\times m}, \;\; \textit{order } N, \tag{4.15}$$

$$\widetilde{\mathbf{G}}^{\Sigma,L}_{r,k}(s) = \left[\begin{array}{c|c} \mathbf{E}^{\Sigma}_{r,k}, \mathbf{A}^{\Sigma}_{r,k} & \widetilde{\mathbf{B}}^{\Sigma,L}_{r,k} \\ \hline \mathbf{C}^{\Sigma,L}_{r,k} & \mathbf{I}_p \end{array}\right] \in \mathbb{C}^{p\times p}, \;\; \textit{order } n^{\Sigma} = \sum_{i=1}^{k} n_i, \tag{4.16}$$

$$\widetilde{\mathbf{G}}^{\Sigma,R}_{r,k}(s) = \left[\begin{array}{c|c} \mathbf{E}^{\Sigma}_{r,k}, \mathbf{A}^{\Sigma}_{r,k} & \mathbf{B}^{\Sigma,R}_{r,k} \\ \hline \widetilde{\mathbf{C}}^{\Sigma,R}_{r,k} & \mathbf{I}_m \end{array}\right] \in \mathbb{C}^{m\times m}, \;\; \textit{order } n^{\Sigma} = \sum_{i=1}^{k} n_i, \tag{4.17}$$

with

$$\mathbf{E}^{\Sigma}_{r,k} = \begin{bmatrix} \mathbf{E}^{\Sigma}_{r,k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_{r,k} \end{bmatrix}, \qquad \mathbf{E}_{r,k} = \mathbf{W}_k^T \mathbf{E} \mathbf{V}_k,$$

$$\mathbf{A}^{\Sigma}_{r,k} = \begin{bmatrix} \mathbf{A}^{\Sigma}_{r,k-1} & \widetilde{\mathbf{B}}^{\Sigma,L}_{r,k-1}\mathbf{C}_{r,k} \\ \mathbf{B}_{r,k}\widetilde{\mathbf{C}}^{\Sigma,R}_{r,k-1} & \mathbf{A}_{r,k} \end{bmatrix}, \quad \mathbf{A}_{r,k} = \mathbf{W}_k^T \mathbf{A} \mathbf{V}_k,$$

$$\mathbf{B}^{\Sigma}_{r,k} = \begin{bmatrix} \mathbf{B}^{\Sigma}_{r,k-1} \\ \mathbf{B}_{r,k} \end{bmatrix}, \qquad \mathbf{B}_{r,k} = \mathbf{W}_k^T \mathbf{B}_{\perp,k-1},$$

$$\mathbf{C}^{\Sigma}_{r,k} = \begin{bmatrix} \mathbf{C}^{\Sigma}_{r,k-1} & \mathbf{C}_{r,k} \end{bmatrix}, \qquad \mathbf{C}_{r,k} = \mathbf{C}_{\perp,k-1} \mathbf{V}_k.$$

Also, for $\mathbf{V}_k$-based decomposition, $\widetilde{\mathbf{C}}_{r,k}$ is known from Sylvester *equation (3.19), and*

$$\mathbf{B}^{\Sigma,R}_{r,k} = \begin{bmatrix} \mathbf{B}^{\Sigma,R}_{r,k-1} \\ \mathbf{B}_{r,k} \end{bmatrix}, \widetilde{\mathbf{C}}^{\Sigma,R}_{r,k} = \begin{bmatrix} \widetilde{\mathbf{C}}^{\Sigma,R}_{r,k-1} & \widetilde{\mathbf{C}}_{r,k} \end{bmatrix}, \widetilde{\mathbf{B}}^{\Sigma,L}_{r,k} = \begin{bmatrix} \widetilde{\mathbf{B}}^{\Sigma,L}_{r,k-1} \\ \mathbf{0} \end{bmatrix}, \mathbf{C}^{\Sigma,L}_{r,k} = \begin{bmatrix} \mathbf{C}^{\Sigma,L}_{r,k-1} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{B}_{\perp,k} = \mathbf{B}_{\perp,k-1} - \mathbf{E}\mathbf{V}_k \mathbf{E}^{-1}_{r,k} \mathbf{B}_{r,k}, \qquad \mathbf{C}_{\perp,k} = \mathbf{C}_{\perp,k-1}.$$

For $\mathbf{W}_k$-based decomposition, $\widetilde{\mathbf{B}}_{r,k}$ follows from Sylvester *equation (3.4), and*

$$\mathbf{B}^{\Sigma,R}_{r,k} = \begin{bmatrix} \mathbf{B}^{\Sigma,R}_{r,k-1} \\ \mathbf{0} \end{bmatrix}, \widetilde{\mathbf{C}}^{\Sigma,R}_{r,k} = \begin{bmatrix} \widetilde{\mathbf{C}}^{\Sigma,R}_{r,k-1} & \mathbf{0} \end{bmatrix}, \widetilde{\mathbf{B}}^{\Sigma,L}_{r,k} = \begin{bmatrix} \widetilde{\mathbf{B}}^{\Sigma,L}_{r,k-1} \\ \widetilde{\mathbf{B}}_{r,k} \end{bmatrix}, \mathbf{C}^{\Sigma,L}_{r,k} = \begin{bmatrix} \mathbf{C}^{\Sigma,L}_{r,k-1} & \mathbf{C}_{r,k} \end{bmatrix},$$

$$\mathbf{B}_{\perp,k} = \mathbf{B}_{\perp,k-1}, \qquad \mathbf{C}_{\perp,k} = \mathbf{C}_{\perp,k-1} - \mathbf{C}_{r,k} \mathbf{E}^{-1}_{r,k} \mathbf{W}_k^T \mathbf{E}.$$

Proof. It is probably more constructive to prove the theorem with the help of an example. An assistant visualization of the first three steps is depicted in Figure 4.2. Therein, a red background marks a system of high order N; blue stands for a reduced model; the small-scale factors with feedthrough are painted orange.

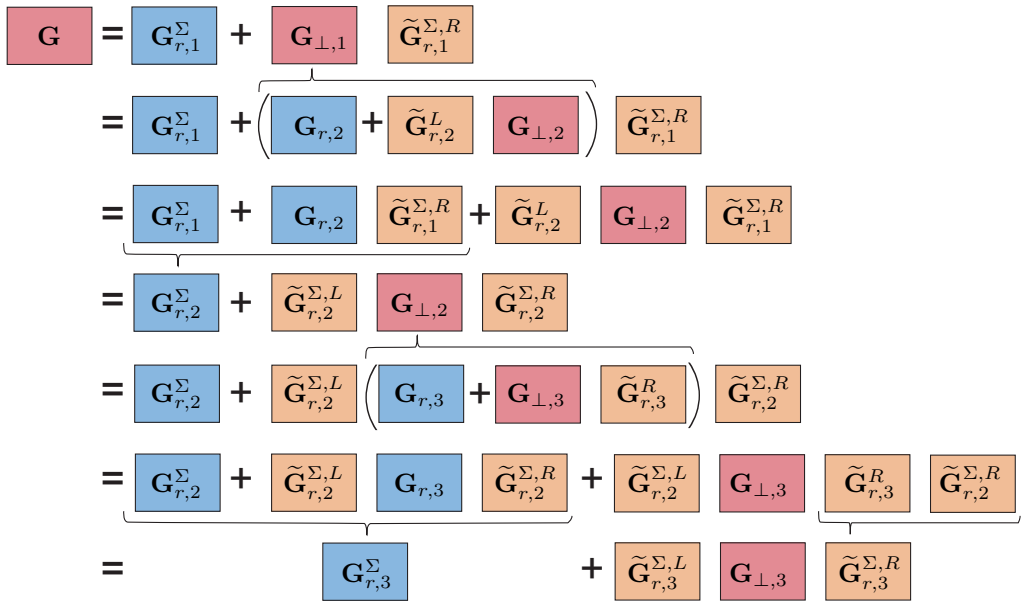


Figure 4.2.: The CURE Framework: Three Alternating Reduction Steps (**V**–**W**–**V**)

In this example we perform a first reduction step such that $\mathbf{V}_1$ solves Sylvester equation (3.3) and compute the input-type error decomposition. Then we reduce $\mathbf{G}_{\perp,1}$ in a second step, where $\mathbf{W}_2$ solves Sylvester equation (3.4), and factorize the error model in the output-type way. Then we perform a third reduction, again input-sided, and a fourth one, output-sided. Then according to the above formulas, the matrices read

$$\mathbf{A}_{r,4}^{\Sigma} = \begin{bmatrix} \mathbf{A}_{r,1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_{r,2}\widetilde{\mathbf{C}}_{r,1} & \mathbf{A}_{r,2} & \widetilde{\mathbf{B}}_{r,2}\mathbf{C}_{r,3} & \widetilde{\mathbf{B}}_{r,2}\mathbf{C}_{r,4} \\ \mathbf{B}_{r,3}\widetilde{\mathbf{C}}_{r,1} & \mathbf{0} & \mathbf{A}_{r,3} & \mathbf{0} \\ \mathbf{B}_{r,4}\widetilde{\mathbf{C}}_{r,1} & \mathbf{0} & \mathbf{B}_{r,4}\widetilde{\mathbf{C}}_{r,3} & \mathbf{A}_{r,4} \end{bmatrix}, \quad \mathbf{B}_{r,4}^{\Sigma} = \begin{bmatrix} \mathbf{B}_{r,1} \\ \mathbf{B}_{r,2} \\ \mathbf{B}_{r,3} \\ \mathbf{B}_{r,4} \end{bmatrix},$$

$$\mathbf{C}_{r,4}^{\Sigma} = \begin{bmatrix} \mathbf{C}_{r,1} & \mathbf{C}_{r,2} & \mathbf{C}_{r,3} & \mathbf{C}_{r,4} \end{bmatrix},$$

$$\mathbf{B}_{r,4}^{\Sigma,R} = \begin{bmatrix} \mathbf{B}_{r,1} \\ \mathbf{0} \\ \mathbf{B}_{r,3} \\ \mathbf{0} \end{bmatrix}, \quad \widetilde{\mathbf{B}}_{r,4}^{\Sigma,L} = \begin{bmatrix} \mathbf{0} \\ \widetilde{\mathbf{B}}_{r,2} \\ \mathbf{0} \\ \widetilde{\mathbf{B}}_{r,4} \end{bmatrix},$$

$$\widetilde{\mathbf{C}}_{r,4}^{\Sigma,R} = \begin{bmatrix} \widetilde{\mathbf{C}}_{r,1} & \mathbf{0} & \widetilde{\mathbf{C}}_{r,3} & \mathbf{0} \end{bmatrix},$$
$$\mathbf{C}_{r,4}^{\Sigma,L} = \begin{bmatrix} \mathbf{0} & \mathbf{C}_{r,2} & \mathbf{0} & \mathbf{C}_{r,4} \end{bmatrix},$$

while $\mathbf{E}_{r,4}^{\Sigma}$ is simply the block-diagonal concatenation of $\mathbf{E}_{r,1}$ to $\mathbf{E}_{r,4}$.

Now let us verify (4.13) recursively. After the first step, $\mathbf{G}_{r,1}$ and $\widetilde{\mathbf{G}}_{r,1}^{R}$ take the standard form of (4.6), while $\widetilde{\mathbf{G}}_{r,1}^{L}(s) \equiv \mathbf{I}_p$, because its input matrix is zero, so the feedthrough makes the system an identity element and indeed (4.13) holds.

After the second step, we basically have the situation of (4.12), so we need to validate that the quantities that appear in (4.12) match with the formulas given in Theorem 4.2.

According to (4.13), the new left-handed factor $\widetilde{\mathbf{G}}_{r,2}^{\Sigma,L}$ is defined by the matrices

$$\mathbf{E}_{r,2}^{\Sigma} = \begin{bmatrix} \mathbf{E}_{r,1} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_{r,2} \end{bmatrix}, \quad \mathbf{A}_{r,2}^{\Sigma} = \begin{bmatrix} \mathbf{A}_{r,1} & \mathbf{0} \\ \mathbf{B}_{r,2}\widetilde{\mathbf{C}}_{r,1} & \mathbf{A}_{r,2} \end{bmatrix}, \quad \widetilde{\mathbf{B}}_{r,2}^{\Sigma,L} = \begin{bmatrix} \mathbf{0} \\ \widetilde{\mathbf{B}}_{r,2} \end{bmatrix}, \quad \mathbf{C}_{r,2}^{\Sigma,L} = \begin{bmatrix} \mathbf{0} & \mathbf{C}_{r,2} \end{bmatrix}.$$

Obviously this realization is not minimal, as the upper part of the system is uncontrollable. In fact, a minimal realization is given by

$$\widetilde{\mathbf{G}}_{r,2}^{\Sigma,L}(s) = \left[\begin{array}{c|c} \mathbf{E}_{r,2}, \mathbf{A}_{r,2} & \widetilde{\mathbf{B}}_{r,2} \\ \hline \mathbf{C}_{r,2} & \mathbf{I}_p \end{array}\right],$$

which is indeed simply the small-scale factor resulting from the output-sided error decomposition in the second step. Similarly, the lower part of the right-handed factor $\widetilde{\mathbf{G}}_{r,2}^{\Sigma,R}$ is unobservable, such that $\widetilde{\mathbf{G}}_{r,2}^{\Sigma,R}(s)$ remains unchanged as expected, because in the second step no right-handed system is factored out:

$$\widetilde{\mathbf{G}}_{r,2}^{\Sigma,R}(s) = \widetilde{\mathbf{G}}_{r,1}^{\Sigma,R}(s) = \left[\begin{array}{c|c} \mathbf{E}_{r,1}, \mathbf{A}_{r,1} & \mathbf{B}_{r,1} \\ \hline \widetilde{\mathbf{C}}_{r,1} & \mathbf{I}_m \end{array}\right].$$

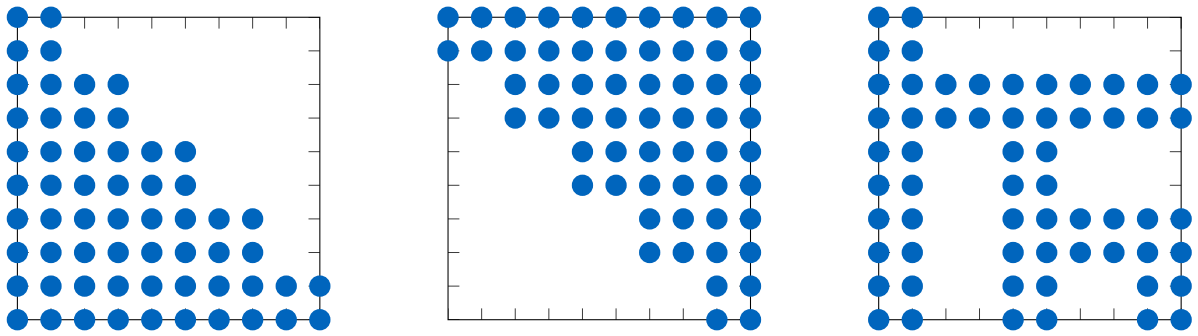
It remains to show that $\mathbf{G}_{r,2}^{\Sigma}(s) = \mathbf{G}_{r,1}(s) + \mathbf{G}_{r,2}(s) \cdot \widetilde{\mathbf{G}}_{r,1}^{R}(s)$, which follows from straightforward calculations. The further reduction steps can be proven similarly by induction. □

Accordingly, the resulting realizations of $\widetilde{\mathbf{G}}_{r,k+1}^{\Sigma,L}(s)$ and $\widetilde{\mathbf{G}}_{r,k+1}^{\Sigma,R}(s)$ are not minimal, which seems disadvantageous at first sight. However, the implementation is very much simplified by the formulation of Theorem 4.2, as in every step the matrices only have to be augmented by some columns and rows (cf. Section 4.3.2). Besides, due to the clear structure, the uncontrollable and unobservable state variables can easily be truncated after the last step of the iteration.

Proposition 4.4. *The entries of the reduced state vector arising in* $\mathbf{V}$*-based error decomposition during the Cumulative Reduction (CURE) framework are uncontrollable in the left-sided factor* $\widetilde{\mathbf{G}}_{r,k}^{\Sigma,L}$ *and can be removed by truncation of all respective rows and columns in the model.*

In the dual way, reduced state variables related to $\mathbf{W}$*-based error factorization are unobservable in the right-hand factor* $\widetilde{\mathbf{G}}_{r,k}^{\Sigma,R}$ *and can be truncated.*

The extreme example of purely $\mathbf{V}$-sided or purely $\mathbf{W}$-sided decomposition naturally leads to a static model $\widetilde{\mathbf{G}}_{r,k}^{\Sigma,L}$ or $\widetilde{\mathbf{G}}_{r,k}^{\Sigma,R}$, respectively. Furthermore, if in such a case only ROMs of order $q_i = 2$ are computed in every step, then the matrix $\mathbf{A}_{r,k}^{\Sigma}$ of the overall ROM is lower or upper Hessenberg, see Figure 4.3, and $\mathbf{E}_{r,k}^{\Sigma}$ is tridiagonal.



a) $\mathbf{V}$-based decomposition b) $\mathbf{W}$-based decomposition c) Alternating decomposition

Figure 4.3.: Pattern of ROM Matrix $\mathbf{A}_r^{\Sigma}$ after Five Reduction Steps to Order $q_i = 2$

4.3.2. Implementation

A possible ready-to-run implementation of the CURE scheme can be seen in Source 4.1. The essential part of the source code provides full generality, but the actual reduction performed in the example (lines 11–14) is only a stub; here, two block moments about $\sigma = 0$ are matched in every iteration. In the presented form, $\mathbf{V}$ is used for the decomposition; to perform $\mathbf{W}$-based factorization, comment line 12 instead of 14. Of course, any other reduction can be included instead, as long as Sylvester equation (3.3) or (3.4), respectively, is fulfilled.

Source 4.1: Matlab Implementation of CURE

```
load CDPlayer; E = speye(120); D = zeros(2,2);
sys = dss(A, B, C, D, E);  % caution: large-scale!

[N,m] = size(B);  p = size(C,1);
Er_tot = []; Ar_tot = []; Br_tot = []; Cr_tot = []; B_ = B; C_ = C;
BrL_tot = zeros(0,p); CrL_tot = zeros(p,0);
BrR_tot = zeros(0,m); CrR_tot = zeros(m,0);

while (1)
    % compute V and W and decide:
    % V-based decomposition, if A*V - E*V*S - B*Crt = 0
    V = [A\B_ A\(A\B_)]; W = V; Crt = [eye(m), zeros(m)];   mode = 'V';
    % W-based decomposition, if A'*W - E'*W*SW - Brt*Cr = 0
%     W = [A'\C_', A'\(A'\C_)]; V = W; Brt = [eye(m); zeros(m)]; mode = 'W';

    n = size(V,2);  Er = W.'*E*V;  Ar = W.'*A*V;  Br = W.'*B_;  Cr = C_*V;
    Er_tot = blkdiag(Er_tot, Er);
    Ar_tot = [Ar_tot, BrL_tot*Cr; Br*CrR_tot, Ar]; %#ok<*AGROW>
    Br_tot = [Br_tot; Br];  Cr_tot = [Cr_tot, Cr];
    if mode=='V'
        B_ = B_ - E*(V*(Er\Br));    % B_bot
        BrL_tot = [BrL_tot; zeros(n,p)];    BrR_tot = [BrR_tot; Br];
        CrL_tot = [CrL_tot, zeros(p,n)];    CrR_tot = [CrR_tot, Crt];
    elseif mode=='W'
        C_ = C_ - Cr/Er*W.'*E;      % C_bot
        BrL_tot = [BrL_tot; Brt];   BrR_tot = [BrR_tot; zeros(n,m)];
        CrL_tot = [CrL_tot, Cr];    CrR_tot = [CrR_tot, zeros(m,n)];
    end

    if size(Ar_tot,1)>=20, break; end   % some stopping criterion
end

sysr    = dss(Ar_tot, Br_tot, Cr_tot, zeros(p,m), Er_tot);
sysbot  = dss(A, B_, C_, zeros(p,m), E);    % caution: large-scale!

% truncate non controllable/observable states in sysrL, sysrR
i = find(any(Ar_tot(any(BrL_tot~=0,2),:),1));
sysrL   = dss(Ar_tot(i,i), BrL_tot(i,:), CrL_tot(:,i), eye(p), Er_tot(i,i));
i = find(any(Ar_tot(:,any(CrR_tot~=0,1)),2));
sysrR   = dss(Ar_tot(i,i), BrR_tot(i,:), CrR_tot(:,i), eye(m), Er_tot(i,i));
```

Note that the code is poor with regard to aspects of memory allocation; the respective warning in MATLAB is suppressed by the comment `%#ok<*AGROW>`. However, in practice, the `while` loop is not run through very often (at most, presumably, a hundred times), so the wasted time is expected to be in the range of milliseconds. But naturally the CURE scheme can probably be implemented in a better way, Source 4.1 is rather optimized for readability, cf. Section 1.2.5.

The example produces a ROM of order $n = 20$ which matches ten block moments about $s = 0$. $\widetilde{\mathbf{G}}_r^{\Sigma,L}(s)$ (`sysrL` in Source 4.1) is a proportional element, $\widetilde{\mathbf{G}}_r^{\Sigma,L}(s) \equiv \mathbf{I}_2$. $\widetilde{\mathbf{G}}_r^{\Sigma,R}(s)$ (`sysrR` in Source 4.1) has a transfer zero of multiplicity 20 at $s = 0$ and the poles of $\mathbf{G}_r^{\Sigma}(s)$. Please note that lines 2 and 34 implement the large-scale systems $\mathbf{G}(s)$ and $\mathbf{G}_\perp(s)$, respectively, as `ss`-objects of MATLAB's Control Toolbox, which stores the full matrices disregarding sparsity. The lines should therefore be replaced for high-dimensional ROMs and are only intended to demonstrate how $\mathbf{G}(s)$ and $\mathbf{G}_\perp(s)$ look.

4.3.3. Properties

First of all, we recall the following observation:

Proposition 4.5. *The spectrum of the overall ROM $\mathbf{G}_{r,k}^{\Sigma}(s)$ is the union of the eigenvalues of the single ROMs $\mathbf{G}_{r,i}(s)$, $i = 1 \ldots (k-1)$.*

Accordingly, the subsequent reduction steps do not affect the poles of the previous iterations, but eigenvalues are added to the overall ROM in a cumulative way.

This is an important property of the CURE scheme and also implies the following consequence: concatenation of the single projection matrices in one common projector delivers a different ROM than CURE. In particular, the overall ROM is *not* guaranteed to converge towards the HFM as its order $n^{\Sigma} = \sum_i n_i$ tends towards the order of the HFM, as it would be the case in standard projective MOR (there, the projection turns into a state transformation if $n = N$).

With regard to moment matching, however, one should note the following cutback of what was derived in Proposition 4.2. Although in the k-th reduction step moments can be matched both with $\mathbf{V}_k$ and $\mathbf{W}_k$, in general half of them is changed in the following step $k+1$. The reason is that in $\mathbf{V}$-based decompositions, the moments matched via $\mathbf{V}$ correspond to invariant zeros of $\widetilde{\mathbf{G}}_{r,k}^{R}(s)$, which still occur in the preceding steps as $\widetilde{\mathbf{G}}_{r,k}^{R}(s)$ always remains a factor in $\widetilde{\mathbf{G}}_{r,k+i}^{\Sigma,R}(s)$ for any $i \geq 0$. The moments matched via $\mathbf{W}_k$, on the other hand, lead to invariant zeros of $\mathbf{G}_{\perp,k}(s)$ whose input or output matrix is changed in the subsequent iteration. As this may completely change its zero configuration, the moments may no longer match. Accordingly, $\mathbf{W}_k$ only influences the poles of the momentary and overall ROM, but cannot be used to explicitly match given moments. Of

course the dual considerations hold true for **W**-based factorization.

To sum up, with purely KRYLOV-based reduction during CURE, one can match a total of $\sum_i^k (n_i) + n_k$ moments in k steps, when n_i are the dimensions of the single ROMs.

One important consequence is the following: If $\mathcal{H}_2$ optimal reduction is performed in every step, then the resulting overall ROM will *not* be a local $\mathcal{H}_2$ optimum, but only pseudo-optimal. One moment is still matched at the mirror images of the poles [122], because $\widetilde{\mathbf{G}}_{r,k}^{\Sigma,L}(s)$ or $\widetilde{\mathbf{G}}_{r,k}^{\Sigma,R}(s)$ still contains the corresponding invariant zero. But the second moment is no longer matched as the zeros of $\mathbf{G}_{\perp,k}(s)$ change in every iteration.

Accordingly, the CURE framework constitutes a kind of greedy algorithm. It is very well suited for choosing an optimal set of shifts for its *momentary* configuration. The sum of the single decisions, however, must not be expected to yield an optimum, but hopefully something close to that.

Note, in this context, that Proposition 4.3 implies a major feature of the CURE scheme:

Corollary 4.1. *If during CURE $\mathcal{H}_2$ pseudo-optimal reduction is performed in every iteration, such that $\widetilde{\mathbf{G}}_{r,k}^{L}(s)$ in **W**-based decomposition or $\widetilde{\mathbf{G}}_{r,k}^{R}(s)$ in **V**-based decomposition, respectively, are unity all-pass elements, then the $\mathcal{H}_2$ norm of the error model decays strictly monotonically, $\|\mathbf{G}_{e,k}\|_{\mathcal{H}_2} < \|\mathbf{G}_{e,k-1}\|_{\mathcal{H}_2}$, unless $\mathbf{G}_{r,k}(s)$ is a zero element.*

An open question remains which type of factorization should be performed after two-sided KRYLOV reduction, when both decompositions are valid. This question is significant in the context of the error bounds presented in Chapter 5 and will be taken up again therein.

4.3.4. CUREd IRKA

So far, we have ignored the problem of finding suitable ROMs in each of the reduction steps. As a first attempt to this end, we incorporate IRKA into the CURE framework ("CUREd IRKA"), so we search $\mathcal{H}_2$ optimal ROMs of order $n = 2$ and cumulate them iteratively to an overall ROM of order $n^\Sigma = 2, 4, 6, \ldots$, whose quality gets better in every step, because it is also $\mathcal{H}_2$ pseudo-optimal (cf. Corollary 4.1).

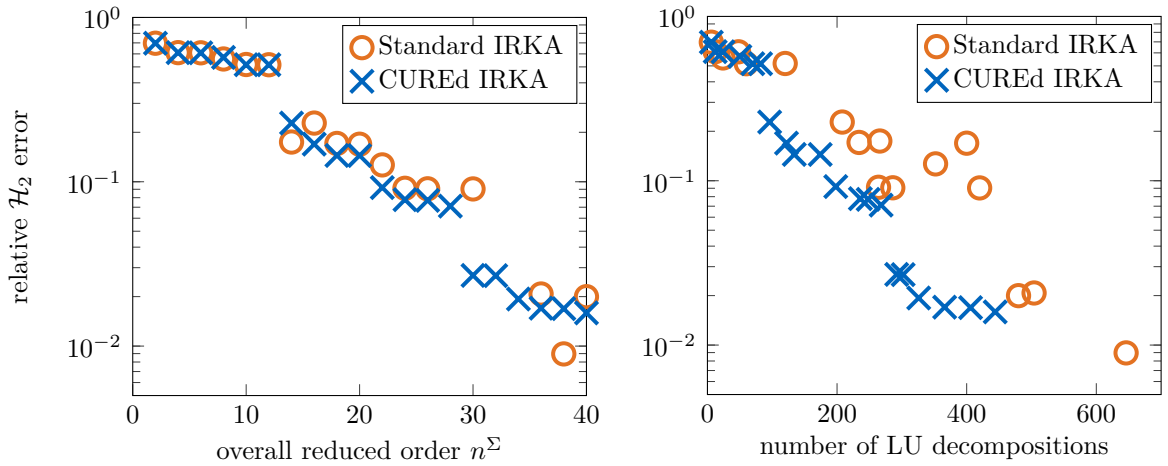
As a demonstrating example, we consider the reduction of the ISS benchmark model. We run through ten iterations of CUREd IRKA and compare the results to the outcome

of standard IRKA (reducing to the overall order n^Σ directly). One can see in Figure 4.4 that both approaches yield equally good approximation of the HFM for a given reduced order n^Σ. However, while the error decays monotonically for CUREd IRKA, worse approximants may result for higher reduced order in standard IRKA—depending on how good the identified local minimum actually is. Missing entries for some (even) n^Σ indicate orders for which IRKA did not converge within 20 steps, but returned an unstable ROM.

Also, Figure 4.4b) shows that the number of LSE solves required to run standard IRKA *once* for some given order n^Σ is typically higher than performing the *whole* CUREd IRKA process from $n = 2$, $n = 4$, etc. up to $n = n^\Sigma$. But CURE offers the possibility to continue or stop at any time, while a new run of IRKA starts from scratch. Accordingly, CUREd IRKA procedure offers more flexibility without loss of precision.

However, Figure 4.4b) also shows that the number of steps required during the single steps of CUREd IRKA varies. In fact, there was no guarantee IRKA would converge at all in each and every step of CURE, but unstable ROMs might have resulted as well.

To conclude: Although CURE offers a possibility to choose the reduced order on the fly without additional effort, the stability and convergence problems of IRKA are not remedied.



a) $\mathcal{H}_2$ error over reduced order

b) $\mathcal{H}_2$ error over number of LSE solves

Figure 4.4.: Comparison of Standard IRKA and CUREd IRKA

4.4. SPARK: A Stability-Preserving, Adaptive Rational Krylov Algorithm

We have seen that the use of IRKA inside the CURE scheme as illustrated in Section 4.3.4 brings about the convergence and stability issues related to IRKA. It is therefore the goal of this section to develop a descent algorithm with more favorable properties.

Such an alternative was published in [122] in the form of the Stability-Preserving, Adaptive Rational Krylov Algorithm (SPARK), which is bound to deliver ROMs of order $n = 2$. So although it is not suitable for finding useful approximations of the HFM in a single step, it very well fits into the CURE framework where the overall reduced model is built incrementally out of many low-order models.[2]

So far, SPARK only applies to SISO systems; for this section we will therefore assume $m = p = 1$, changing $\mathbf{B} \to \mathbf{b}$ and $\mathbf{C} \to \mathbf{c}$ to vectors.

4.4.1. Optimization-Based Computation of Shifts

The key idea that was presented in [122] to improve existing optimization based $\mathcal{H}_2$ model reduction schemes from [20, 21] was to restrict the search space to $\mathcal{H}_2$ pseudo-optimal ROMs that match moments at the mirror images of its poles, instead of modifying poles and residues of the ROM independently. This approach is valid because any local optimum is necessarily a pseudo-optimum; and a pseudo-optimum in whose vicinity there is no better ROM is a local optimum and therefore a Hermite interpolant about the mirror images of its poles [76].

This circumstance is sketched in Figure 4.5. While IRKA and the method presented in [21] search among the ROMs resulting from Hermite interpolation, until hopefully shifts and reduced poles lie vis-à-vis (i. e., a pseudo-optimum is found), SPARK seeks within the set of pseudo-optima until *two* moments match at the expansion points. The

[2] Please note that in [122], "SPARK" referred to the union of the cumulative reduction framework (which has been introduced under the name "CURE" in Section 4.3) and the descent shift selection strategy which will be presented in this section. The reason for the renaming was that these two components have partly been misconceived; in truth, cumulative reduction constitutes a concept on its own and can not only be applied in combination with the descent strategy, as was shown above.

intersection of these respective sets of ROMs contains precisely the candidates for $\mathcal{H}_2$ optima, so both algorithms look for the same thing but in different search spaces.

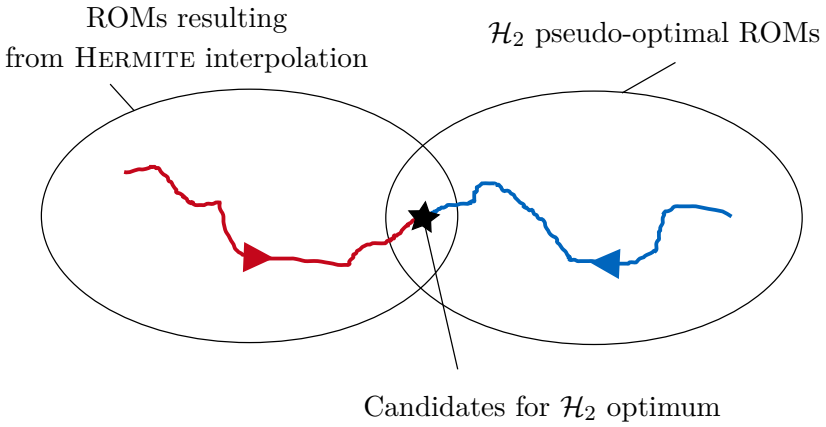


Figure 4.5.: IRKA vs. SPARK: Comparison of Search Space

The search among pseudo-optima has, however, a crucial advantage: The $\mathcal{H}_2$ error expression simplifies according to (3.27); besides, as $\|\mathbf{G}\|_{\mathcal{H}_2}$ is independent of the reduction, we can equivalently use the very convenient cost functional

$$\mathcal{J} := \|\mathbf{G}_e\|_{\mathcal{H}_2}^2 - \underbrace{\|\mathbf{G}\|_{\mathcal{H}_2}^2}_{const.} = -\|\mathbf{G}_r\|_{\mathcal{H}_2}^2 , \tag{4.18}$$

which only depends on the reduced order matrices.

In comparison to [20], the number of optimization variables is halved, as poles and residues are manipulated together in a judicious way. In fact, for $n = 2$, the parameter space reduces to the first quadrant of $\mathbb{R}^2$:

Lemma 4.1. *The set of all minimal, asymptotically stable, and pseudo-optimal ROMs of order $n = 2$ can be parametrized by two real positive parameters $a, b \in \mathbb{R}^+$.*

Proof. An order 2 ROM is uniquely determined by two poles $\lambda_{r,1}, \lambda_{r,2}$ (with JORDAN block if $\lambda_{r,1} = \lambda_{r,2}$, otherwise it would not be minimal) and the respective residuals. According to [76], for given poles, the optimal residual configuration is uniquely determined due to HILBERT's projection theorem. Accordingly, the set of asymptotically stable pole configurations and the set of pseudo-optimal ROMs are in a one-to-one relation. Now without loss of generality, let the characteristic polynomial of the ROM be given by $\det(s\mathbf{E}_r - \mathbf{A}_r) = s^2 + as + 4b$. Then following HURWITZ' criterion, for an asymptotically stable pole configuration, $a, b > 0$ is necessary and sufficient. □

In [122], unaware of the projection-based algorithm PORK [163] (cf. Section 3.6.5), this approach was realized not in a projective way, but the reduced transfer function was constructed in the frequency domain to feature the required properties, i. e. to match moments at

$$\sigma_1 = a + \sqrt{a^2 - b} \quad \text{and} \quad \sigma_2 = a - \sqrt{a^2 - b}, \tag{4.19}$$

and to have poles at the mirror images, $\lambda_{r,i} = -\sigma_{r,i}$. This basically worked well for the actual optimization procedure, yet had two minor drawbacks. Firstly, the case $\sigma_1 = \sigma_2$ had to be excluded due to a singularity (numerical instabilities, however, could occur even for roughly equal shifts $\sigma_1 \approx \sigma_2$). Secondly, and perhaps more importantly, the determined $\mathcal{H}_2$ optimal ROM could not be directly related to the high-dimensional state space for lack of associated projection matrices $\mathbf{V}$ and $\mathbf{W}$. These are, however, necessary for the factorization of the error model according to Section 4.2 (in particular, to compute $\mathbf{B}_\perp$ or $\mathbf{C}_\perp$) and for the application of the CURE framework.

Therefore, after convergence of SPARK, in [122] the $\mathcal{H}_2$ optimal shifts were used to re-compute the ROM as a HERMITE interpolant by two-sided KRYLOV reduction, which—in theory—is equivalent, as discussed above: matching two moments at each of the $\mathcal{H}_2$ optimal shifts implicitly sites poles at their mirror images, while for a pseudo-optimum (*one* moment is matched per shift and poles are explicitly placed vis-à-vis) the second moments match unsolicited.

In practice, however, round-off errors and incomplete convergence can cause non-optimal shifts as a result of the optimization, so the above considerations do not hold true anymore, and HERMITE interpolation delivers a ROM which is neither $\mathcal{H}_2$ optimal *nor pseudo-optimal*! Therefore, the factors $\widetilde{\mathbf{G}}_r(s)$ in the error factorization are not all-pass and the monotonicity of the $\mathcal{H}_2$ error (cf. Corollary 4.1) is lost.

For that reason, the procedure of [122] was enhanced by incorporating PORK, which delivered a projective embedding of the pseudo-optimal ROM. The whole approach is presented in the following.

4.4.2. Enhanced Formulation of SPARK

We use again the abbreviation $\mathbf{A}_\sigma := (\mathbf{A} - \sigma\mathbf{E})$.

Lemma 4.2. *Let a, b be real positive numbers and define $\sigma_{1,2} := a \pm \sqrt{a^2 - b}$, such that $\sigma_1, \sigma_2 \in \mathbb{R}$ or $\sigma_1, \sigma_2 = \overline{\sigma_1} \in \mathbb{C}$. Then, a real basis of the corresponding* Krylov *subspace is given by*

$$\mathbf{V} = \left[\ \tfrac{1}{2}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} + \tfrac{1}{2}\mathbf{A}_{\sigma_2}^{-1}\mathbf{b} \ , \quad \mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} \ \right] \in \mathbb{R}^{N\times 2} \tag{4.20}$$

which solves Sylvester *equation (3.3) with*

$$\mathbf{S}_V = \begin{bmatrix} \frac{\sigma_1+\sigma_2}{2} & 1 \\ \left(\frac{\sigma_1-\sigma_2}{2}\right)^2 & \frac{\sigma_1+\sigma_2}{2} \end{bmatrix} = \begin{bmatrix} a & 1 \\ a^2-b & a \end{bmatrix}, \qquad \widetilde{\mathbf{C}}_r = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

The eigenvalues of $\mathbf{S}_V$ are σ_1 and σ_2; if $\sigma_1 = \sigma_2$, $\mathbf{S}_V$ contains a Jordan *structure.*

Proof. To show that $\mathbf{V}$ is real even for complex σ_1, σ_2, we notice that then $\mathbf{A}_{\sigma_2} = \overline{\mathbf{A}_{\sigma_1}}$, so the imaginary parts in $\frac{1}{2}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}$ and $\frac{1}{2}\mathbf{A}_{\sigma_2}^{-1}$ cancel in the first column $\mathbf{v}_1$. The second column $\mathbf{v}_2$ is also real because it solves the purely real equation

$$\mathbf{b} = (\mathbf{A} - \sigma_1\mathbf{E})\,\mathbf{E}^{-1}\,(\mathbf{A} - \sigma_2\mathbf{E})\,\mathbf{v}_2 = \Big(\mathbf{A}\mathbf{E}^{-1}\mathbf{E} - \underbrace{(\sigma_1 + \overline{\sigma_1})}_{\in\mathbb{R}}\mathbf{A} + \underbrace{\sigma_1\overline{\sigma_1}}_{\in\mathbb{R}}\mathbf{E}\Big)\mathbf{v}_2.$$

Now we consider the two columns of the Sylvester equation separately:

$$\begin{aligned}
\mathbf{A}\mathbf{v}_1 &= \mathbf{A}\left[\tfrac{1}{2}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} + \tfrac{1}{2}\mathbf{A}_{\sigma_2}^{-1}\right] \\
&= \tfrac{1}{2}\mathbf{A}_{\sigma_1}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} + \tfrac{\sigma_1}{2}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} + \tfrac{1}{2}\mathbf{A}_{\sigma_2}\mathbf{A}_{\sigma_2}^{-1}\mathbf{b} + \tfrac{\sigma_2}{2}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{b} \\
&= \mathbf{b} + \mathbf{E}\left[\tfrac{\sigma_1}{2}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} + \tfrac{\sigma_1}{2}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{b}\right] \\
&= \mathbf{b} + \mathbf{E}\left[\tfrac{\sigma_1+\sigma_2}{4}\left(\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} + \mathbf{A}_{\sigma_2}^{-1}\mathbf{b}\right) + \tfrac{\sigma_1-\sigma_2}{4}\left(\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} - \mathbf{A}_{\sigma_2}^{-1}\mathbf{b}\right)\right] \\
&= \mathbf{b} + \tfrac{\sigma_1+\sigma_2}{2}\mathbf{E}\mathbf{v}_1 + \tfrac{\sigma_1-\sigma_2}{4}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\left[\mathbf{A}_{\sigma_2} - \mathbf{A}_{\sigma_1}\right]\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} \\
&= \mathbf{b} + \tfrac{\sigma_1+\sigma_2}{2}\mathbf{E}\mathbf{v}_1 + \tfrac{\sigma_1-\sigma_2}{4}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\left[-\sigma_2\mathbf{E} + \sigma_1\mathbf{E}\right]\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} \\
&= \mathbf{b} + \tfrac{\sigma_1+\sigma_2}{2}\mathbf{E}\mathbf{v}_1 + \left(\tfrac{\sigma_1-\sigma_2}{2}\right)^2\mathbf{E}\mathbf{v}_2 \\
\mathbf{A}\mathbf{v}_2 &= \mathbf{A}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} \\
&= \mathbf{A}_{\sigma_2}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} + \sigma_2\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} \\
&= \tfrac{1}{2}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} + \tfrac{1}{2}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} + \tfrac{1}{2}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{b} - \tfrac{1}{2}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{b} + \sigma_2\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} \\
&= \mathbf{E}\mathbf{v}_1 + \tfrac{1}{2}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\left[\mathbf{A}_{\sigma_2} - \mathbf{A}_{\sigma_1}\right]\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} + \sigma_2\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} \\
&= \mathbf{E}\mathbf{v}_1 + \tfrac{\sigma_1-\sigma_2}{2}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} + \sigma_2\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} = \mathbf{E}\mathbf{v}_1 + \tfrac{\sigma_1+\sigma_2}{2}\mathbf{E}\mathbf{v}_2
\end{aligned}$$

The proof for the eigenvalues is straightforward. □

Corollary 4.2. *Let a, b be real positive numbers, define $\sigma_{1,2} = a \pm \sqrt{a^2 - b}$, and compute $\mathbf{V} \in \mathbb{R}^{N\times 2}$ according to (4.20). Then the ROM defined by*

$$\mathbf{E}_r = \mathbf{I}_2, \quad \mathbf{A}_r = \begin{bmatrix} -3a & 1 \\ -3a^2 - b & a \end{bmatrix}, \quad \mathbf{B}_r = \begin{bmatrix} -4a \\ -4a^2 \end{bmatrix}, \quad \mathbf{C}_r = \mathbf{CV} \tag{4.21}$$

is an $\mathcal{H}_2$ pseudo-optimal approximant of the HFM whose $\mathcal{H}_2$ norm is given by $\|\mathbf{G}_r\|^2_{\mathcal{H}_2} = \mathbf{C}_r\mathbf{P}_r\mathbf{C}_r^T$ with the reduced Controllability Gramian

$$\mathbf{P}_r = \begin{bmatrix} 4a & 4a^2 \\ 4a^2 & 4a(a^2+b) \end{bmatrix}. \tag{4.22}$$

Proof. The proof is straightforward following the PORK algorithm. □

It is easy to verify that the eigenvalues of $\mathbf{A}_r$ are given by $-\sigma_1$ and $-\sigma_2$, so they are indeed the mirror images of the expansion points.

