Lehrstuhl für Regelungstechnik

Technische Universität München

A Black-Box Method for Parametric Model Order Reduction based on Matrix Interpolation with Application to Simulation and Control

Matthias Martin Georg Geuß

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. Wolfgang Polifke, Ph.D.
Prüfer der Dissertation:	1. Univ.-Prof. Dr.-Ing. habil. Boris Lohmann
	2. Prof. Karen Willcox, Ph.D. Massachusetts Institute of Technology/USA

Die Dissertation wurde am 02.07.2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 25.09.2015 angenommen.

Abstract

Accurate modeling of dynamical systems can lead to large-scale systems of ordinary differential equations. For many applications, these systems additionally depend on parameters, which, for example, describe geometric attributes. In order to be able to perform tasks such as optimization, simulation or control, it is necessary to reduce the computational effort of solving the system. For that reason, methods of parametric model order reduction have been developed which reduce the order of the large-scale system and at the same time preserve its parametric dependencies.

This thesis deals with a novel approach for model order reduction of parameter-dependent systems. The method calculates a set of reduced systems with individual projection matrices for a number of grid points. Subsequently, the bases of the local systems are adjusted. In order to obtain a model for a new parameter value, the reduced system matrices are interpolated. A general framework is proposed which can be applied by the user as a construction kit. It illustrates the necessary steps and presents the different options for each step. In particular, three approaches for adjusting the bases are proposed and it is pointed out that the user can choose between a large number of candidate interpolation methods and manifolds. In many cases, the user lacks insight into the physics of the model in order to decide on the options leading to the most accurate reduced model. Hence, a Black-Box method is proposed that automatically determines the optimal interpolation method and grid points and delivers a reduced system with the desired accuracy. In addition, the general framework is extended to stability preservation. For this, a low-order optimization problem based on semidefinite programming is solved for every grid point. Moreover, as common grid-based approaches lead to costs that grow exponentially with the number of parameters, sparse-grid-based interpolation is introduced for the interpolation of system matrices. This allows the user to apply the framework to high-dimensional parameter spaces. The general framework is also extended to the interpolation of differently-sized reduced models. For this, a resizing procedure, which is based on pseudoinverses, is presented.

The effectiveness of the proposed methods is shown for examples from microsystems technology, mechatronics and structural mechanics. In addition, the application of the interpolation-based framework to control is demonstrated for a system on a test rig.

Acknowledgments

The results presented in this thesis were developed during my time as a research assistant at the Institute of Automatic Control, Technische Universität München.

My first thanks go to Prof. Boris Lohmann. He offered me the opportunity to be part of his research group and was the first examiner of my thesis. Second, I would like to thank Prof. Karen Willcox for her interest in my research and for being my second examiner. I am also very glad that she gave me the opportunity to be a member of her research group at MIT. A lot of new ideas originated from this collaboration.

My thanks also go to the members of the Institute of Automatic Control for the excellent atmosphere on and off the job. Especially, I would like to thank my office mates Thomas Wolf and Heiko Panzer who introduced me into the research field of model order reduction. I am also grateful to my students Jürgen Hödl, Cholwoo Choi, Sebastian Carter and Tobias Englmaier. I would also like to thank Maria Cruz Varona, Alessandro Castagnotto, Alexander Lange and Benedikt Jäger for carefully reading the manuscript of this thesis.

I would also like to thank Benjamin Peherstorfer and Daniel Butnaru for all the fruitful discussions about sparse grids and machine learning.

Last but not least, I want to thank my parents Kunigunda and Ottmar and my brother Michael for their love, care and support throughout all my years of education.

Contents

List of Figures

List of Tables

Glossary

Abbreviations

IRKA	Iterative rational Krylov algorithm
LMI	Linear matrix inequality
LTI	Linear time-invariant
MAC	Modal assurance criterion
MIMO	Multiple-input multiple-output
MPC	Model predictive control
MOR	Model order reduction
ODE	Ordinary differential equation
pMOR	Parametric model order reduction
PDE	Partial differential equation
POD	Proper orthogonal decomposition
RBF	Radial basis function
ROB	Reduced order basis
ROM	Reduced order model
SISO	Single-input single-output
STABLE	Stability algorithm based on linear matrix inequalities
SVD	Singular value decomposition
TBR	Truncated balanced realization

Symbols

$\mathbb{C}$	Set of complex numbers
$\mathbb{C}^{n \times n}$	Set of complex $n \times n$ matrices
$\mathbb{E}$	Expectation of a random variable
$\mathbb{GL}_q$	Set of nonsingular $n \times n$ matrices
$\mathbb{N}$	Set of natural numbers
$\mathbb{O}_n$	Set of orthogonal $n \times n$ matrices

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^+$	Set of real positive numbers
$\mathbb{R}_0^+$	Set of real nonnegative numbers
$\mathbb{R}^{n\times n}$	Set of real $n \times n$ matrices
$\mathbb{S}_n$	Set of symmetric $n \times n$ matrices
$\mathbb{S}_n^+$	Set of symmetric positive semidefinite $n \times n$ matrices
$\mathbb{S}_n^{++}$	Set of symmetric positive definite $n \times n$ matrices
$\mathbb{Z}_0^+$	Set of nonnegative integers
$\mathcal{D}$	Parameter domain
$\mathbf{G}$	MIMO LTI system
G	SISO LTI system
$\mathbf{G}(\mathbf{p})$	Parameter-dependent high-order system, order n
$\hat{\mathbf{G}}(\mathbf{p})$	Parameter-dependent reduced order model, order $q \ll n$
$\tilde{\mathbf{G}}(\mathbf{p})$	Interpolated reduced system, order $q \ll n$
$\mathbf{G}_i$	Local system $\mathbf{G}_i := \mathbf{G}(\mathbf{p}_i)$ at parameter vector $\mathbf{p}_i$
$\mathbf{I}_n$	Identity matrix of size n

Matrix Operations

$\mathbf{A}(\mathbf{p})$	Parameter-dependent matrix
$\mathbf{A}_i$	Matrix $\mathbf{A}_i := \mathbf{A}(\mathbf{p}_i)$ at parameter vector $\mathbf{p}_i$
$\mathbf{A} \geq \mathbf{0}$	Symmetric positive semidefinite matrix
$\mathbf{A} > \mathbf{0}$	Symmetric positive definite matrix
$\mathbf{A} \leq \mathbf{0}$	Symmetric negative semidefinite matrix
$\mathbf{A} < \mathbf{0}$	Symmetric negative definite matrix
$\mathbf{A}^T$	Matrix transposition
$\mathbf{A}^{-T}$	Matrix transposition and inversion
$\mathbf{A}^{-1}$	Matrix inversion
$\mathbf{P}_{\mathbf{A}\perp\mathbf{B}}$	Projector into the subspace spanned by $\mathbf{A}$ orthogonal to the subspace spanned by $\mathbf{B}$
$\ker(\mathbf{A})$	Kernel of the matrix
$\operatorname{rank}(\mathbf{A})$	Rank of the matrix
$\operatorname{span}(\mathbf{A})$	Linear hull of the matrix
$\operatorname{tr}(\mathbf{A})$	Trace of the matrix

Part I

Preliminaries

1 Introduction

1.1 Motivation

Modern applications in engineering make high demands on the accuracy of dynamical models. In many cases, technical or physical systems are modeled using partial differential equations (PDE) such as in the context of heat transfer, fluid dynamics or structural mechanics. This is often followed by a spatial discretization using e.g. the finite element method which results in a system of ordinary differential equations (ODE). Obviously, a finer grid leads to a more accurate model, but on the other hand it increases the number of ODEs. Therefore, the need for accurate models often leads to large-scale systems. If these systems are used for the purpose of optimization, simulation or control, even powerful computer systems reach their limitations, especially concerning execution time and available memory. As a remedy, methods of model order reduction (MOR) have been developed which approximate the input-output behavior of the original system by a low-order one so that the error is supposed to be small with respect to a certain norm. In addition, properties of the high-dimensional system like stability should be preserved during the reduction step. For many engineering applications, the high-order system additionally depends on parameters, for example geometry or material parameters. For this case, methods of parametric model order reduction (pMOR) have been developed. They construct a low-order system which approximates the large-scale one for the entire parameter domain. Hence, one can obtain a reduced system for every parameter value of the domain without the need to repeat the reduction procedure. The field of pMOR has become increasingly important in the last years for both scientific and industrial applications. Especially for the latter case, there is a demand for automatic and computationally cheap procedures which deliver a reduced system with a desired accuracy. For that reason, a method of pMOR is developed in this thesis which is particularly efficient as it only requires the interpolation of low-order system matrices when a reduced system is requested for a new parameter value. Firstly, the different steps of this method and the options for each step including properties like stability preservation are presented. Secondly, a Black-Box method is proposed which automatically determines the optimal design parameters and hence, can easily be applied by the user. Thirdly, the benefit of this method is demonstrated for examples from simulation and control.