As stated above, the cost functional (4.18), $\mathcal{J}(a,b) = -\|\mathbf{G}_r\|^2_{\mathcal{H}_2} = -\mathbf{C}_r\mathbf{P}_r\mathbf{C}_r^T$ is well suited to find an $\mathcal{H}_2$ optimum by means of minimization. Note that given $a, b > 0$, the only quantities in (4.21) that depend on the HFM are the elements of $\mathbf{c}_r$:

$$\mathbf{c}_r = \begin{bmatrix} c_{r,1}, & c_{r,1} \end{bmatrix} = \mathbf{cV} = \begin{bmatrix} \mathbf{cv}_1, & \mathbf{cv}_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\mathbf{c}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} + \frac{1}{2}\mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{b}\ , & \mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} \end{bmatrix}. \tag{4.23}$$

These are not moments of the HFM, but encode the respective information differently. The advantage is that complex conjugated shifts as well as real shifts and even double shifts are incorporated in this formulation; also, it is robust to situations when $\sigma_1 \approx \sigma_2$.

If $\mathbf{V}$ is not required explicitly, two LSE solves suffice to compute $\mathbf{c}_r$, because both entries of $\mathbf{c}_r$ can be written as *products* containing the factors $\mathbf{l}_1 := \mathbf{c}\mathbf{A}_{\sigma_2}^{-1}$ and $\mathbf{r}_1 := \mathbf{A}_{\sigma_1}^{-1}\mathbf{b}$. So once one has solved for $\mathbf{l}_1$ and $\mathbf{r}_1$, the vector $\mathbf{c}_r$ can be computed as

$$\mathbf{c}_r = \begin{bmatrix} \frac{1}{2}\mathbf{c}\mathbf{r}_1 + \frac{1}{2}\mathbf{l}_1\mathbf{b}, & \mathbf{l}_1\mathbf{E}\mathbf{r}_1 \end{bmatrix}. \tag{4.24}$$

4.4.3. Analytic Gradient and Hessian

To efficiently run optimization algorithms solving the constrained minimization problem

$$\arg\min_{a>0, b>0} \mathcal{J}(a,b), \tag{4.25}$$

it is often beneficial to supply analytic gradient and Hessian matrix. Though not aesthetic in their mathematical form, the respective formulas presented in the following theorem provide an efficient way to compute the required information (cf. Remark 4.1 below).

Source 4.2: Computation of Cost Functional, Gradient, and Hessian

```
function [J, g, H] = CostFunctionH2(A, B, C, E, p, r, l)
% Enhanced SPARK Cost Functional
%   Input:  A,B,C,E: HFM matrices;
%           p:       parameter vector [a,b];
%           r,l:     left an right rational Krylov sequence;
%   Output: cost functional J; gradient g; Hessian H
%

    a = p(1); b = p(2);  l1 = l(1,:); r1 = r(:,1);
    Pr = [4*a, 4*a^2; 4*a^2 4*a*(a^2+b)];  Cr = [0.5*(C*r1 + l1*B), l1*E*r1];
    J = real(-Cr*Pr*Cr');
    if nargout==1, return, end
    l2 = l(2,:); r2 = r(:,2);  l3 = l(3,:); r3 = r(:,3);

    dPrda =     [4, 8*a; 8*a, 12*a^2 + 4*b];    dPrdb =     [0, 0; 0,  4*a];
    ddPrdada =  [0, 8; 8, 24*a];                ddPrdadb =  [0, 0; 0, 4];
    dcrda =     [0.5*(C*r2 + l2*B) + a*(l2*E*r1 + l1*E*r2), 2*l2*A*r2];
    dcrdb =     [-0.5*(l2*E*r1 + l1*E*r2), -l2*E*r2];
    ddcrdada =  [C*r3 + l3*B + 4*a*l1*E*r3 + 4*a*l3*E*r1 + 2*a*l2*E*r2 + 2*l2*A*r2 ...
                      + 4*a^2*l2*E*r3 + 4*a^2*l3*E*r2, ...
                 4*l3*A*r2 + 4*l2*A*r3 + 8*a*l3*A*r3];
    ddcrdadb =  [-l3*E*r1 - l1*E*r3 - l2*E*r2 - 2*a*l2*E*r3 - 2*a*l3*E*r2, -4*l3*A*r3];
    ddcrdbdb =  [ l3*E*r2 + l2*E*r3,  2*l3*E*r3];

    g = real([-Cr*dPrda*Cr', -Cr*dPrdb*Cr'] - 2*Cr*Pr*[dcrda; dcrdb]');
    H = [-2*ddcrdada*Pr*Cr'-4*dcrda*dPrda*Cr'-2*dcrda*Pr*dcrda'-Cr*ddPrdada*Cr', ...
         -2*ddcrdadb*Pr*Cr'-2*dcrdb*dPrda*Cr'-2*dcrda*dPrdb*Cr'-2*dcrda*Pr*dcrdb'-...
         Cr*ddPrdadb*Cr'; 0,-2*ddcrdbdb*Pr*Cr'-4*dcrdb*dPrdb*Cr'-2*dcrdb*Pr*dcrdb'];
    H(2,1)=H(1,2); H=real(H);
end
```

Theorem 4.3. *Given a parameter vector* $\mathbf{p} = [a,\ b]$ *with* $a, b \in \mathbb{R}^+$, *define* $\sigma_{1,2} := a \pm \sqrt{a^2 - b}$ *as before and*

$$\begin{aligned} \mathbf{l}_i &:= \mathbf{c}\left(\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\right)^{i-1}\mathbf{A}_{\sigma_2}^{-1} \quad \in \mathbb{C}^{1\times N} \\ \mathbf{r}_i &:= \left(\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\right)^{i-1}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} \quad \in \mathbb{C}^{N\times 1} \end{aligned} \qquad \text{for } i \in \{1,2,3\}.$$

Then the cost functional $\mathcal{J} = -\|\mathbf{G}_r\|_{\mathcal{H}_2}^2 = -\mathbf{c}_r\mathbf{P}_r\mathbf{c}_r^T$, *its gradient vector* $\mathbf{g}$, *and the Hessian matrix* $\mathbf{H}$ *follow from matrix-vector products and small-scale operations according to the following formulas:*

$$\mathbf{g} = \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\mathbf{p}} = \left[\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}a},\ \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}b}\right] = \left[-2\tfrac{\partial \mathbf{c}_r}{\partial a}\mathbf{P}_r\mathbf{c}_r - \mathbf{c}_r\tfrac{\partial \mathbf{P}_r}{\partial a}\mathbf{c}_r^T,\quad -2\tfrac{\partial \mathbf{c}_r}{\partial b}\mathbf{P}_r\mathbf{c}_r - \mathbf{c}_r\tfrac{\partial \mathbf{P}_r}{\partial b}\mathbf{c}_r^T\right],$$

$$\mathbf{H} = \frac{\mathrm{d}^2\mathcal{J}}{\mathrm{d}\mathbf{p}^2} = \begin{bmatrix} \frac{\mathrm{d}^2\mathcal{J}}{\mathrm{d}a^2} & \frac{\mathrm{d}^2\mathcal{J}}{\mathrm{d}a\mathrm{d}b} \\ \frac{\mathrm{d}^2\mathcal{J}}{\mathrm{d}a\mathrm{d}b} & \frac{\mathrm{d}^2\mathcal{J}}{\mathrm{d}b^2} \end{bmatrix},$$

$$\begin{aligned}
\frac{\mathrm{d}^2\mathcal{J}}{\mathrm{d}a^2} &= -2\tfrac{\partial^2\mathbf{c}_r}{\partial a^2}\mathbf{P}_r\mathbf{c}_r^T - 4\tfrac{\partial\mathbf{c}_r}{\partial a}\tfrac{\partial\mathbf{P}_r}{\partial a}\mathbf{c}_r^T - 2\tfrac{\partial\mathbf{c}_r}{\partial a}\mathbf{P}_r\tfrac{\partial\mathbf{c}_r^T}{\partial a} - \mathbf{c}_r\tfrac{\partial^2\mathbf{P}_r}{\partial a^2}\mathbf{c}_r^T \\
\frac{\mathrm{d}^2\mathcal{J}}{\mathrm{d}b^2} &= -2\tfrac{\partial^2\mathbf{c}_r}{\partial b^2}\mathbf{P}_r\mathbf{c}_r^T - 4\tfrac{\partial\mathbf{c}_r}{\partial b}\tfrac{\partial\mathbf{P}_r}{\partial b}\mathbf{c}_r^T - 2\tfrac{\partial\mathbf{c}_r}{\partial b}\mathbf{P}_r\tfrac{\partial\mathbf{c}_r^T}{\partial b} - \mathbf{c}_r\tfrac{\partial^2\mathbf{P}_r}{\partial b^2}\mathbf{c}_r^T \\
\frac{\mathrm{d}^2\mathcal{J}}{\mathrm{d}a\mathrm{d}b} &= -2\tfrac{\partial^2\mathbf{c}_r}{\partial a\partial b}\mathbf{P}_r\mathbf{c}_r^T - 2\tfrac{\partial\mathbf{c}_r}{\partial b}\tfrac{\partial\mathbf{P}_r}{\partial a}\mathbf{c}_r^T - 4\tfrac{\partial\mathbf{c}_r}{\partial a}\tfrac{\partial\mathbf{P}_r}{\partial b}\mathbf{c}_r^T - 2\tfrac{\partial\mathbf{c}_r}{\partial a}\mathbf{P}_r\tfrac{\partial\mathbf{c}_r^T}{\partial b} - \mathbf{c}_r\tfrac{\partial^2\mathbf{P}_r}{\partial a\partial b}\mathbf{c}_r^T.
\end{aligned}$$

Therein,

$$\mathbf{c}_r = \begin{bmatrix} \frac{1}{2}\mathbf{l}_1\cdot\mathbf{b} + \frac{1}{2}\mathbf{c}\cdot\mathbf{r}_1, & \mathbf{l}_1\mathbf{E}\mathbf{r}_1, \end{bmatrix} \in \mathbb{R}^{1\times 2}.$$

$$\frac{\partial\mathbf{P}_r}{\partial a} = \begin{bmatrix} 4 & 8a \\ 8a & 12a^2+4b \end{bmatrix}, \frac{\partial\mathbf{P}_r}{\partial b} = \begin{bmatrix} 0 & 0 \\ 0 & 4a \end{bmatrix}, \frac{\partial^2\mathbf{P}_r}{\partial a^2} = \begin{bmatrix} 0 & 8 \\ 8 & 24a \end{bmatrix}, \frac{\partial^2\mathbf{P}_r}{\partial a\partial b} = \begin{bmatrix} 0 & 0 \\ 0 & 4 \end{bmatrix},$$

$$\begin{aligned}
\frac{\partial c_{r,1}}{\partial a} &= \tfrac{1}{2}\mathbf{l}_2\cdot\mathbf{b} + \tfrac{1}{2}\mathbf{c}\cdot\mathbf{r}_2 + a\left(\mathbf{l}_1\mathbf{E}\mathbf{r}_2 + \mathbf{l}_2\mathbf{E}\mathbf{r}_1\right), & \frac{\partial c_{r,1}}{\partial b} &= -\tfrac{1}{2}\mathbf{l}_1\mathbf{E}\mathbf{r}_2 - \tfrac{1}{2}\mathbf{l}_2\mathbf{E}\mathbf{r}_1 \\
\frac{\partial c_{r,2}}{\partial a} &= 2\mathbf{l}_2\mathbf{A}\mathbf{r}_2, & \frac{\partial c_{r,2}}{\partial b} &= -\mathbf{l}_2\mathbf{E}\mathbf{r}_2.
\end{aligned}$$

Proof. The proof is straightforward except for the derivatives of the entries of $\mathbf{c}_r$ with respect to a and b; those include some lengthy linear algebraic computations and are therefore shifted to Appendix A.1. □

Remark 4.1. *Note that* $\mathbf{l}_i$, $\mathbf{r}_i$ *are recursively given by*

$$\begin{aligned}
\mathbf{l}_1 &= \left[\mathbf{c}\,\mathbf{U}_2^{-1}\right]\mathbf{L}_2^{-1} \\
\mathbf{r}_1 &= \mathbf{U}_1^{-1}\left[\mathbf{L}_1^{-1}\mathbf{b}\right]
\end{aligned} \quad \textit{and} \quad \begin{aligned}
\mathbf{l}_i &= \left[(\mathbf{l}_{i-1}\cdot\mathbf{E})\,\mathbf{U}_2^{-1}\right]\mathbf{L}_2^{-1} \\
\mathbf{r}_i &= \mathbf{U}_1^{-1}\left[\mathbf{L}_1^{-1}(\mathbf{E}\cdot\mathbf{r}_{i-1})\right]
\end{aligned} \quad \textit{for } i\in\{2,3\}$$

if $\mathbf{L}_1,\mathbf{U}_1$ *and* $\mathbf{L}_2,\mathbf{U}_2$ *describe LU-decompositions of* $\mathbf{A}_{\sigma_1}$ *and* $\mathbf{A}_{\sigma_2}$*, respectively. So the total of six LSE solves requires two LU-decompositions and twelve backward substitutions. If* $\sigma_1=\sigma_2$*, one LU suffices. If* $\sigma_2=\overline{\sigma_1}\in\mathbb{C}$*, one complex LU instead of two real ones is needed, the total numerical effort of which is comparable.*

Source 4.2 shows how the enhanced computation of cost functional, gradient and Hessian can be realized in MATLAB. The function accepts the two-dimensional parameter vector $\mathbf{p}$ containing a and b, the vectors $\mathbf{l}_i$ and $\mathbf{r}_i$, each concatenated in one matrix, and the HFM matrices.

A possible implementation of the whole extended SPARK (ESPARK) algorithm is given in Source 4.3. Note that this enhanced version delivers almost the same result as the first formulation presented in [122], as the underlying cost functional is the same, just more general and more robust in the new (ESPARK) formulation. Figure 4.6 shows simulation results for the "beam" benchmark model; in fact, this figure looks very much alike the result printed in [122].

Source 4.3: Enhanced Stability Preserving Adaptive Rational Krylov (ESPARK)

```
function [V,S_V,Crt,k] = ESPARK(A,B,C,E,s0)
% Enhanced Stability Preserving Adaptive Rational Krylov
%   Input:  A,B,C,E:   HFM matrices;
%           s0:        Initial shifts
%   Output: V,S_V,Crt: Input Krylov subspace,  A*V - E*V*S_V - B*Crt = 0
%

p0 = [(s0(1)+s0(2))/2, s0(1)*s0(2)];
N = size(A,1); precond = eye(2); t = tic;
opts=optimset('TolFun',1e-15*abs(CostFunction(p0)),'TolX',1e-20, ...
    'Display','none', 'Algorithm','trust-region-reflective', ...
    'GradObj','on','Hessian','on', 'MaxFunEvals',100,'MaxIter',100);
precond = diag(p0);
[p_opt,~,~,output] = fmincon(@CostFunction,p0/precond,[],[],[],[],[0;0],[inf;inf],[],opts);
p_opt = p_opt*precond; k=output.funcCount;
disp(['ESPARK required ' num2str(k) ' LUs and ' num2str(toc(t),'%.1f') 'seconds.'])
v1  = Q1*(U1\(L1\(P1*B))); v12= Q2*(U2\(L2\(P2*B))); v2 = Q2*(U2\(L2\(P2*(E*v1))));
V   = full(real([v1/2 + (v12/2+p_opt(1)*v2), v2*sqrt(p_opt(2))]));
S_V = [2*p_opt(1), sqrt(p_opt(2)); -sqrt(p_opt(2)), 0]; Crt = [1 0];

function [J, g, H] = CostFunction(p)
    % compute Krylov sequences l_i, r_i
    p = p*precond;  s = p(1)+[1 -1]*sqrt(p(1)^2-p(2)); r = zeros(N,3); l = r.';
    [L1,U1,P1,Q1] = lu(sparse(A-s(1)*E));
    if real(s(1))==real(s(2)) % complex conjugated or double shifts
        L2=conj(L1);U2=conj(U1);P2=P1;Q2=Q1;
    else                      % two different real shifts
        [L2,U2,P2,Q2] = lu(sparse(A-s(2)*E));
    end
    solveLSE1 = @(x) Q1*(U1\(L1\(P1*x)));  solveLSE2 = @(x) x*Q2/U2/L2*P2;
    r(:,1) = solveLSE1(B(:,1)); l(1,:) = solveLSE2(C(1,:));
    r(:,2) = solveLSE1(r(:,1)); l(2,:) = solveLSE2(l(1,:));
    r(:,3) = solveLSE1(r(:,2)); l(3,:) = solveLSE2(l(2,:));
    % compute cost, gradient, and Hessian
    [J,g,H] = CostFunctionH2(A, B(:,1), C(1,:), E, p, r, l);
    g = g * precond; H = precond * H * precond;
end
end
```

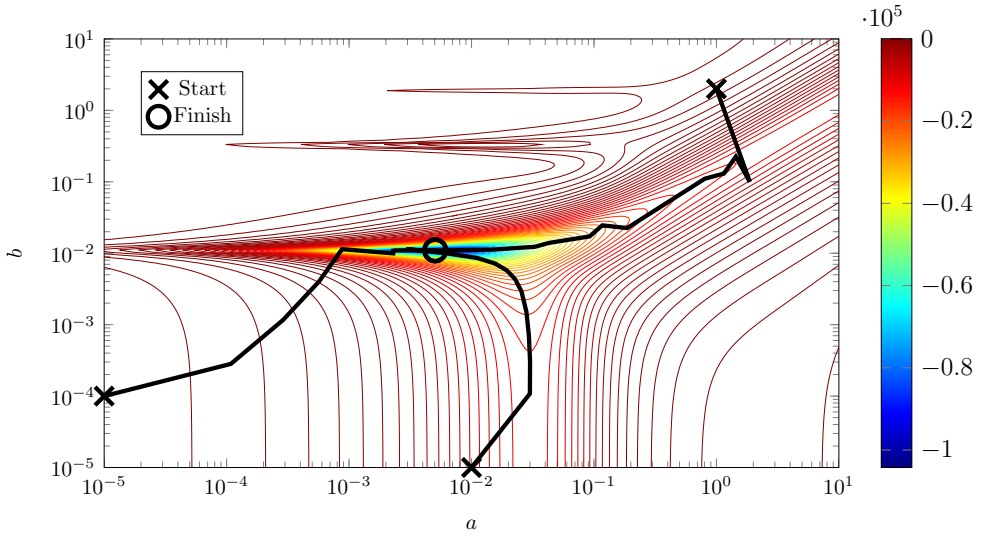


Figure 4.6.: Process of Enhanced SPARK for Various Initial Parameter Values

4.4.4. Speed-Up due to Model Function

Although SPARK in its enhanced formulation describes a globally convergent Trust Region method (at least under mild assumptions), its convergence requires a rather high number of LSE solves or LU-decompositions.

In the following, we will therefore exploit the benefits of a so-called model function.

As a start, we note that the trust region algorithm locally approximates the cost functional with a quadric defined by $\mathcal{J}$, gradient $\mathbf{g}$, and Hessian matrix $\mathbf{H}$, i. e. a quadratic TAYLOR polynomial. The overall shape of the cost functional, however, is not at all of polynomial character, but rather a rational function, as it tends to zero or finite values at the rim of the parameter domain ("trampoline" shape). Figure 4.7, for instance, shows $\mathcal{J}$ as in (4.18) over a and b for the Clamped Beam benchmark model (in fact, this is a 3D-visualization of Figure 4.6). Accordingly, the polynomial approximation handed to the optimizer is likely to fit only very locally.

The question is, whether the "expensive" information on the HFM which is contained in the KRYLOV sequences $\mathbf{l}_i$ and $\mathbf{r}_i$ might not be used in a more effective way than by just building $\mathcal{J}$, $\mathbf{g}$, and $\mathbf{H}$ from them. What is more, we know from Section 3.4 that computing more than three columns of a rational KRYLOV subspace increases the numerical effort only slightly once an LU-decomposition is available anyway. Therefore, the following procedure is suggested.

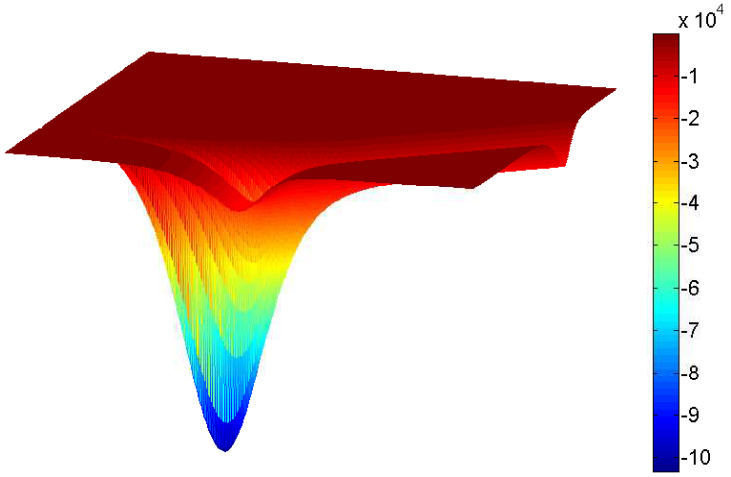


Figure 4.7.: Typical Shape of Cost Functional $\mathcal{J}(a, b)$

Given an initial parameter value $\mathbf{p}_0 = (a_0, b_0)$, we reduce the large-scale model by PADÉ approximation about the respective shifts defined by $\mathbf{p}_0$ to quite small order, say $n = 6$. Then, we run the Trust Region optimizer as before, but supply it not with the *true* values of $\mathcal{J}$, $\mathbf{g}$, and $\mathbf{H}$ (which require the solution of N-dimensional linear systems of equations), but we compute the cost functional and its derivatives based on the intermediate low-order ROM (this yields the simplified so-called "model function"). Of course this massively speeds up the procedure, and in fact the optimizer may require as many steps as it wants to—the numerical effort will be quite manageable.

Once the Trust Region algorithm has converged to some new parameter $\mathbf{p}_1 = (a_1, b_1)$, this point constitutes a local minimum of the model function, i. e. of the *approximated* cost functional. Now, we compute the *true* cost functional $\mathcal{J}(a_1, b_1)$, which requires the solution of two LSEs and typically involves an LU-decomposition about the shifts that correspond to $\mathbf{p}_1$. Then we decide:

- If $\mathcal{J}(a_1, b_1) < \mathcal{J}(a_0, b_0)$, then $\mathbf{p}_1$ yields an improvement and we are getting closer to a minimum. Therefore, we use the available LU-factors to compute additional KRYLOV vectors about the newly found shifts. These vectors are incorporated in our projection matrices with which we update the model function by standard projection. The model function now matches moments about the initial shifts (corresponding to $\mathbf{p}_0$) and the determined shifts (corresponding to $\mathbf{p}_1$), so it approximates the true cost functional well in both vicinities. Accordingly, we can restart the trust

region optimization with initial position $\mathbf{p}_1$, and proceed with the newly found local minimum $\mathbf{p}_2$ of the updated model function.

- If, on the other hand, $\mathcal{J}(a_1, b_1) > \mathcal{J}(a_0, b_0)$, then the model function was unfit, because its minimum corresponds to a degradation of the true cost functional. In this case, the update of the intermediate ROM will strongly change the model function in the neighborhood of $\mathbf{p}_1$. So as we restart the trust algorithm from the unchanged initial parameter $\mathbf{p}_0$, this time it will not converge to $\mathbf{p}_1$ but hopefully to a point $\mathbf{p}_2$ which is better suited.
- If at some step the relative change between $\mathbf{p}_i$ and $\mathbf{p}_{i-1}$ is very small, we seem to have found a minimum of the true cost functional and can stop the algorithm. The final parameter $\mathbf{p}$ is then converted towards two shifts and the discovered ROM is incorporated into the overall ROM in the CURE scheme.

We call this basic algorithm MESPARK (Model function based Extended SPARK). The philosophy behind it is that the information contained in the Krylov subspace vectors, whose computation is the most expensive part of the algorithm, should be used to the full extent. During IRKA and (standard) SPARK, the Krylov subspaces computed in one iteration are immediately overwritten in the following one. Due to the model function, on the other hand, it is now well possible to incorporate all available information on the HFM transfer function in order to find a local optimum as quickly as possible.

We will demonstrate the effectiveness with the help of the Clamped Beam benchmark example, for which we have already considered the standard SPARK algorithm before. Again, we start from the initial shifts $1 \pm 1i$ and run the code given in Source 4.4. Firstly, a Padé approximant of order $n = 6$ about the initial shifts is computed. Its contour plot can be seen in the upper plot of Figure 4.8. Then the trust region algorithm is run and requires 18 steps to find a minimum of the approximated cost function (the path is also printed in Figure 4.8). Note that no large-scale operations are performed during the optimization, so it lasts only milliseconds. Now, the true cost functional at $\mathbf{p}_1$ is evaluated (to this end, two real LU-decomposition about the shifts are performed) and indeed constitutes an improvement. Therefore, the model function is updated and the optimizer restarted from $\mathbf{p}_1$. It converges in another 17 steps (see middle plot in

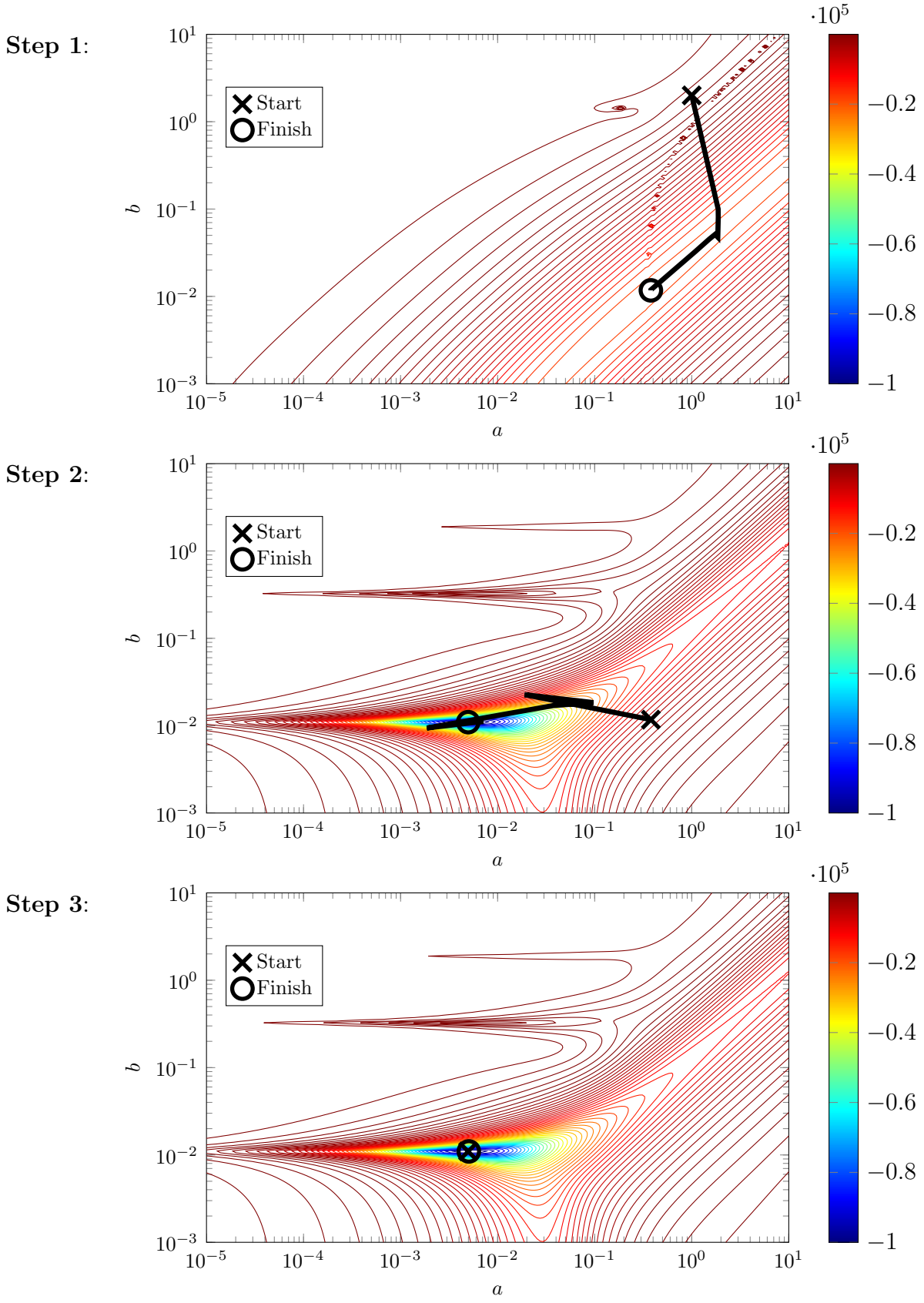


Figure 4.8.: Process of MESPARK

Figure 4.8) and again delivers an improvement. The model function is again updated and the procedure continued. After one more step the parameter does not change any more—in fact, one can see in Figure 4.8 that after the third step, the model function hardly changes, so a minimum of the true cost functional could indeed be found in only four steps.

4.4.5. Preconditioning and further Numerical Aspects

Preconditioning is a powerful technique to improve the convergence behavior of optimizers. Although the details of optimization go beyond the scope of this thesis, it is worth mentioning that even elementary preconditioning can have a strong effect. In Source 4.4, two ways of diagonal scaling are suggested. The first one (line 22, in comment) is based on the initial parameter value and scales the a- and b-axes such that the initial value is $(1, 1)$; this mends the influence of significantly different magnitudes in a and b. The other approach (line 23, active) uses the Hessian matrix at the initial parameter. More sophisticated preconditioning techniques may of course strongly support convergence.

Another numerical aspect of the ESPARK algorithm must be highlighted. Looking at the reduced order matrices

$$\mathbf{A}_r = \begin{bmatrix} -3a & 1 \\ -3a^2 - b & a \end{bmatrix} \quad \text{and} \quad \mathbf{B}_r = \begin{bmatrix} -4a \\ -4a^2 \end{bmatrix}$$

as defined by Corollary 4.2, it is clear that large values of a lead to numbers of very different magnitude, which compromises the numerical condition. This is illustrated in Figure 4.9a), which shows the common logarithm of the condition number of $\mathbf{A}_r$ over a and b.

One must note that this is not related to the considered model, but an inherent property of the formulation in Corollary 4.2: $\mathbf{A}_r$ only depends on a and b. Although the projection matrix $\mathbf{V} = \left[\frac{1}{2}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} + \frac{1}{2}\mathbf{A}_{\sigma_2}^{-1}\mathbf{b}, \quad \mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}\right]$ as defined in Lemma 4.2 provides a nice analytical parametrization, it is obviously not expedient from a numerical point of view. Other parametrizations following from a change of basis $\hat{\mathbf{V}} := \mathbf{V}\mathbf{T}$ may lead to more numerical robustness.

Source 4.4: Model Function Based Extended SPARK

```
function [V,S_V,Crt,k] = MESPARK(A,B,C,E,s0)
% Model Function based Enhanced Stability Preserving Adaptive Rational Krylov
%   Input:  A,B,C,E:   HFM matrices;
%           s0:        Initial shifts
%   Output: V,S_V,Crt: Input Krylov subspace,  A*V - E*V*S_V - B*Crt = 0
%

    if size(B,2)>1 || size(C,1)>1, error('System must be SISO.'), end
    p0 = [(s0(1)+s0(2))/2, s0(1)*s0(2)];    % convert shifts to parameter
    t = tic; k = 0; precond = eye(2);
    % compute initial model function and cost function at p0
    computeLU(s0);  V = newColV([],3);  W = newColW([],3);
    Am=W'*A*V; Bm=W'*B; Cm=C*V; Em=W'*E*V;
    J_old = CostFunction(p0);
    options=optimset('TolFun',1e-16,'TolX',1e-16, ...
        'Display','none', 'Algorithm','trust-region-reflective', ...
        'GradObj','on','Hessian','on','MaxFunEvals',100,'MaxIter',100);

    while(1)
        k = k + 1;
        disp(['Iteration ' num2str(k) ': q = ' num2str(size(V,2))])
%         precond = diag(p0);
        [~,~,H] = CostFunction(p0); precond = diag(1./abs(diag(H).^0.25));
        % run trust region algorithm to find minimum of model function
        p_opt = fmincon(@CostFunction,p0/precond,[],[],[],[],[0;0],[inf;inf],[],options);
        p_opt = p_opt*precond; precond = eye(2);
        % convert parameter to shifts and perform LU decompositions
        s_opt=p_opt(1)+[1,-1]*sqrt(p_opt(1)^2-p_opt(2)); computeLU(s_opt);
        % update model function by two-sided (Hermite) projection
        V = newColV(V, 2);  W = newColW(W, 2);  Am=W'*A*V; Bm=W'*B; Cm=C*V; Em=W'*E*V;
        % evaluate cost functional at new parameter point
        J = CostFunction(p_opt);

        disp(['  relative change:      ' num2str(norm((p0-p_opt)./p0), '%1.2e')]);
        disp(['  relative improvement: ' num2str((J-J_old)/J, '%1.2e')]);
        disp(['  absolute J = ' num2str(J, '%1.12e')]);

        % decide how to proceed
        if abs((J-J_old)/J) < 1e-10 || norm((p0-p_opt)./p0) < 1e-10 || size(Am,1)>=20
            break;                      % convergence in J or in p  => stop
        elseif J<J_old
             J_old = J;  p0 = p_opt;    % improvement: continue with p_opt
        end
    end
    % supply output variables
    v1  = Q1*(U1\(L1\(P1*B))); v12= Q2*(U2\(L2\(P2*B))); v2 = Q2*(U2\(L2\(P2*(E*v1))));
    V   = full(real([v1/2 + (v12/2+p_opt(1)*v2), v2*sqrt(p_opt(2))]));
    S_V = [2*p_opt(1), sqrt(p_opt(2)); -sqrt(p_opt(2)), 0]; Crt = [1 0];
    disp(['MESPARK required ca. ' num2str(2*(k+1)) ' LUs ', ...
        ' and converged in ' num2str(toc(t),'%.1f') 'sec.'])
```

```
51
52      function [J, g, H] = CostFunction(p)
53          % H2 cost functional, gradient and Hessian
54          p = p*precond;  a = p(1); b = p(2);  s1 = a+sqrt(a^2-b); s2 = a-sqrt(a^2-b);
55          r1 = (Am-s1*Em)\Bm;  r2 = (Am-s1*Em)\(Em*r1); r3 = (Am-s1*Em)\(Em*r2);
56          l1 = Cm/(Am-s2*Em);  l2 = l1*Em/(Am-s2*Em);   l3 = l2*Em/(Am-s2*Em);
57          [J, g, H] = CostFunctionH2(Am, Bm, Cm, Em, p, [r1,r2,r3], [l1;l2;l3]);
58          g = g * precond;  H = precond * H * precond;
59      end
60      function computeLU(s0)
61          % compute new LU decompositions
62          if real(s0(1))==real(s0(2))  % complex conjugated or double shift
63              [L1,U1,P1,Q1] = lu(sparse(A-s0(1)*E));  L2=conj(L1);U2=conj(U1);P2=P1;Q2=Q1;
64          else                         % two real shifts
65              [L1,U1,P1,Q1] = lu(sparse(A-s0(1)*E));  [L2,U2,P2,Q2] = lu(sparse(A-s0(2)*E));
66          end
67      end
68      function V = newColV(V, k)
69          % add columns to input Krylov subspace
70          for i=(size(V,2)+1):2:(size(V,2)+2*k)
71              if i==1, x=B; else x=E*V(:,i-1); end
72              r1  = Q1*(U1\(L1\(P1*x)));   tmp = Q2*(U2\(L2\(P2*x)));
73              v1 = real(0.5*r1 + 0.5*tmp); v2  = real(Q2*(U2\(L2\(P2*(E*r1)))));
74              V = GramSchmidt([V,v1,v2],[],[],[i,i+1]);
75          end
76      end
77      function W = newColW(W, k)
78          % add columns to output Krylov subspace
79          for i=(size(W,2)+1):2:(size(W,2)+2*k)
80              if i==1, x=C; else x=W(:,i-1)'*E; end
81              l1  = x*Q1/U1/L1*P1;          tmp = x*Q2/U2/L2*P2;
82              w1 = real(0.5*l1 + 0.5*tmp);  w2  = real(l1*E*Q2/U2/L2*P2);
83              W = GramSchmidt([W,w1',w2'],[],[],[i,i+1]);
84          end
85      end
86  end
```

In fact, $\mathbf{T} = \begin{bmatrix} 1 & 0 \\ a & \sqrt{b} \end{bmatrix}$ yields $\hat{\mathbf{A}}_r = \mathbf{T}^{-1}\mathbf{A}_r\mathbf{T} = \begin{bmatrix} -3a & \sqrt{b} \\ -\sqrt{b} & 0 \end{bmatrix}$ and $\hat{\mathbf{B}}_r = \mathbf{T}^{-1}\mathbf{B}_r = \begin{bmatrix} -4a \\ 0 \end{bmatrix}$, which seems to be a more suitable realization from a numerical point of view; the condition number of $\hat{\mathbf{A}}$ is shown in Figure 4.9b).[3] The corresponding matrices $\hat{\mathbf{V}}$ and $\hat{\mathbf{S}}_V$ read

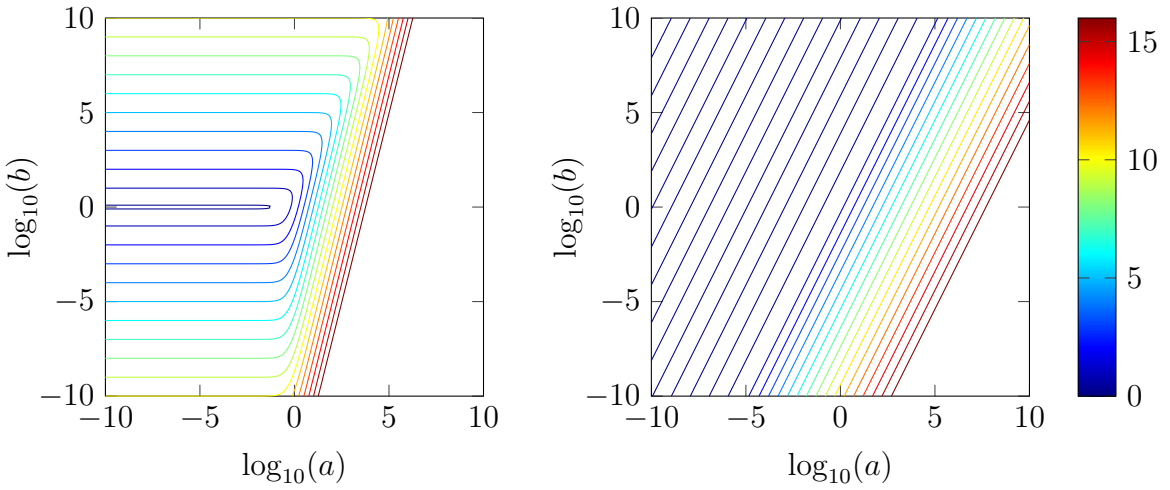
$$\hat{\mathbf{V}} = \mathbf{V}\mathbf{T} = \left[\tfrac{1}{2}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} + \tfrac{1}{2}\mathbf{A}_{\sigma_2}^{-1}\mathbf{b} + a \cdot \mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}, \quad \sqrt{b} \cdot \mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}\right] \tag{4.26}$$

$$= \left[\mathbf{A}_{\sigma_2}^{-1}\mathbf{A}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}, \quad \sqrt{b} \cdot \mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}\right] \tag{4.27}$$

and $\hat{\mathbf{S}}_V = \begin{bmatrix} 2a & \sqrt{b} \\ -\sqrt{b} & 0 \end{bmatrix}$. $\hat{\tilde{\mathbf{c}}}_r = [1,\ 0]$ remains unchanged.