1.2 Thesis Accomplishments and Outline

The following overview gives the outline of this thesis and briefly summarizes its accomplishments.

Part I: Preliminaries

In Section 2, the system of interest is introduced. In addition, background information on model order reduction is presented and common methods are reviewed. Afterwards, parametric model order reduction is introduced and a literature overview is given for existing methods. Finally, the contributions of this thesis are arranged into the state of the art of parametric model order reduction.

Part II: A Black-Box Method for Parametric Model Order Reduction by Matrix Interpolation

In Section 3, a general framework for parametric model order reduction based on interpolating reduced system matrices is proposed. It combines and extends existing methods to a construction kit which illustrates the necessary steps and presents the different options for each step. Parts of the results presented in this section have been published in [76]. Selecting the options which lead to the most accurate reduced system often confronts the user with difficult tasks. In order to provide the user a tool which selects the best options, a Black-Box method is proposed in Section 4. It automatically selects the best interpolation method, performs the refinement in the parameter space iteratively and stops when the maximum predicted error of the interpolated system falls below a given error tolerance.

Part III: Further Results for Interpolating Reduced System Matrices

In Section 5, the proposed framework for parametric model order reduction is extended to stability preservation. The approach is based on semidefinite programming and results in low-order optimization problems which can efficiently be solved. Parts of the results presented in this section have been published in [77]. In Section 6, the general framework is combined with sparse grids in order to extend it to multidimensional parameter spaces. The results presented in this section have been published in [72]. In Section 7, the framework is extended to the interpolation of differently-sized reduced systems. This approach applies pseudoinverses which resize the reduced system matrices to the same size. The results presented in this section have been published in [75]. In

Section 8, the interpolation-based framework is applied to the control of a system at a test rig using a two-degree-of-freedom control and a model predictive controller. Parts of the results presented in this section have been published in [73]. In Section 9, concluding remarks are presented and possible future directions of research are suggested.

Appendix

Supplementary material is provided in several appendices, referenced at appropriate places. Appendix A gives a review of different interpolation methods. Appendix B provides the proof of a proposition and in Appendix C the notion of convex cones is introduced. In Appendix D, background information on pseudoinverses is presented.

2 Model Order Reduction

2.1 Parameter-Dependent Linear Time-Invariant Systems

2.1.1 System Representation

In this thesis a parameter-dependent, high-dimensional LTI system of order n is considered. Its representation in the time domain is given by

$$\mathbf{G}(t,\mathbf{p}) : \begin{cases} \mathbf{E}(\mathbf{p})\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{p})\mathbf{x}(t) + \mathbf{B}(\mathbf{p})\mathbf{u}(t) \\ \qquad\;\; \mathbf{y}(t) = \mathbf{C}(\mathbf{p})\mathbf{x}(t), \end{cases} \tag{2.1}$$