[3] For $\frac{b}{a} \to 0$, one of the two eigenvalues of the ROM tends to zero, so bad condition in the lower right part of the plot is unavoidable.

Note that these formulas have been incorporated in Source 4.4 (cf. lines 46–48) due to their superiority, but not exploited in the derivation of the cost functional and its derivatives (cf. Source 4.2), yet.



a) Standard Formulation b) Transformed Formulation

Figure 4.9.: Condition Number of $\mathbf{A}_r$ in ESPARK ($\log_{10}(\cdot)$)

4.5. Generalization of MESPARK to MIMO Systems

The ESPARK and MESPARK algorithms so far only work for SISO systems, because the very particular parametrization of the underlying optimization problem exploits the fact that here an $\mathcal{H}_2$ pseudo-optimal approximant of order $n = 2$ is uniquely determined through two real positive numbers a and b. For MIMO systems, however, the situation is more complicated. Although no final answer to the problem of generalizing the algorithms to multivariable systems can be given so far, some ideas will be sketched in the following.

Thinking of tangential interpolation, two problems arise in the context of ESPARK for MIMO systems. Firstly, to choose (possibly complex!) tangential directions together with the shifts, one must increase the number k of optimization variables—and k will then depend on the number of inputs and outputs of the system. And secondly, the computation of gradient and Hessian matrix above was derived from a very particular formulation of the ROM in which its $\mathcal{H}_2$ norm could be suitably expressed to obtain manageable terms.

For that reason, the only solution seems to be through a direct use of the available SISO technique by defining real tangential vectors $\mathbf{t}_B$ and $\mathbf{t}_C$ *a priori* which transform the MIMO system temporarily into a SISO model. The discovered optimal shifts σ_1, σ_2 are then used together with one of the tangential vectors $\mathbf{t}_B$ or $\mathbf{t}_C$ to construct the solution of a SYLVESTER equation and run the PORK algorithm.

On the plus side, this guarantees a monotonic decay of the $\mathcal{H}_2$ error. However, performance is strongly compromised if the tangential vectors are not chosen suitably. In fact, one can show that tangential interpolation including complex $\mathbf{t}_B$ and $\mathbf{t}_C$ can not be constituted in this way at all. So even if one knew shifts and tangential vectors belonging to $\mathcal{H}_2$ optimal reduction, this solution could not necessarily be reproduced by the described procedure.

However, first tests suggest that at least for the SIMO and MISO case, acceptable results may be obtained. The automatic choice of the respective tangential vector remains an open problem, but an *ad hoc* approach could be to choose unity vectors and simply alternate between the m inputs or p outputs, respectively.

A more sophisticated idea is to hand over *two* tangential vectors each, $\mathbf{t}_{B,1}, \mathbf{t}_{B,2}$ and $\mathbf{t}_{C,1}, \mathbf{t}_{C,2}$ such that ESPARK can recombine them to two-dimensional real or complex subspaces. The number of optimization variables would then amount to $k = 4$, independently of m and p. The choice of the tangential vectors could again be carried out with a model function to accelerate convergence.

To conclude: MESPARK can be applied to MIMO systems only in a rudimentary way so far by choosing tangential vectors manually. Automatic selection schemes are a current topic of research.

5. Rigorous Error Estimation in Krylov Subspace Methods

"Truth will sooner come out from error than from confusion."

— Francis Bacon

One major open issue in linear MOR is reliable error estimation in scenarios where the true error cannot be computed any more due to excessive size of the HFM. The problem of error estimation is naturally associated with Krylov subspace methods, as in balanced truncation techniques the (full-rank) Gramians (computed by direct methods) are available anyway, so here the computation of the $\mathcal{H}_2$ error is quite easily possible; an $\mathcal{H}_\infty$ error bound is even available *a priori*.

In Padé type approximation, on the other hand, all one knows about the HFM is *local* information in the form of moments, which does not readily imply statements on the *global* approximation quality.

Over the years, several error estimators have been presented in the literature, none of which, however, seems to constitute a convincing rigorous and global bound. Therefore, to the best of the author's knowledge, the work presented in [124] is the first generic approach that delivers global and rigorous $\mathcal{H}_2$ and $\mathcal{H}_\infty$ error bounds for purely Krylov-based reduction without further costly information on the HFM (like sampled frequency response values etc.). The only assumptions required are the strict definiteness properties

$$\mathbf{E} = \mathbf{E}^T > \mathbf{0} \qquad \text{and} \qquad \mathbf{A} + \mathbf{A}^T < \mathbf{0}.$$

Before the new approach from [124] is presented, we briefly review existing methods from the literature.

5.1. State of the Art

Grimme suggested two error estimation procedures in his thesis [73]. The first is the comparison of two "completely different", "complementary" ROMs of the same HFM. The basic assumption is that they are both close to the HFM if they are close to each other. This doubles the reduction effort, and although it is a good indicator, no rigorous error information can be deduced.

Grimme's second approach is through residual expressions

$$\mathbf{r}_b(s) := \mathbf{b} - (s\mathbf{E} - \mathbf{A})\mathbf{V}(s\mathbf{E}_r - \mathbf{A}_r)^{-1}\mathbf{b}_r \qquad \text{and} \tag{5.1}$$

$$\mathbf{r}_c(s) := \mathbf{c} - \mathbf{c}_r(s\mathbf{E}_r - \mathbf{A}_r)^{-1}\mathbf{W}^T(s\mathbf{E} - \mathbf{A}), \tag{5.2}$$

which can be easily shown to fulfill

$$\mathbf{G}_e(s) = \mathbf{G}(s) - \mathbf{G}_r(s) = \mathbf{r}_c(s)(\mathbf{A} - s\mathbf{E})^{-1}\mathbf{r}_b(s). \tag{5.3}$$

Even if the computation of $\mathbf{G}_e(s)$ is still expensive due to the large-scale inverse (or the related high-dimensional LSE), the residuals themselves can be calculated very easily.

But although "sufficiently small $\mathbf{r}_b$ and $\mathbf{r}_c$ at some s_0 implies a small error at that frequency by itself, [...] monitoring $\mathbf{r}_b$ and/or $\mathbf{r}_c$ does not directly lead to an estimate for the modeling error." [73] This is because controllability and observability effects inherent in the structure of $\mathbf{A}$ and $\mathbf{E}$ are disregarded. In the presence of weakly damped poles along the imaginary axis, for instance, small residuals may still lead to very large error values. The residuals alone may therefore indicate convergence of the ROM, but cannot be used as rigorous bounds either. Please note, at this point, that many of the adaptive shift selection strategies that were mentioned in Section 4.1 try to minimize residual expressions, which is the reason for their heuristic nature.

Meanwhile, however, more sophisticated approaches have been described in the literature which are also based on the above residuals. The basic idea is to use the Cauchy-Schwarz inequality

$$\mathbf{G}_e(s) = \mathbf{r}_c(s)(\mathbf{A} - s\mathbf{E})^{-1}\mathbf{r}_b(s) \quad \Rightarrow \quad \|\mathbf{G}_e(s)\| \leq \|\mathbf{r}_c(s)\| \cdot \|(\mathbf{A} - s\mathbf{E})^{-1}\| \cdot \|\mathbf{r}_b(s)\| \tag{5.4}$$

and to find an upper bound on $\|(\mathbf{A} - s\mathbf{E})^{-1}\|$—or, equivalently, a lower bound on the smallest singular value $\sigma_{min}\big(\mathbf{A} - s\mathbf{E}\big)$—to estimate the error at frequency s. The main

problem is to control the overestimation introduced by the CAUCHY-SCHWARZ inequality on the one side, and by the estimate of the norm of $\|(\mathbf{A} - s\mathbf{E})^{-1}\|$, on the other.

BAI ET AL. first presented such local error bounds in the context of the LANCZOS algorithm [15, 16] and used the inequality

$$\left\|(\mathbf{I} - s\mathbf{A})^{-1}\right\| \leq \frac{1}{1 - |s| \cdot \|\mathbf{A}\|}, \tag{5.5}$$

which holds for $|s| < \|\mathbf{A}\|^{-1}$, i. e. in a quite narrow range for systems with high frequent dynamics.

ODABASIOGLU ET AL. therefore suggested approximate error measures in [121], which were valid in the whole spectrum, but not rigorous anymore. More recent results are due to FENG and BENNER, who considered the symmetric case in [27]. AMSALLEM and HETMANIUK proposed a tight error estimator in [6] to avoid overestimation, which is, however, not rigorous.

One should generally note that the local nature of these bounds limits their use to the estimation of accuracy within a certain frequency range and requires dense sampling. Global error bounds, on the other hand, need to be evaluated only once and offer valuable possibilities like, for instance, time domain envelopes of the output signal (cf. Section 5.3.4).

KONKEL, FARLE, and DICZIJ-EDLINGER presented a provable error bound for lossless systems whose eigenvalues are known in a frequency range of interest [94, 95]. The eigenvalues of the HFM which lie within this interval are included in the ROM; the influence of the remaining eigenvalues is then upper bounded under certain assumptions and exploiting orthogonality relations. The idea has also been translated into second order systems by FEHR in [53, 54].

Please note that for (almost) lossless systems the use of a frequency-limited error measure is indeed very sensible, because the eigenvalues of the HFM which are neglected in the ROM still lead to high peaks in the amplitude response of the error model (they lie close to or even on the imaginary axis) and therefore to large or infinite global error norms. For that reason, $\mathcal{H}_2$ and $\mathcal{H}_\infty$ error bounds cannot be expected to yield helpful results for such models.

An early error estimation technique similar to GRIMME's first suggestion was due to BECHTOLD, RUDNYI, and KORVINK [24, 25]. They observed that during an ARNOLDI process—i. e. a rational KRYLOV method about a single expansion point whose multiplicity is iteratively increased—the difference between two consecutive ("neighboring") ROMs $\mathbf{G}_{r,k}(s)$ and $\mathbf{G}_{r,k+1}(s)$ reflects the actual error:

$$\left|\frac{\mathbf{G}(s)-\mathbf{G}_{r,k}(s)}{\mathbf{G}(s)}\right| \approx \left|\frac{\mathbf{G}_{r,k}(s)-\mathbf{G}_{r,k+1}(s)}{\mathbf{G}_{r,k}(s)}\right|. \tag{5.6}$$

Also, they suggested to monitor the HANKEL Singular Values (HSV) of the ROM and stop the iteration as soon as no more significant changes occur in the largest HSVs, as the dominant parts of the transfer behavior are then likely to be captured by the ROM.

Finally, they came up with the idea of sequential model reduction, which means a two-step procedure. The very high-dimensional HFM is first reduced to an intermediate model $\mathbf{G}_i(s)$ which is large enough to capture all relevant dynamics of the HFM, but small enough to be reduced efficiently in a second step including exact computation of the error. The error of the second step is then assumed to be the error between the final ROM and the HFM, so the intermediate ROM is supposed to be extremely close to the HFM.

All these ideas constitute only heuristic estimates, no rigorous bounds.

SORENSEN, TENG, and ANTOULAS presented an $\mathcal{H}_2$ error bound for model reduction of second order systems (cf. Chapter 6), which, however, requires knowledge of the dominant eigenspace of the controllability Gramian, whose computation is expensive [148].

5.2. Exploiting the Factorization of the Error System

In the following, we will see that the factorization of the error model which holds in SYLVESTER-based model reduction is an important step towards global error bounds. Consider the following lemma.

Lemma 5.1. *Let $\mathbf{G}(s)$ be an LTI system which is reduced by* SYLVESTER*-based model reduction, so that eventually it takes the general form (4.13)[1]:*

$$\mathbf{G}(s) = \mathbf{G}_r(s) \; + \; \widetilde{\mathbf{G}}_r^L(s) \cdot \mathbf{G}_\perp(s) \cdot \widetilde{\mathbf{G}}_r^R(s). \tag{5.7}$$

Then, the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norm of the respective error model $\mathbf{G}_e(s)$ are upper bounded by

$$\begin{aligned} \left\|\mathbf{G}_e\right\|_{\mathcal{H}_2} &\leq \left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_2} \cdot \left\|\widetilde{\mathbf{G}}_r^L\right\|_{\mathcal{H}_\infty} \cdot \left\|\widetilde{\mathbf{G}}_r^R\right\|_{\mathcal{H}_\infty} \qquad \textit{and} \\ \left\|\mathbf{G}_e\right\|_{\mathcal{H}_\infty} &\leq \left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_\infty} \cdot \left\|\widetilde{\mathbf{G}}_r^L\right\|_{\mathcal{H}_\infty} \cdot \left\|\widetilde{\mathbf{G}}_r^R\right\|_{\mathcal{H}_\infty}. \end{aligned} \tag{5.8}$$

In the $\mathcal{H}_2$ pseudo-optimal case, when $\widetilde{\mathbf{G}}_r^R(s)$ and $\widetilde{\mathbf{G}}_r^L(s)$ are unity all-pass systems, equality holds:

$$\left\|\mathbf{G}_e\right\|_{\mathcal{H}_2} = \left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_2} \qquad \textit{and} \qquad \left\|\mathbf{G}_e\right\|_{\mathcal{H}_\infty} = \left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_\infty}. \tag{5.9}$$

Proof. This lemma was partly presented in [124] for purely $\mathbf{V}$-based decomposition (where $\widetilde{\mathbf{G}}_r^L(s) \equiv \mathbf{I}$), but the proof carries over to the general case.

To start with, the equalities for unity all-pass systems $\widetilde{\mathbf{G}}_r^L(s)$ and $\widetilde{\mathbf{G}}_r^R(s)$ follow directly with Theorem 4.1, because both $\widetilde{\mathbf{G}}_r^R(s)$ and $\widetilde{\mathbf{G}}_r^R(s)$ are unitary matrices along the imaginary axis in this case. The $\mathcal{H}_\infty$ inequality is a direct consequence of the submultiplicativity of the $\mathcal{H}_\infty$ norm [46]. From definition (2.11) of the $\mathcal{H}_2$ norm it follows generally for some product $\mathbf{G}_e(s) = \mathbf{G}_\perp(s) \cdot \widetilde{\mathbf{G}}_r(s)$:

$$\begin{aligned} \left\|\mathbf{G}_e\right\|_{\mathcal{H}_2}^2 &= \tfrac{1}{2\pi} \int_{-\infty}^{\infty} \left\|\mathbf{G}_e(i\omega)\right\|_F^2 d\omega = \tfrac{1}{2\pi} \int_{-\infty}^{\infty} \left\|\widetilde{\mathbf{G}}_r^L(i\omega) \cdot \mathbf{G}_\perp(i\omega) \cdot \widetilde{\mathbf{G}}_r^R(i\omega)\right\|_F^2 d\omega \\ &\leq \tfrac{1}{2\pi} \int_{-\infty}^{\infty} \left\|\widetilde{\mathbf{G}}_r^L(i\omega)\right\|_2^2 \cdot \left\|\mathbf{G}_\perp(i\omega)\right\|_F^2 \cdot \left\|\widetilde{\mathbf{G}}_r^R(i\omega)\right\|_2^2 d\omega \qquad [108,\ 8.5.2(6)] \\ &\leq \tfrac{1}{2\pi} \int_{-\infty}^{\infty} \sup_\omega \left\|\widetilde{\mathbf{G}}_r^L(i\omega)\right\|_2^2 \cdot \left\|\mathbf{G}_\perp(i\omega)\right\|_F^2 \cdot \sup_\omega \left\|\widetilde{\mathbf{G}}_r^R(i\omega)\right\|_2^2 d\omega \\ &= \tfrac{1}{2\pi} \int_{-\infty}^{\infty} \operatorname{tr}\left[\mathbf{G}_\perp^H(i\omega)\mathbf{G}_\perp(i\omega)\right] d\omega \cdot \left\|\widetilde{\mathbf{G}}_r^L\right\|_{\mathcal{H}_\infty}^2 \cdot \left\|\widetilde{\mathbf{G}}_r^R\right\|_{\mathcal{H}_\infty}^2 \\ &= \left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_2}^2 \cdot \left\|\widetilde{\mathbf{G}}_r^L\right\|_{\mathcal{H}_\infty}^2 \cdot \left\|\widetilde{\mathbf{G}}_r^R\right\|_{\mathcal{H}_\infty}^2. \end{aligned} \tag{5.10}$$

□

[1] For better readability, we omit the indices $^\Sigma$ and $_k$ in the following; the use of the error bounds is not restricted to the iterative CURE scheme, anyway.

Please note that after exclusively **V**- or exclusively **W**-based reduction, $\widetilde{\mathbf{G}}_r^L(s)$ or $\widetilde{\mathbf{G}}_r^R(s)$, respectively, are identity proportional systems, so in (5.8) the corresponding factor $\left\|\widetilde{\mathbf{G}}_r^L\right\|_{\mathcal{H}_\infty}$ or $\left\|\widetilde{\mathbf{G}}_r^R\right\|_{\mathcal{H}_\infty}$ amounts to one and can be omitted.

The above lemma is essential for the actual error bounds presented in the next section, because instead of looking at the error model $\mathbf{G}_e(s)$, we only need to consider the system $\mathbf{G}_\perp(s)$ which is much more familiar due to the fact that it inherits the matrices **E** and **A** from the HFM. In particular, a strictly dissipative realization of the HFM carries over to $\mathbf{G}_\perp(s)$. What is more: If, for instance, only **V**-based factorization of the error model is performed (possibly several times), $\mathbf{G}_\perp(s)$ also contains the original output matrix **C** and therefore even shares its observability Gramian with the HFM.

It is true that a certain overestimation of the error can be introduced by the factorization and Lemma 5.1. This is, for instance, the case, when the amplitude response of $\widetilde{\mathbf{G}}_r^L(s)$ or $\widetilde{\mathbf{G}}_r^R(s)$ exhibits distinct peaks. The highest of them defines the $\mathcal{H}_\infty$ norm of the system and enters the integral in (5.10) as a worst-case estimate for the whole frequency range. However, the lemma also states that in $\mathcal{H}_2$ pseudo-optimal reduction, no overestimation occurs at all.

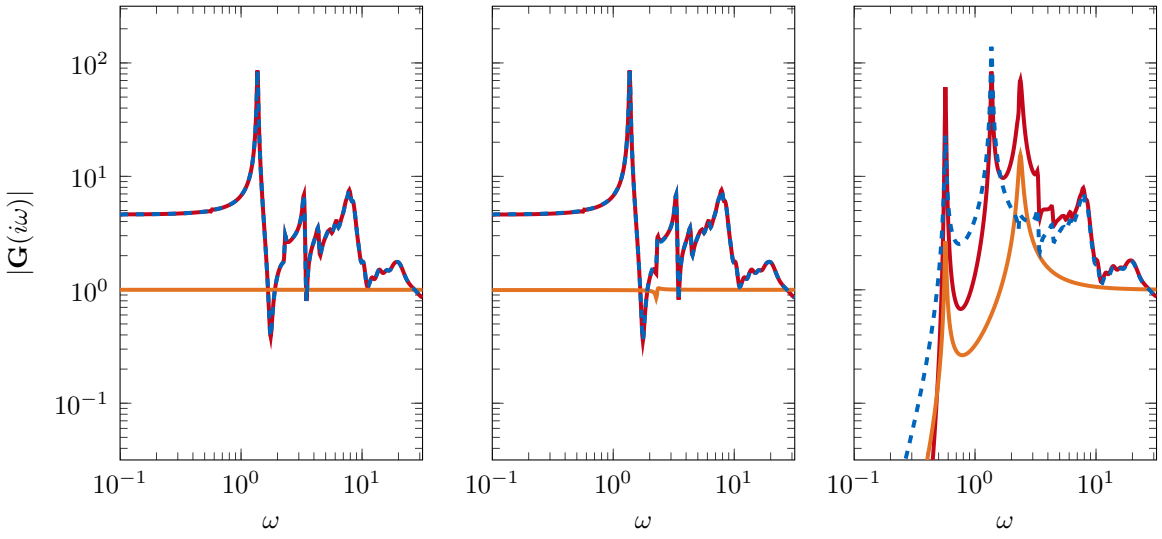
In fact, three basic situations have been observed in practice and will be presented with the help of a numerical example. Figure 5.1 shows typical results for **V**-based factorizations after three different reductions of the Clamped Beam benchmark model to order $n = 6$. We can see the amplitude responses of the true error $\mathbf{G}_e(s)$ in red, of $\mathbf{G}_\perp(s)$ in dashed blue and of $\widetilde{\mathbf{G}}_r^R(s)$ in orange; $\widetilde{\mathbf{G}}_r^L(s)$ is ignored because $|\widetilde{\mathbf{G}}_r^L(s)| \equiv 1$ (**V**-based decomposition).

The first ROM (Figure 5.1a)) is the result of IRKA. $\widetilde{\mathbf{G}}_r^R(s)$ is all-pass and $\mathbf{G}_\perp(s)$ exactly describes the amplitude of $\mathbf{G}_e(s)$, so the bounds from Lemma 5.1 deliver the exact error norm.

In the second case (Figure 5.1b)), IRKA was stopped prematurely. Although $\widetilde{\mathbf{G}}_r^R(s)$ is not an all-pass system, the induced overestimation is acceptable—it amounts to about 4 % in both $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norm—because the amplitude response of $\widetilde{\mathbf{G}}_r^R(s)$ only exhibits a minor jitter.

Figure 5.1c), however, shows the result for Padé approximation about $\sigma = 0$. Here, the actual $\mathcal{H}_\infty$ norm of $\mathbf{G}_e(s)$ amounts to about 88 (39dB), whereas the product of the

$\mathcal{H}_\infty$ norms of $\mathbf{G}_\perp(s)$ and $\widetilde{\mathbf{G}}_r^R(s)$ delivers 2200 (69dB), so one has an overestimation of about 25 due to the peaks of $\widetilde{\mathbf{G}}_r^R(s)$ that do not correlate with the peaks of $\mathbf{G}_\perp(s)$; the $\mathcal{H}_2$ overestimation amounts to about 12. For that reason, it seems indeed advisable to perform $\mathcal{H}_2$ pseudo-optimal reduction in order to avoid overestimation by the error decomposition.



a) No Overestimation b) Slight Overestimation c) Large Overestimation

Figure 5.1.: Various Overestimation of $\mathcal{H}_\infty$ Error Norm due to Factorization of Error Model

Although Lemma 5.1 provides a first advance towards rigorous error estimation, the problem remains that for high-order systems we cannot compute the norms $\|\mathbf{G}_\perp\|_{\mathcal{H}_2}$ and $\|\mathbf{G}_\perp\|_{\mathcal{H}_\infty}$. For the special case of strictly dissipative systems, however, global upper bounds on the respective norms have recently been derived in [124] and will be discussed in the following sections.

Before that, we remark that the local bound (5.4) is given by

$$\|\mathbf{G}_e(s)\| \leq \|\widetilde{\mathbf{G}}^L(s)\mathbf{C}_\perp\| \cdot \|(\mathbf{A} - s\mathbf{E})^{-1}\| \cdot \|\mathbf{B}_\perp\widetilde{\mathbf{G}}^R(s)\|, \tag{5.11}$$

if the error model is factorized as in (5.7). In $\mathcal{H}_2$ pseudo-optimal reduction, this simplifies to

$$\|\mathbf{G}_e(s)\| \leq \|\mathbf{C}_\perp\| \cdot \|(\mathbf{A} - s\mathbf{E})^{-1}\| \cdot \|\mathbf{B}_\perp\|, \tag{5.12}$$

so the time-dependent residuals $\mathbf{r}_b(t)$ and $\mathbf{r}_c(t)$ do not have to be evaluated at all.

5.3. Global $\mathcal{H}_2$ Error Bound for Systems in Strictly Dissipative Realization

5.3.1. Upper Bound on $\mathcal{H}_2$ Norm of $\mathbf{G}_\perp$

Theorem 5.1 (cf. [124]). *Let $\mathbf{G}(s)$ be given in strictly dissipative realization: $\mathbf{E} = \mathbf{L}^T\mathbf{L}$ is positive definite and $\mu_{\mathbf{E}}(\mathbf{A}) < 0$ holds. Then an upper bound on the $\mathcal{H}_2$ norm of $\mathbf{G}_\perp(s) = (\mathbf{A}, \mathbf{B}_\perp, \mathbf{C}_\perp, \mathbf{0}, \mathbf{E})$ can be found in the following way.*

Let $\hat{\mathbf{P}} \in \mathbb{R}^{N\times N}$ be a positive semidefinite approximation of the controllability Gramian of $\mathbf{G}_\perp(s)$ and define the residual

$$\mathbf{R}_C := \mathbf{A}\hat{\mathbf{P}}\mathbf{E}^T + \mathbf{E}\hat{\mathbf{P}}\mathbf{A}^T + \mathbf{B}_\perp\mathbf{B}_\perp^T. \tag{5.13}$$

Then an upper bound on the $\mathcal{H}_2$ norm of $\mathbf{G}_\perp(s)$ is given by

$$\left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_2} \leq \sqrt{\operatorname{tr}\left[\mathbf{C}_\perp\hat{\mathbf{P}}\mathbf{C}_\perp^T\right] + \frac{1}{-2\mu_{\mathbf{E}}(\mathbf{A})} \cdot \left\|\mathbf{L}^{-T}\mathbf{R}_C\mathbf{L}^{-1}\right\|_2 \cdot \left\|\mathbf{L}^{-T}\mathbf{C}_\perp^T\right\|_{\mathrm{F}}^2}. \tag{5.14}$$

The dual form also holds true: For some positive semidefinite matrix $\hat{\mathbf{Q}}$, define

$$\mathbf{R}_O := \mathbf{A}^T\hat{\mathbf{Q}}\mathbf{E} + \mathbf{E}^T\hat{\mathbf{Q}}\mathbf{A} + \mathbf{C}_\perp^T\mathbf{C}_\perp.$$

Then an upper bound on the $\mathcal{H}_2$ norm of $\mathbf{G}_\perp(s)$ is given by

$$\left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_2} \leq \sqrt{\operatorname{tr}\left[\mathbf{B}_\perp^T\hat{\mathbf{Q}}\mathbf{B}_\perp\right] + \frac{1}{-2\mu_{\mathbf{E}}(\mathbf{A})} \cdot \left\|\mathbf{L}^{-T}\mathbf{R}_O\mathbf{L}^{-1}\right\|_2 \cdot \left\|\mathbf{L}^{-T}\mathbf{B}_\perp\right\|_{\mathrm{F}}^2}. \tag{5.15}$$

Proof. We start from the formulation of the $\mathcal{H}_2$ norm given in (2.12), which reads

$$\left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_2}^2 = \operatorname{tr}\left[\mathbf{C}_\perp\mathbf{P}\mathbf{C}_\perp^T\right], \tag{5.16}$$

where $\mathbf{P}$ is the controllability Gramian solving

$$\mathbf{A}\hat{\mathbf{P}}\mathbf{E}^T + \mathbf{E}\hat{\mathbf{P}}\mathbf{A}^T + \mathbf{B}_\perp\mathbf{B}_\perp^T = \mathbf{0}. \tag{5.17}$$

Given some arbitrary, positive semidefinite approximation $\hat{\mathbf{P}}$ of $\mathbf{P}$, we split expression (5.16) into two summands

$$\left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_2}^2 = \operatorname{tr}\left[\mathbf{C}_\perp\hat{\mathbf{P}}\mathbf{C}_\perp^T\right] + \operatorname{tr}\left[\mathbf{C}_\perp\left(\mathbf{P} - \hat{\mathbf{P}}\right)\mathbf{C}_\perp^T\right]. \tag{5.18}$$

Let $\mathbf{L}$ be a CHOLESKY factor of $\mathbf{E}$ such that $\mathbf{L}^T\mathbf{L} = \mathbf{E}$. Then, the second summand fulfills

$$\begin{aligned}
\operatorname{tr}\left[\mathbf{C}_\perp(\mathbf{P}-\widehat{\mathbf{P}})\mathbf{C}_\perp^T\right] &= \operatorname{tr}\left[\mathbf{C}_\perp\mathbf{L}^{-1}\mathbf{L}(\mathbf{P}-\widehat{\mathbf{P}})\mathbf{L}^T\mathbf{L}^{-T}\mathbf{C}_\perp^T\right] \\
&\leq \left|\operatorname{tr}\left[\mathbf{L}(\mathbf{P}-\widehat{\mathbf{P}})\mathbf{L}^T\cdot\mathbf{L}^{-T}\mathbf{C}_\perp^T\mathbf{C}_\perp\mathbf{L}^{-1}\right]\right| \\
&\leq \sum_{i=1}^{N}\sigma_i\left[\mathbf{L}(\mathbf{P}-\widehat{\mathbf{P}})\mathbf{L}^T\right]\cdot\sigma_i\left[\mathbf{L}^{-T}\mathbf{C}_\perp^T\mathbf{C}_\perp\mathbf{L}^{-1}\right] \quad \text{(cf. [112],[113],[72])} \\
&\leq \sum_{i=1}^{N}\max_j\sigma_j\left[\mathbf{L}(\mathbf{P}-\widehat{\mathbf{P}})\mathbf{L}^T\right]\cdot\sigma_i^2\left[\mathbf{L}^{-T}\mathbf{C}_\perp^T\right] \\
&= \left\|\mathbf{L}(\mathbf{P}-\widehat{\mathbf{P}})\mathbf{L}^T\right\|_2\cdot\left\|\mathbf{L}^{-T}\mathbf{C}_\perp^T\right\|_{\mathrm{F}}^2.
\end{aligned}$$

The remaining part of the proof is an extension of a result from [81] that allows to upper bound the factor $\left\|\mathbf{L}(\mathbf{P}-\widehat{\mathbf{P}})\mathbf{L}^T\right\|_2$—which contains the unknown Gramian $\mathbf{P}$—with some term that depends on the *residual* corresponding to $\widehat{\mathbf{P}}$. To this end, we subtract (5.13) from (5.17):

$$\mathbf{A}\left(\mathbf{P}-\widehat{\mathbf{P}}\right)\mathbf{E}^T+\mathbf{E}\left(\mathbf{P}-\widehat{\mathbf{P}}\right)\mathbf{A}^T+\mathbf{R}_C=\mathbf{0}. \tag{5.19}$$

Now multiply (5.19) by $\mathbf{L}^{-T}$ from the left and by $\mathbf{L}^{-1}$ from the right,

$$\begin{aligned}
&\mathbf{L}^{-T}\mathbf{A}\left(\mathbf{P}-\widehat{\mathbf{P}}\right)\mathbf{E}^T\mathbf{L}^{-1}+\mathbf{L}^{-T}\mathbf{E}\left(\mathbf{P}-\widehat{\mathbf{P}}\right)\mathbf{A}^T\mathbf{L}^{-1}+\mathbf{L}^{-T}\mathbf{R}_C\mathbf{L}^{-1}=\mathbf{0} \\
\Leftrightarrow\ &\mathbf{L}^{-T}\mathbf{A}\mathbf{L}^{-1}\underbrace{\mathbf{L}\left(\mathbf{P}-\widehat{\mathbf{P}}\right)\mathbf{L}^T}_{\mathbf{X}}+\underbrace{\mathbf{L}\left(\mathbf{P}-\widehat{\mathbf{P}}\right)\mathbf{L}^T}_{\mathbf{X}}\mathbf{L}^{-T}\mathbf{A}^T\mathbf{L}^{-1}+\mathbf{L}^{-T}\mathbf{R}_C\mathbf{L}^{-1}=\mathbf{0}
\end{aligned} \tag{5.20}$$

This is again a LYAPUNOV equation whose solution $\mathbf{X}$ is unique and fulfills

$$\mathbf{L}(\mathbf{P}-\widehat{\mathbf{P}})\mathbf{L}^T=\mathbf{X}=\int_0^\infty e^{\mathbf{L}^{-T}\mathbf{A}\mathbf{L}^{-1}\,t}\,\mathbf{L}^{-T}\mathbf{R}_C\mathbf{L}^{-1}\,e^{\mathbf{L}^{-T}\mathbf{A}^T\mathbf{L}^{-1}\,t}\,dt$$

$$\begin{aligned}
\Longrightarrow\ \|\mathbf{L}(\mathbf{P}-\widehat{\mathbf{P}})\mathbf{L}^T\|_2 &\leq \int_0^\infty\left\|e^{\mathbf{L}^{-T}\mathbf{A}\mathbf{L}^{-1}\,t}\right\|_2^2dt\cdot\left\|\mathbf{L}^{-T}\mathbf{R}_C\mathbf{L}^{-1}\right\|_2 \\
&\leq \int_0^\infty\left[e^{\mu_2\left(\mathbf{L}^{-T}\mathbf{A}\mathbf{L}^{-1}\right)\,t}\right]^2dt\cdot\left\|\mathbf{L}^{-T}\mathbf{R}_C\mathbf{L}^{-1}\right\|_2 \qquad \text{(Theorem 2.1)} \\
&= \int_0^\infty e^{2\mu_{\mathbf{E}}(\mathbf{A})\,t}\,dt\cdot\left\|\mathbf{L}^{-T}\mathbf{R}_C\mathbf{L}^{-1}\right\|_2 \\
&= \frac{1}{-2\,\mu_{\mathbf{E}(\mathbf{A})}}\cdot\left\|\mathbf{L}^{-T}\mathbf{R}_C\mathbf{L}^{-1}\right\|_2 \qquad (\mu<0)
\end{aligned}$$

With this, the overall inequality (5.14) is shown. The dual formulation of the theorem can be proven analogously (cf. [124]). □

Together with Lemma 5.1 this theorem provides a global upper bound on the $\mathcal{H}_2$ norm of the error system resulting from SYLVESTER-based model reduction.

5.3.2. Analysis and Remarks on Implementation

For the following analysis, let us introduce some abbreviations in (5.14):

$$\left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_2}^2 \leq \underbrace{\operatorname{tr}\left[\mathbf{C}_\perp \hat{\mathbf{P}} \mathbf{C}_\perp^T\right]}_{k_1} + \frac{1}{-2\mu_{\mathbf{E}}(\mathbf{A})} \cdot \underbrace{\left\|\mathbf{L}^{-T}\mathbf{R}_C\mathbf{L}^{-1}\right\|_2}_{k_2} \cdot \underbrace{\left\|\mathbf{L}^{-T}\mathbf{C}_\perp^T\right\|_{\mathrm{F}}^2}_{k_3}.$$

The clue behind the $\mathcal{H}_2$ bound is to split the system norm into two parts: the first—i. e., k_1—follows from some arbitrary approximation $\hat{\mathbf{P}}$ of the controllability Gramian, and the second accounts for the deviation of $\hat{\mathbf{P}}$ from the real Gramian. The estimation of this latter part, $\operatorname{tr}\left[\mathbf{C}_\perp\left(\mathbf{P}-\hat{\mathbf{P}}\right)\mathbf{C}_\perp^T\right]$, is only valid for systems in strictly dissipative realization and is eventually derived from the worst case estimation (2.25) of the "contraction speed" of the system (cf. Section 2.2). It includes several (rigorous) estimation steps, so it is clearly the major source for possible overestimation. In fact, if $\hat{\mathbf{P}}$ exactly solves the Lyapunov equation, then the residual $\mathbf{R}_C$ accounts to zero, $k_2 = 0$ holds and the bound delivers the true norm. If, as the other extreme, one sets $\hat{\mathbf{P}} = \mathbf{0}$, then $k_1 = 0$ and $\mathbf{R}_C = \mathbf{B}_\perp\mathbf{B}_\perp^T$ and the whole bound rests upon the dissipativity-based estimation. Indeed we will see in the example below that huge overestimation can appear in this case.

Let us now consider the numerical aspects of the bounds. A MATLAB implementation of both the controllability-based (5.14) and the observability-based (5.15) formulation of the bound can be seen in Source 5.1. Some explanations and comments are in order.

We firstly note that k_1 can be found purely by matrix-vector products. In addition, the approximate Gramian is typically given by

$$\hat{\mathbf{P}} = \mathbf{Z}\hat{\mathbf{P}}_r\mathbf{Z}^H = \mathbf{Z}\mathbf{L}_{\hat{\mathbf{P}}_r}^H\mathbf{L}_{\hat{\mathbf{P}}_r}\mathbf{Z}^H, \quad \text{where} \quad \mathbf{Z} \in \mathbb{C}^{N\times\widetilde{q}}, \quad \hat{\mathbf{P}}_r, \mathbf{L}_{\hat{\mathbf{P}}_r} \in \mathbb{C}^{\widetilde{q}\times\widetilde{q}}, \tag{5.21}$$

so one can write k_1 even simpler

$$k_1 = \operatorname{tr}\left[\left(\mathbf{C}_\perp\mathbf{Z}\right)\hat{\mathbf{P}}_r\left(\mathbf{C}_\perp\mathbf{Z}\right)^H\right] = \left\|\mathbf{C}_\perp\mathbf{Z}\mathbf{L}_{\hat{\mathbf{P}}_r}^H\right\|_{\mathrm{F}}^2. \tag{5.22}$$

The constant k_2 looks hard to compute at first sight. However, thanks to the low-rank structure (5.21) of $\hat{\mathbf{P}}$, we do not need to calculate $\mathbf{R}_C$ explicitly according to (5.13) nor perform expensive operations on it (multiply by $\mathbf{L}^{-1}$ and compute 2-norm). Instead, we note that due to symmetry, we can rewrite

$$\begin{aligned}
k_2 &= \left\|\mathbf{L}^{-T}\mathbf{R}_C\mathbf{L}^{-1}\right\|_2 = \max_i \sigma_i\left(\mathbf{L}^{-T}\mathbf{R}_C\mathbf{L}^{-1}\right) && \text{(definition of 2-norm)}\\
&= \max_i \left|\lambda_i\left(\mathbf{L}^{-T}\mathbf{R}_C\mathbf{L}^{-1}\right)\right| && \text{(symmetry)}\\
&= \max_i \left|\lambda_i\left(\mathbf{R}_C, \mathbf{E}\right)\right| && \text{(generalized eigenvalue problem)}
\end{aligned}$$

So all we need to do is find the "largest magnitude" solution of a symmetric-definite generalized eigenvalue problem formulated by products of sparse matrices. As was pointed out in [124], this can be quite easily achieved by a power method. For more information on large sparse eigenvalue problems, please refer to [136].

The computation of k_3, finally, mainly requires p backward substitutions of high dimension. Should the CHOLESKY factor $\mathbf{L}$ not be available, k_3 can be rewritten as

$$k_3 = \left\|\mathbf{L}^{-T}\mathbf{C}_\perp^T\right\|_{\mathrm{F}}^2 = \operatorname{tr}\left[\mathbf{C}_\perp\mathbf{E}^{-1}\mathbf{C}_\perp^T\right]. \tag{5.23}$$

Note that in the original work [124], the factor corresponding to k_3 read $p \cdot \left\|\mathbf{C}_\perp\mathbf{E}^{-1}\mathbf{C}_\perp^T\right\|_2$ instead, where p is the number of output variables. The new term, however, is tighter and leads to a smaller bound in the presence of multiple outputs.

Please note that if zero is used as approximate Gramian, the bounds read

$$\left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_2}^2 \leq \frac{1}{-2\mu_{\mathbf{E}}(\mathbf{A})} \cdot \left\|\mathbf{L}^{-T}\mathbf{B}_\perp\right\|_2^2 \cdot \left\|\mathbf{L}^{-T}\mathbf{C}_\perp^T\right\|_{\mathrm{F}}^2 \quad \text{and}$$

$$\left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_2}^2 \leq \frac{1}{-2\mu_{\mathbf{E}}(\mathbf{A})} \cdot \left\|\mathbf{L}^{-T}\mathbf{B}_\perp\right\|_{\mathrm{F}}^2 \cdot \left\|\mathbf{L}^{-T}\mathbf{C}_\perp^T\right\|_2^2$$

and are identical in the SISO case.

5.3.3. Relative $\mathcal{H}_2$ Error Bound

The upper bound presented so far provides an estimate of the *absolute* error norm $\|\mathbf{G}_e\|_{\mathcal{H}_2}$. In practice, however, one is often also interested in *relative* error information, i. e.

$$\epsilon_{\mathcal{H}_2,\mathrm{rel}} := \frac{\|\mathbf{G}_e\|_{\mathcal{H}_2}}{\|\mathbf{G}\|_{\mathcal{H}_2}}.$$

To obtain such a relative error bound, it is suggested to take use of (3.27), according to which an $\mathcal{H}_2$-pseudo-optimal ROM provides a *lower* bound on the $\mathcal{H}_2$ norm of the HFM.

Corollary 5.1. *Given an upper bound $\overline{\epsilon_{\mathcal{H}_2}}$ on the absolute $\mathcal{H}_2$ norm of an error model $\mathbf{G}_e(s)$, and an $\mathcal{H}_2$-pseudo-optimal ROM $\mathbf{G}_r^*(s)$—which is not necessarily associated with $\mathbf{G}_e(s)$—then a relative $\mathcal{H}_2$ bound is given by*

$$\epsilon_{\mathcal{H}_2,rel} \leq \overline{\epsilon_{\mathcal{H}_2,rel}} := \frac{\overline{\epsilon_{\mathcal{H}_2}}}{\|\mathbf{G}_r^*\|_{\mathcal{H}_2}}.$$

Proof. The claim follows immediately with (3.27). □

Note that the benefit of $\mathcal{H}_2$ pseudo-optimal reduction is therefore threefold. First of all, we avoid overestimation caused by the error factorization in Lemma 5.1 due to the

Source 5.1: Evaluation of $\mathcal{H}_2$ Error Bound

```
function bndH2Con = BoundH2Con(A,B,C,E,mu,L_E,P_E,Z,L_Prh)
% Bound on H2 norm of strictly dissipative system - Approx. Controllability Gramian
%   Input:  A,B,C,E:  HFM matrices;
%           mu:       Generalized Spectral Abscissa (must be negative!)
%           L_E,P_E:  Cholesky factors of E
%           Z,L_Prh:  Low-Rank Cholesky factor of approx. Gramian: P ~ Z*L_Prh'*L_Prh*Z'
%   Output: bndH2Con: Upper bound on H2 norm of (A,B,C,0,E)
%

k_1 = norm(C*Z*L_Prh','fro')^2;
R_C = @(x) (A*(Z*((L_Prh'*L_Prh)*(Z'*(E*x))))+(x'*A*Z*(L_Prh'*L_Prh)*Z'*E)'+B*(B'*x));
k_2 = abs(eigs(R_C, size(A,1), E, 1, 'LM', struct('issym', true)));
C_E = (L_E'\(P_E'*C'))';  k_3 = norm(C_E, 'fro')^2;
bndH2Con = sqrt(k_1 + k_2*k_3/(-2*mu));
```

```
function bndH2Obs = BoundH2Obs(A,B,C,E,mu,L_E,P_E,Z,L_Qrh)
% Bound on H2 norm of strictly dissipative system - Approx. Observability Gramian
%   Input:  A,B,C,E:  HFM matrices;
%           mu:       Generalized Spectral Abscissa (must be negative!)
%           L_E,P_E:  Cholesky factors of E
%           Z,L_Qrh:  Low-Rank Cholesky factor of approx. Gramian: Q ~ Z*L_Qrh'*L_Qrh*Z'
%   Output: bndH2Obs: Upper bound on H2 norm of (A,B,C,0,E)
%

k_1 = norm(L_Qrh*Z'*B,'fro')^2;
R_O = @(x) (E*(Z*((L_Qrh'*L_Qrh)*(Z'*(A*x))))+(x'*E*Z*(L_Qrh'*L_Qrh)*Z'*A)'+C'*(C*x));
k_2 = abs(eigs(R_O, size(A,1), E, 1, 'LM', struct('issym', true)));
B_E = L_E'\(P_E'*B); k_3 = norm(B_E, 'fro')^2;
bndH2Obs = sqrt(k_1 + k_2*k_3/(-2*mu));
```

all-pass property of $\widetilde{\mathbf{G}}_r^L(s)$ and $\widetilde{\mathbf{G}}_r^R(s)$. Secondly, we simply have $\|\mathbf{G}_e(s)\|_{\mathcal{H}_2} = \|\mathbf{G}_\perp\|_{\mathcal{H}_2}$. And finally, the resulting ROM provides a relative error bound as soon as an absolute one is known. In fact, the additional overestimation introduced by Corollary 5.1 is very minor even if $\mathbf{G}_r^*(s)$ is only a mediocre approximant of $\mathbf{G}_r(s)$. Assume, for instance, a considerable deviation of 20 %: $\|\mathbf{G} - \mathbf{G}_r^*\|_{\mathcal{H}_2} = 0.2 \cdot \|\mathbf{G}\|_{\mathcal{H}_2}$. Then,

$$\|\mathbf{G}_r^*\|_{\mathcal{H}_2} = \sqrt{\|\mathbf{G}\|_{\mathcal{H}_2}^2 - \|\mathbf{G} - \mathbf{G}_r^*\|_{\mathcal{H}_2}^2} = \sqrt{0.96} \cdot \|\mathbf{G}\|_{\mathcal{H}_2} \approx 0.98 \cdot \|\mathbf{G}\|_{\mathcal{H}_2}\,.$$

Accordingly, the additional overestimation amounts to $\frac{1}{\sqrt{0.96}} - 1 \approx 2\,\%$. For a more "serious" ROM with an error of 1 % or less, the overestimation is below 10^{-4}.

To conclude, in practical cases the additional overestimation introduced by the relative $\mathcal{H}_2$ bound is perfectly negligible in comparison to the conservativeness of the absolute upper bound.

5.3.4. Time Domain Envelopes

Having found an upper bound on the $\mathcal{H}_2$ norm of the error system, we can use this bound for the derivation of envelopes in the time domain, understanding the reduction as an uncertainty.

More precisely, if we use the ROM to simulate the system output $\mathbf{y}_r(t)$ resulting from a given input signal $\mathbf{u}(t)$, we are interested in the maximal deviation between the high fidelity output signal $\mathbf{y}(t)$ and its approximant $\mathbf{y}_r(t)$. As their difference $\mathbf{y}_e(t) = \mathbf{y}(t) - \mathbf{y}_r(t)$ is the output of the error system when $\mathbf{u}(t)$ is applied, we can use the fact that the $\mathcal{H}_2$ norm is related to induced norms which allow us to upper bound some norm on $\mathbf{y}_e(t)$ given some norm of $\mathbf{u}(t)$. In fact, there are several induced norms [8, 39], which may be suitable for a given application. We only consider one of them in the following.

Proposition 5.1 (cf. [10])**.** *Let $\mathbf{u}(t)$ be a finite energy input signal with $\mathbf{u}(t) = \mathbf{0}$ for $t < 0$ and let $\mathbf{y}(t)$, $\mathbf{y}_r(t)$, and $\mathbf{y}_e(t)$ be the corresponding output signals of the original, reduced, and error model, respectively. Then,*

$$\mathbf{y}(t) \in \Big[\mathbf{y}_r(t) - \Delta,\ \mathbf{y}_r(t) + \Delta\Big] \quad \forall t \geq 0, \tag{5.24}$$

where

$$\Delta := \|\mathbf{y}_e(\cdot)\|_{(\infty,\infty)} = \max_t \|\mathbf{y}_e(t)\|_\infty \leq \overline{\epsilon_{\mathcal{H}_2}} \cdot \|\mathbf{u}(\cdot)\|_{(2,2)} := \overline{\epsilon_{\mathcal{H}_2}} \cdot \sqrt{\textstyle\int_0^\infty \|\mathbf{u}(t)\|_2 dt}. \tag{5.25}$$

Proof. The $\mathcal{H}_2$ norm is related to the induced $\|\mathbf{G}(\cdot)\|_{2,\infty}$ norm (see [8] for details).

$$\begin{aligned} \|\mathbf{y}_e(\cdot)\|_{(\infty,\infty)} = \max_t \|\mathbf{y}_e(t)\|_\infty &\leq \sqrt{\max_i \operatorname{diag}_i\Big[\mathbf{C}\mathbf{P}\mathbf{C}^T\Big]} \cdot \|\mathbf{u}(\cdot)\|_{(2,2)} \\ &\leq \sqrt{\operatorname{tr}\Big[\mathbf{C}\mathbf{P}\mathbf{C}^T\Big]} \cdot \|\mathbf{u}(\cdot)\|_{(2,2)} = \|\mathbf{G}_e\|_{\mathcal{H}_2} \cdot \|\mathbf{u}(\cdot)\|_{(2,2)} \\ &\leq \overline{\epsilon_{\mathcal{H}_2}} \cdot \sqrt{\textstyle\int_0^\infty \|\mathbf{u}(t)\|_2 dt}. \end{aligned}$$

□

Accordingly, time domain simulations with a ROM whose associated $\mathcal{H}_2$ error is upper bounded by $\overline{\epsilon_{\mathcal{H}_2}}$ can be used to envelope the output signal of the HFM; the smaller and tighter the error bound, the thinner the envelope.

5.4. Global $\mathcal{H}_\infty$ Error Bound for Systems in Strictly Dissipative Realization

5.4.1. Upper Bound on $\mathcal{H}_\infty$ Norm

Theorem 5.2 ([124]). *Let $\mathbf{G}(s)$ be an LTI system in strictly dissipative realization and define $\mathbf{S} := -\mathbf{A} - \mathbf{A}^T$. Then $\mathbf{G}_\perp(s)$ in (5.7) is strictly dissipative as well and its $\mathcal{H}_\infty$ norm is upper bounded by*

$$\left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_\infty} \leq \left\|\mathbf{C}_\perp\mathbf{S}^{-1}\mathbf{B}_\perp\right\|_2 + \sqrt{\left\|\mathbf{B}_\perp^T\mathbf{S}^{-1}\mathbf{B}_\perp\right\|_2\left\|\mathbf{C}_\perp\mathbf{S}^{-1}\mathbf{C}_\perp^T\right\|_2}. \tag{5.26}$$

Proof. The elaborate proof was given in [124], but it is quite technical and had to be presented in a very compact form due to space limitations. For that reason, and also in order to be self-contained, it is repeated in Appendix A.2 with some additional explanations. □

5.4.2. Remarks and Implementation

In contrast to the $\mathcal{H}_2$ upper bound, the $\mathcal{H}_\infty$ upper bound in Theorem 5.2 is unique and offers no additional degrees of freedom.

As to the implementation, please note that the inverse $\mathbf{S}^{-1}$ is, of course, not required explicitly, but the numerical effort reduces to the solution of linear systems of equations. The best way to do so with regard to effort and precision is to compute the Cholesky factor of $\mathbf{S} = \mathbf{L}_S^T\mathbf{L}_S$ *once* and *a priori*—if possible. Then,

$$\begin{aligned}\left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_\infty} &\leq \left\|\underbrace{\mathbf{C}_\perp\mathbf{L}_S^{-1}}_{\mathbf{C}_{\perp,S}}\underbrace{\mathbf{L}_S^{-T}\mathbf{B}_\perp}_{\mathbf{B}_{\perp,S}}\right\|_2 + \sqrt{\left\|\underbrace{\mathbf{B}_\perp^T\mathbf{L}_S^{-1}}_{\mathbf{B}_{\perp,S}^T}\underbrace{\mathbf{L}_S^{-T}\mathbf{B}_\perp}_{\mathbf{B}_{\perp,S}}\right\|_2 \cdot \left\|\underbrace{\mathbf{C}_\perp\mathbf{L}_S^{-1}}_{\mathbf{C}_{\perp,S}}\underbrace{\mathbf{L}_S^{-T}\mathbf{C}_\perp}_{\mathbf{C}_{\perp,S}^T}\right\|_2} \\ &= \left\|\mathbf{C}_{\perp,S}\mathbf{B}_{\perp,S}\right\|_2 + \left\|\mathbf{B}_{\perp,S}\right\|_2 \cdot \left\|\mathbf{C}_{\perp,S}\right\|_2.\end{aligned} \tag{5.27}$$

MATLAB code can be seen in Source 5.2. It is assumed that the Cholesky factor $\mathbf{L}_S$ of $\mathbf{S}$ is known, which is for instance a side product of Source 2.1.

5.4.3. Relative $\mathcal{H}_\infty$ Error Bound

Similarly to the $\mathcal{H}_2$ case, a bound on the *relative* $\mathcal{H}_\infty$ error $\epsilon_{\mathcal{H}_\infty,\mathrm{rel}} := \frac{\|\mathbf{G}_e\|_{\mathcal{H}_\infty}}{\|\mathbf{G}\|_{\mathcal{H}_\infty}}$ can be quite easily derived as soon as an absolute upper bound is known.

Source 5.2: Evaluation of $\mathcal{H}_\infty$ Error Bound

```
function bndHinf = BoundHinf(L_S,P_S,B,C)
% Upper bound on H-infinity norm of strictly dissipative system
%   Input:  L_S,P_S: Cholesky factor of S=-A-A', and permutation matrix;
%           B,C    : Input and output matrix
%   Output: bndHinf: Upper bound
%

B_S = L_S'\(P_S'*B);
C_S = (L_S'\(P_S'*C'))';
bndHinf = norm(full(C_S*B_S)) + norm(full(B_S))*norm(full(C_S));
```

Corollary 5.2. *Let* $\overline{\epsilon_{\mathcal{H}_\infty}}$ *be an upper bound on the absolute* $\mathcal{H}_\infty$ *norm of an error model* $\mathbf{G}_e(s)$*, and let* $\omega^* \in \mathbb{R}$ *be the frequency for which the amplitude response of the ROM* $\mathbf{G}_r(s)$ *reaches its maximum, i. e.* $\|\mathbf{G}_r(i\omega^*)\|_2 = \|\mathbf{G}_r\|_{\mathcal{H}_\infty}$*. Then the relative* $\mathcal{H}_\infty$ *error* $\epsilon_{\mathcal{H}_\infty,rel}$ *is upper bounded by*

$$\epsilon_{\mathcal{H}_\infty,rel} \leq \overline{\epsilon_{\mathcal{H}_\infty,rel}} := \frac{\overline{\epsilon_{\mathcal{H}_\infty}}}{\|\mathbf{G}(i\omega^*)\|_2}.$$

Proof. The proof is obvious as $\|\mathbf{G}(i\omega)\|_2 \leq \|\mathbf{G}\|_{\mathcal{H}_\infty}$ for all ω including ω^*. □

In fact one may use *any* real frequency ω, evaluate the HFM at $i\omega$ and use the 2-norm of the resulting block moments instead of $\|\mathbf{G}\|_{\mathcal{H}_\infty}$ to obtain an upper bound on $\epsilon_{\mathcal{H}_\infty,\mathrm{rel}}$. However, the farer ω is from the frequency where $\|\mathbf{G}(i\omega)\|$ exhibits its peak, the greater is the overestimation introduced. For that reason, the idea here is to approximate the peak frequency of the HFM by the peak frequency of the ROM, which, of course, will fairly coincide for a reasonably good approximant $\mathbf{G}_r(s)$.

5.5. Error-Controlled Model Reduction

5.5.1. Change of Paradigm

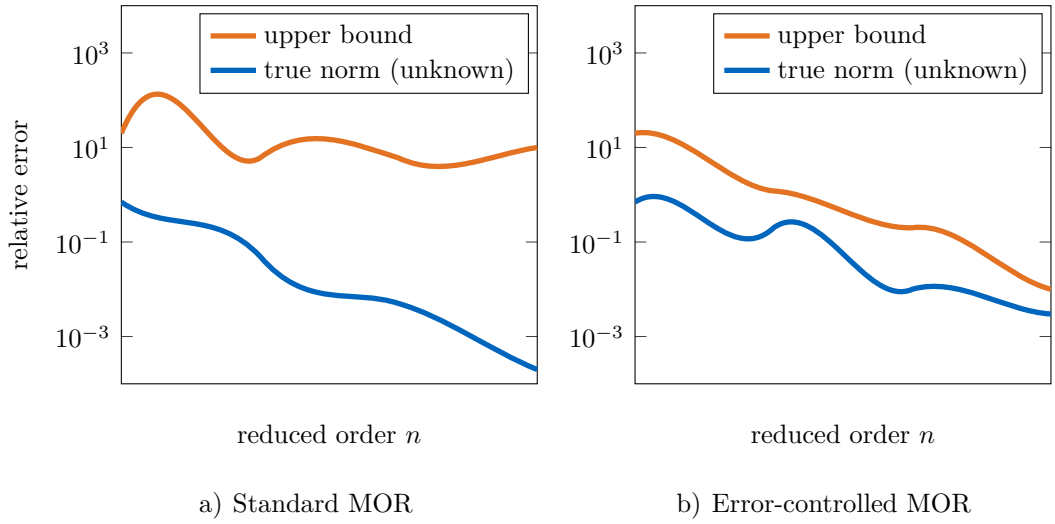
So far in this chapter, global *a posteriori* upper bounds on the absolute and relative $\mathcal{H}_2$ and $\mathcal{H}_\infty$ error have been presented. These bounds are rigorous and easy to compute, yet in practical settings it is not quite satisfactory to find out *a posteriori* that the computed ROM does not comply with the requirements such that the reduction process has to be repeated in order to find a better approximant. Rather, one would like to *a priori* define certain specifications which the ROM has to fulfill, like, for instance, a maximum of 1 % relative deviation with respect to the $\mathcal{H}_\infty$ norm.

Remember that the CURE framework can partially mend this problem, because in fact one does not have to start from scratch with the search for a ROM, but one can incorporate the revealed model into the overall ROM and perform additional reduction steps until the given condition is satisfied. In fact, the main motivation for the incremental CURE framework was its potential to choose the reduced order "on the fly".

But unfortunately, the error bounds suffer from a crucial drawback: they introduce substantial overestimation for standard reduction techniques, as could already be seen in [124]. This means that the true error norm values can be orders of magnitude smaller than what the error bounds suggest. Though this is not a problem per sé, one may easily end up with an error expression certifying that some relative error is below 1000 %, which of course is of no practical value at all.

Also, there is no guarantee that during cumulative reduction the error bounds do decay at all (see below Figure 5.3a)). Indeed, it was suggested in Sections 4.3.4 and 4.4 to perform $\mathcal{H}_2$ optimal model reduction in each step of the CURE framework, which has the positive side effect that according to Lemma 5.1, one source of overestimation is eliminated. But otherwise, $\mathcal{H}_2$ model reduction turns out to mostly yield even more overestimation than less sophisticated methods like Padé approximation about $\sigma = 0$.

Accordingly, the policy that was followed in this thesis so far—finding shifts that minimize the true $\mathcal{H}_2$ error—does not seem to be constructive with regard to the goal of finding a ROM which satisfies certain conditions on the error. In addition, given the fact



a) Standard MOR b) Error-controlled MOR

Figure 5.2.: Change of Paradigm for Error-Controlled Model Reduction by CURE (Schematic)

that the true error remains unknown, and the only reliable criteria to hold on to are the error bounds, the following change of paradigm seems advisable:

During cumulative reduction, find ROMs such that the error bounds decrease as effectively as possible, no matter how this affects the true error norms.

This concept will be referred to as *error-controlled model reduction*. Figure 5.2 is an attempt to illustrate it. The left part schematically shows results as they follow from usual model reduction: The true error norm is effectively decreased in each iteration, but the error bound—which is the only available quantity—does not constitute useful information. In Figure 5.2b), on the other hand, cumulative model order reduction is performed such that the *bound* decays in each step. The decline of the true error may not be as fast as in Figure 5.2a), but it is known to lie below the bound.

To sum up: For effective error-controlled model reduction, one must concentrate on iteratively lowering the bound of interest, even if this may diminish the decay of the true error.

5.5.2. How to Control Overestimation of $\mathcal{H}_2$ Error Bound

We start with considerations on the $\mathcal{H}_2$ case and assume without loss of generality that $\mathbf{V}$-based decomposition of the error is performed; all results carry over to $\mathbf{W}$-sided factorization.

Recall that the error bound (5.8) is given by

$$\left\|\mathbf{G}_e\right\|_{\mathcal{H}_2} \leq \left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_2} \cdot \left\|\widetilde{\mathbf{G}}_r^R\right\|_{\mathcal{H}_\infty}$$

and we can use (5.14) and (5.15) to upper bound the first factor while the second is easy to obtain. Let us now exemplarily concentrate on formulation (5.15) of the $\mathcal{H}_2$ bound:

$$\left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_2}^2 \leq \operatorname{tr}\left[\mathbf{B}_\perp^T \widehat{\mathbf{Q}} \mathbf{B}_\perp\right] + \frac{1}{-2\mu_{\mathbf{E}}(\mathbf{A})} \cdot \left\|\mathbf{L}^{-T}\mathbf{R}_O\mathbf{L}^{-1}\right\|_2 \cdot \left\|\mathbf{L}^{-T}\mathbf{B}_\perp\right\|_{\mathrm{F}}^2 \tag{5.28}$$

It turns out that for reasonable approximate Gramians $\widehat{\mathbf{Q}}$ (and even for $\widehat{\mathbf{Q}} = \mathbf{0}$), the first summand is comparably small and does therefore not essentially contribute to the overestimation the bound exhibits. In fact, even for mediocre $\widehat{\mathbf{Q}}$ the term is often close to the true squared error norm.

Indeed, it is the second summand which is responsible for the overestimation; we shall therefore look at it more closely. We note that its first two factors are not dependent on the ROM: $\frac{1}{-2\mu_{\mathbf{E}}(\mathbf{A})}$ is a property of the HFM matrices $\mathbf{A}$ and $\mathbf{E}$ only, while $\left\|\mathbf{L}^{-T}\mathbf{R}_O\mathbf{L}^{-1}\right\|_2$ is determined by the approximate Gramian $\widehat{\mathbf{Q}}$, which we choose independently of the ROM. Accordingly, it is only the third factor we can influence, namely by finding a ROM such that the singular values of the resulting term $\mathbf{B}_\perp^T\mathbf{E}^{-1}\mathbf{B}_\perp$ become as small as possible.

Proposition 5.2. *Let* $\mathbf{V}$ *solve* Sylvester *equation* (3.3) *and set* $\mathbf{W} := \mathbf{V}$. *Then the input matrix* $\mathbf{B}_\perp$ *of* $\mathbf{G}_\perp(s)$ *in the error factorization* (4.6) *fulfills* $\left\|\mathbf{L}^{-T}\mathbf{B}_\perp\right\|_{\mathrm{F}} \leq \left\|\mathbf{L}^{-T}\mathbf{B}\right\|_{\mathrm{F}}$.

Proof. $\mathbf{W} := \mathbf{V}$ leads to orthogonal projection with respect to the $\mathbf{E}^{-1}$ inner product:

$$\begin{aligned}
\left\|\mathbf{L}^{-T}\mathbf{B}_\perp\right\|_{\mathrm{F}}^2 &= \operatorname{tr}\left[\mathbf{B}_\perp^T\mathbf{E}^{-1}\mathbf{B}_\perp\right] \\
&= \operatorname{tr}\left[(\mathbf{B} - \mathbf{E}\mathbf{V}\mathbf{E}_r^{-1}\mathbf{B}_r)^T\mathbf{E}^{-1}(\mathbf{B} - \mathbf{E}\mathbf{V}\mathbf{E}_r^{-1}\mathbf{B}_r)\right] \\
&= \operatorname{tr}\left[\mathbf{B}^T\mathbf{E}^{-1}\mathbf{B} - 2\mathbf{B}^T\mathbf{E}^{-1}\mathbf{E}\mathbf{V}\mathbf{E}_r^{-1}\mathbf{B}_r + \mathbf{B}_r^T\mathbf{E}_r^{-T}\mathbf{V}^T\mathbf{E}^T\mathbf{E}^{-1}\mathbf{E}\mathbf{V}\mathbf{E}_r^{-1}\mathbf{B}_r\right] \\
&= \operatorname{tr}\left[\mathbf{B}^T\mathbf{E}^{-1}\mathbf{B} - 2\underbrace{\mathbf{B}^T\mathbf{V}}_{\mathbf{B}_r^T}\mathbf{E}_r^{-1}\mathbf{B}_r + \mathbf{B}_r^T\mathbf{E}_r^{-T}\underbrace{\mathbf{V}^T\mathbf{E}\mathbf{V}}_{\mathbf{E}_r}\mathbf{E}_r^{-1}\mathbf{B}_r\right] \\
&= \operatorname{tr}\left[\mathbf{B}^T\mathbf{E}^{-1}\mathbf{B}\right] - \underbrace{\operatorname{tr}\left[\mathbf{B}_r^T\mathbf{E}_r^{-1}\mathbf{B}_r\right]}_{\geq 0}.
\end{aligned}$$

□

Accordingly, applying this $\mathbf{E}^{-1}$-orthogonal projection in every iteration of the CURE framework delivers monotonic decay of $\left\|\mathbf{L}^{-T}\mathbf{B}_\perp\right\|_{\mathrm{F}}^2$ and therefore—under the above assumptions—of the upper bound on $\|\mathbf{G}_\perp\|_{\mathcal{H}_2}$. However, the bound on the actual error norm $\|\mathbf{G}_e\|_{\mathcal{H}_2}$ also contains the factor $\|\widetilde{\mathbf{G}}_r^R\|_{\mathcal{H}_\infty}$. It was already discussed in Section 5.2 that in general this term can also induce massive overestimation. For $\mathcal{H}_2$ pseudo-optimal reduction, however, it amounts to one and introduces no additional overestimation at all.

So in order to guarantee monotonic decay of the error bound in the CURE framework, we could—in addition to the condition $\mathbf{V} = \mathbf{W}$—demand that the Krylov subspace must fulfill

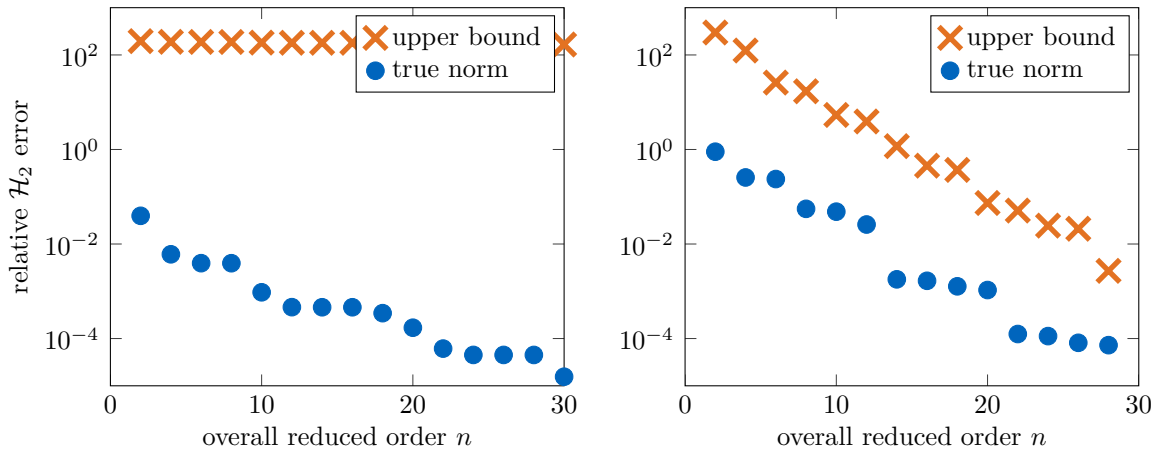
$$\lambda_i(\mathbf{S}_V) = -\lambda_i(\mathbf{V}^T\mathbf{A}\mathbf{V}, \mathbf{V}^T\mathbf{E}\mathbf{V}), \tag{5.29}$$

where $\mathbf{S}_V$ follows from Sylvester equation (3.3). Remember that the eigenvalues of $\mathbf{S}_V$ are precisely the shifts used for the Krylov subspace, so eventually the conditions read as follows: We must find a Krylov subspace such that the employed expansion points are the mirror images of the eigenvalues of the ROM which results from one-sided projection.

It is quite clear that this is not fulfilled for arbitrary shifts. Indeed, it sounds very much alike what one does when running IRKA, with the only difference that in IRKA, both $\mathbf{V}$ and $\mathbf{W}$ span Krylov subspaces, while now the input Krylov subspace must be used for either of the matrices $\mathbf{V}$ and $\mathbf{W}$. However, we can very easily modify IRKA appropriately.

In fact, it turns out that this one-sided version of IRKA exhibits similar convergence properties as its ancestor, so a shift configuration fulfilling the given conditions can be found as well as a local $\mathcal{H}_2$ optimum. This means that, unfortunately, the same difficulties as in standard IRKA may arise: convergence is not monotonic and may take very long or sometimes not occur at all. In such a case, one may also stop IRKA prematurely and follow up PORK to obtain an $\mathcal{H}_2$ pseudo-optimum. The ROM is then not exactly the result of an orthogonal projection, but $\mathbf{B}_\perp$ may still be "shorter" than $\mathbf{B}$ such that the error bound decreases.

For proof of concept, we consider the numerical example of the continuous heat equation [38]. It only has order $N = 200$, so we can easily compare the values of the error



a) CUREd IRKA b) CUREd Modified IRKA ($\mathbf{W} = \mathbf{V}$)

Figure 5.3.: Simulation Results for $\mathcal{H}_2$ Error-Controlled MOR of Continuous Heat Equation

bound to the true respective error. We use the CURE scheme during which we compute ROMs of order $n_i = 2$ that are iteratively accumulated based on $\mathbf{V}$-type error decomposition. For the actual reduction we use two different approaches: firstly standard IRKA and secondly the modified version of IRKA where we set $\mathbf{W} = \mathbf{V}$. After each step, the $\mathcal{H}_2$ error bound is evaluated, where for simplicity we just set $\widehat{\mathbf{Q}} = \mathbf{0}$.

The results after some iterations are shown in Figure 5.3. In fact, they very much resemble the plots in Figure 5.2: While standard $\mathcal{H}_2$ optimal reduction improves the true error, the bound does not decay. The modified IRKA version with orthogonal projection, however, achieves fast decay of the bound with slightly deteriorated true error.

So far we assumed $\mathbf{V}$ to be a Krylov subspace and chose $\mathbf{W} = \mathbf{V}$. Therefore, we were only allowed to apply $\mathbf{V}$-based error decomposition (4.6). Of course, everything carries over to $\mathbf{W}$ solving Sylvester equation (3.4) and the corresponding factorization (4.7) of the error model. In fact, it turns out that in practice, it is most efficient to use both variants and alternate between them, because then both $\mathbf{B}_\perp$ and $\mathbf{C}_\perp$ are shortened during CURE, while in purely input type factorization, for instance, $\mathbf{C}_\perp$ would always remain the output matrix $\mathbf{C}$ of the HFM.

5.5.3. How to Control Overestimation of $\mathcal{H}_\infty$ Error Bound

Let us now turn to the $\mathcal{H}_\infty$ case and again consider $\mathbf{V}$-based decomposition. Here, the upper bound (5.26) on $\|\mathbf{G}_\perp\|_{\mathcal{H}_\infty}$ takes an even simpler form than in the $\mathcal{H}_2$ case:

$$\left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_\infty} \leq \left\|\mathbf{C}\mathbf{S}^{-1}\mathbf{B}_\perp\right\|_2 + \sqrt{\left\|\mathbf{B}_\perp^T\mathbf{S}^{-1}\mathbf{B}_\perp\right\|_2} \cdot \sqrt{\left\|\mathbf{C}\mathbf{S}^{-1}\mathbf{C}^T\right\|_2}. \tag{5.30}$$

Obviously, only $\mathbf{B}_\perp$ depends on the ROM, but it is not straightforward to say how one should try to influence it, because one would like to minimize $\left\|\mathbf{C}\mathbf{S}^{-1}\mathbf{B}_\perp\right\|_2$ and $\left\|\mathbf{B}_\perp^T\mathbf{S}^{-1}\mathbf{B}_\perp\right\|_2$ at the same time. However, one can easily show that

$$\left\|\mathbf{C}\mathbf{S}^{-1}\mathbf{B}_\perp\right\|_2 \leq \sqrt{\left\|\mathbf{B}_\perp^T\mathbf{S}^{-1}\mathbf{B}_\perp\right\|_2} \cdot \sqrt{\left\|\mathbf{C}\mathbf{S}^{-1}\mathbf{C}^T\right\|_2} \tag{5.31}$$

holds because of the Cauchy-Schwarz inequality. Introducing the Cholesky decomposition $\mathbf{S} = \mathbf{L}_S^T\mathbf{L}_S$, we can therefore replace the upper bound by yet another upper bound,

$$\left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_\infty} \leq 2 \cdot \sqrt{\left\|\mathbf{B}_\perp^T\mathbf{S}^{-1}\mathbf{B}_\perp\right\|_2} \cdot \sqrt{\left\|\mathbf{C}\mathbf{S}^{-1}\mathbf{C}^T\right\|_2} = 2 \cdot \|\mathbf{L}_S^{-T}\mathbf{B}_\perp\|_2 \cdot \|\mathbf{L}_S^{-T}\mathbf{C}^T\|_2, \tag{5.32}$$

the second factor of which is independent of the reduction. Note that the worst case additional overestimation introduced by this step is a factor of two.

However, if we succeed in lowering $\|\mathbf{L}_S^{-T}\mathbf{B}_\perp\|_2$, then eventually we also diminish the actual bound and obtain more precise information on the error.

Remarkably, the task is highly similar to the problem in the $\mathcal{H}_2$ case: We want to find an $\mathcal{H}_2$ pseudo-optimal ROM (otherwise, the factor $\|\widetilde{\mathbf{G}}_r^R\|_{\mathcal{H}_\infty}$ can prevent the bound from being tight) which at the same time guarantees that $\mathbf{B}_\perp$ becomes shorter with respect to some norm. The only difference is that now we need to consider the $\mathbf{S}^{-1}$-norm instead of the $\mathbf{E}^{-1}$-norm as above. But this can be done in a very similar way.

Proposition 5.3. *Let* $\mathbf{V}$ *solve* Sylvester *equation* (3.3) *and set* $\mathbf{W} := \mathbf{S}^{-1}\mathbf{E}\mathbf{V}$*, such that* $\mathbf{W}^T = \mathbf{V}^T\mathbf{E}\mathbf{S}^{-1}$*. Then the input vector* $\mathbf{B}_\perp$ *of* $\mathbf{G}_\perp(s)$ *in the error factorization* (4.6) *fulfills* $\left\|\mathbf{B}_\perp^T\mathbf{S}^{-1}\mathbf{B}_\perp\right\|_2 \leq \left\|\mathbf{B}^T\mathbf{S}^{-1}\mathbf{B}\right\|_2$.

Proof.

$$\begin{aligned}
\mathbf{B}_\perp^T\mathbf{S}^{-1}\mathbf{B}_\perp &= (\mathbf{B} - \mathbf{E}\mathbf{V}\mathbf{E}_r^{-1}\mathbf{B}_r)^T\mathbf{S}^{-1}(\mathbf{B} - \mathbf{E}\mathbf{V}\mathbf{E}_r^{-1}\mathbf{B}_r) \\
&= \mathbf{B}^T\mathbf{S}^{-1}\mathbf{B} - 2\underbrace{\mathbf{B}^T\mathbf{S}^{-1}\mathbf{E}\mathbf{V}}_{\mathbf{B}_r^T}\mathbf{E}_r^{-1}\mathbf{B}_r + \mathbf{B}_r^T\underbrace{\mathbf{E}_r^{-T}}_{\mathbf{E}_r^{-1}}\underbrace{\mathbf{V}^T\mathbf{E}^T\mathbf{S}^{-1}\mathbf{E}\mathbf{V}}_{\mathbf{E}_r}\mathbf{E}_r^{-1}\mathbf{B}_r \\
&= \mathbf{B}^T\mathbf{S}^{-1}\mathbf{B} - \underbrace{\mathbf{B}_r^T\mathbf{E}_r^{-1}\mathbf{B}_r}_{\geq \mathbf{0}}
\end{aligned}$$

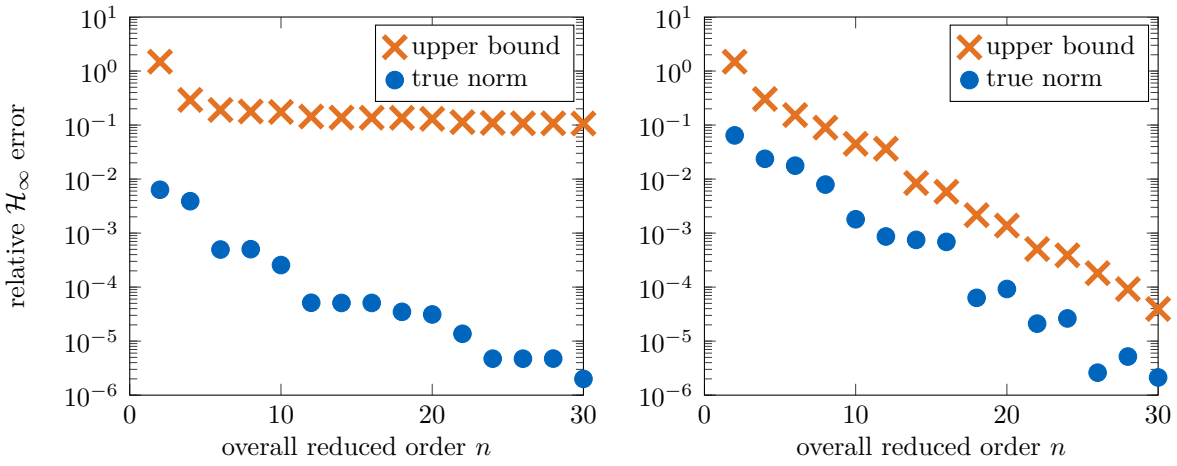
As $\mathbf{B}_\perp^T\mathbf{S}^{-1}\mathbf{B}_\perp \geq \mathbf{0}$, it must hold $\left\|\mathbf{B}_\perp^T\mathbf{S}^{-1}\mathbf{B}_\perp\right\|_2 \leq \left\|\mathbf{B}^T\mathbf{S}^{-1}\mathbf{B}\right\|_2$. □

Quite like before, one way to suitably influence the error bound is to find a shift configuration for which the ROM resulting from the given oblique projection is $\mathcal{H}_2$ pseudo-optimal, i. e. the KRYLOV subspace $\mathbf{V}$ must fulfill (3.3) such that

$$\lambda_i(\mathbf{S}_V) = -\lambda_i\left(\mathbf{V}^T\mathbf{E}\mathbf{S}^{-1}\mathbf{A}\mathbf{V}, \mathbf{V}^T\mathbf{E}\mathbf{S}^{-1}\mathbf{E}\mathbf{V}\right) \qquad \forall i = 1\ldots n. \tag{5.33}$$

So again, we can use a modification of IRKA, this time by defining `W:=S\(E*V)`. Of course, one can improve computational efficiency by calculating the CHOLESKY factor of $\mathbf{S}$ only once and then using it during IRKA's iterations as well as for the calculation of the error bound. The computational overhead incurred into IRKA is then again minor.

Again, we consider the continuous heat equation for proof of concept. The procedure is as above in Section 5.5.2, just that we set $\mathbf{W} := \mathbf{S}^{-1}\mathbf{E}\mathbf{V}$ in the second run, and compute the $\mathcal{H}_\infty$ norm instead of the $\mathcal{H}_2$ norm. The outcome is depicted in Figure 5.4 and exhibits the same behavior as before; the upper bound on the $\mathcal{H}_\infty$ error norm in the second case is very tight, as Figure 5.4b) shows.



a) CUREd IRKA b) CUREd Modified IRKA ($\mathbf{W} = \mathbf{S}^{-1}\mathbf{E}\mathbf{V}$)

Figure 5.4.: Simulation Results for $\mathcal{H}_\infty$ Error-Controlled MOR of Continuous Heat Equation

5.6. Optimization-based Decrease of Error Bounds

It was shown in the previous section that certain modifications of IRKA yield guaranteed decrease of the error bounds, in case the respective algorithm converges. However, as was already commented on in Section 4.4, IRKA (and its derivates, too) suffers from unsteady convergence behavior. This was the motivation for the optimization-based descent algorithm SPARK (Section 4.4.2), during which we concentrated on $\mathcal{H}_2$ pseudo-optimal ROMs and used the cost functional $\mathcal{J} = \|\mathbf{G}_e\|_{\mathcal{H}_2}^2 - \|\mathbf{G}\|_{\mathcal{H}_2}^2 = -\|\mathbf{G}_r\|_{\mathcal{H}_2}^2$ together with a model function (Section 4.4.4) for efficiency.