where $\mathbf{E}(\mathbf{p}) \in \mathbb{R}^{n\times n}$, $\mathbf{A}(\mathbf{p}) \in \mathbb{R}^{n\times n}$, $\mathbf{B}(\mathbf{p}) \in \mathbb{R}^{n\times r}$ and $\mathbf{C}(\mathbf{p}) \in \mathbb{R}^{m\times n}$ are the system matrices which depend on the parameter vector $\mathbf{p} \in \mathcal{D}$ with domain $\mathcal{D} \subseteq \mathbb{R}^{d}$. The vectors $\mathbf{u}(t) \in \mathbb{R}^{r}$, $\mathbf{y}(t) \in \mathbb{R}^{m}$ and $\mathbf{x}(t) \in \mathbb{R}^{n}$ denote the inputs, outputs and states of the system at time t with initial value $\mathbf{x}(0) = \mathbf{x}_0$. The parameter dependency of the states and the outputs is omitted. The system $\mathbf{G}(t,\mathbf{p})$ realizes a *multiple-input multiple-output* (MIMO) model, whereas $G(t,\mathbf{p})$ is a *single-input single-output* (SISO) system with $m = r = 1$. For the SISO case we have $\mathbf{B}(\mathbf{p}) \to \mathbf{b}(\mathbf{p}) \in \mathbb{R}^{n}$, $\mathbf{C}(\mathbf{p}) \to \mathbf{c}(\mathbf{p})^T \in \mathbb{R}^{1\times n}$, $\mathbf{u}(t) \to u(t) \in \mathbb{R}$ and $\mathbf{y}(t) \to y(t) \in \mathbb{R}$. Performing a Laplace transformation on system (2.1) and setting $\mathbf{x}_0 = \mathbf{0}$, we obtain the representation of the system in the frequency domain

$$\mathbf{G}(s,\mathbf{p}) = \mathbf{C}(\mathbf{p})\Big[s\mathbf{E}(\mathbf{p}) - \mathbf{A}(\mathbf{p})\Big]^{-1}\mathbf{B}(\mathbf{p}), \tag{2.2}$$

where $s \in \mathbb{C}$ is the complex frequency. When we refer in the following to the system independent from its representation, it is denoted by $\mathbf{G}(\mathbf{p})$.

2.1.2 Stability

Stability is an important property of system (2.1). There exist various stability definitions [15, 99, 100]. In this thesis, the following concept of stability is examined.

Definition 2.1. The system (2.1) is said to be *asymptotically stable* for parameter vector $\mathbf{p}$ if one finds $\lim\limits_{t\to\infty} \mathbf{x}(t) = \mathbf{0}$ for all initial conditions $\mathbf{x}_0$.

The following theorem gives necessary and sufficient conditions.

Theorem 2.1 ([155])**.** *The system (2.1) with nonsingular* $\mathbf{E}(\mathbf{p})$ *is asymptotically stable for parameter vector* $\mathbf{p}$ *if and only if . . .*

- ∘ *S1: . . . there exists a Lyapunov function, i.e. a function* $V(\mathbf{x}(t), \mathbf{p}) : \mathbb{R}^n \to \mathbb{R}$ *with* $V(\mathbf{x}(t), \mathbf{p}) > 0$ *and* $\dot{V}(\mathbf{x}(t), \mathbf{p}) < 0$ *for all* $\mathbf{x}(t) \neq \mathbf{0}$.
- ∘ *S2: . . . for any given* $\mathbf{Q} \in \mathbb{S}_n^{++}$ *there exists a unique solution* $\mathbf{P}(\mathbf{p}) \in \mathbb{S}_n^{++}$ *to the generalized Lyapunov equation*

  $$\mathbf{E}^T(\mathbf{p})\mathbf{P}(\mathbf{p})\mathbf{A}(\mathbf{p}) + \mathbf{A}^T(\mathbf{p})\mathbf{P}(\mathbf{p})\mathbf{E}(\mathbf{p}) = -\mathbf{Q}. \tag{2.3}$$

  Then, a Lyapunov function is $V(\mathbf{x}(t), \mathbf{p}) = \mathbf{x}(t)^T\mathbf{E}^T(\mathbf{p})\mathbf{P}(\mathbf{p})\mathbf{E}(\mathbf{p})\mathbf{x}(t)$.
- ∘ *S3: . . . the eigenvalues of the pencil* $(\mathbf{A}(\mathbf{p}), \mathbf{E}(\mathbf{p}))$ *lie in the open left half of the complex plane.*

A special class of systems is of particular interest in this thesis.