The remainder of this section is therefore intended to present ideas for an optimization-based alternative to the modified versions of IRKA, similarly to $\mathcal{H}_2$ model reduction by MESPARK. The goal is to obtain a descent algorithm for fast and reliable reduction of the error bounds during MOR with the CURE framework.

Again, we will focus on $\mathcal{H}_2$ pseudo-optimal reduction, because $\widetilde{\mathbf{G}}_r^L$ and $\widetilde{\mathbf{G}}_r^R$ are unity all-pass elements then, and the error norm is bounded above by the respective upper bound on $\|\mathbf{G}_\perp\|_{\mathcal{H}_2}$ or $\|\mathbf{G}_\perp\|_{\mathcal{H}_\infty}$.

5.6.1. Optimization of $\mathcal{H}_2$ Error Bound

For a start, we assume a SISO system. Then, the plain idea is to replace the cost functional $\mathcal{J} = -\|\mathbf{G}_r\|_{\mathcal{H}_2}^2$ by

$$\mathcal{J}_{\mathcal{H}_2} := \frac{\mathbf{b}_\perp^T \mathbf{E}^{-1} \mathbf{b}_\perp}{\mathbf{b}^T \mathbf{E}^{-1} \mathbf{b}} - 1, \tag{5.34}$$

which describes the relative improvement of the squared upper bound on the $\mathcal{H}_2$ norm in $\mathbf{V}$-based error decomposition, assuming $\widehat{\mathbf{P}} = \widehat{\mathbf{Q}} = \mathbf{0}$.

Given some $\widehat{\mathbf{P}} \neq \mathbf{0}$ or $\widehat{\mathbf{Q}} \neq \mathbf{0}$, one might of course also use one of the actual error bounds as $\mathcal{J}$, but this cost functional would increase complexity and not be smooth with respect to the optimization variables a, b. Remember at this point, that the one-sided IRKA algorithm described in Section 5.5.2 had a very similar objective (lowering $\mathbf{b}_\perp^T \mathbf{E}^{-1} \mathbf{b}_\perp$) and worked out well, too.

In fact, using the new cost functional (5.34) does not change much in comparison to what was presented in Section 4.4. Again, PORK is used to find ROMs of order $q = 2$, and we parametrize all $\mathcal{H}_2$ pseudo-optimal ROMs by two positive real numbers a, b, like

in Section 4.4.2. This yields $\sigma_{1,2} := a \pm \sqrt{a^2 - b}$, $\mathbf{V} = \left[\frac{1}{2}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} + \frac{1}{2}\mathbf{A}_{\sigma_2}^{-1}\mathbf{b}, \quad \mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}\right]$, and $\mathbf{b}_r = [-4a, \ -4a^2]^T$ as in Lemma 4.2 and Corollary 4.2. Accordingly, (5.34) can be directly evaluated.

Gradients and Hessian, too, can be derived similarly to Section 4.4.3. The terms become slightly more complicated, because we need the partial derivatives of the $N \times 2$-matrix $\mathbf{V}$, and not only those of the 2-dimensional vector $\mathbf{c}_r = \mathbf{c}\mathbf{V}$ as in Theorem 4.3. But the derivatives are feasible; in fact, the formulas of Theorem 4.3 can be adapted in a straightforward way by omitting the factor $\mathbf{c}$ on the left.

However, the computation of gradient and Hessian requires several LSE solves, therefore it is even more important than during $\mathcal{H}_2$ model reduction to use a model function which avoids as many costly large-scale operations as possible. To guarantee stability of the model function, one-sided projection is recommended.

The new cost functional then also has a nice feature: it is independent of the output matrix $\mathbf{C}$ of the HFM. Therefore, it directly extends to SIMO systems without any changes. And of course, it can be used in the MISO case as well by using the dual formulation

$$\mathcal{J}_{\mathcal{H}_2} := \frac{\mathbf{c}_\perp \mathbf{E}^{-1} \mathbf{c}_\perp^T}{\mathbf{c}\,\mathbf{E}^{-1}\mathbf{c}^T} - 1. \tag{5.35}$$

As to the implementation, the few necessary modifications starting from the standard MESPARK algorithm Source 4.4 are summed up in Source A.1; the implementation of the cost functional `CostFunctionH2Bound` is given in Source A.2; both can be found in the appendix.

Remember that the $\mathcal{H}_2$ objective function has the nice property that for $\mathcal{H}_2$ pseudo-optimal reduction it is negative in the whole parameter range; so even if no optimum is found and the algorithm is stopped before convergence, PORK makes sure that the error norm decays anyway due to Proposition 4.3. In fact, the new cost functional (5.34) has the same property:

Proposition 5.4. *The upper bound on the $\mathcal{H}_2$ norm of the error decreases monotonically in the CURE framework if $\mathcal{H}_2$ pseudo-optimal reduction is applied in every step and $\hat{\mathbf{Q}} = \mathbf{0}$:*

$$\mathcal{J}_{\mathcal{H}_2}(a, b) < 0 \quad \forall a, b \in \mathbb{R}^+.$$

Proof. We consider the numerator $\mathcal{J}^*_{\mathcal{H}_2}$ of $\mathcal{J}_{\mathcal{H}_2}$, as the denominator is positive.

$$\begin{aligned}
\mathcal{J}^*_{\mathcal{H}_2} &= (\mathbf{b} - \mathbf{E}\mathbf{V}\mathbf{E}_r^{-1}\mathbf{b}_r)^T\mathbf{E}^{-1}(\mathbf{b} - \mathbf{E}\mathbf{V}\mathbf{E}_r^{-1}\mathbf{b}_r) - \mathbf{b}^T\mathbf{E}^{-1}\mathbf{b} \\
&= -2\mathbf{b}_r^T\mathbf{V}^T\mathbf{b} + \mathbf{b}_r^T\mathbf{V}^T\mathbf{E}\mathbf{V}\mathbf{b}_r \\
&= -2\operatorname{tr}\left[\mathbf{V}^T\mathbf{b}\mathbf{b}_r^T\right] + \operatorname{tr}\left[\mathbf{b}_r^T\mathbf{V}^T\mathbf{E}\mathbf{V}\mathbf{b}_r\right] \\
&= 2\operatorname{tr}\left[\mathbf{V}^T\mathbf{A}\mathbf{V}\mathbf{P}_r\right] + 2\operatorname{tr}\left[\mathbf{V}^T\mathbf{E}\mathbf{V}\mathbf{P}_r\mathbf{A}_r^T\right] + \operatorname{tr}\left[\mathbf{V}^T\mathbf{E}\mathbf{V}\mathbf{b}_r\mathbf{b}_r^T\right] \quad \text{(because of (3.37))} \\
&= 2\operatorname{tr}\left[\mathbf{V}^T\mathbf{A}\mathbf{V}\mathbf{P}_r\right] + \operatorname{tr}\left[\mathbf{V}^T\mathbf{E}\mathbf{V}\mathbf{P}_r\mathbf{A}_r^T\right] + \operatorname{tr}\left[\mathbf{A}_r\mathbf{P}_r\mathbf{V}^T\mathbf{E}\mathbf{V}\right] + \operatorname{tr}\left[\mathbf{V}^T\mathbf{E}\mathbf{V}\mathbf{b}_r\mathbf{b}_r^T\right] \\
&= 2\operatorname{tr}\left[\mathbf{V}^T\mathbf{A}\mathbf{V}\mathbf{P}_r\right] + \operatorname{tr}\left[\mathbf{V}^T\mathbf{E}\mathbf{V}\left(\mathbf{P}_r\mathbf{A}_r^T + \mathbf{A}_r\mathbf{P}_r + \mathbf{b}_r\mathbf{b}_r^T\right)\right] \\
&= 2\operatorname{tr}\left[\mathbf{V}^T\mathbf{A}\mathbf{V}\mathbf{L}_{\widehat{\mathbf{P}}_r}^T\mathbf{L}_{\widehat{\mathbf{P}}_r}\right] = 2\operatorname{tr}\left[\mathbf{L}_{\widehat{\mathbf{P}}_r}\mathbf{V}^T\mathbf{A}\mathbf{V}\mathbf{L}_{\widehat{\mathbf{P}}_r}^T\right] \\
&= \operatorname{tr}\left[\mathbf{L}_{\widehat{\mathbf{P}}_r}\mathbf{V}^T(\mathbf{A} + \mathbf{A}^T)\mathbf{V}\mathbf{L}_{\widehat{\mathbf{P}}_r}^T\right] < 0
\end{aligned}$$

□

The new cost functional therefore resembles the one discussed in Section 4.4 and typically looks like Figure 4.7. Accordingly, starting from an arbitrary initial value, optimization should yield a local minimum under mild assumptions. In fact, a fast decrease of the upper $\mathcal{H}_2$ error bound has been observed in many applications.

5.6.2. Optimization of $\mathcal{H}_\infty$ Error Bound

To reduce the upper bound on the $\mathcal{H}_\infty$ norm iteratively, one can use the cost functional

$$\mathcal{J}_{\mathcal{H}_\infty} := \frac{\mathbf{b}_\perp^T\mathbf{S}^{-1}\mathbf{b}_\perp}{\mathbf{b}^T\mathbf{S}^{-1}\mathbf{b}} - 1 = \frac{-2\mathbf{b}^T\mathbf{S}^{-1}\mathbf{E}\mathbf{V}\mathbf{b}_r + \mathbf{b}_r^T\mathbf{V}^T\mathbf{E}\mathbf{S}^{-1}\mathbf{E}\mathbf{V}\mathbf{b}_r}{\mathbf{b}^T\mathbf{S}^{-1}\mathbf{b}}, \tag{5.36}$$

which describes the relative improvement of the squared upper bound (5.32) on the $\mathcal{H}_\infty$ norm,

$$\left\|\mathbf{G}_\perp\right\|_{\mathcal{H}_\infty} \leq 2 \cdot \|\mathbf{L}_S^{-T}\mathbf{B}_\perp\|_2 \cdot \|\mathbf{L}_S^{-T}\mathbf{C}^T\|_2, \tag{5.37}$$

in $\mathbf{V}$-based error decomposition. Again, we do not use the tighter bound (5.32) because of discontinuities; minimization of $\mathcal{J}_{\mathcal{H}_\infty}$ is also related to the objective of the modified IRKA with $\mathbf{W} = \mathbf{S}^{-1}\mathbf{E}\mathbf{V}$ in Section 5.5.3.

The formulas and considerations from the previous subsection carry over quite similarly, because for the computation of $\mathcal{J}_{\mathcal{H}_\infty}$, its gradient, and Hessian matrix, one needs $\mathbf{b}_\perp$ and its partial derivatives.

Unfortunately, there is a crucial difference to the cost functionals that minimize the $\mathcal{H}_2$ error or the upper $\mathcal{H}_2$ error bound: $\mathcal{J}_{\mathcal{H}_\infty}(a, b)$ is not necessarily negative for all a, b. In

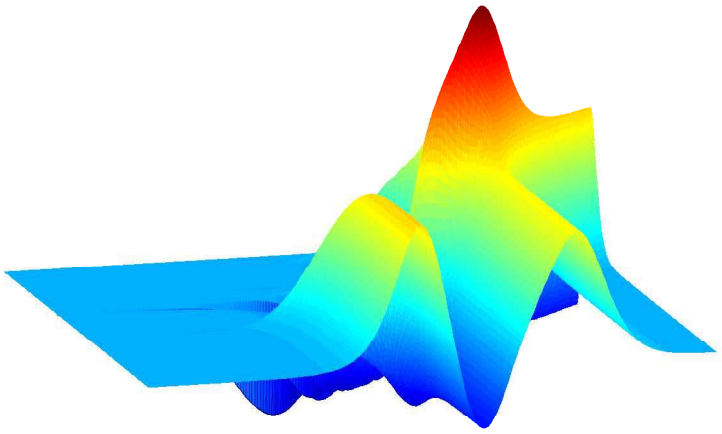


Figure 5.5.: Typical Shape of Cost Functional $\mathcal{J}_{\mathcal{H}_\infty}(a, b)$

fact, two highly problematic situations can occur: Firstly, there may be no configuration at all which would lead to a decrease of the $\mathcal{H}_\infty$ error bound; so any $\mathcal{H}_2$ pseudo-optimal reduction leads to an increase of the error bound, which is counterproductive, of course. Secondly, the cost functional may be shaped as can be seen in Figure 5.5 (the light blue plane indicates zero level). Even though there are regions where $\mathcal{J}_{\mathcal{H}_\infty}$ is negative, they are not found by the optimizer if the initial position is chosen disadvantageously.[2]

Obviously, minimizing the upper bound on the $\mathcal{H}_\infty$ norm is therefore a more challenging task and will need further research.

[2]Typically, the optimizer does not move towards a minimum, but towards the boundary. The reason is that for $a \to \infty$, $b \to 0$, or $b \to \infty$, the cost functional approaches zero, because the ROM tends to zero if the expansion points move towards infinity or the imaginary axis. Clearly the optimizer prefers zero to a positive $\mathcal{J}_{\mathcal{H}_\infty}$ at an unsuitable initial condition.

6. Example of Use: Second Order Systems

The error bounds presented in the previous chapter only apply to strictly dissipative state space models and are therefore restrictive in their assumptions. This chapter is dedicated to certain second order systems, which can often be formulated in such a state space realization and thus constitute an interesting area of application. We will see how state space methodology (including the newly introduced CURE framework) can be applied efficiently to second order systems in general, and how the strictly dissipative realization derived in [123, 127] leads to $\mathcal{H}_\infty$ and $\mathcal{H}_2$ error bounds for second order systems with positive definite mass, damping, and stiffness matrices.

6.1. Preliminaries on Second Order Systems

Definition 6.1. *A second order system is given by*

$$\mathbf{G}(s) : \begin{cases} \mathbf{M}\,\ddot{\mathbf{z}}(t) + \mathbf{D}\,\dot{\mathbf{z}}(t) + \mathbf{K}\,\mathbf{z}(t) = \mathbf{F}\,\mathbf{u}(t)\,, \\ \mathbf{y}(t) = \mathbf{C}_\mathrm{p}\,\mathbf{z}(t) + \mathbf{C}_\mathrm{v}\,\dot{\mathbf{z}}(t), \end{cases} \tag{6.1}$$

where $\mathbf{z}(t) \in \mathbb{R}^{\widehat{N}}$, $\mathbf{u}(t) \in \mathbb{R}^m$, *and* $\mathbf{y}(t) \in \mathbb{R}^p$ *contain the* $\widehat{N}$ *displacement variables,* m *inputs, and* p *outputs of the system, respectively.* $\mathbf{F} \in \mathbb{R}^{\widehat{N}\times m}$ *and* $\mathbf{C}_\mathrm{p}, \mathbf{C}_\mathrm{v} \in \mathbb{R}^{p\times\widehat{N}}$ *denote the input, position-based and velocity-based output matrix, respectively.* $\mathbf{M}, \mathbf{D}, \mathbf{K} \in \mathbb{R}^{\widehat{N}\times\widehat{N}}$ *are called mass, damping, and stiffness matrix.*

In the following, we assume $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ to be symmetric positive definite:

$$\mathbf{M} = \mathbf{M}^T > \mathbf{0}, \quad \mathbf{K} = \mathbf{K}^T > \mathbf{0}, \quad \mathbf{D} = \mathbf{D}^T > \mathbf{0}. \tag{6.2}$$

Admittedly, these assumptions are restrictive. In the electrical domain, for instance, a full-rank high-dimensional damping matrix belongs to a system with extremely many dampening (resistive) elements—and in fact, this is not necessary for asymptotic stability;

a rank one damping matrix can also be pervasive, as it is e. g. the case in the telegraph equation.

In structural mechanics, on the other hand, friction is often modeled as RAYLEIGH damping (also: proportional damping), meaning

$$\mathbf{D} = \alpha\,\mathbf{K} + \beta\,\mathbf{M}, \qquad \alpha, \beta \geq 0. \tag{6.3}$$

For positive definite mass and stiffness matrices, this implies $\mathbf{D} > \mathbf{0}$ unless $\alpha = \beta = 0$. Accordingly, many FEM models of microelectronic-mechanical (MEMS) devices and lightweight structures fulfill the definiteness conditions (6.2).

6.2. Strictly Dissipative State Space Realizations of Second Order Systems

A standard realization of second order systems in state space is given by

$$\begin{aligned} \overbrace{\begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix}}^{\mathbf{E}} \begin{bmatrix} \dot{\mathbf{z}}(t) \\ \ddot{\mathbf{z}}(t) \end{bmatrix} &= \overbrace{\begin{bmatrix} \mathbf{0} & \mathbf{R} \\ -\mathbf{K} & -\mathbf{D} \end{bmatrix}}^{\mathbf{A}} \begin{bmatrix} \mathbf{z}(t) \\ \dot{\mathbf{z}}(t) \end{bmatrix} + \overbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{F} \end{bmatrix}}^{\mathbf{B}} \mathbf{u}(t), \\ \mathbf{y}(t) &= \underbrace{\begin{bmatrix} \mathbf{C}_\mathrm{p} & \mathbf{C}_\mathrm{v} \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} \mathbf{z}(t) \\ \dot{\mathbf{z}}(t) \end{bmatrix}. \end{aligned} \tag{6.4}$$

where $\mathbf{R} \in \mathbb{R}^{\widehat{N}\times\widehat{N}}$ is an arbitrary regular matrix, so that the state vector consists of the positions $\mathbf{z}$ and velocities $\dot{\mathbf{z}}$ [138]. The order of the state space model is of course $N = 2\widehat{N}$.

The simplest choice is naturally $\mathbf{R} = \mathbf{I}_{\widehat{N}}$, but SALIMBAHRAMI observed that $\mathbf{R} = \mathbf{K}$ delivered a realization with favorable properties with respect to stability preservation [138]. In fact, the resulting matrix $\mathbf{E}$ is positive definite while the symmetric part of $\mathbf{A}$,

$$\operatorname{sym} \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{D} \end{bmatrix},$$

is clearly negative semidefinite, which characterizes a (not strictly) dissipative realization with $\mu_{\mathbf{E}}(\mathbf{A}) = 0$ and guarantees preservation of stability in one-sided projection.

But the application of the error bounds from Chapter 5 requires a *strictly* dissipative realization, i. e. $\mu = \mu_{\mathbf{E}}(\mathbf{A}) < 0$. According to Lemma 2.2, finding such a realization is related to solving an N-dimensional LYAPUNOV inequality and therefore not possible for

general large-scale systems. For the special case of second order systems, however, the problem was reconsidered in [127] and [123]. Indeed, it is possible to find a strictly dissipative state space formulation in this case. The results are summarized in the following.

Starting from realization (6.4) above, one pre-multiplies the state equation from the left by a matrix

$$\mathbf{T} := \begin{bmatrix} \mathbf{I} & \gamma\mathbf{I} \\ \gamma\mathbf{M}\mathbf{K}^{-1} & \mathbf{I} \end{bmatrix} \in \mathbb{R}^{N\times N}, \tag{6.5}$$

which depends on the real positive scalar $\gamma \in \mathbb{R}^+$. This does not affect the solution $\mathbf{x}(t)$ and is neither a state transformation, but merely a change of realization, as only the "row information" is re-ordered.

Theorem 6.1 ([123, 127])**.** *The matrices*

$$\widetilde{\mathbf{A}} = \begin{bmatrix} -\gamma\,\mathbf{K} & \mathbf{K} - \gamma\,\mathbf{D} \\ -\mathbf{K} & -\mathbf{D} + \gamma\,\mathbf{M} \end{bmatrix}, \quad \widetilde{\mathbf{E}} = \begin{bmatrix} \mathbf{K} & \gamma\,\mathbf{M} \\ \gamma\,\mathbf{M} & \mathbf{M} \end{bmatrix}, \quad \widetilde{\mathbf{B}} = \begin{bmatrix} \gamma\,\mathbf{F} \\ \mathbf{F} \end{bmatrix}, \quad \widetilde{\mathbf{C}} = \begin{bmatrix} \mathbf{C}_\mathrm{p} & \mathbf{C}_\mathrm{v} \end{bmatrix} \tag{6.6}$$

define a strictly dissipative realization $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{C}}, \mathbf{0}, \widetilde{\mathbf{E}})$ *of the second order system* (6.1)*, if the second order matrices* $\mathbf{M}, \mathbf{D}$*, and* $\mathbf{K}$ *are symmetric positive definite and*

$$0 < \gamma < \gamma^* := \lambda_{\min}\left[\mathbf{D}\left(\mathbf{M} + \tfrac{1}{4}\,\mathbf{D}\mathbf{K}^{-1}\mathbf{D}\right)^{-1}\right] \tag{6.7}$$

is fulfilled, where γ^* *is the smallest solution of the generalized eigenvalue problem*

$$\mathbf{D}\mathbf{v} = \lambda \cdot \left(\mathbf{M} + \tfrac{1}{4}\,\mathbf{D}\mathbf{K}^{-1}\mathbf{D}\right)\mathbf{v}, \qquad \lambda \in \mathbb{R},\ \mathbf{v} \in \mathbb{R}^{\widehat{N}} \setminus \{\mathbf{0}\}. \tag{6.8}$$

Although the inverse stiffness matrix appears in the above formula, it is not necessary to compute $\mathbf{K}^{-1}$ in order to determine $\left(\mathbf{M} + \frac{1}{4}\,\mathbf{D}\mathbf{K}^{-1}\mathbf{D}\right)\mathbf{v}$. Instead, one can perform a Cholesky decomposition of $\mathbf{K}$ and equip the sparse eigen-solver with an efficient routine to compute LSE solves. As we only need to find one extremal eigenvalue and the problem is real symmetric, it can be solved very easily with the help of a standard power method [136].

A possible implementation is given in Source 6.1. It configures the `eigs` command to solve the inverse problem of (6.8) and to exploit the realness and symmetry; the start vector is determined to avoid random influences (see [150] `eigs`). For the Butterfly Gyroscope, the computation of γ^* required only $1.1s$.

Source 6.1: Computation of γ^* in MATLAB

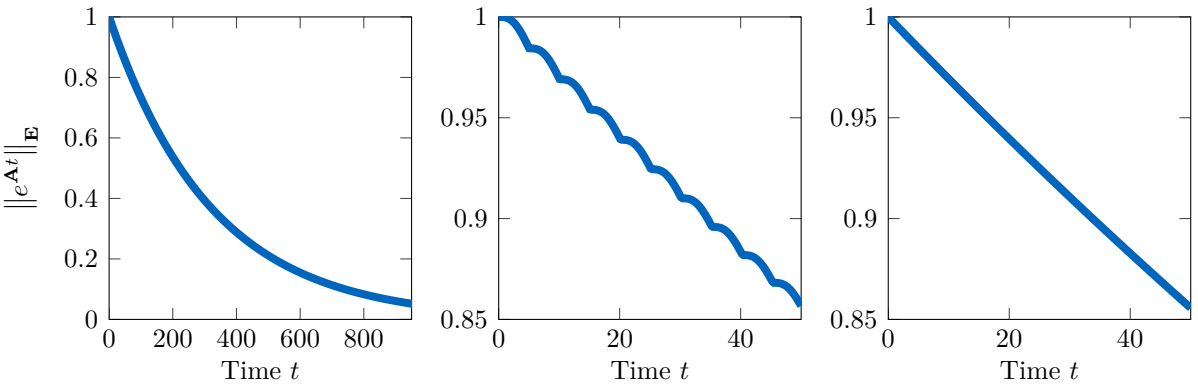
```
function gma = gamma_max(M,D,K)
% Maximal gamma for Strictly Dissipative Realization of 2nd Order System
%   Input:  M,D,K:      matrices of second order system
%   Output: gamma_max:  maximal gamma. The recommended choice for gamma
%                           in weakly damped systems is gamma/2.
%

[L_K,x,P_K] = chol(sparse(K));      % direct solver
if x, warning('Cholesky decomposition not successful.'); end
Kinv = @(x) P_K*(L_K\(L_K'\(P_K'*x)));
% Kinv = @(x) pcg(K,x,1e-8,20,K);       % iterative solver
gamma_fun   = @(x) (M*x + D*Kinv(D*x/4));
opts        = struct('issym', true, 'isreal', true, 'v0', diag(K));
gma         = 1/eigs(gamma_fun, size(D,1), sparse(D), 1, 'LA', opts);
```

Accordingly, we have found a convenient way how second order systems with positive definite mass, damping, and stiffness matrices can be expressed in strictly dissipative state space realizations, because any value of γ within the valid interval $]0;\gamma^*[$ delivers a realization with $\mu_{\mathbf{E}}(\mathbf{A}) < 0$.

For proof of concept, consider once more the ISS benchmark model we have already used in Section 2.2.3. In fact, it takes the form of the standard realization (6.4) with $\mathbf{R} = \mathbf{I}$ ($\mathbf{M} = \mathbf{I}$), which is why it is not dissipative. Setting $\mathbf{R} := \mathbf{K}$ changes the plot $\|e^{\mathbf{A}t}\|_{\mathbf{E}}$ from what we saw in Figure 2.1b) to a macroscopically smooth curve as in Figure 6.1a), which however shows horizontal tangents in microscopic scale (see Figure 6.1b)), and is therefore not *strictly* dissipative. Using realization (6.6) with $\gamma = \frac{1}{2}\gamma^*$ yields the perfectly smooth curve in Figure 6.1c).

Generalizations to singular mass matrices and to damping matrices with skew-symmetric components were presented in [123], but will be omitted in the following. The case of singular damping, however, remains an open problem.

With regard to model reduction, the knowledge of a strictly dissipative realization has two important consequences. It allows us to apply the error bounds of Chapter 5 (and benefit from the stability preservation property, cf. Lemma 2.4). But on the other hand, the dimension of the model is doubled and, in addition, sparsity of the new state space matrices (6.6) is compromised by the additional entries in comparison to the standard realization (6.4), which, of course, can have massive impact on the numerical effort.



a) Dissipative realization b) Dissipative realization c) Strictly dissipative realization

Figure 6.1.: Matrix Exponential of ISS Benchmark Model in Dissipative Realization

The remainder of this chapter therefore shows how the increase in numerical complexity can be inhibited and how the error bounds can be used efficiently.

To conclude this section, it is noted that there exists a generalization of the transformation (6.5). One can in fact introduce an additional parameter $\vartheta \in \mathbb{R}^+$ and replace $\mathbf{T}$ from (6.5) by

$$\mathbf{T} := \begin{bmatrix} \mathbf{I} & \gamma\mathbf{I} + \vartheta\mathbf{K}\mathbf{M}^{-1} \\ \gamma\mathbf{M}\mathbf{K}^{-1} + \vartheta\mathbf{I} & \mathbf{I} \end{bmatrix}, \tag{6.9}$$

which leads to state space matrices

$$\begin{aligned} \widetilde{\mathbf{A}} &= \begin{bmatrix} -(\gamma\,\mathbf{M} + \vartheta\,\mathbf{K})\mathbf{M}^{-1}\mathbf{K} & \mathbf{K} - (\gamma\,\mathbf{M} + \vartheta\,\mathbf{K})\mathbf{M}^{-1}\mathbf{D} \\ -\mathbf{K} & -\mathbf{D} + (\gamma\,\mathbf{M} + \vartheta\,\mathbf{K}) \end{bmatrix}, \\ \widetilde{\mathbf{E}} &= \begin{bmatrix} \mathbf{K} & \gamma\,\mathbf{M} + \vartheta\,\mathbf{K} \\ \gamma\,\mathbf{M} + \vartheta\,\mathbf{K} & \mathbf{M} \end{bmatrix}, \qquad \widetilde{\mathbf{B}} = \begin{bmatrix} (\gamma\,\mathbf{M} + \vartheta\,\mathbf{K})\mathbf{M}^{-1}\,\mathbf{F} \\ \mathbf{F} \end{bmatrix}. \end{aligned} \tag{6.10}$$

For simplicity, ϑ is set to zero in what follows.

6.3. Efficient Application of State Space Methods

We will now try to apply the methods presented in Chapters 4 and 5 to the reduction of second order systems, exploiting their particular structure. In a first step, we recall how KRYLOV subspaces for standard state space realizations can be computed efficiently. Due to invariance properties, these KRYLOV subspaces can be used directly for the reduction of the strictly dissipative state space model, so that the computational overhead due to the new realization is minimal. Finally, we focus on judicious ways to evaluate the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ error bounds.

6.3.1. Computation of Krylov Subspaces

In the following, we therefore recall the well-known fact that KRYLOV subspaces of models in the particular form (6.4) can be computed far more easily with a two-level approach than in the general unstructured case [54, 97, 137].

Consider, for instance, the SISO case or tangential interpolation as in Section 3.2.2. The columns of $\mathbf{V}$ are defined recursively by equations of the form

$$\left(\widetilde{\mathbf{A}} - \sigma\widetilde{\mathbf{E}}\right)\mathbf{v}_i = \widetilde{\mathbf{E}}\mathbf{v}_{i-1} \quad \Leftrightarrow \quad \left(\mathbf{A} - \sigma\mathbf{E}\right)\mathbf{v}_i = \mathbf{E}\mathbf{v}_{i-1},$$

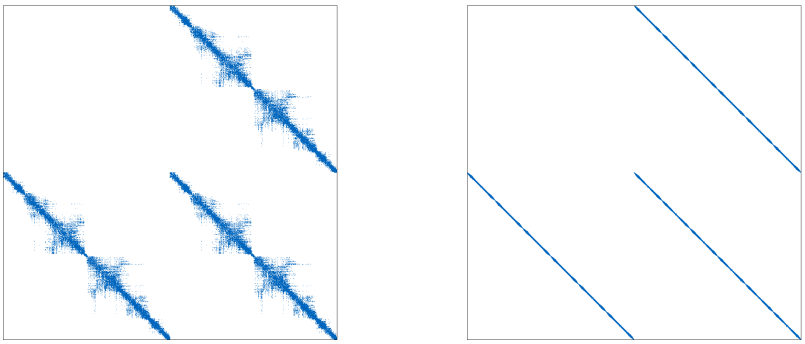
where for $i = 1$ the right hand side is replaced by the input vector $\widetilde{\mathbf{b}}$ or $\widetilde{\mathbf{B}}\mathbf{t}$ or by $\mathbf{b}$ or $\mathbf{B}\mathbf{t}$, respectively. To solve such an LSE, one might perform an LU-decomposition of $(\mathbf{A} - \sigma\mathbf{E})$; yet its dimension is $N = 2\widehat{N}$ and the decomposition suffers from many fill-ins due to the structure (see Figure 6.2), which drastically increases the demands on memory and time; iterative solvers experience difficulties as well.

Instead, for matrices $\mathbf{A}$ and $\mathbf{E}$ as in (6.4) with $\mathbf{R} = \mathbf{K}$, we can divide $\mathbf{v}_i$ into an upper ("position") and a lower ("velocity") component and obtain

$$\left(\begin{bmatrix} \mathbf{0} & \mathbf{K} \\ -\mathbf{K} & -\mathbf{D} \end{bmatrix} - \sigma \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix}\right) \begin{bmatrix} \mathbf{v}_{i,p} \\ \mathbf{v}_{i,v} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{i-1,p} \\ \mathbf{v}_{i-1,v} \end{bmatrix}. \tag{6.11}$$

From the first line, it follows that $\mathbf{v}_{i,v} = \sigma\mathbf{v}_{i,p} + \mathbf{K}^{-1}\mathbf{v}_{i-1,p}$, and thus

$$-\left(\mathbf{K} + \sigma\mathbf{D} + \sigma^2\mathbf{M}\right)\mathbf{v}_{i,p} = \mathbf{v}_{i-1,v} + (\mathbf{D} + \sigma\mathbf{M})\,\mathbf{K}^{-1}\mathbf{v}_{i-1,p}. \tag{6.12}$$



a) Butterfly Gyroscope b) Wineglass

Figure 6.2.: Sparsity Pattern of Matrix $\mathbf{A}$ in Standard Realization of Second Order Systems

Defining $\mathbf{h} := \mathbf{K}^{-1}\mathbf{v}_{i-1,p}$, the solution $\mathbf{v}_i = (\mathbf{A} - \sigma\mathbf{E})^{-1}\,\mathbf{v}_{i-1}$ is given by

$$\mathbf{v}_i = \begin{bmatrix} \mathbf{v}_{i,p} \\ \sigma \cdot \mathbf{v}_{i,p} + \mathbf{K}^{-1}\mathbf{v}_{i-1,p} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{i,p} \\ \sigma \cdot \mathbf{v}_{i,p} + \mathbf{h} \end{bmatrix}, \tag{6.13}$$

where both $\mathbf{h}$ and $\mathbf{v}_{i,p} = -\left(\mathbf{K} + \sigma\mathbf{D} + \sigma^2\mathbf{M}\right)^{-1}\left(\mathbf{v}_{i-1,v} + (\mathbf{D} + \sigma\mathbf{M})\,\mathbf{h}\right)$ solve symmetric linear systems of equations of dimension $\hat{N}$. For real σ (preserving positive definiteness), the two problems can be solved with a CHOLESKY decomposition of $\mathbf{K}$ and of $(\mathbf{K} + \sigma\mathbf{D} + \sigma^2\mathbf{M})$, respectively; for complex σ, an LU-decomposition is mandatory for the latter, increasing the effort slightly. Of course, indirect solvers may be applied as well (cf. Section 3.4). Note that for the particular case that the right hand side is given by $\mathbf{Bt}$ or $\mathbf{b}$, the computation simplifies even more as $\mathbf{h} = \mathbf{0}$; however, in the light of the CURE framework, where the structured input $\mathbf{B}$ is replaced by some $\mathbf{B}_\perp$ after the first reduction step, the general procedure was presented intentionally.

Anyway, the numerical effort is by far lower than for solving the problem in state space, so for both the standard and the strictly dissipative realization, the above advance delivers input KRYLOV subspaces efficiently.

An *output* KRYLOV subspace of the standard realization (6.4) can be found similarly. Here, the vector $\mathbf{w}_i$ solving $(\mathbf{A} - \sigma\mathbf{E})^T\,\mathbf{w}_i = \mathbf{E}^T\mathbf{w}_{i-1}$ is given by

$$\mathbf{w}_i = \begin{bmatrix} \mathbf{w}_{i,p} \\ \mathbf{w}_{i,v} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_{i,p} \\ -\sigma \cdot \mathbf{w}_{i,p} - \mathbf{K}^{-1}\mathbf{w}_{i-1,p} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_{i,p} \\ -\sigma \cdot \mathbf{w}_{i,p} - \mathbf{h}_W \end{bmatrix}, \tag{6.14}$$

with $\mathbf{h}_W := \mathbf{K}^{-1}\mathbf{w}_{i-1,p}$ and $\mathbf{w}_{i,p} = -\left(\mathbf{K} + \sigma\mathbf{D} + \sigma^2\mathbf{M}\right)^{-1}\left(\mathbf{w}_{i-1,v} + (\mathbf{D} + \sigma\mathbf{M})\,\mathbf{h}_W\right)$.

Source 6.2: Multipoint Tangential Rational Krylov for Second Order Systems

```
function [V,S_V,Crt,W,S_W,Brt] = TangentialKrylov2nd(M,D,K,B,C,s0,t_B,t_C)
% Two-sided Rational Krylov
%   Input:  M,D,K  : second order matrices;
%           B,C:     input and output matrix of standard state space model;
%           s0:      vector of shifts;
%           t_B,t_C: tangential directions
%   Output: A*V - E*V*S_V - B*Crt = 0,  W.'*A - S_W*W.'*E - Brt*C = 0
%

% initialization and preallocation
N_=size(M,1); n=length(s0); m=size(B,2); p=size(C,1); i=1;
V=zeros(2*N_,n); W=V; S_V=zeros(n,n); Crt=zeros(m,n); Brt=zeros(n,p); S_W=S_V;
[L_K,e,P_K] = chol(sparse(K));
if e, warning('Cholesky decomposition not successful.'); end
Kinv = @(x) P_K*(L_K\(L_K'\(P_K'*x)));

while i<=n
    s = s0(i);
    h   = Kinv(B(1:N_,:)*t_B(:,i));  h_W = Kinv(C(:,1:N_).'*t_C(i,:).');
    % compute new basis vectors
    if ~isreal(s)   % complex conjugated pair of shifts -> two new columns
        [L,U,P,Q,R] = lu(sparse(K+s*D+s^2*M));
        Vip = -Q*(U\(L\(P*(R\(B(N_+1:end,:)*t_B(:,i)+D*h+M*h*s)))));
        tempV = [Vip;s*Vip+h];
        Wip = Q*(U\(L\(P*(R\(C(:,N_+1:end).'*t_C(i,:).'-D*h_W-M*h_W*s)))));
        tempW = [Wip;-h_W-s*Wip];
        V(:,i:(i+1))    = [real(tempV), imag(tempV)];
        Crt(:,i:(i+1))  = [real(t_B(:,i)), imag(t_B(:,i))];
        S_V(i:(i+1),i:(i+1)) = [real(s), imag(s); -imag(s), real(s)];
        W(:,i:(i+1))    = [real(tempW), imag(tempW)];
        Brt(i:(i+1),:)  = [real(t_C(i,:)); imag(t_C(i,:))];
        S_W(i:(i+1),i:(i+1)) = [real(s), -imag(s); imag(s), real(s)];
        i = i+2;
    else            % real shift -> one new column
        [L,e,P] = chol(sparse(K+s*D+s^2*M));
        if e, warning('Cholesky decomposition not successful.'); end
        Vip    = -P*(L\(L'\(P'*(B(N_+1:end,:)*t_B(:,i)+D*h+M*h*s))));
        tempV  = [Vip;s*Vip+h];
        Wip    = P*(L\(L'\(P'*(C(:,N_+1:end).'*t_C(i,:).'-D*h_W-M*h_W*s))));
        tempW  = [Wip;-h_W-s*Wip];
        V(:,i) = real(tempV);  Crt(:,i) = real(t_B(:,i)); S_V(i,i) = s;
        W(:,i) = real(tempW);  Brt(i,:) = real(t_C(i,:)); S_W(i,i) = s;
        i = i+1;
    end
end
% orthogonalization
[V,S_V,Crt] = GramSchmidt(V,S_V,Crt);
[W,S_W,Brt] = GramSchmidt(W,S_W',Brt'); S_W=S_W.'; Brt=Brt.';
end
```

A MATLAB implementation of the KRYLOV subspaces for second order systems can be seen in Source 6.2. The code was applied to the Butterfly Gyroscope and compared to the generic code in Source 3.3. First, eight eigenvalues closest to zero were computed with the `eigs` command and mirrored with respect to the imaginary axis to create some shifts. Then, both routines Sources 3.3 and 6.2 were run. The second order code required less than 30% of the execution time of the standard code, without loss of precision. For eight purely real shifts the speed-up factor even amounted to more than five.

When the CHOLESKY and LU factors of $\mathbf{K}$ and $(\mathbf{K}+\sigma\mathbf{D}+\sigma^2\mathbf{M})$ cannot be computed any more due to shortage of RAM, the direct solvers can be replaced by iterative methods. This typically lasts longer, but requires far less memory. In Source 6.2, one could replace lines 13–15 by a preconditioned conjugate gradient method to solve LSEs with $\mathbf{K}$,

```
13  Kinv = @(x) pcg(K,x,1e-8,20,K);
```

lines 22, 23, and 25 by

```
22  Vip = bicg(K+s*D+s^2*M,B(N_+1:end,:)*t_B(:,i)+D*h+M*h*s,1e-8,20,K);
23  Wip = bicg(K+s*D+s^2*M,C(:,N_+1:end).'*t_C(i,:).'-D*h_W-M*h_W*s,1e-8,20,K);
```

and lines 35–37 and 39 by

```
35  Vip = -pcg(K+s*D+s^2*M,B(N_+1:end,:)*t_B(:,i)+D*h+M*h*s,1e-8,20,K);
36  Wip = pcg(K+s*D+s^2*M,C(:,N_+1:end).'*t_C(i,:).'-D*h_W-M*h_W*s,1e-8,20,K);
```

6.3.2. Invariance Properties in Sylvester Model Reduction

The strictly dissipative realization (6.6) of a second order system arises from the standard state space formulation (6.4) by pre-multiplication with a regular matrix $\mathbf{T}$. As a direct consequence of Lemma 3.2, any solution of an *input* Sylvester equation (3.3)—in particular: the basis of a rational input KRYLOV subspace—carries over to the rather complicated looking dissipative realization. Accordingly, we can follow the procedure presented in the previous section and need not explicitly solve complicated LSEs of the type $(\widetilde{\mathbf{A}}-\sigma\widetilde{\mathbf{E}})\mathbf{v}=\widetilde{\mathbf{b}}$.

Unfortunately, the *output* KRYLOV subspace is not equal to the one of the strictly dissipative realization (6.6), but it must be multiplied by $\mathbf{T}^{-1}$ from the right to solve

$$\widetilde{\mathbf{W}}^T\widetilde{\mathbf{A}}+(-\mathbf{S}_W)\widetilde{\mathbf{W}}^T\widetilde{\mathbf{E}}+(-\widetilde{\mathbf{B}}_r)\mathbf{C}=\mathbf{0}, \tag{6.15}$$

i. e. $\widetilde{\mathbf{W}}^T=\mathbf{W}^T\mathbf{T}^{-1}$ is required in theory. In two-sided reduction, however, the resulting

ROM is the same as for the standard realization, because its matrices

$$\widetilde{\mathbf{W}}^T\widetilde{\mathbf{E}} = \mathbf{W}^T\mathbf{E}, \quad \widetilde{\mathbf{W}}^T\widetilde{\mathbf{A}} = \mathbf{W}^T\mathbf{A}, \quad \text{and} \quad \widetilde{\mathbf{W}}^T\widetilde{\mathbf{B}} = \mathbf{W}^T\mathbf{B}$$

remain unchanged as well as $\mathbf{V}$, so both HFMs deliver the same ROM.

Also the result of the PORK algorithm, which determines all degrees of freedom of the ROM, is invariant under the transformation. If $\mathbf{V}$ spans an input Krylov subspace and solves (3.3)—no matter, whether for the matrices $\mathbf{A}$, $\mathbf{E}$, $\mathbf{B}$ as in (6.4) or for the transformed $\widetilde{\mathbf{A}}$, $\widetilde{\mathbf{E}}$, $\widetilde{\mathbf{B}}$ in (6.6)—the reduced order matrices (3.31) are given independently of the high order matrices except for $\mathbf{C}$, which is not affected by the change of realization. In the dual case, if $\mathbf{W}$ spans an output Krylov subspace and solves (3.4) for the standard realization, then $\widetilde{\mathbf{W}}^T = \mathbf{W}^T\mathbf{T}^{-1}$ solves (6.15). As the reduced order matrices $\mathbf{A}_r$, $\mathbf{E}_r$, and $\mathbf{C}_r$ in (3.39) only depend on $\mathbf{S}_W$ and $\widetilde{\mathbf{B}}_r$, they remain unchanged by the transformation; only $\mathbf{B}_r$ is influenced by the left-handed projection matrix $\widetilde{\mathbf{W}}$, yet we can easily see that $\mathbf{B}_r = \widetilde{\mathbf{W}}^T\widetilde{\mathbf{B}} = \mathbf{W}^T\mathbf{T}^{-1}\mathbf{T}\mathbf{B} = \mathbf{W}^T\mathbf{B}$ is also the same as when applying PORK to the standard realization.

Only in one-sided reduction, where $\widetilde{\mathbf{V}} := \widetilde{\mathbf{W}}$ is defined or vice versa, the ROM does change due to the transformation $\mathbf{T}$. This also is to be expected as we know that orthogonal projection of a strictly dissipative model always yields an asymptotically stable ROM, which is not generally the case. In fact, choosing $\mathbf{R} = \mathbf{I}_{\widehat{N}}$ or arbitrarily in (6.4) can easily yield an unstable ROM if the basis of an input Krylov subspace is used both for $\mathbf{V}$ and $\mathbf{W} := \mathbf{V}$. If the same matrix is applied from both sides to the transformed realization, the ROM is always stable.

6.3.3. Error Decomposition

According to the previous subsection, due to invariance properties the actual Krylov subspaces of the strictly dissipative realization (6.6) do not have to be computed by solving linear systems of equations in $N = 2\widehat{N}$ dimensional state space. In fact, for two-sided reduction, the ROM is not even affected by the change of realization, but can be found with the help of the standard realization (6.4) according to Section 6.3.1.

The transformation does, however, affect the factorized formulation of the error model

presented in Section 4.2. Let $\mathbf{V}$ solve Sylvester equation (3.3) or, equivalently,

$$\widetilde{\mathbf{A}}\mathbf{V} - \widetilde{\mathbf{E}}\mathbf{V}\mathbf{S} - \widetilde{\mathbf{B}}\widetilde{\mathbf{C}}_r = \mathbf{0}. \tag{6.16}$$

Then, the factorization (4.6) of the error model holds true with $\widetilde{\mathbf{E}}$ instead of $\mathbf{E}$, $\widetilde{\mathbf{A}}$ instead of $\mathbf{A}$ and

$$\widetilde{\mathbf{B}}_\perp = \widetilde{\mathbf{B}} - \widetilde{\mathbf{E}}\mathbf{V}\mathbf{E}_r^{-1}\mathbf{B}_r = \mathbf{T}\left(\mathbf{B} - \mathbf{E}\mathbf{V}\mathbf{E}_r^{-1}\mathbf{B}_r\right) = \mathbf{T}\mathbf{B}_\perp$$

instead of $\mathbf{B}_\perp$, with $\mathbf{C}$ unchanged. If, on the other hand, $\mathbf{W}$ solves the Sylvester equation (6.15), the output type error factorization (4.7) holds, again with $\widetilde{\mathbf{E}}$ instead of $\mathbf{E}$, $\widetilde{\mathbf{A}}$ instead of $\mathbf{A}$, $\widetilde{\mathbf{B}}$ instead of $\mathbf{B}$, and with $\mathbf{C}_\perp$ unchanged.

Either way, the transformation of the HFM affects the realization of the high order model $\mathbf{G}_\perp(s)$ in the same way: all matrices of the ODE are pre-multiplied by the matrix $\mathbf{T}$, but no further changes occur.

To conclude, to make use of the advantages of the strictly dissipative realization, in two-sided reduction (including PORK) it is often not necessary to actually work with the matrices in (6.6). Instead, one can perform MOR using the standard realization and transform the $\mathbf{G}_\perp(s)$ model in the error decomposition to strictly dissipative realization *afterwards* in order to evaluate the error bounds. In fact, γ does not even have to be fixed *a priori*, but can be chosen or modified *a posteriori* to optimize the error bounds from Chapter 5.

6.3.4. Evaluation of $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Error Bounds

To compute the upper bounds on the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ error norm presented in Chapter 5, one must solve LSEs of dimension $N = 2\hat{N}$. The matrices which represent the LSEs—$\widetilde{\mathbf{E}}$ and $\mathbf{S}$, respectively—are symmetric, positive definite, and sparse, but have significantly more entries and less convenient structure than the matrices of the second order system (see Figure 6.2). For that reason, the direct evaluation of the bounds in state space may be unfeasible (just like the direct computation of Krylov subspaces, see above), but can be avoided in the following way.

Proposition 6.1. *Given $\gamma \in \mathbf{R}^+$ fulfilling (6.7) and a realization $(\mathbf{A}, \mathbf{B}_\perp, \mathbf{C}_\perp, \mathbf{0}, \mathbf{E})$ with $\mathbf{E} = \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix}$ and $\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{K} \\ -\mathbf{K} & -\mathbf{D} \end{bmatrix}$, the respective expressions of the $\mathcal{H}_2$ error bound*

evaluated for the strictly dissipative realization (6.6) *read*

$$\widetilde{\mathbf{B}}_\perp^T \widetilde{\mathbf{E}}^{-1} \widetilde{\mathbf{B}}_\perp = \mathbf{B}_\perp^T \begin{bmatrix} \mathbf{K}^{-1} & \gamma \mathbf{K}^{-1} \\ \gamma \mathbf{K}^{-1} & \mathbf{M}^{-1} \end{bmatrix} \mathbf{B}_\perp \qquad \textit{and} \tag{6.17}$$

$$\widetilde{\mathbf{C}}_\perp \widetilde{\mathbf{E}}^{-1} \widetilde{\mathbf{C}}_\perp^T = \mathbf{C}_\perp \begin{bmatrix} (\mathbf{K} - \gamma^2 \mathbf{M})^{-1} & -\gamma (\mathbf{K} - \gamma^2 \mathbf{M})^{-1} \\ -\gamma (\mathbf{K} - \gamma^2 \mathbf{M})^{-1} & \mathbf{M}^{-1} + \gamma^2 (\mathbf{K} - \gamma^2 \mathbf{M})^{-1} \end{bmatrix} \mathbf{C}_\perp^T. \tag{6.18}$$

Proof. The proof for the first term is straightforward.

$$\widetilde{\mathbf{B}}_\perp^T \widetilde{\mathbf{E}}^{-1} \widetilde{\mathbf{B}}_\perp = \mathbf{B}_\perp^T \mathbf{T}^T \mathbf{E}^{-1} \mathbf{T}^{-1} \mathbf{T} \mathbf{B}_\perp = \mathbf{B}_\perp^T \begin{bmatrix} \mathbf{I} & \gamma \mathbf{K}^{-1} \mathbf{M} \\ \gamma \mathbf{I} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{K}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^{-1} \end{bmatrix} \mathbf{B}_\perp.$$

For the second expression, we need to invert the transformation matrix $\mathbf{T}$ from (6.5):

$$\begin{aligned} &\widetilde{\mathbf{C}}_\perp \widetilde{\mathbf{E}}^{-1} \widetilde{\mathbf{C}}_\perp^T = \mathbf{C}_\perp \mathbf{E}^{-1} \mathbf{T}^{-1} \mathbf{C}_\perp^T = \\ &= \mathbf{C}_\perp \begin{bmatrix} \mathbf{K}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^{-1} \end{bmatrix} \cdot \begin{bmatrix} (\mathbf{I} - \gamma^2 \mathbf{M} \mathbf{K}^{-1})^{-1} & -\gamma (\mathbf{I} - \gamma^2 \mathbf{M} \mathbf{K}^{-1})^{-1} \\ -\gamma \mathbf{M} \mathbf{K}^{-1} (\mathbf{I} - \gamma^2 \mathbf{M} \mathbf{K}^{-1})^{-1} & \mathbf{I} + \gamma \mathbf{M} \mathbf{K}^{-1} (\mathbf{I} - \gamma^2 \mathbf{M} \mathbf{K}^{-1})^{-1} \end{bmatrix} \cdot \mathbf{C}_\perp^T. \end{aligned}$$

□

If Cholesky factors of $\mathbf{M}$, $\mathbf{K}$, and $(\mathbf{K} - \gamma^2 \mathbf{M})$ are available, the evaluation works out particularly fast. Otherwise, the new formulations still allows to compute the bound by solving sparse symmetric $\widehat{N}$-dimensional LSEs only, which can be also be achieved by iterative solvers.

Now let us turn to the upper bound on the $\mathcal{H}_\infty$ norm. Unfortunately, there is no closed formulation as in the $\mathcal{H}_2$ case, but the explicit solution of $2\widehat{N}$-dimensional LSEs can still be avoided.

Let $\mathbf{G}_\perp(s)$ be given in a standard state space realization $(\mathbf{A}, \mathbf{B}_\perp, \mathbf{C}_\perp, \mathbf{0}, \mathbf{E})$, and $\gamma \in \mathbf{R}^+$ fulfilling (6.7). Then, the upper bound (5.26) on the $\mathcal{H}_\infty$-norm reads

$$\begin{aligned} \left\| \mathbf{G}_\perp \right\|_{\mathcal{H}_\infty} &\leq \left\| \widetilde{\mathbf{C}}_\perp \mathbf{S}^{-1} \widetilde{\mathbf{B}}_\perp \right\|_2 + \sqrt{\left\| \widetilde{\mathbf{B}}_\perp^T \mathbf{S}^{-1} \widetilde{\mathbf{B}}_\perp \right\|_2 \left\| \widetilde{\mathbf{C}}_\perp \mathbf{S}^{-1} \widetilde{\mathbf{C}}_\perp^T \right\|_2} \\ &= \left\| \mathbf{C}_\perp \mathbf{S}^{-1} (\mathbf{T} \mathbf{B}_\perp) \right\|_2 + \sqrt{\left\| (\mathbf{T} \mathbf{B}_\perp)^T \mathbf{S}^{-1} (\mathbf{T} \mathbf{B}_\perp) \right\|_2 \left\| \mathbf{C}_\perp \mathbf{S}^{-1} \mathbf{C}_\perp^T \right\|_2}. \end{aligned}$$

For its evaluation we firstly need to find $\widetilde{\mathbf{B}}_\perp = \mathbf{T} \mathbf{B}_\perp$, and secondly we must solve LSEs including the $2\widehat{N}$-dimensional matrix $\mathbf{S}$. To this end, we write

$$\mathbf{B}_\perp = \begin{bmatrix} \mathbf{B}_{\perp,p} \\ \mathbf{B}_{\perp,v} \end{bmatrix}, \quad \mathbf{C}_\perp = \begin{bmatrix} \mathbf{C}_{\perp,p} & \mathbf{C}_{\perp,v} \end{bmatrix}. \tag{6.19}$$

Source 6.3: Computation of Upper Bound on $\mathcal{H}_2$ Norm for Second Order Systems

```
function bndH2 = BoundH22nd(L_K,P_K,L_M,P_M,L_KM,P_KM,mu,gma,B,C)
% Upper bound on H2 norm of Second Order System
%   Input:  L_K,P_K; L_M,P_M: Cholesky factor of mass and stiffness matrix
%           L_KM,P_KM:        Cholesky factor of K-gma^2*M
%           mu:               Generalized Spectral Abscissa, mu<0!
%           gma:              transformation parameter, 0<gma<gamma_max!
%           B,C:              Input and output matrices of standard state space realization
%   Output: bndH2:            Upper bound on H2 norm
%

N_ = size(B,1)/2;
% compute B'*inv(E)*B
B_1 = L_K'\(P_K'*B(1:N_,:));  B_2 = L_K'\(P_K'*B(N_+1:end,:));
tmp = L_M'\(P_M'*B(N_+1:end,:));
k_3 = norm(full(B_1'*B_1 + 2*gma*B_1'*B_2 + tmp'*tmp), 'fro');

% compute C*inv(E)*C'
C_1 = C(:,1:N_)*P_KM/L_KM;     C_2 = -gma*C(:,N_+1:end)*P_KM/L_KM;
tmp = C(:,N_+1:end)*P_M/L_M;
k_2 = norm(full(C_1*C_1' + 2*C_2*C_1' + tmp*tmp' + C_2*C_2'));

% compute bound for P_hat=0
bndH2 = sqrt(k_3*k_2/(-2*mu));
```

Then, for the first task, we note that the computation of

$$\widetilde{\mathbf{B}}_\perp = \begin{bmatrix} \widetilde{\mathbf{B}}_{\perp,p} \\ \widetilde{\mathbf{B}}_{\perp,v} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \gamma\mathbf{I} \\ \gamma\mathbf{M}\mathbf{K}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{B}_{\perp,p} \\ \mathbf{B}_{\perp,v} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{\perp,p} + \gamma\mathbf{B}_{\perp,v} \\ \gamma\mathbf{M}\mathbf{K}^{-1}\mathbf{B}_{\perp,v} + \mathbf{B}_{\perp,v} \end{bmatrix} \tag{6.20}$$

only requires solutions of $\widehat{N}$-dimensional LSEs to find $\mathbf{K}^{-1}\mathbf{B}_{\perp,v}$, so $\widetilde{\mathbf{B}}_\perp$ can be found quite easily. Secondly, we must solve for $\mathbf{R} := \mathbf{S}^{-1}\widetilde{\mathbf{B}}_\perp$, which can be accomplished with the Schur complement of $\mathbf{S}$.

$$\begin{aligned} &\begin{bmatrix} 2\gamma\mathbf{K} & \gamma\mathbf{D} \\ \gamma\mathbf{D} & 2\mathbf{D} - 2\gamma\mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{R}_p \\ \mathbf{R}_v \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{B}}_{\perp,p} \\ \widetilde{\mathbf{B}}_{\perp,v} \end{bmatrix} \\ \Leftrightarrow \quad &\begin{cases} \text{I.} \quad (2\mathbf{D} - 2\gamma\mathbf{M} - \frac{\gamma}{2}\mathbf{D}\mathbf{K}^{-1}\mathbf{D})\mathbf{R}_v = \widetilde{\mathbf{B}}_{\perp,v} - \frac{1}{2}\mathbf{D}\mathbf{K}^{-1}\widetilde{\mathbf{B}}_{\perp,p}, \\ \text{II.} \quad \mathbf{R}_p = \frac{1}{2\gamma}\mathbf{K}^{-1}\widetilde{\mathbf{B}}_{\perp,p} - \gamma\mathbf{D}\mathbf{R}_v. \end{cases} \end{aligned} \tag{6.21}$$

While equation II. can again be solved in a straightforward way, the first equation I. contains a term $\mathbf{D}\mathbf{K}^{-1}\mathbf{D}$ which cannot be factored out. It turns out, however, that with the help of iterative solvers, this equation can indeed be solved; in all considered cases, convergence occurred when $\mathbf{D}$ was used as preconditioner. Note that $\mathbf{S}^{-1}\mathbf{C}_\perp^T$ can be computed similarly.

Source 6.4: Computation of Upper Bound on $\mathcal{H}_\infty$ Norm for Second Order Systems

```
function bndHinf = BoundHinf2nd(M,D,K,gma,B,C,L_K,P_K)
% Upper bound on H-infinity norm of second order system
%   Input:  M,D,K:   Mass, damping, and stiffness matrix
%           gma:     transformation parameter, 0<gma<gamma_max!
%           B,C:     Input and output matrices of standard state space realization
%           L_K,P_K: Cholesky factor and pivoting matrix of K
%   Output: bndHinf: Upper bound
%

N_ = size(B,1)/2;
Kinv = @(x) P_K*(L_K\(L_K'\(P_K'*x)));  % direct solver
% Kinv = @(x) pcg(K,x,1e-8,10,K);       % iterative solver

% compute Btilde (of strictly dissipative realization)
Bt = [B(1:N_,:)+gma*B(N_+1:end,:); gma*M*Kinv(B(1:N_,:))+B(N_+1:end,:)];
% Schur complement of S (anonymous function for PCG)
S22 = @(x) D*x*2 - M*x*2*gma - D*Kinv(D*x)*gma/2;

% find inv(S)*Btilde by Preconditioned Conjugate Gradients
tmp = Bt(N_+1:end,:) - D*(Kinv(Bt(1:N_,:)))/2;
SinvB2 = pcg(S22,tmp,1e-8,10,D,[]);
SinvB1 = Kinv(Bt(1:N_,:) - D*SinvB2*gma )/2/gma;

% find inv(S)*C' by Preconditioned Conjugate Gradients
tmp = C(:,N_+1:end)' - D*(Kinv( C(:,1:N_)'))/2;
SinvC2 = pcg(S22,tmp,1e-8,10,D,[]);
SinvC1 = Kinv(C(:,1:N_)' - D*SinvC2*gma )/2/gma;

bndHinf = norm(SinvB1'*C(:,1:N_)' + SinvB2'*C(:,N_+1:end)') +  ...
    sqrt(SinvB1'*Bt(1:N_,:) + SinvB2'*Bt(N_+1:end,:)) * ...
    sqrt(SinvC1'*C(:,1:N_)' + SinvC2'*C(:,N_+1:end)');
end
```

An implementation for SISO systems is given in Source 6.4. For the Butterfly Gyroscope, its execution required 2.9 seconds; in comparison, the state space based source Source 5.2, which uses a CHOLESKY factor of $\mathbf{S}$, lasted only 0.83 sec, of which 0.62 sec fell upon the CHOLESKY decomposition of $\mathbf{S}$. However, the peak amount of memory required by Source 6.4 was only 5.3 MB compared to 77 MB for the CHOLESKY factor. Of course, within this scale of model dimension, memory is not the limiting factor.

For the Wineglass model (first output), on the other hand, Source 5.2 loaded 1.7 GB of memory, while Source 6.4 required only 0.1 GB peak memory. The calculation of the CHOLESKY factor lasted 26 sec, the evaluation of the bound in state space additional 13 sec; Source 6.4 took 57 sec. In both methods, a CHOLESKY factorization of $\mathbf{K}$ was

performed *a priori*; it lasted only 2.4 sec, but required 0.4 GB. For even larger models, when the memory demands even of this $\widehat{N}$-dimensional triangular factors exceed the available memory, one must also do without direct methods to solve LSEs of $\mathbf{K}$; Source 6.4 can then be modified accordingly by changing the comments of lines 11 and 12 (cf. end of 6.3.1). In this case, the computation of the bound lasted 210 sec, but required less than 21 MB (!) additional memory.

To conclude, direct methods are typically superior with regard to computational time. With increasing model complexity, the use of indirect methods becomes mandatory due to excessive storage requirements of direct methods. Note, by the way, that in both the examples the relative difference of the error bound values amounted to less than $2 \cdot 10^{-11}$.

6.3.5. Computation of Generalized Spectral Abscissa

To compute the generalized spectral abscissa μ, Source 6.5 can be used instead of the state space based code Source 2.1. As we know that $\mu < 0$, we can spare the effort to test positive definiteness of $-\mathbf{A} - \mathbf{A}^H$ via a CHOLESKY decomposition, cf. Section 2.2.4.

In fact, for the benchmark model of the Butterfly Gyroscope, calculating μ for some given γ lasted $3.5s$—less than half the time as in the state space based formulation, cf. Table 2.1. If the CHOLESKY factorization is no more feasible, the solution of LSEs must be performed by iterative solvers as described above (cf. (6.21)).

Source 6.5: Computation of $\mu_{\widetilde{\mathbf{E}}}(\widetilde{\mathbf{A}})$ in MATLAB

```
1  function mu = SpectralAbscissa2nd(M,D,K,gamma)
2  % Compute mu of second order system in strictly dissipative realization
3  %   Input:  M,D,K,F:    matrices of second order system
4  %           gamma:      transformation parameter, 0<gamma<gamma_max!
5  %   Output: mu:         generalized spectral abscissa
6  %
7
8  E    = [K, gamma*M; gamma*M, M];
9  symA = [-gamma*K, -gamma/2*D; -gamma/2*D, -D+gamma*M];
10
11 p    = 20;      % number of Lanczos vectors
12 tol  = 1e-10;   % convergence tolerance
13
14 opts = struct('issym',true, 'isreal',true, 'p', p, 'tol',tol, 'v0', diag(E));
15 mu   = eigs(symA, E, 1, 0, opts);
```

6.3.6. Dependency of the Error Bounds on γ

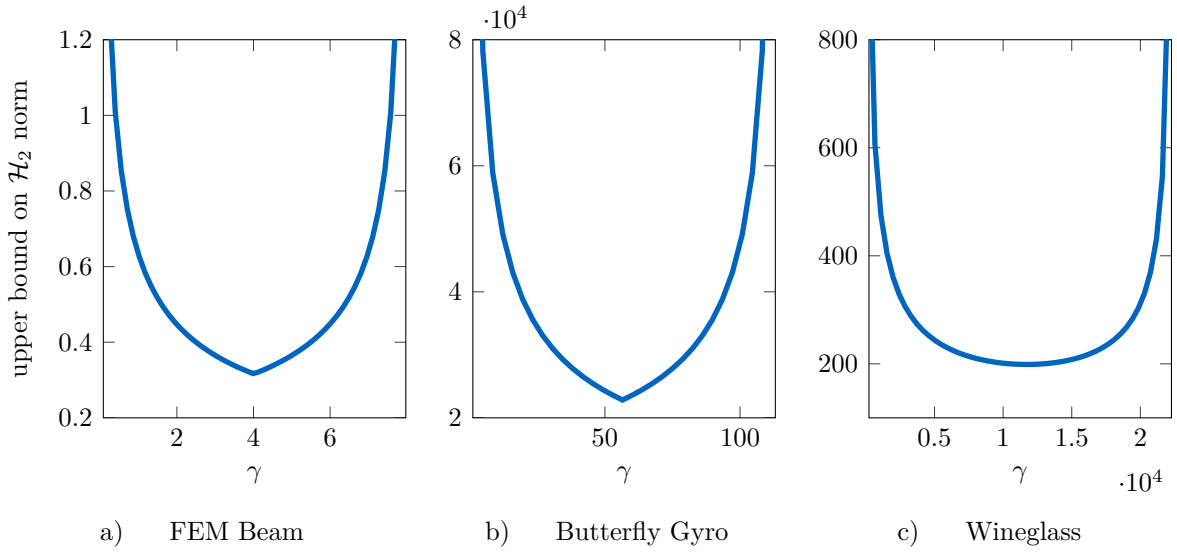
The global error bounds presented in Chapter 5 hold true for arbitrary strictly dissipative state space models. As we have seen in Section 6.2, however, any real number γ fulfilling (6.7) defines such a realization of a second order system. Therefore, the question is *where* in the valid interval γ should be chosen to obtain the tightest error bounds.

In [123, 127] it was shown that the dependency $\mu(\gamma)$ of the generalized matrix measure $\mu_{\mathbf{E}}(\mathbf{A})$ on γ resembles shifted absolute value functions and has its minimum very close to $\frac{1}{2}\gamma$ for lightly damped systems. Unfortunately, both the $\mathcal{H}_2$ and the $\mathcal{H}_\infty$ bounds depend on γ in a more complicated way. Although no substantiated analytical propositions on their shape and the location of their minima could be found so far, a definite trend is evident and presented in the following with the help of numerical examples. For simplicity, we simply consider the upper bound on the norm of $\mathbf{G}(s)$, not of an error model $\mathbf{G}_\perp(s)$.

Figure 6.3 shows simulation results for the FEM beam, the Butterfly Gyro, and the Wineglass model. One can see that in all three cases the upper bound on the $\mathcal{H}_2$ norm is U- or V-shaped and more or less symmetric; for $\gamma \to 0$ and $\gamma \to \gamma^*$, the bounds tend to infinity. The global minimum is reached for $\gamma \approx \frac{1}{2}\gamma^*$. This form has in fact been observed for all considered cases.

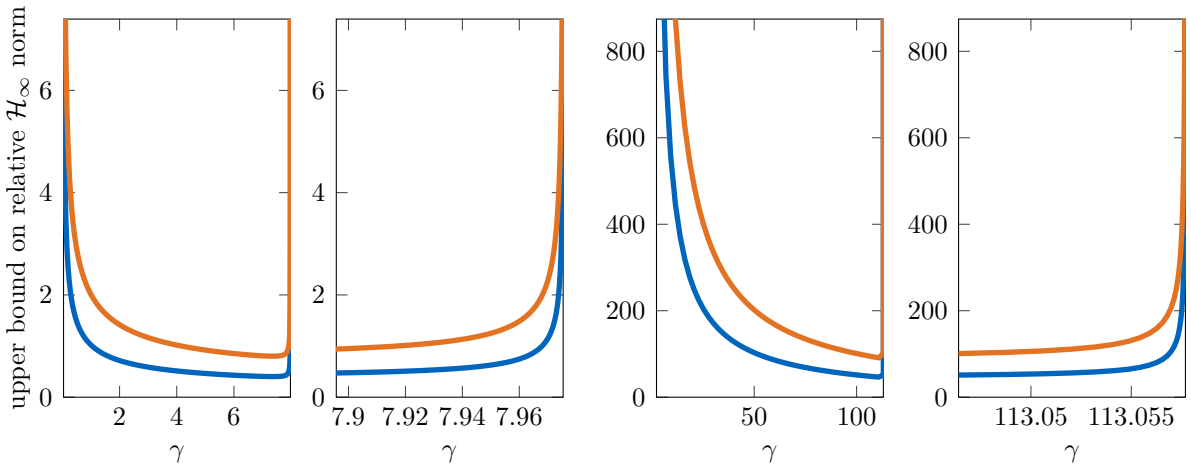
Results for the upper bounds on the $\mathcal{H}_\infty$ norm can be seen in Figure 6.4. The orange curve is the simpler upper bound (5.32), the tighter bound (5.26) is depicted in blue. Both tend to infinity at the boundary, yet typically in a very abrupt way on one of the two sides.

Accordingly, it seems advisable to choose γ somewhere in the middle of the valid interval in any case. The $\mathcal{H}_2$ bound thus becomes almost as tight as possible, and the $\mathcal{H}_\infty$ bounds are also not far from their optimal value.



a) FEM Beam b) Butterfly Gyro c) Wineglass

Figure 6.3.: Upper Bounds on $\mathcal{H}_2$ Norm of Second Order System over γ



a) FEM Beam b) Butterfly Gyro

Figure 6.4.: Upper Bound on $\mathcal{H}_\infty$ Norm of Second Order System over γ

7. Numerical Examples

"Hic Rhodus! Hic salta!"

— Aesopus

The numerous methods presented in the preceding chapters will now be applied to selected benchmark models in order to present their behavior in practical situations.

7.1. Spiral Inductor

We start with the PEEC model of a spiral inductor of order $N = 1434$, see Section 1.2.6. It is a strictly dissipative SISO system with generalized spectral abscissa $\mu \approx -1.4e7$.

Performing ten iterations of the model function based MESPARK algorithm during the cumulative reduction (CURE) scheme leads to the blue curve of the $\mathcal{H}_\infty$ and $\mathcal{H}_2$ error bounds in Figure 7.1. As the bounds—in particular, the $\mathcal{H}_2$ bound—decay only hesitantly, the cost functional in MESPARK is replaced by $\mathcal{J}_{\mathcal{H}_2}$ as defined in Section 5.6.1. This changes the course of the error bounds to the values depicted in orange. Both error bounds decay very fast now, and the final ROM of order $n = 20$ is guaranteed to imply relative errors of about $1 \cdot 10^{-4}$ or less. In fact, the $\mathcal{H}_2$ error amounts to $\overline{\epsilon_{\mathcal{H}_2}} \approx 1.4 \cdot 10^{-5}$, so the overestimation is below ten.

A verification of the actual physical quantities of interest is provided in Figure 7.2. One can see that both resistance and inductance are perfectly approximated by the ROM resulting from the latter algorithm, which surprisingly outperforms the standard MESPARK version, whose cost functional aims at minimizing the true error norm. For some reasons, however, probably because of unsuitable initial values, the standard MESPARK selects inferior local minima, and requires more iterations to reach similar approximation quality.

Both algorithms required a total of about 100 real LU decompositions, as in each of the ten CURE steps, MESPARK needed four steps to converge.

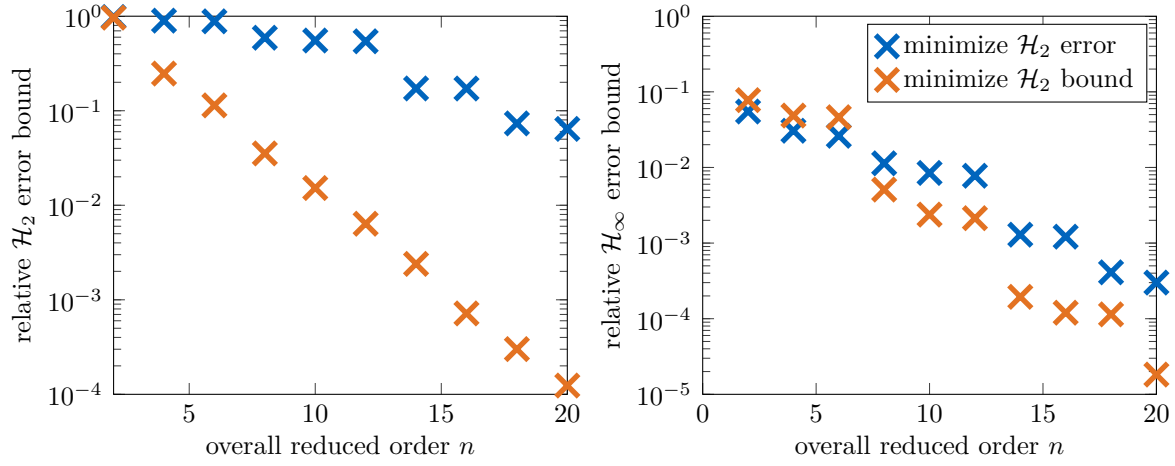


Figure 7.1.: Error Bounds during CURE of Spiral Inductor

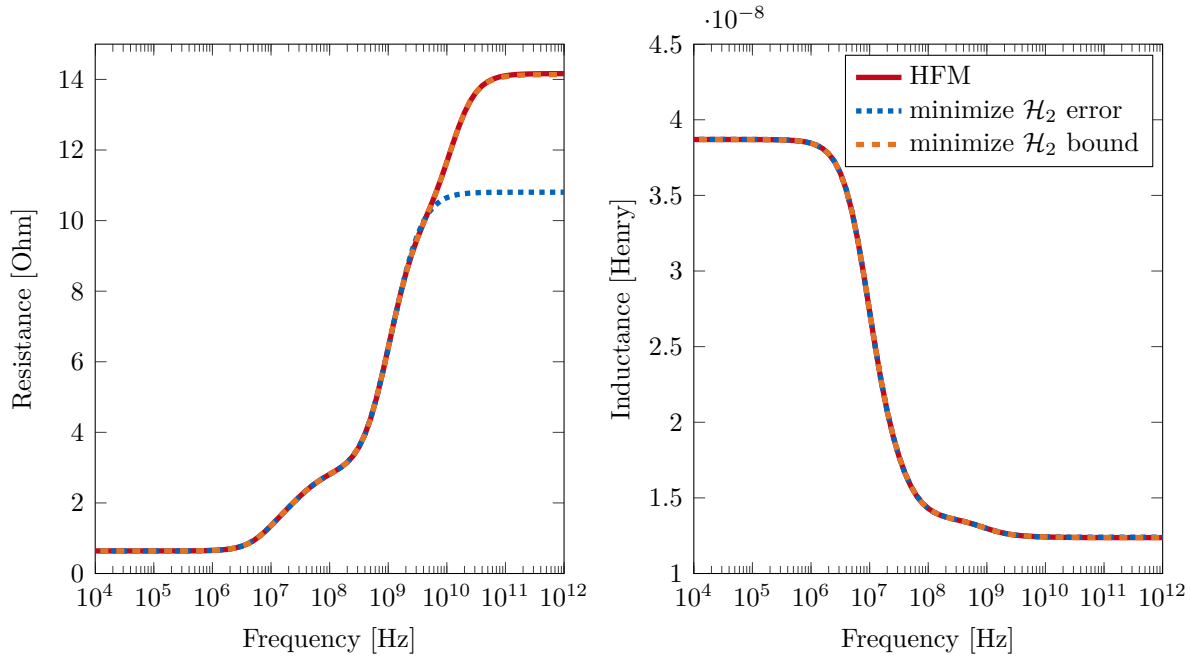


Figure 7.2.: Resistance and Inductance of Spiral Inductor

7.2. Flow Meter

Next, we consider the Flow Meter (v=0), whose order is $N = 9669$. We run the CURE framework and perform 30 reduction steps towards order $n_i = 2$. For shift selection, we use IRKA, orthogonal IRKA ($\mathbf{W} := \mathbf{V}$), $\mathcal{H}_\infty$-IRKA ($\mathbf{W} := \mathbf{S}^{-1}\mathbf{E}\mathbf{V}$), and MESPARK minimizing the $\mathcal{H}_2$ bound cost function $\mathcal{J}_{\mathcal{H}_2}$. The standard MESPARK algorithm cannot be used as the system has five outputs. We monitor the error bounds, setting $\widehat{\mathbf{P}} = \widehat{\mathbf{Q}} = \mathbf{0}$ in the $\mathcal{H}_2$ upper bound. Results are given in Figure 7.3.

This time, even $\mathcal{H}_2$ model reduction by (standard) IRKA yields a decay of the error bounds, but it is obvious that its derivates accomplish significantly faster decrease in their respective bound. MESPARK compares well to all other methods, and what is more, only required about four steps per iteration (in each of which an LU decomposition was performed), while orthogonal IRKA, for instance, required an average of 18 steps to converge. Stopping orthogonal IRKA after four iterations (implying a similar number of LU decompositions as MESPARK) has a massive impact on the performance; the error bounds after 30 steps are then orders of magnitude higher than in Figure 7.3.

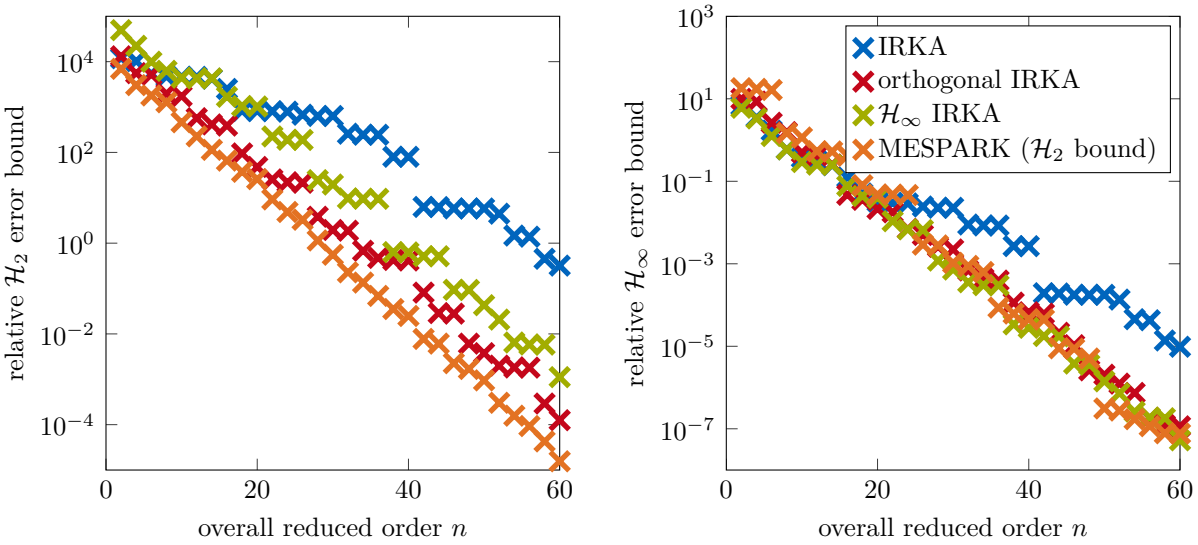
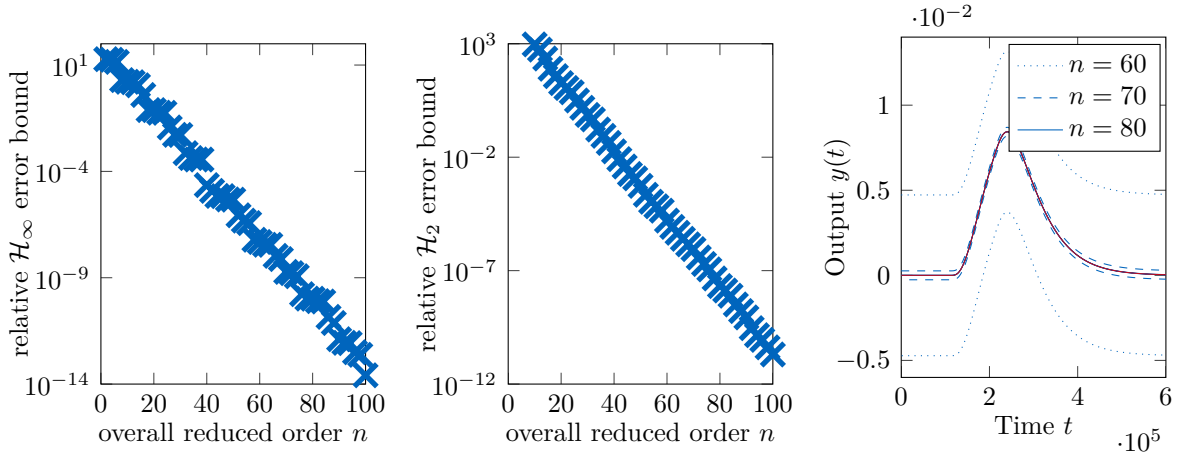


Figure 7.3.: Error Bounds during CURE of Flow Meter (v=0)

7.3. Steel Profile

We turn to the Steel Profile from Section 1.2.6. It is strictly dissipative, but has $m = 7$ inputs and $p = 6$ outputs, so it is truly MIMO and one cannot benefit from the convenient SIMO or MISO special case as above.

As a start, we therefore focus on the SISO case, and only consider the transfer behavior from the first input to the first output of the $N = 1357$ version of the model. In this scenario, CUREd MESPARK with the $\mathcal{J}_{\mathcal{H}_2}$ cost functional leads to very rapid decay of both the $\mathcal{H}_\infty$ and the $\mathcal{H}_2$ error bound, even though $\hat{\mathbf{P}} = \hat{\mathbf{Q}} = \mathbf{0}$; see Figure 7.4. To demonstrate the time domain envelope derived in Section 5.3.4, a rectangular bang-bang signal $u_1(t) = \sigma(t - 1e5) - \sigma(t - 2e5)$ is applied to the system. Both the HFM and several ROMs are simulated and the corresponding envelopes in time domain are evaluated. One can see in Figure 7.4c) that with decreasing $\mathcal{H}_2$ error bound, the envelope gets very tight (the $n = 80$ version is almost indistinguishable from the true output $y(t)$).



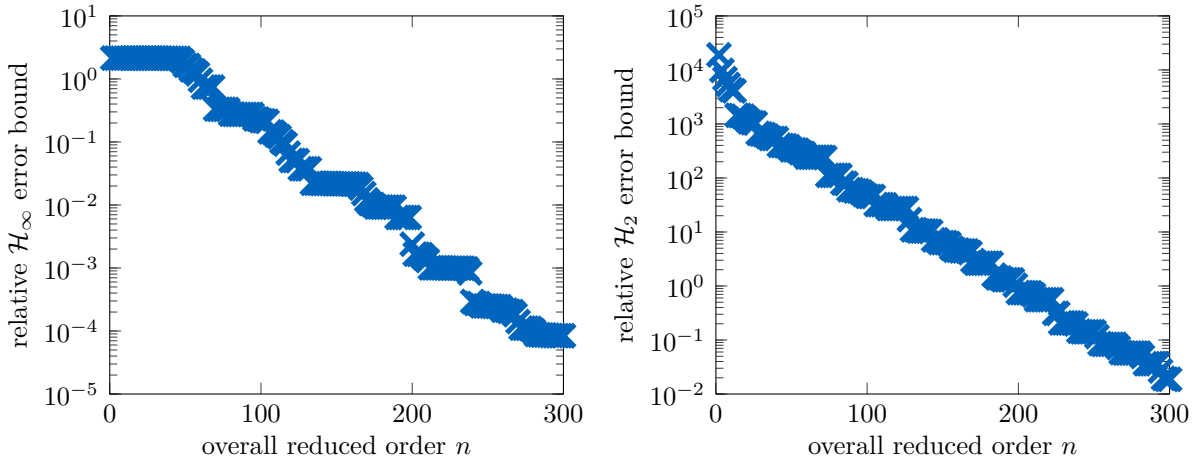
a) Relative $\mathcal{H}_\infty$ bound b) Relative $\mathcal{H}_2$ bound c) Time Domain Envelope

Figure 7.4.: Reduction of SISO Steel Profile 1357 by CUREd MESPARK with $\mathcal{J}_{\mathcal{H}_2}$

Now we move on to the MIMO case. We apply the strategy that was suggested in Section 4.5 and switch between the input channels; thus we feed only SIMO systems into the CUREd MESPARK ($\mathcal{J}_{\mathcal{H}_2}$) algorithm (the output matrix $\mathbf{C}$ does not enter anyway, so the number of outputs does not matter). Results can be seen in Figure 7.5. Despite the primitive strategy of selecting tangential vector, both error bounds decrease more or less constantly. The decay rate is, however, not quite convincing, in particular when compared

to the SISO result. One must run through 150 iterations to obtain a relative $\mathcal{H}_2$ error bound of about 1 %.

In fact, most of the studied multivariable systems led to considerably less performance than when choosing a single transmission path from it. Efficient strategies for the reduction of MIMO systems are therefore clearly an open problem.



a) Relative $\mathcal{H}_\infty$ bound

b) Relative $\mathcal{H}_2$ bound

Figure 7.5.: Reduction of MIMO Steel Profile 20209 by CUREd MESPARK with $\mathcal{J}_{\mathcal{H}_2}$

7.4. Acoustic Field in Gas Turbine Combustor

Let us now consider the model of the acoustic field in a gas turbine (cf. Section 1.2.6). It turns out that the given model exhibits unstable modes. The reason are so-called Kelvin-Helmholtz instabilities, which are a result of the linearization of Navier-Stokes equations. As a matter of fact, these modes are unwanted in the model and do not have to be mimicked by the ROM. Furthermore, they contribute little to the amplitude response $|G(s)|$ in the interesting frequency range, so simply "deleting" them is the best one can do.

To remove the unstable eigenvalues from the transfer function, they must be made uncontrollable and/or unobservable before the actual reduction process. It is shown in the sequel that this procedure can be interpreted by means of the CURE framework. If we perform modal truncation as a first reduction step (extracting the unstable components

from the HFM), and perform an input type error factorization (4.6), then the unstable eigenvalues in $\mathbf{G}_\perp(s)$ become uncontrollable according to Section 4.2. In fact, we can then write the HFM as

$$\mathbf{G}(s) = \mathbf{G}_{r,1}(s) + \mathbf{G}_{\perp,1}(s),$$

where $\mathbf{G}_r(s)$ contains the entire unstable dynamics, and $\mathbf{G}_\perp(s) = (\mathbf{A}, \mathbf{B}_\perp, \mathbf{C}_\perp, \mathbf{0}, \mathbf{E})$ is BIBO stable. The unstable ROM $\mathbf{G}_{r,1}(s) = (\mathbf{W}_S^T\mathbf{A}\mathbf{V}_S, \mathbf{W}_S^T\mathbf{B}, \mathbf{C}\mathbf{V}_S, \mathbf{0}, \mathbf{W}_S^T\mathbf{E}\mathbf{V}_S)$ might also be kept as the first component of the cumulatively built overall ROM, but is simply discarded here, so basically we replace $\mathbf{G}(s)$ by $\mathbf{G}_\perp(s)$. $\mathbf{B}_\perp = (\mathbf{I} - \mathbf{\Pi})\mathbf{B}$ and $\mathbf{C}_\perp = \mathbf{C}(\mathbf{I} - \mathbf{\Pi}_W)$ are given with the help of the spectral projector as described in Section 4.2. To find $\mathbf{\Pi}_W = \mathbf{E}\mathbf{V}_S(\mathbf{W}_S^T\mathbf{E}\mathbf{V}_S)^{-1}\mathbf{W}_S^T$ and $\mathbf{\Pi} = \mathbf{V}_S(\mathbf{W}_S^T\mathbf{E}\mathbf{V}_S)^{-1}\mathbf{W}_S^T\mathbf{E}$, bases $\mathbf{W}_S$ and $\mathbf{V}_S$ of the unstable left and right invariant subspaces of $\mathbf{A}$,$\mathbf{E}$ must be identified, i. e. solutions of Sylvester equations

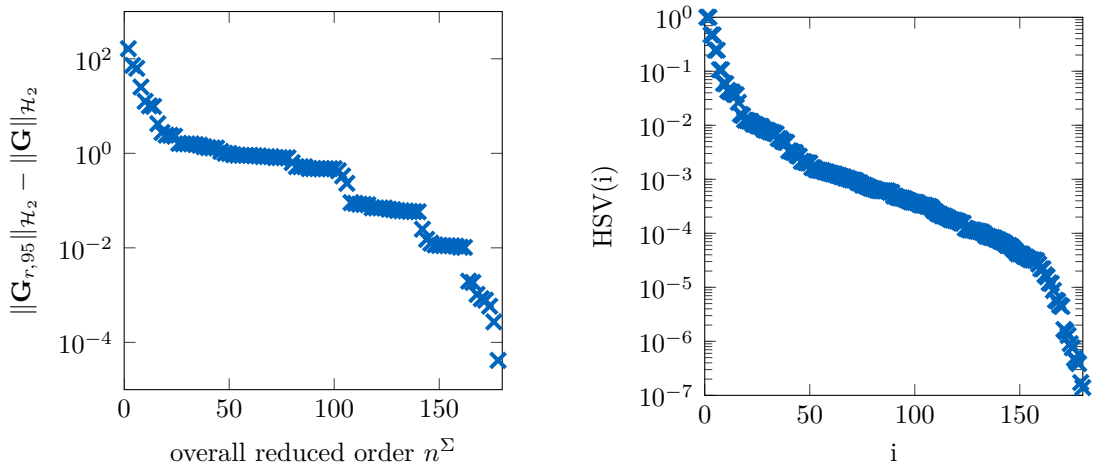
$$\mathbf{A}\mathbf{V}_S = \mathbf{E}\mathbf{V}_S\mathbf{S}_V \quad \text{and} \quad \mathbf{W}_S^T\mathbf{A} = \mathbf{S}_W\mathbf{W}_S^T\mathbf{E}, \tag{7.1}$$

where $\mathbf{S}_V = \mathbf{S}_W$ are diagonal matrices containing all unstable eigenvalues. This can be achieved with the `eigs` command in MATLAB.

The existence of unstable eigenvalues implies $\alpha > 0$ and therefore $\mu > 0$, so the model cannot be strictly dissipative and the error bounds do not apply. However, we will consider its reduction by means of the CUREd MESPARK algorithm in the following.

In the absence of rigorous error bounds, heuristic stopping criteria were used and CURE was interrupted after 95 iterations, because then the ROM hardly changed anymore. Figure 7.6a) shows the $\mathcal{H}_2$ norm of the ROM during CURE[1]; it grows monotonically and approaches the $\mathcal{H}_2$ norm of the HFM, which it cannot exceed because of Proposition 4.3. The Hankel singular values of the ROM are depicted in Figure 7.6b) for $\mathbf{G}_{r,50}^\Sigma$. Both plots indicate that after an order of about 180, the "learning curve" of the ROM flattens. The amplitude responses of the HFM, the stabilized high-dimensional model, and the resulting ROM can be seen in Figure 7.7 and show a highly satisfactory result. The whole cumulative reduction process required 94 minutes.

[1]Please mind the vertical scaling; the graph shows the difference of the norm of the last ROM ($n = 190$) to the previous ones.



a) Change of $\left\|\mathbf{G}^{\Sigma}_{r,k}\right\|_{\mathcal{H}_2}$ during CURE

b) Normalized HSVs of final ROM

Figure 7.6.: Stopping Criteria for Acoustic Field Model during CURE

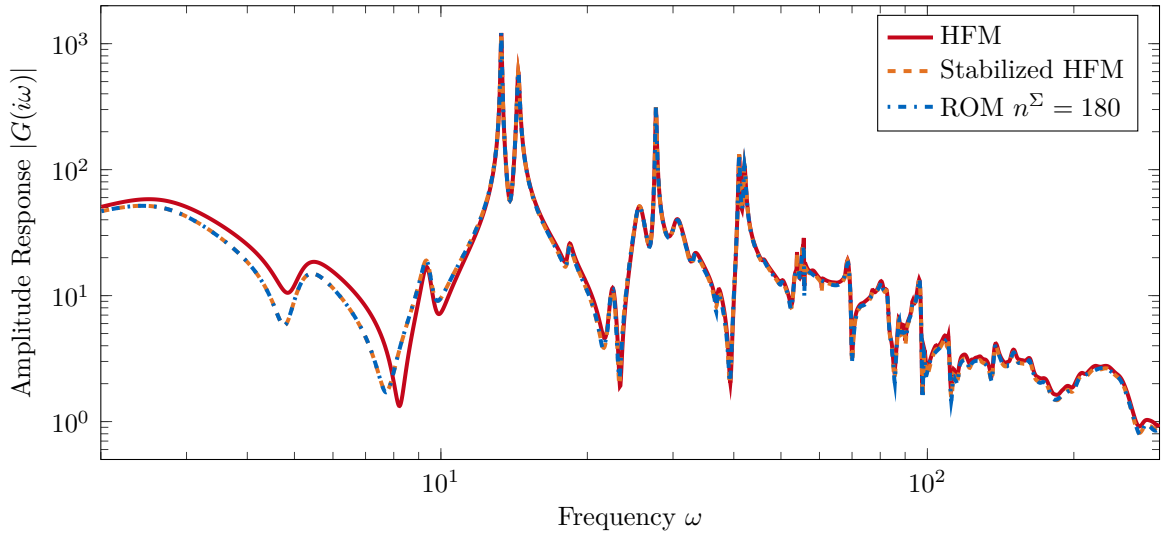


Figure 7.7.: Comparison of Amplitude Responses for Acoustic Field Model

7.5. Power System

The model described in Section 1.2.6 was considered to show that some DAE models may, in principle, be reduced within the CURE scheme, as well. The model is obviously not dissipative, but was reduced by CUREd MESPARK successfully. Figure 7.8 shows the amplitude response of the HFM and of three different ROMs, whose associated error amplitude response can additionally be seen on the right hand side. Running CURE up to an order of 150 yields a very good ROM, which may also be compressed to order $n = 50$ in a second reduction step by TBR without great loss of accuracy. Stopping CURE at order 50 leads to mediocre approximation, which is however acceptable in the frequency range of interest.

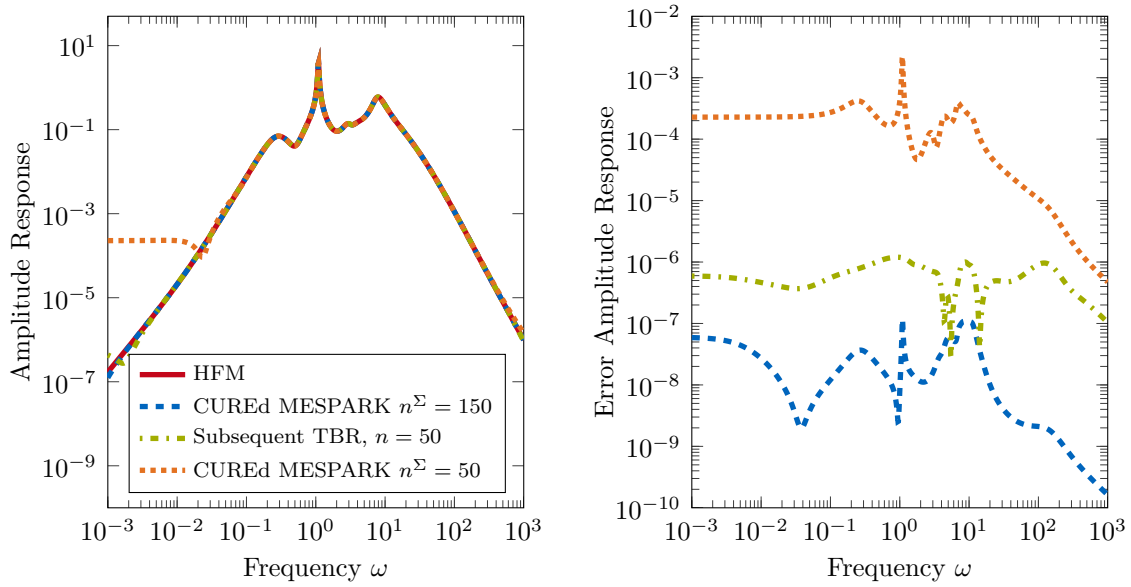


Figure 7.8.: Amplitude Response for Power System Model

8. Summary, Conclusions, and Outlook

> *"The hope is that there will be devised a method which combines the best attributes of the SVD and Krylov methods. Such a method does not exist yet. The quest, however, continues."*
>
> — A. C. Antoulas and D. Sorensen [11]

Several new methods for SYLVESTER-based model reduction of high-dimensional LTI state space models have been presented in this thesis:

- By means of the Cumulative Reduction (CURE) scheme it is possible to perform several consecutive reduction steps, during which the overall reduced model is iteratively accumulated. This framework requires almost no additional computational effort, but opens up new possibilities for adaptive MOR. Firstly, the final order of the ROM can be chosen on the fly instead of *a priori*. Secondly, it is sufficient to determine a small number of expansion points at a time.
- The shift selection can be carried out with the stability-preserving descent algorithm MESPARK, which uses trust region optimization to find two $\mathcal{H}_2$ optimal expansion points in a very little number of steps, thus minimizing the count of high-dimensional operations.
- For systems in strictly dissipative realization, rigorous upper bounds on the global $\mathcal{H}_2$ and $\mathcal{H}_\infty$ error have been presented, which hold for both modal truncation and KRYLOV subspace methods. Both bounds are cheap to evaluate and can be monitored during CURE to determine a suitable reduced order.

- To avoid excessive overestimation, methods for so-called error controlled MOR have been presented. During CURE, they follow the goal of reducing the error bounds instead of the true error which is unknown anyway. This can be achieved either by special ways of projection or by adapting the cost functional in MESPARK.
- Second order systems with positive definite mass, damping, and stiffness matrices were shown to be an interesting application possibility, as they can be described in strictly dissipative state space models. The induced additional numerical complexity can essentially be avoided by judicious implementation.

Several numerical experiments verified the general applicability of the new techniques; most algorithms are included in form of ready-to-run MATLAB source code.

Many questions, however, remain to be investigated and are specified in the sequel.

In classical projective MOR, the reduced state vector can be used to estimate the full-order state via $\mathbf{x}(t) \approx \mathbf{V}\mathbf{x}_r(t)$. This direct relationship is lost during the CURE framework. Can it be restored somehow?

How can the SISO method MESPARK and its derivatives be effectively extended to multivariable systems? Can the parametrization of the optimization problem be improved to obtain more robust and faster convergence? How about preconditioning?

How should initial values be chosen in order to obtain fast convergence to a (possibly global) optimum? One idea to this end might be to collect all moments of the HFM (moments of $\mathbf{G}_\perp(s)$ can be translated to moments of $\mathbf{G}(s)$ due to (4.13)) and use the frequency where the highest deviation between $\mathbf{G}_r^\Sigma(s)$ and $\mathbf{G}(s)$ occurs, because this location in the complex LAPLACE plane has potential for improvement.

Can the CURE scheme be adapted to preserve structural properties like passivity?

What are the numerical properties of the error bounds, and how about error propagation? (When) are the bounds really reliable from a numerical point of view?

Is it possible to derive tighter bounds on the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms of a large-scale system? Inspiration might come from results on tighter bounds on the matrix exponential [152]. Also, an interesting (rigorous) estimate of eigenvalues of Gramian matrices can be found in [93]. Instead of a worst-case estimate based on the largest eigenvalue (i. e., the norm) of

the Gramian, this might enable more sophisticated estimates together with linear algebraic results like [108, 9.7.3.(3)]:

$$\mathbf{P} = \mathbf{P}^T > \mathbf{0},\ \mathbf{X}^T\mathbf{X} = \mathbf{I}_m \quad \Rightarrow \quad \operatorname{tr}(\mathbf{X}^T\mathbf{P}\mathbf{X}) \leq \sum_{i=1}^{m} \lambda_{N-m+i}(\mathbf{P}). \tag{8.1}$$

Also, during the CURE framework, it seems reasonable to recycle all the projection matrices $\mathbf{V}_i$ and $\mathbf{W}_i$ that appear during the reduction of the small-scale models, for the projection of the LYAPUNOV equations, such that the approximate Gramian becomes better and better, while in addition $B_\perp$ becomes shorter and shorter, which hopefully speeds up the decrease of the error bound.

How can MESPARK be extended to the $\mathcal{J}_{\mathcal{H}_\infty}$ cost functional such that the determination of a local minimum is still guaranteed? Does a local minimum with $\mathcal{J}(a^*, b^*) < 0$ always exist at all? Can other adaptive MOR methods be used for the shift selection to obtain tight error bounds? In the light of the $\mathcal{H}_2$ error bound, minimizing the length of the residual vector $\mathbf{b}_\perp$ becomes more than a heuristic. How can the error bounds be exploited in practical applications like, for instance, robust control?

In the context of second order systems: Is it possible to find a strictly dissipative realization also for singular damping matrices, i. e. $\det \mathbf{D} = 0$? Can the state space based procedure for the reduction of second order systems be avoided by applying a structure-preserving projection? In other words, is there a similar error decomposition for second order KRYLOV subspaces which allows for error bounds and cumulative reduction?

> *"There can be no universal model reduction algorithm.*
> *The best one can hope for is a good set of tools*
> *and some reliable guidelines for using them."*
>
> — B. C. Moore [117]

To conclude: In this work, the toolbox has been extended by a couple of new instruments. Their potential is not quite clear yet, it is hoped that the reliable guidelines for their usage will emerge belatedly over time—indeed, the quest continues.

A. Appendix

A.1. Proof of Theorem 4.3

To begin with, note that

$$\frac{\partial \sigma_1}{\partial a} = 1 + \frac{a}{\sqrt{a^2-b}}, \quad \frac{\partial \sigma_2}{\partial a} = 1 - \frac{a}{\sqrt{a^2-b}}, \quad \frac{\partial \sigma_1}{\partial b} = \frac{-1}{2\sqrt{a^2-b}}, \quad \frac{\partial \sigma_2}{\partial b} = \frac{1}{2\sqrt{a^2-b}}.$$

Also, we use the relationship $\frac{\partial \mathbf{A}^{-1}}{\partial a} = -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial a}\mathbf{A}^{-1}$ from [128] to derive

$$\frac{\partial \mathbf{A}_{\sigma_1}^{-1}}{\partial a} = -\mathbf{A}_{\sigma_1}^{-1}\left(-\sigma_1\mathbf{E}\cdot\frac{\partial \sigma_1}{\partial a}\right)\mathbf{A}_{\sigma_1}^{-1} = \mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\left(1+\frac{a}{\sqrt{a^2-b}}\right)$$

and similarly $\frac{\partial \mathbf{A}_{\sigma_1}^{-1}}{\partial b}$, $\frac{\partial \mathbf{A}_{\sigma_2}^{-1}}{\partial a}$, and $\frac{\partial \mathbf{A}_{\sigma_2}^{-1}}{\partial b}$. Then, we obtain:

$$\begin{aligned}
\frac{\partial c_{r,1}}{\partial a} &= \tfrac{1}{2}\mathbf{c}\left[\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\cdot\left(1+\tfrac{a}{\sqrt{a^2-b}}\right)+\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\cdot\left(1-\tfrac{a}{\sqrt{a^2-b}}\right)\right]\mathbf{b} \\
&= \tfrac{1}{2}\mathbf{c}\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}+\tfrac{1}{2}\mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{b}+ \\
&\quad +a\cdot\mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\cdot\underbrace{\left[\mathbf{A}_{\sigma_2}\mathbf{E}^{-1}\mathbf{A}_{\sigma_2}-\mathbf{A}_{\sigma_1}\mathbf{E}^{-1}\mathbf{A}_{\sigma_1}\right]\cdot\tfrac{1}{2\sqrt{a^2-b}}}\cdot\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} \\
&\qquad [(\mathbf{A}-\sigma_2\mathbf{E})\,\mathbf{E}^{-1}\,(\mathbf{A}-\sigma_2\mathbf{E})-(\mathbf{A}-\sigma_1\mathbf{E})\,\mathbf{E}^{-1}\,(\mathbf{A}-\sigma_1\mathbf{E})]\cdot\tfrac{1}{\sigma_1-\sigma_2} \\
&\qquad = [2\,(\sigma_1-\sigma_2)\,\mathbf{A}-(\sigma_1^2-\sigma_2^2)\,\mathbf{E}]\cdot\tfrac{1}{\sigma_1-\sigma_2} \\
&\qquad = [\mathbf{A}-\sigma_1\mathbf{E}+\mathbf{A}-\sigma_2\mathbf{E}] = \mathbf{A}_{\sigma_1}+\mathbf{A}_{\sigma_2} \\
&= \tfrac{1}{2}\mathbf{c}\cdot\mathbf{r}_2+\tfrac{1}{2}\mathbf{l}_2\cdot\mathbf{b}+a\,\mathbf{l}_1\mathbf{E}\mathbf{r}_2+a\,\mathbf{l}_2\mathbf{E}\mathbf{r}_1
\end{aligned}$$

$$\begin{aligned}
\frac{\partial c_{r,2}}{\partial a} &= \tfrac{1}{2}\mathbf{c}\left[\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\cdot\left(1-\tfrac{a}{\sqrt{a^2-b}}\right)+\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\cdot\left(1+\tfrac{a}{\sqrt{a^2-b}}\right)\right]\mathbf{b} \\
&= \mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\cdot\underbrace{\left[\mathbf{A}_{\sigma_1}\left(1-\tfrac{a}{\sqrt{a^2-b}}\right)+\mathbf{A}_{\sigma_2}\left(1+\tfrac{a}{\sqrt{a^2-b}}\right)\right]}\cdot\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b} \\
&\qquad \mathbf{A}-\sigma_1\mathbf{E}+\mathbf{A}-\sigma_2\mathbf{E}+\tfrac{a}{\sqrt{a^2-b}}\,(\mathbf{A}-\sigma_2\mathbf{E}-(\mathbf{A}-\sigma_1\mathbf{E})) \\
&\qquad = 2\mathbf{A}-(\sigma_1+\sigma_2)\mathbf{E}+\tfrac{a}{\sqrt{a^2-b}}(\sigma_1-\sigma_2)\mathbf{E} = 2\mathbf{A} \\
&= 2\mathbf{l}_2\mathbf{A}\mathbf{r}_2
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial c_{r,1}}{\partial b} &= \tfrac{1}{2}\mathbf{c}\left[\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\cdot\tfrac{-1}{2\sqrt{a^2-b}}+\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\cdot\tfrac{1}{2\sqrt{a^2-b}}\right]\mathbf{b}\\
&= \tfrac{1}{2}\mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\cdot \underbrace{\left[\mathbf{A}_{\sigma_1}\mathbf{E}^{-1}\mathbf{A}_{\sigma_1}-\mathbf{A}_{\sigma_2}\mathbf{E}^{-1}\mathbf{A}_{\sigma_2}\right]\cdot\tfrac{1}{2\sqrt{a^2-b}}}\quad\cdot\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}\\
&\qquad\left[\left(\mathbf{A}-\sigma_1\mathbf{E}\right)\mathbf{E}^{-1}\left(\mathbf{A}-\sigma_1\mathbf{E}\right)-\left(\mathbf{A}-\sigma_2\mathbf{E}\right)\mathbf{E}^{-1}\left(\mathbf{A}-\sigma_2\mathbf{E}\right)\right]\cdot\tfrac{1}{\sigma_1-\sigma_2}\\
&= \left[-2\left(\sigma_1-\sigma_2\right)\mathbf{A}+\left(\sigma_1^2-\sigma_2^2\right)\mathbf{E}\right]\cdot\tfrac{1}{\sigma_1-\sigma_2}\\
&= \left[-\left(\mathbf{A}-\sigma_1\mathbf{E}\right)-\left(\mathbf{A}-\sigma_2\mathbf{E}\right)\right]=-\mathbf{A}_{\sigma_1}-\mathbf{A}_{\sigma_2}\\
&= -\tfrac{1}{2}\mathbf{l}_1\mathbf{E}\mathbf{r}_2-\tfrac{1}{2}\mathbf{l}_2\mathbf{E}\mathbf{r}_1
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial c_{r,2}}{\partial b} &= \mathbf{c}\left[\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\cdot\tfrac{1}{2\sqrt{a^2-b}}+\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\cdot\tfrac{-1}{2\sqrt{a^2-b}}\right]\mathbf{b}\\
&= \mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\cdot\left[\mathbf{A}_{\sigma_1}-\mathbf{A}_{\sigma_2}\right]\tfrac{1}{2\sqrt{a^2-b}}\cdot\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}\\
&= \mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\cdot\left[\mathbf{A}-\sigma_1\mathbf{E}-\left(\mathbf{A}-\sigma_2\mathbf{E}\right)\right]\cdot\tfrac{1}{\sigma_1-\sigma_2}\cdot\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}\\
&= -\mathbf{l}_2\mathbf{E}\mathbf{r}_2
\end{aligned}
$$

The second derivatives which enter the Hessian matrix can be computed similarly.

$$
\begin{aligned}
\frac{\partial^2 c_{r,1}}{\partial a^2} &= \mathbf{c}\left[\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\cdot\left(1+\tfrac{a}{\sqrt{a^2-b}}\right)+\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\cdot\left(1-\tfrac{a}{\sqrt{a^2-b}}\right)\right]\mathbf{b}\\
&\quad+\mathbf{l}_1\mathbf{E}\mathbf{r}_2+\mathbf{l}_2\mathbf{E}\mathbf{r}_1\\
&\quad+2a\cdot\mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\cdot\left(1-\tfrac{a}{\sqrt{a^2-b}}\right)\\
&\quad+a\cdot\mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\cdot\left(1+\tfrac{a}{\sqrt{a^2-b}}\right)\\
&\quad+a\cdot\mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\cdot\left(1-\tfrac{a}{\sqrt{a^2-b}}\right)\\
&\quad+2a\cdot\mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\cdot\left(1+\tfrac{a}{\sqrt{a^2-b}}\right)\\
&= \mathbf{c}\mathbf{r}_3+\mathbf{l}_3\mathbf{b}+\mathbf{l}_3\underbrace{\left[\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}-\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\right]\cdot\tfrac{a}{\sqrt{a^2-b}}}\mathbf{r}_3\\
&\qquad\qquad=2a\cdot\left[\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}-\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}+\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\right]\\
&\quad+\mathbf{l}_1\mathbf{E}\mathbf{r}_2+\mathbf{l}_2\mathbf{E}\mathbf{r}_1+2a\cdot\mathbf{l}_3\mathbf{E}\mathbf{r}_1+2a\cdot\mathbf{l}_1\mathbf{E}\mathbf{r}_r+2a\cdot\mathbf{l}_2\mathbf{E}\mathbf{r}_2\\
&\quad+2a\cdot\mathbf{l}_3\underbrace{\left[\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}-\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\right]\cdot\tfrac{a}{\sqrt{a^2-b}}}\mathbf{r}_3\\
&\qquad\qquad=2a\left(\mathbf{A}_{\sigma_1}^{-1}+\mathbf{A}_{\sigma_2}^{-1}\right)\\
&= \mathbf{c}\mathbf{r}_3+\mathbf{l}_3\mathbf{b}+2a\cdot\mathbf{l}_1\mathbf{E}\mathbf{r}_3+2a\cdot\mathbf{l}_3\mathbf{E}\mathbf{r}_1+2a\cdot\mathbf{l}_2\mathbf{E}\mathbf{r}_2\\
&\quad+\mathbf{l}_1\mathbf{E}\mathbf{r}_2+\mathbf{l}_2\mathbf{E}\mathbf{r}_1+2a\cdot\mathbf{l}_3\mathbf{E}\mathbf{r}_1+2a\cdot\mathbf{l}_1\mathbf{E}\mathbf{r}_3+2a\cdot\mathbf{l}_2\mathbf{E}\mathbf{r}_2\\
&\quad+4a^2\cdot\mathbf{l}_2\mathbf{E}\mathbf{r}_3+4a^2\cdot\mathbf{l}_3\mathbf{E}\mathbf{r}_2\\
&= \mathbf{c}\mathbf{r}_3+\mathbf{l}_3\mathbf{b}+4a\cdot\mathbf{l}_1\mathbf{E}\mathbf{r}_3+4a\cdot\mathbf{l}_3\mathbf{E}\mathbf{r}_1+2a\cdot\mathbf{l}_2\mathbf{E}\mathbf{r}_2+2a\cdot\mathbf{l}_2\mathbf{A}\mathbf{r}_2\\
&\quad+4a^2\cdot\mathbf{l}_2\mathbf{E}\mathbf{r}_3+4a^2\cdot\mathbf{l}_3\mathbf{E}\mathbf{r}_2
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 c_{r,2}}{\partial a^2} &= 4\,\mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\cdot\mathbf{A}\cdot\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}\cdot\left(1-\tfrac{a}{\sqrt{a^2-b}}\right)+\\
&\quad +4\,\mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\cdot\mathbf{A}\cdot\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\cdot\left(1+\tfrac{a}{\sqrt{a^2-b}}\right)\\
&= 4\,\mathbf{l}_3\mathbf{A}\mathbf{r}_2+4\,\mathbf{l}_2\mathbf{A}\mathbf{r}_3+4\,\mathbf{l}_3\left[-\mathbf{A}\mathbf{E}^{-1}\mathbf{A}_{\sigma_1}+\mathbf{A}_{\sigma_2}\mathbf{E}^{-1}\mathbf{A}\right]\cdot\tfrac{a}{\sqrt{a^2-b}}\cdot\mathbf{r}_3\\
&= 4\mathbf{l}_3\mathbf{A}\mathbf{r}_2+4\mathbf{l}_2\mathbf{A}\mathbf{r}_3+4\mathbf{l}_3\left[-\mathbf{A}\mathbf{E}^{-1}\mathbf{A}+\sigma_1\mathbf{A}+\mathbf{A}\mathbf{E}^{-1}\mathbf{A}-\sigma_2\mathbf{A}\right]\tfrac{2a}{\sigma_1-\sigma_2}\mathbf{r}_3\\
&= 4\,\mathbf{l}_3\mathbf{A}\mathbf{r}_2+4\,\mathbf{l}_2\mathbf{A}\mathbf{r}_3+8\,\mathbf{l}_3\mathbf{A}\mathbf{r}_3
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 c_{r,2}}{\partial a\partial b} &= 4\mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{A}\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}\cdot\tfrac{1}{2\sqrt{a^2-b}}-\\
&\quad 4\mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{A}\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}\cdot\tfrac{-1}{2\sqrt{a^2-b}}\\
&= 4\,\mathbf{l}_3\left[\mathbf{A}\mathbf{E}^{-1}\left(\mathbf{A}-\sigma_1\mathbf{E}\right)-\left(\mathbf{A}-\sigma_1\mathbf{E}\right)\mathbf{E}^{-1}\mathbf{A}\right]\tfrac{1}{2\sqrt{a^2-b}}\mathbf{r}_3\\
&= -4\,\mathbf{l}_3\,\mathbf{A}\,\mathbf{r}_3
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 c_{r,2}}{\partial b^2} &= -2\mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}\cdot\tfrac{1}{2\sqrt{a^2-b}}-\\
&\quad -2\mathbf{c}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_2}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{E}\mathbf{A}_{\sigma_1}^{-1}\mathbf{b}\cdot\tfrac{-1}{2\sqrt{a^2-b}}\\
&= -2\,\mathbf{l}_3\left[\mathbf{A}_{\sigma_1}-\mathbf{A}_{\sigma_2}\right]\tfrac{1}{2\sqrt{a^2-b}}\mathbf{r}_3\\
&= 2\,\mathbf{l}_3\,\mathbf{E}\,\mathbf{r}_3
\end{aligned}
$$

□

A.2. Proof of Theorem 5.2

According to [32], the $\mathcal{H}_\infty$ norm of an LTI system is correlated with asymptotic stability of the perturbed system, namely by the so-called structured complex stability radius $r_{\mathbb{C}} \in \mathbb{R}^+$: unless $\mathbf{G}(s) \equiv \mathbf{0}$, it holds that $\|\mathbf{G}\|_{\mathcal{H}_\infty} = \frac{1}{r_{\mathbb{C}}}$.

$r_{\mathbb{C}}$ is the smallest number $\epsilon \in \mathbb{R}^+$ that admits a matrix $\mathbf{\Delta} \in \mathbb{C}^{m\times p}$ with $\|\mathbf{\Delta}\|_2 < \epsilon$ such that the perturbed system

$$\mathbf{G}_\Delta(s) := \mathbf{C}\Big(s\mathbf{E} - (\mathbf{A} + \mathbf{B}\mathbf{\Delta}\mathbf{C})\Big)^{-1}\mathbf{B} \tag{A.1}$$

is no longer asymptotically stable, but has at least one purely imaginary or zero pole.

The key idea was to use strict dissipativity as a sufficient criterion for asymptotic stability in order to compute a lower bound ϵ^* on $r_{\mathbb{C}}$. More precisely: If for every $\mathbf{\Delta}$ fulfilling $\|\mathbf{\Delta}\| < \epsilon^*$ the perturbed system (A.1) is strictly dissipative, then it is also asymptotically stable according to Lemma 2.1, so ϵ^* is surely less or equal to $r_{\mathbb{C}}$, so its inverse is greater or equal to $\|\mathbf{G}\|_{\mathcal{H}_\infty}$.

It remains to find a number ϵ^* for which the perturbed system is guaranteed to stay strictly dissipative, i. e. for which $\mu_{\mathbf{E}}(\mathbf{A} + \mathbf{B}\mathbf{\Delta}\mathbf{C})$ is strictly negative as long as $\|\mathbf{\Delta}\| < \epsilon^*$. Because of Corollary 2.1, it must hold

$$\begin{aligned}
\mu_2(\mathbf{A} + \mathbf{B}\mathbf{\Delta}\mathbf{C}) < 0 \quad &\Leftrightarrow \quad \mathbf{x}^H\Big(\mathbf{A}+\mathbf{A}^H + \mathbf{B}\mathbf{\Delta}\mathbf{C}+(\mathbf{B}\mathbf{\Delta}\mathbf{C})^H\Big)\mathbf{x} < 0 \quad \forall \mathbf{x} \in \mathbb{C}^N \\
&\Leftrightarrow \quad \mathbf{x}^H\Big(\mathbf{B}\mathbf{\Delta}\mathbf{C} + (\mathbf{B}\mathbf{\Delta}\mathbf{C})^H\Big)\mathbf{x} \;<\; \mathbf{x}^H(-\mathbf{A} - \mathbf{A}^T)\mathbf{x} \quad \forall \mathbf{x} \\
&\Leftrightarrow \quad \frac{\mathbf{x}^H\Big(\mathbf{B}\mathbf{\Delta}\mathbf{C} + (\mathbf{B}\mathbf{\Delta}\mathbf{C})^H\Big)\mathbf{x}}{\mathbf{x}^H(-\mathbf{A} - \mathbf{A}^T)\mathbf{x}} \;<\; 1 \quad \forall \mathbf{x}
\end{aligned}$$

for all $\mathbf{\Delta}$ with $\|\mathbf{\Delta}\| < \epsilon^*$. Define $\widetilde{\mathbf{\Delta}} := \frac{\mathbf{\Delta}}{\|\mathbf{\Delta}\|}$. Then, $\left\|\widetilde{\mathbf{\Delta}}\right\| = 1$ holds and the above can be equivalently written as

$$\mu_2(\mathbf{A} + \mathbf{B}\mathbf{\Delta}\mathbf{C}) < 0 \quad \Leftrightarrow \quad \frac{\mathbf{x}^H\Big(\mathbf{B}\widetilde{\mathbf{\Delta}}\mathbf{C} + (\mathbf{B}\widetilde{\mathbf{\Delta}}\mathbf{C})^H\Big)\mathbf{x}}{\mathbf{x}^H\mathbf{S}\mathbf{x}} \;<\; \frac{1}{\|\mathbf{\Delta}\|} \quad \forall \mathbf{x}. \tag{A.2}$$

In order to guarantee that (A.2) is fulfilled, we must choose ϵ^* such that for any $\mathbf{x} \in \mathbb{C}^N$ and for any $\widetilde{\mathbf{\Delta}} \in \mathbb{C}^{m\times p}$ with $\left\|\widetilde{\mathbf{\Delta}}\right\| = 1$, the condition

$$\frac{\mathbf{x}^H\Big(\mathbf{B}\widetilde{\mathbf{\Delta}}\mathbf{C} + (\mathbf{B}\widetilde{\mathbf{\Delta}}\mathbf{C})^H\Big)\mathbf{x}}{\mathbf{x}^H\mathbf{S}\mathbf{x}} \leq \frac{1}{\epsilon^*} \tag{A.3}$$

is satisfied. (A.3) implies (A.2), because by definition

$$\frac{1}{\epsilon^*} < \frac{1}{\|\mathbf{\Delta}\|}. \tag{A.4}$$

The difficulty is that the term in (A.3) depends on two independent quantities, namely an N-dimensional complex vector $\mathbf{x}$ and an $m \times p$ complex matrix $\widetilde{\mathbf{\Delta}}$ whose 2-norm amounts to one. The key to bound the expression in (A.3) anyway is to firstly fix $\mathbf{x}$ and find the particular $\widetilde{\mathbf{\Delta}}^*(\mathbf{x})$ which maximizes the constrained problem.

A first simplification to this end was shown in [124]: Given some $\mathbf{x}$, the numerator in (A.3) reaches its maximum for

$$\widetilde{\mathbf{\Delta}} = \widetilde{\mathbf{\Delta}}^*(\mathbf{x}) = \frac{\mathbf{B}^T\mathbf{x}\mathbf{x}^H\mathbf{C}^T}{\|\mathbf{B}^T\mathbf{x}\|_2\|\mathbf{C}\mathbf{x}\|_2} \tag{A.5}$$

whose rank is one. It turns out, however, that even though we know the exact value of $\widetilde{\mathbf{\Delta}}^*(\mathbf{x})$, it is helpful to ignore this knowledge and only exploit the rank-one property. More precisely, we write $\widetilde{\mathbf{\Delta}}^*(\mathbf{x})$ as a product of two normalized vectors $\mathbf{u} \in \mathbb{C}^m$ and $\mathbf{v} \in \mathbb{C}^p$,

$$\widetilde{\mathbf{\Delta}}^*(\mathbf{x}) = \mathbf{u}\mathbf{v}^H, \qquad \text{where} \quad \|\mathbf{u}\|_2 = 1, \|\mathbf{v}\|_2 = 1. \tag{A.6}$$

Also, we use the relationship of the Rayleigh quotient and the eigenvalues of a matrix [108], to turn (A.3) into the generalized eigenvalue problem

$$\begin{aligned}
\frac{\mathbf{x}^H\left(\mathbf{B}\widetilde{\mathbf{\Delta}}\mathbf{C} + (\mathbf{B}\widetilde{\mathbf{\Delta}}\mathbf{C})^H\right)\mathbf{x}}{\mathbf{x}^H\mathbf{S}\mathbf{x}} &\leq \max_{i,\widetilde{\mathbf{\Delta}}} \lambda_i\left[\mathbf{B}\widetilde{\mathbf{\Delta}}\mathbf{C} + (\mathbf{B}\widetilde{\mathbf{\Delta}}\mathbf{C})^H, \mathbf{S}\right] \\
&\leq \max_{i,\mathbf{u},\mathbf{v}} \lambda_i\left[\mathbf{B}\mathbf{u}\mathbf{v}^H\mathbf{C} + (\mathbf{B}\mathbf{u}\mathbf{v}^H\mathbf{C})^H, \mathbf{S}\right] \\
&= \max_{i,\mathbf{u},\mathbf{v}} \lambda_i\left[\begin{bmatrix}\mathbf{B}\mathbf{u} & \mathbf{C}^T\mathbf{v}\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\mathbf{u}^H\mathbf{B}^T\\ \mathbf{v}^H\mathbf{C}\end{bmatrix}\mathbf{S}^{-1}\right] = \\
&= \max_{i,\mathbf{u},\mathbf{v}} \lambda_i\left[\underbrace{\begin{bmatrix}\mathbf{u}^H\mathbf{B}^T\\ \mathbf{v}^H\mathbf{C}\end{bmatrix}\mathbf{S}^{-1}\begin{bmatrix}\mathbf{B}\mathbf{u} & \mathbf{C}^T\mathbf{v}\end{bmatrix}}_{\in\mathbb{R}^{2\times 2}}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\right] = \\
&= \max_{i,\mathbf{u},\mathbf{v}} \lambda_i\left[\begin{bmatrix}\mathbf{u}^H\mathbf{B}^T\mathbf{S}^{-1}\mathbf{B}\mathbf{u} & \mathbf{u}^H\mathbf{B}^T\mathbf{S}^{-1}\mathbf{C}^T\mathbf{v}\\ \mathbf{v}^H\mathbf{C}\mathbf{S}^{-1}\mathbf{B}\mathbf{u} & \mathbf{v}^H\mathbf{C}\mathbf{S}^{-1}\mathbf{C}^T\mathbf{v}\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\right] \\
&= \max_{\mathbf{u},\mathbf{v}} \left[\mathbf{v}^H\mathbf{C}\mathbf{S}^{-1}\mathbf{B}\mathbf{u} + \sqrt{\mathbf{u}^H\mathbf{B}^T\mathbf{S}^{-1}\mathbf{B}\mathbf{u}\cdot\mathbf{v}^H\mathbf{C}\mathbf{S}^{-1}\mathbf{C}^T\mathbf{v}}\right] \\
&\leq \left\|\mathbf{C}_\perp\mathbf{S}^{-1}\mathbf{B}_\perp\right\|_2 + \sqrt{\left\|\mathbf{B}_\perp^T\mathbf{S}^{-1}\mathbf{B}_\perp\right\|_2 \cdot \left\|\mathbf{C}_\perp\mathbf{S}^{-1}\mathbf{C}_\perp^T\right\|_2} =: \frac{1}{\epsilon^*}.
\end{aligned}$$

□

A.3. MATLAB Source Code Files

Source A.1: MESPARK for Minimization of $\mathcal{H}_2$ Error Bound

```
1  function [V,S_V,Crt] = MESPARK_H2Bound(A,B,E,s0,L_E,P_E)
2  % MESPARK for Minimization of H2 Error Bound
3  %   Input:  A,B,E; L_E,P_E: HFM matrices; Cholesky decomposition of E
4  %           s0:             Initial shifts
5  %   Output: V,S_V,Crt:      Input Krylov subspace, A*V - E*V*S_V - B*Crt = 0
6  %
7
8      if size(B,2)>1, error('System must be SIMO.'), end
9      p0 = [(s0(1)+s0(2))/2, s0(1)*s0(2)];    % convert shifts to parameter
10     t = tic; k = 0; precond = eye(2);
11     % compute initial model function and cost function at p0
12     computeLU(s0); V = newColV([],3);
13     Am=V'*A*V; Bm=V'*B; Em=V'*E*V; Em=(L_E*(P_E'*V)); Em=Em'*Em;
14     J0 = B'*P_E*(L_E\(L_E'\(P_E'*B)));  J_old = CostFunction(p0);
⋮
30     V = newColV(V, 2); Am=V'*A*V; Bm=V'*B; Em=(L_E*(P_E'*V)); Em=Em'*Em;
⋮
52     function [J, g, H] = CostFunction(p)
53         [J,g,H] = CostFunctionH2Bound(Am, Bm, Em, p*precond);
54         J = full(J/J0); g = g*precond/J0; H = precond*H*precond/J0;
55     end
```

Source A.2: MESPARK for $\mathcal{H}_2$ Error Bound: Cost Functional, Gradient, and Hessian

```
function [J, g, H] = CostFunctionH2Bound(A, B, E, p)
% Cost Functional for H2 Error Bound Minimization, by A. Kohl
%   Input:  A,B,E: HFM matrices;
%           p:       parameter vector [a,b];
%   Output: cost functional J; gradient g; Hessian H
%

a = p(1); b = p(2); s0 = p(1)+[1 -1]*sqrt(p(1)^2-p(2));
As1 = A-s0(1)*E; As2 = A-s0(2)*E; L_E = chol(E);
r1 = As1\B; r2 = As1\(E*r1); r3 = As1\(E*r2);
LeB = L_E'\B;
LeB_ = LeB + L_E*((r1 + (As2\(B+E*r1*2*a)))*2*a );
J = LeB_'*LeB_ - LeB'*LeB;
if (nargout==1), return; end

lAr1 = E*(As2\(A*r1)); lAr2 = E*(As2\(E*(As2\(A*r2))));
lAr3 = E*(As2\(E*(As2\(E*(As2\(A*r3))))));
lEr1 = E*(As2\(E*r1)); lEr2 = E*(As2\(E*(As2\(E*r2))));
lAEAr2 = E*(As2\(E*(As2\(A *(L_E\(L_E'\(A*r2)))))));
tmp = A *(L_E\(L_E'\(A*r3)));
lAEAr3 = E*(As2\(E*(As2\(E*(As2\tmp)))));
lAEAEAr3 = E*(As2\(E*(As2\(E*(As2\(A*(L_E\(L_E'\tmp))))))));
g_Ba = 4*a*lAEAr2 + 8*a^2*lAr2 - 4*a*b*lEr2 + 4*lAr1 + 4*a*lEr1;  g_Bb = -4*a*lAr2;

LeB_a = L_E'\g_Ba; LeB_b = L_E'\g_Bb; g = 2*real([LeB_'*LeB_a, LeB_'*LeB_b]);
g_Baa = 16*a*lAEAEAr3+32*a^2*lAEAr3-16*a*b*lAr3+12*lAEAr2+24*a*lAr2-4*b*lEr2+4*lEr1;
g_Bbb = 8*a*lAr3; g_Bab = -4*lAr2 -16*a*lAEAr3;

H = [LeB_a'*LeB_a + LeB_'*(L_E'\g_Baa), LeB_a'*LeB_b + LeB_'*(L_E'\g_Bab); ...
     0, LeB_b'*LeB_b + LeB_'*(L_E'\g_Bbb)];
H(2,1)=H(1,2); H=2*real(H);
end
```

References

[1] M. I. Ahmad, M. Frangos, and I. M. Jaimoukha. "Second order $\mathcal{H}_2$ optimal approximation of linear dynamical systems". In: *18th IFAC World Congress*. Milano, Italy, 2011.

[2] M. I. Ahmad, I. M. Jaimoukha, and M. Frangos. "$\mathcal{H}_2$ optimal model reduction of linear dynamical systems". In: *49th IEEE Conference on Decision and Control*. Atlanta, USA, 2010.

[3] K. Ahuja. "Recycling Bi-Lanczos Algorithms: BiCG, CGS, and BiCGSTAB". PhD thesis. Virginia Polytechnic Institute and State University, 2009.

[4] K. Ahuja, E. de Sturler, S. Gugercin, and E. R. Chang. "Recycling BiCG with an application to model reduction". In: *SIAM Journal on Scientific Computing* 34.4 (2012), A1925–A1949.

[5] D. Amsallem and C. Farhat. "An online method for interpolating linear parametric reduced-order models". In: *SIAM Journal on Scientific Computing* 33.5 (2011), pp. 2169–2198.

[6] D. Amsallem and U. Hetmaniuk. "A posteriori error estimators for linear reduced order models using Krylov-based integrators". Preprint. 2014.

[7] B. Anderson and A. C. Antoulas. "Rational interpolation and state-variable realizations". In: *Linear Algebra and Its Applications* 137 (1990), pp. 479–509.

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

[9] A. C. Antoulas. "On pole placement in model reduction". In: *at-Automatisierungstechnik* 9 (2007), pp. 443–448.

[10] A. C. Antoulas, C. A. Beattie, and S. Gugercin. "Interpolatory model reduction of large-scale dynamical systems". In: *Efficient Modeling and Control of Large-Scale Systems*. Springer, 2010, pp. 3–58.

[11] A. C. Antoulas and D. C. Sorensen. "Approximation of large-scale dynamical systems: An overview". In: *International Journal of Applied Mathematics and Computer Science* 11 (2001), pp. 1093–1121.

[12] A. C. Antoulas, D. C. Sorensen, and S. Gugercin. “A survey of model reduction methods for large-scale systems”. In: *Contemporary mathematics* 280 (2001), pp. 193–220.

[13] A. Astolfi. “Model reduction by moment matching for linear and nonlinear systems”. In: *IEEE Transactions on Automatic Control* 55.10 (2010), pp. 2321–2336.

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

[15] Z. Bai, R. D. Slone, W. T. Smith, and Q. Ye. “Error bound for reduced system model by Padé approximation via the Lanczos process”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 18.2 (1999), pp. 133 –141.

[16] Z. Bai and Q. Ye. “Error estimation of the Padé approximation of transfer functions via the Lanczos process”. In: *Electronic Transactions on Numerical Analysis* 7 (1998), pp. 1–17.

[17] U. Baur and P. Benner. “Model reduction for parametric systems using balanced truncation and interpolation”. In: *at-Automatisierungstechnik* 57.8 (2009), pp. 411–419.

[18] U. Baur, P. Benner, and L. Feng. “Model order reduction for linear and nonlinear systems: a system-theoretic perspective”. In: *Archives of Computational Methods in Engineering* (2014).

[19] C. A. Beattie and S. Gugercin. “Inexact Solves in Krylov-based Model Reduction”. In: *45th IEEE Conference on Decision and Control*. Dec. 2006, pp. 3405–3411.

[20] C. A. Beattie and S. Gugercin. “A trust region method for optimal $\mathcal{H}_2$ model reduction”. In: *IEEE Conference on Decision and Control*. 2009.

[21] C. A. Beattie and S. Gugercin. “Krylov-based minimization for optimal $\mathcal{H}_2$ model reduction”. In: *46th IEEE Conference on Decision and Control*. 2007, pp. 4385–4390.

[22] C. A. Beattie and S. Gugercin. “Model reduction by rational interpolation”. 2014. Apr. 2014.

[23] C. A. Beattie and S. Gugercin. “Realization-independent $\mathcal{H}_2$-approximation”. In: *51st IEEE Conference on Decision and Control*. IEEE. 2012, pp. 4953–4958.

[24] T. Bechtold. “Model order reduction of electro-thermal MEMS”. PhD thesis. Albert-Ludwigs Universität Freiburg, 2005.

[25] T. Bechtold, E. B. Rudnyi, and J. G. Korvink. "Error indicators for fully automatic extraction of heat-transfer macromodels for MEMS". In: *Journal of Micromechanics and Microengineering* 15.3 (2005).

[26] *Benchmark Examples for Model Reduction*. SLICOT, Niconet e. V. URL: http://slicot.org/20-site/126-benchmark-examples-for-model-reduction.

[27] P. Benner and L. Feng. "Some a posteriori error bounds for reduced order modelling of parametrized linear systems". Oct. 2013.

[28] P. Benner, S. Gugercin, and K. Willcox. *A survey of model reduction methods for parametric systems*. Preprint MPIMD/13-14. Max Planck Institute Magdeburg, 2013.

[29] P. Benner, V. Mehrmann, and D. C. Sorensen. *Dimension reduction of large-scale systems*. Vol. 45. Springer, 2005.

[30] P. Benner and E. S. Quintana-Ortí. "Solving stable generalized Lyapunov equations with the matrix sign function". In: *Numerical Algorithms* 20.1 (1999), pp. 75–100.

[31] P. Benner and J. Saak. "A semi-discretized heat transfer model for optimal cooling of steel profiles". In: *Dimension Reduction of Large-Scale Systems*. Springer, 2005, pp. 353–356.

[32] P. Benner and M. Voigt. *A Structured Pseudospectral Method for $\mathcal{H}_\infty$-Norm Computation of Large-Scale Descriptor Systems*. Tech. rep. MPIMD/12-10. Max Planck Institute Magdeburg Preprint, May 2012. URL: http://www.mpi-magdeburg.mpg.de/preprints/2012/MPIMD12-10.pdf.

[33] N. F. Benninger. "Die Reduktionsdominanz als Ausgangspunkt für neue modale Masszahlen bei der Ordnungsreduktion". In: *Automatisierungstechnik* 35.1 (1987), pp. 19–26.

[34] D. Billger. *The Butterfly Gyro*. The Imego Institute, Sweden. URL: http://portal.uni-freiburg.de/imteksimulation/downloads/benchmark/TheButterflyGyro(35889).

[35] A. Bodendiek and M. Bollhöfer. "A modified adaptive-order rational Arnoldi method for model order reduction". In: *PAMM* (2013).

[36] A. Bodendiek and M. Bollhöfer. "Adaptive-order rational Arnoldi-type methods in Computational Electromagnetism". In: *BIT Numerical Mathematics* (Nov. 2013), pp. 1–24.

[37] A. Bunse-Gerstner, D. Kubalińska, G. Vossen, and D. Wilczek. "H2-norm optimal model reduction for large scale discrete dynamical MIMO systems". In: *Journal of computational and applied mathematics* 233.5 (2010), pp. 1202–1216.

[38] Y. Chahlaoui and P. Van Dooren. *A collection of benchmark examples for model reduction of linear time invariant dynamical systems*. SLICOT. Feb. 2002. URL: http://www.icm.tu-bs.de/NICONET/index.html.

[39] V. Chellaboina, W. M. Haddad, D. S. Bernstein, and D. A. Wilson. "Induced convolution operator norms of linear dynamical systems". In: *Mathematics of Control, Signals and Systems* 13.3 (2000), pp. 216–239.

[40] T.-Y. Chen. "Preconditioning sparse matrices for computing eigenvalues and solving linear systems of equations". PhD thesis. University of California at Berkeley, 2001.

[41] W. A. Coppel. *Stability and Asymptotic Behavior of Differential Equations*. D. C. Heath and Company Boston, 1965.

[42] G. Dahlquist. "Stability and error bounds in the numerical integration of ordinary differential equations". Transactions of the Royal Institute of Technology. Stockholm, Sweden, 1959.

[43] C. De Villemagne and R. E. Skelton. "Model reductions using a projection formulation". In: *International Journal of Control* 46.6 (1987), pp. 2141–2169.

[44] K. Dekker and J. Verwer. *Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations*. CWI Monograph. Elsevier Science Publishers B.V, Amsterdam, 1984.

[45] C. Desoer and H. Haneda. "The measure of a matrix as a tool to analyze computer algorithms for circuit analysis". In: *IEEE Transactions on Circuit Theory* 19.5 (1972), pp. 480–486.

[46] J. Doyle, B. Francis, and A. Tannenbaum. *Feedback Control Theory*. Macmillan Publishing Co., 1990.

[47] V. Druskin, V. Simoncini, and M. Zaslavsky. "Adaptive tangential interpolation in rational Krylov subspaces for MIMO dynamical systems". In: *SIAM Journal on Matrix Analysis and Applications* 35.2 (2014), pp. 476–498.

[48] V. Druskin, L. Knizhnerman, and V. Simoncini. "Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation". In: *SIAM Journal on Numerical Analysis* 49.5 (2011), pp. 1875–1898.

[49] V. Druskin, C. Lieberman, and M. Zaslavsky. "On adaptive choice of shifts in rational Krylov subspace reduction of evolutionary problems". In: *SIAM Journal on Scientific Computing* 32.5 (2010), pp. 2485–2496.

[50] R. Eid. "Time domain model reduction by moment matching". PhD thesis. Technische Universität München, 2009.

[51] R. Eid, H. Panzer, and B. Lohmann. *How to choose a single expansion point in Krylov-based model reduction?* Technical Reports on Automatic Control 2. Institute of Automatic Control, Technical University of Munich, Sept. 2009. URL: http://mediatum.ub.tum.de/node?id=1072354.

[52] K. Fan. "On strictly dissipative matrices". In: *Linear Algebra and Its Applications* 9 (1974), pp. 223–241.

[53] J. Fehr, M. Fischer, B. Haasdonk, and P. Eberhard. "Greedy-based approximation of frequency-weighted Gramian matrices for model reduction in multibody dynamics". In: *ZAMM - Zeitschrift für Angewandte Mathematik und Mechanik* 93.8 (2013), pp. 501–519.

[54] J. Fehr. "Automated and Error Controlled Model Reduction in Elastic Multibody Systems". PhD thesis. Universität Stuttgart, 2011.

[55] P. Feldmann and R. Freund. "Efficient linear circuit analysis by Padé approximation via the Lanczos process". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 14.5 (1995), pp. 639–649.

[56] L. Feng and P. Benner. "Automatic model order reduction by moment-matching according to an efficient output error bound". In: *Scientific Computing in Electrical Engineering*. Zürich, Switzerland, Sept. 2012.

[57] R. P. Feynman, R. B. Leighton, and M. Sands. *The Feynman Lectures on Physics, BD 1: Mainly Mechanics, Radiation and Heat*. Addison-Wesley, 1963.

[58] M. Fischer and P. Eberhard. "Automatisierte Modellreduktion großer elastischer Mehrkörpersysteme durch die Greedy-basierte Approximation der Gramschen Matrizen". In: *at-Automatisierungstechnik* 8 (2013), pp. 557–566.

[59] G. M. Flagg, C. A. Beattie, and S. Gugercin. "Convergence of the iterative rational Krylov algorithm". In: *Systems & Control Letters* 61 (2012), pp. 688–691.

[60] G. M. Flagg, C. A. Beattie, and S. Gugercin. "Interpolatory $\mathcal{H}_\infty$ model reduction". In: *Systems & Control Letters* 62.7 (2013), pp. 567–574.

[61] M. Frangos and I. M. Jaimoukha. "Adaptive rational Krylov algorithms for model reduction". In: *European Control Conference*. Kos, Greece, July 2007.

[62] M. Frangos and I. M. Jaimoukha. “Adaptive rational interpolation: restarting methods for a modified rational arnoldi algorithm”. In: *European Control Conference*. Budapest, Hungary, Aug. 2009.

[63] M. Frangos and I. M. Jaimoukha. “Adaptive rational interpolation: Arnoldi and Lanczos-like equations”. In: *European Journal of Control* 14 (2008), pp. 342–354.

[64] M. Frangos and I. M. Jaimoukha. “Rational interpolation: modified rational Arnoldi algorithm and Arnoldi-like equations”. In: *46th IEEE Conference on Decision and Control*. New Orleans, USA, Dec. 2007.

[65] R. W. Freund. “Model reduction methods based on Krylov subspaces”. In: *Acta Numerica* 12 (2003), pp. 267–319.

[66] K. Gallivan, A. Vandendorpe, and P. Van Dooren. “Model reduction via truncation: an interpolation point of view”. In: *Linear Algebra and Its Applications* 375 (2003), pp. 115–134.

[67] K. A. Gallivan, A. Vandendorpe, and P. Van Dooren. “Model reduction of MIMO systems via tangential interpolation”. In: *SIAM Journal on Matrix Analysis and Applications* 26.2 (2004), pp. 328–349.

[68] K. A. Gallivan, A. Vandendorpe, and P. Van Dooren. “Model reduction and the solution of Sylvester equations”. In: *17th International Symposium on Mathmatical Theory of Networks and Systems*. Kyoto, Japan, July 2006.

[69] K. A. Gallivan, A. Vandendorpe, and P. Van Dooren. “Sylvester equations and projection-based model reduction”. In: *Journal of Computational and Applied Mathematics* 162.1 (2004), pp. 213–229.

[70] K. Glover. “All optimal Hankel-norm approximations of linear multivariable systems and their L-error bounds”. In: *International Journal of Control* 39.6 (1984), pp. 1115–1193.

[71] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1996.

[72] R. D. Grigorieff. “A Note on von Neumann’s Trace Inequality”. In: *Math. Nachr* 151 (1991), pp. 327–328.

[73] E. J. Grimme. “Krylov Projection Methods for Model Reduction”. PhD thesis. Dep. of Electrical Eng., Uni. Illinois at Urbana Champaign, 1997.

[74] S. Gugercin, C. A. Beattie, and A. C. Antoulas. “Rational Krylov methods for optimal $\mathcal{H}_2$ model reduction”. In: *ICAM Technical Report* (2006).

[75] S. Gugercin and A. C. Antoulas. "A survey of model reduction by balanced truncation and some new results". In: *International Journal of Control* 77.8 (2004), pp. 748–766.

[76] S. Gugercin, A. C. Antoulas, and C. A. Beattie. "$\mathcal{H}_2$ model reduction for large-scale linear dynamical systems". In: *SIAM Journal on Matrix Analysis and Applications* 30.2 (2008), pp. 609–638.

[77] Y. Halevi. "Can any reduced order model be obtained via projection?" In: *American Control Conference*. Boston, Massachusetts, 2004.

[78] J. S. Han. "Efficient frequency response and its direct sensitivity analyses for large-size finite element models using Krylov subspace-based model order reduction". In: *Journal of mechanical science and technology* 26.4 (2012), pp. 1115–1126.

[79] I. Higueras and B. García-Celayeta. "How Close Can the Logarithmic Norm of a Matrix Pencil Come to the Spectral Abscissa". In: *SIAM Journal on Matrix Analysis and Applications* 22.2 (2000), pp. 472–478.

[80] I. Higueras and B. García-Celayeta. "Logarithmic Norms for Matrix Pencils". In: *SIAM J. Matrix Anal. Appl.* 20 (3 May 1999), pp. 646–666.

[81] A. S. Hodel and K. Poolla. "Heuristic approaches to the solution of very large sparse Lyapunov and algebraic Riccati equations". In: *27th IEEE Conference Decision and Control*. Austin, Texas, 1988, pp. 2217–2222.

[82] C. Hsu, U. Desai, and R. Darden. "Reduction of large-scale systems via generalized Gramians". In: *22nd IEEE Conference on Decision and Control*. Vol. 22. 1983, pp. 1409–1410.

[83] D. C. Hyland and D. Bernstein. "The optimal projection equations for model reduction and the relationships among the methods of Wilson, Skelton, and Moore". In: *IEEE Transactions on Automatic Control* 30.12 (1985), pp. 1201–1211.

[84] S. Jaensch. *$\mathcal{H}_2$-optimale Entwicklungspunktwahl bei der Modellordnungsreduktion mit Krylow-Unterraum-Verfahren*. Diplomarbeit, Lehrstuhl für Regelungstechnik, TUM. 2012.

[85] I. M. Jaimoukha and E. M. Kasenally. "Implicitly restarted Krylov subspace methods for stable partial realizations". In: *SIAM Journal on Matrix Analysis and Applications* 18.3 (1997), pp. 633–652.

[86] T. Kailath. *Linear Systems*. Prentice-Hall, Inc., New Jersey, 1980.

[87] M. Kamon, F. Wang, and J. White. “Generating nearly optimally compact models from Krylov-subspace based reduced-order models”. In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 47.4 (2000), pp. 239–248.

[88] H. Kiendl, J. Adamy, and P. Stelzner. “Vector norms as Lyapunov functions for linear systems”. In: *IEEE Transactions on Automatic Control* 37.6 (1992), pp. 839–842.

[89] B. Kleinherne. *Strikt dissipative Modellierung von Systemen zweiter Ordnung im Zustandsraum*. Semesterarbeit, Lehrstuhl für Regelungstechnik, TUM. 2012.

[90] L. Knockaert and D. De Zutter. “Laguerre-SVD Reduced-Order Modeling”. In: *IEEE Transaction on Microwave Theory and Techniques* 48.9 (2000), pp. 1469–1475.

[91] A. Kohl. *Error Controlled Model Order Reduction by Krylov Subspace Methods*. Master’s Thesis, Lehrstuhl für Regelungstechnik, TUM. 2014.

[92] A. Köhler, S. Reitz, and P. Schneider. “Sensitivity analysis and adaptive multi-point multi-moment model order reduction in MEMS design”. In: *Analog Integrated Circuits and Signal Processing* 71.1 (2012), pp. 49–58.

[93] N. Komaroff. “Upper bounds for the eigenvalues of the solution of the Lyapunov matrix equation”. In: *IEEE Transactions on Automatic Control* 35.6 (June 1990), pp. 737–739.

[94] Y. Konkel, O. Farle, and R. Dyczij-Edlinger. “Ein Fehlerschätzer für die Krylov-Unterraum basierte Ordnungsreduktion zeitharmonischer Anregungsprobleme”. In: *Tagungsband GMA-Fachausschuss 1.30*. 2008.

[95] Y. Konkel et al. “A Posteriori Error Bounds for Krylov-Based Fast Frequency Sweeps of Finite-Element Systems”. In: *IEEE Transactions on Magnetics* 50.2 (2014), pp. 441–444.

[96] J. Korvink and E. Rudnyi. “Oberwolfach Benchmark Collection”. In: *Dimension Reduction of Large-Scale Systems*. Ed. by P. Benner, V. Mehrmann, and D. C. Sorensen. Vol. 45. Lecture Notes in Computational Science and Engineering. Springer-Verlag, Berlin/Heidelberg, Germany, 2005, pp. 311–315. URL: http://portal.uni-freiburg.de/imteksimulation/downloads/benchmark.

[97] P. Koutsovasilis. “Model Order Reduction in Structural Mechanics: Coupling the Rigid and Elastic Multi Body Dynamics”. PhD thesis. Technische Universität Dresden, 2009.

[98] S. Lefteriu and A. C. Antoulas. "A new approach to modeling multiport systems from frequency-domain data". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 29.1 (Jan. 2010), pp. 14–27.

[99] S. Lefteriu and A. C. Antoulas. "Modeling multi-port systems from frequency response data via tangential interpolation". In: *IEEE Workshop on Signal Propagation on Interconnects*. May 2009, pp. 1–4.

[100] S. Lefteriu, A. C. Antoulas, and A. C. Ionita. "Parametric model order reduction from measurements". In: *19th IEEE Conference on Electrical Performance of Electronic Packaging and Systems*. IEEE. 2010, pp. 193–196.

[101] M. Lehner. "Modellreduktion in elastischen Mehrkörpersystemen". PhD thesis. Universität Stuttgart, 2007.

[102] J.-R. Li and M. Kamon. "PEEC model of a spiral inductor generated by Fasthenry". In: *Dimension Reduction of Large-Scale Systems*. Springer, 2005, pp. 373–377.

[103] J.-R. Li and J. White. "Low rank solution of Lyapunov equations". In: *SIAM Journal on Matrix Analysis and Applications* 24.1 (2002), pp. 260–280.

[104] J. Lienemann et al. "MEMS compact modeling meets model order reduction: Examples of the application of Arnoldi methods to microsystem devices". In: *The Technical Proceedings of the 2004 Nanotechnology Conference and Trade Show, Nanotech*. Vol. 4. 2004.

[105] L. Litz. "Modale Maße für Steuerbarkeit, Beobachtbarkeit, Regelbarkeit und Dominanz – Zusammenhänge, Schwachstellen, neue Wege". In: *Regelungstechnik* 31 (1983), pp. 148–158.

[106] B. Lohmann and R. Eid. "Efficient Order Reduction of Parametric and Nonlinear Models by Superposition of Locally Reduced Models". In: *Methoden und Anwendungen der Regelungstechnik – Erlangen-Münchener Workshops 2007 und 2008*. Boris Lohmann and Günter Roppenecker (Hrsg.), 2009.

[107] S. M. Lozinskii. "Error estimate for numerical integration of ordinary differential equations". In: *Izvestiya Vysshikh Uchebnykh Zavedenii Matematika* 5 (1958), pp. 52–90.

[108] H. Lütkepohl. *Handbook of matrices*. John Wiley & Sons, 1996.

[109] A. MacFarlane and N Karcanias. "Poles and zeros of linear multivariable systems: a survey of the algebraic, geometric and complex-variable theory". In: *International Journal of Control* 24.1 (1976), pp. 33–74.

[110] A. Mayo and A. C. Antoulas. “A framework for the solution of the generalized realization problem”. In: *Linear Algebra and Its Applications* 425.2–3 (2007), pp. 634–662.

[111] L. Meier and D. G. Luenberger. “Approximation of Linear Constant Systems”. In: *IEEE Transactions on Automatic Control* 12.5 (Oct. 1967), pp. 585–588.

[112] L. Mirsky. “A Trace Inequality of John von Neumann”. In: *Monatshefte für Mathematik* 79.4 (1975), pp. 303–306.

[113] L. Mirsky. “On the Trace of Matrix Products”. In: *Mathematische Nachrichten* 20.3–6 (1959), pp. 171–174.

[114] *Model Order Reduction (MOR) Wiki*. Max-Planck-Institut Magdeburg. URL: http://morwiki.mpi-magdeburg.mpg.de.

[115] C. Moler and C. Van Loan. “Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later”. In: *SIAM Review* 45.1 (2003), pp. 3–49.

[116] C. Moler and C. Van Loan. “Nineteen Dubious Ways to Compute the Exponential of a Matrix”. In: *SIAM Review* 20.4 (Oct. 1978), pp. 801–836.

[117] B. C. Moore. “Principal component analysis in linear systems: controllability, observability and model reduction”. In: *IEEE Transactions on Automatic Control* AC-26 (1981), pp. 17–32.

[118] S. Moschik and N. Dourdoumas. “Steuerbarkeitsmaße für lineare zeitinvariante Systeme-Ein Überblick”. In: *International Journal Automation Austria* 19.1 (2011), pp. 1–14.

[119] A. Odabasioglu, M. Celik, and L. T. Pileggi. “PRIMA: passive reduced-order interconnect macromodeling algorithm”. In: *IEEE/ACM international conference on Computer-Aided Design*. IEEE Computer Society. 1997, pp. 58–65.

[120] A. Odabasioglu, M. Celik, and L. T. Pileggi. “PRIMA: passive reduced-order interconnect macromodeling algorithm”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 17.8 (1998), pp. 645–654.

[121] A. Odabasioglu, M. Celik, and L. T. Pileggi. “Practical considerations for passive reduction of RLC circuits”. In: *IEEE/ACM international Conference on Computer-Aided Design*. IEEE Press. 1999, pp. 214–220.

[122] H. K. F. Panzer, S. Jaensch, T. Wolf, and B. Lohmann. “A greedy rational Krylov method for $\mathcal{H}_2$-pseudooptimal model order reduction with preservation of stability”. In: *American Control Conference*. 2013, pp. 5532–5537.

[123] H. K. F. Panzer, B. Kleinherne, and B. Lohmann. "Analysis, Interpretation and Generalization of a Strictly Dissipative State Space Formulation of Second Order Systems". In: *Methoden und Anwendungen der Regelungstechnik – Erlangen-Münchener Workshops 2011 und 2012*. Boris Lohmann and Günter Roppenecker (Hrsg.), 2013. URL: http://mediatum.ub.tum.de/doc/1175736.

[124] H. K. F. Panzer, T. Wolf, and B. Lohmann. "$\mathcal{H}_2$ and $\mathcal{H}_\infty$ error bounds for model order reduction of second order systems by Krylov subspace methods". In: *European Control Conference*. 2013, pp. 4484–4489.

[125] H. Panzer, J. Hubele, R. Eid, and B. Lohmann. *Generating a Parametric Finite Element Model of a 3D Cantilever Timoshenko Beam Using MATLAB*. Technical Reports on Automatic Control. Lehrstuhl für Regelungstechnik, Technische Universität München, Sept. 2009.

[126] H. Panzer, J. Mohring, R. Eid, and B. Lohmann. "Parametric MModel order reduction by matrix interpolation". In: *at–Automatisierungstechnik* 58.8 (Aug. 2010), pp. 475–484.

[127] H. Panzer, T. Wolf, and B. Lohmann. "A strictly dissipative state space representation of second order systems". In: *at–Automatisierungstechnik* 60 (2012), pp. 392–396.

[128] K. B. Petersen and M. S. Pedersen. *The matrix cookbook*. Tech. rep. Technical University of Denmark, 2005.

[129] J. W. Polderman and J. C. Willems. *Introduction to Mathematical Systems Theory: A Behavioral Approach*. 26. Springer, 1998.

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

[131] M. Rewienski and J. White. "A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 22.2 (2003), pp. 155–170.

[132] J. Rommes. *Homepage of Joost Rommes*. URL: https://sites.google.com/site/rommes/software.

[133] J. Rommes and N. Martins. "Computing large-scale system eigenvalues most sensitive to parameter changes, with applications to power system small-signal stability". In: *IEEE Transactions on Power Systems* 23.2 (2008), pp. 434–442.

[134] H. H. Rosenbrock. *State-space and Multivariable Theory*. Thomas Nelson and Sons LTD., 1970.

[135] A. Ruhe. "Rational Krylov sequence methods for eigenvalue computation". In: *Linear Algebra and Its Applications* 58 (1984), pp. 391–405.

[136] Y. Saad. "Analysis of some Krylov subspace approximations to the matrix exponential operator". In: *SIAM Journal on Numerical Analysis* 29.1 (1992), pp. 209–228.

[137] J. Sabino. "Solution of Large-Scale Lyapunov Equations via the Block Modified Smith Method". PhD thesis. Rice Univ. Houston, 2007.

[138] B. Salimbahrami. "Structure Preserving Order Reduction of Large Scale Second Order Models". PhD thesis. Technische Universität München, 2005.

[139] B. Salimbahrami and B. Lohmann. *Krylov Subspace Methods in Linear Model Order Reduction: Introduction and Invariance Properties*. Tech. rep. Institute of Automation, University of Bremen, 2002.

[140] W. H. Schilders, H. A. Van Der Vorst, and J. Rommes. *Model Order Reduction: Theory, Research Aspects and Applications*. Springer, Berlin, 2008.

[141] G. Söderlind. "The Logarithmic Norm. History and Modern Theory". In: *BIT Numerical Mathematics* 46 (2006), pp. 631–652.

[142] L. M. Silveira, M. Kamon, I. Elfadel, and J. White. "A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits". In: *IEEE/ACM International Conference on Computer-Aided Design*. 1996, pp. 288–294.

[143] L. M. Silveira, M. Kamon, I. Elfadel, and J. White. "A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits". In: *Computer Methods in Applied Mechanics and Engineering* 169.3-4 (1999), pp. 377–389.

[144] V. Simoncini. "A new iterative method for solving large-scale Lyapunov matrix equations". In: *SIAM Journal on Scientific Computing* 29.3 (2007), pp. 1268–1288.

[145] V. Simoncini. "On the Numerical Solution of AX-XB=C". In: *BIT Numerical Mathematics* 36 (4 1996). 10.1007/BF01733793, pp. 814–830.

[146] R. E. Skelton and A. Yousuff. "Component cost analysis of large scale systems". In: *International Journal of Control* 37.2 (1983), pp. 285–304.

[147] A. Sommer, O. Farle, and R. Dyczij-Edlinger. "Efficient finite-element computation of far-fields of phased arrays by order reduction". In: *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering* 32.5 (2013), pp. 1721–1734.

[148] D. Sorensen, C. Teng, and A. C. Antoulas. "Derivation of an H2 error bound for model reduction of second order systems". In: *16th International Symposium on Mathematical Theory of Networks and Systems, Leuven, Belgium*. 2004.

[149] Y. Stürz. *Model Order Reduction of MIMO Systems by Krylov Subspace Methods*. Master's Thesis, Lehrstuhl für Regelungstechnik, TUM. 2014.

[150] The MathWorks, Inc. *Matlab Documentation*.

[151] R. Thompson. "Dissipative Matrices and the Matrix $A^{-1}A^*$". In: *Houston Journal of Mathematics* 1.1 (1975), pp. 137–147.

[152] L. Trefethen and M. Embree. *Spectra and pseudospectra: the behavior of nonnormal matrices and operators*. Princeton Univ Pr, 2005.

[153] E. Tyrtyshnikov. "Mosaic-Skeleton approximations". In: *Calcolo* 33 (1996), pp. 47 –57.

[154] A. Unger and F. Tröltzsch. "Fast solution of optimal control problems in the selective cooling of steel". In: *Zeitschrift für Angewandte Mathematik und Mechanik ZAMM* 81.7 (2001), pp. 447–456.

[155] P. Van Dooren, K. A. Gallivan, and P.-A. Absil. "$\mathcal{H}_2$-optimal model reduction of MIMO systems". In: *Applied Mathematics Letters* 21.12 (2008), pp. 1267–1273.

[156] P. Van Dooren, K. A. Gallivan, and P.-A. Absil. "$\mathcal{H}_2$-optimal model reduction with higher-order poles". In: *SIAM Journal on Matrix Analysis and Applications* 31.5 (2010), pp. 2738–2753.

[157] A. Vandendorpe. "Model Reduction of Linear Systems, an Interpolation Point of View". PhD thesis. Université Catholique De Louvain, Dec. 2004.

[158] A. Varga. "Enhanced modal approach for model reduction". In: *Mathematical Modelling of Systems* 1.2 (1995), pp. 91–105. eprint: http://www.tandfonline.com/doi/pdf/10.1080/13873959508837010.

[159] J. F. Villena and L. M. Silveira. "ARMS – Automatic Residue-minimization based Sampling for Multi-Point Modeling Techniques". In: *46th ACM/IEEE Design Automation Conference*. IEEE. 2009, pp. 951–956.

[160] J. F. Villena and L. M. Silveira. "Multi-dimensional automatic sampling schemes for multi-point modeling methodologies". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30.8 (2011), pp. 1141–1151.

[161] J. C. Willems. "Dissipative dynamical systems Part I: General theory". In: *Archive for Rational Mechanics and Analysis* 45.5 (1972), pp. 321–351.

[162] D. A. Wilson. "Optimum solution of model-reduction problem". In: *Proceedings of the Institution of Electrical Engineers*. Vol. 117. 6. IET. 1970, pp. 1161–1165.

[163] T. Wolf, H. K. F. Panzer, and B. Lohmann. "H2 pseudo-optimality in model order reduction by Krylov subspace methods". In: *European Control Conference*. 2013.

[164] T. Wolf. "$\mathcal{H}_2$ Pseudo-Optimal Model Order Reduction". PhD thesis. Technische Universität München, submitted.

[165] T. Wolf, H. K. F. Panzer, and B. Lohmann. "Gramian-based error bound in model reduction by Krylov-subspace methods". In: *18th IFAC World Congress*. Milano, Italy, 2011, pp. 3587–3592.

[166] T. Wolf, H. K. F. Panzer, and B. Lohmann. "Sylvester equations and a factorization of the error system in Krylov-based model reduction". In: *Vienna Conference on Mathematical Modelling (MATHMOD)*. 2012.

[167] T. Wolf and H. K. F. Panzer. *The ADI iteration for Lyapunov equations implicitly performs $\mathcal{H}_2$ pseudo-optimal model order reduction*. Sept. 2013. URL: http://arxiv.org/abs/1309.3985.

[168] S. Wyatt. "Issues in Interpolatory Model Reduction: Inexact Solves, Second-order Systems and DAEs". PhD thesis. Virginia Polytechnic Institute and State University, 2012.

[169] Y. Xie et al. "UHF micromechanical extensional wine-glass mode ring resonators". In: *IEEE International Electron Devices Meeting*. Dec. 2003, pp. 39.2.1–39.2.4.

[170] W. Zhao, G. K. Pang, and N. Wong. "Automatic adaptive multi-point moment matching for descriptor system model order reduction". In: *VLSI Design, Automation, and Test (VLSI-DAT), 2013 International Symposium on*. 2013, pp. 1–4.

[171] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Vol. 40. Prentice Hall New Jersey, 1996.