Definition 2.2 ([127])**.** The system (2.1) is said to be *strictly dissipative* if it satisfies $\mathbf{E}(\mathbf{p}) > \mathbf{0}$ and $\mathbf{A}(\mathbf{p}) + \mathbf{A}(\mathbf{p})^T < \mathbf{0}$.

These kind of systems have the following stability property.

Corollary 2.1 ([41])**.** *A strictly dissipative system possesses the Lyapunov function* $V(\mathbf{x}(t), \mathbf{p}) = \mathbf{x}(t)^T\mathbf{E}(\mathbf{p})\mathbf{x}(t)$ *and hence, is asymptotically stable.*

Proof. For all $\mathbf{x}(t) \neq \mathbf{0}$ it holds

$$\begin{aligned} V(\mathbf{x}(t), \mathbf{p}) &= \mathbf{x}(t)^T \overbrace{\mathbf{E}(\mathbf{p})}^{>\mathbf{0}} \mathbf{x}(t) > 0, && (2.4)\\ \dot{V}(\mathbf{x}(t), \mathbf{p}) &= \dot{\mathbf{x}}(t)^T\mathbf{E}(\mathbf{p})\mathbf{x}(t) + \mathbf{x}(t)^T\mathbf{E}(\mathbf{p})\dot{\mathbf{x}}(t) && (2.5)\\ &= \mathbf{x}(t)^T\mathbf{A}(\mathbf{p})^T\mathbf{E}(\mathbf{p})^{-1}\mathbf{E}(\mathbf{p})\mathbf{x}(t) + \mathbf{x}(t)^T\mathbf{E}(\mathbf{p})\mathbf{E}(\mathbf{p})^{-1}\mathbf{A}(\mathbf{p})\mathbf{x}(t) && (2.6)\\ &= \mathbf{x}(t)^T \underbrace{\left(\mathbf{A}(\mathbf{p}) + \mathbf{A}(\mathbf{p})^T\right)}_{<\mathbf{0}} \mathbf{x}(t) < 0. && (2.7) \end{aligned}$$

■

Strictly dissipative systems have another interesting property which is used in Section 5 for proposing a stability-preserving interpolation method.

Corollary 2.2 ([27, 55])**.** *Given a set of* $i \in \{1, \ldots, N\}$ *strictly dissipative LTI systems, then the system obtained by superposition of the system matrices with weighting functions* $\omega_i(\mathbf{p}) \in \mathbb{R}_0^+$ *and at least one* $\omega_i(\mathbf{p}) \in \mathbb{R}^+$ *is asymptotically stable.*

Proof. With $\mathbf{E}_i > \mathbf{0}$ and $\mathbf{A}_i + \mathbf{A}_i^T < \mathbf{0}$ the matrices obtained by superposition are $\mathbf{A}(\mathbf{p}) = \sum_{i=1}^N \omega_i(\mathbf{p})\mathbf{A}_i$ and $\mathbf{E}(\mathbf{p}) = \sum_{i=1}^N \omega_i(\mathbf{p})\mathbf{E}_i$. The resulting system has the Lyapunov function $V(\mathbf{x}(t), \mathbf{p}) = \mathbf{x}(t)^T\mathbf{E}(\mathbf{p})\mathbf{x}(t)$ and hence, is asymptotically stable. It holds for all $\mathbf{x}(t) \neq \mathbf{0}$

$$V(\mathbf{x}(t), \mathbf{p}) = \mathbf{x}(t)^T\mathbf{E}(\mathbf{p})\mathbf{x}(t) = \mathbf{x}(t)^T\Big(\sum_{i=1}^N \omega_i(\mathbf{p})\mathbf{E}_i\Big)\mathbf{x}(t) = \sum_{i=1}^N \omega_i(\mathbf{p})\underbrace{\mathbf{x}(t)^T\mathbf{E}_i\mathbf{x}(t)}_{>0} > 0 \tag{2.8}$$

and starting from equation (2.7), we obtain

$$\dot{V}(\mathbf{x}(t), \mathbf{p}) = \mathbf{x}(t)^T\left(\mathbf{A}(\mathbf{p}) + \mathbf{A}(\mathbf{p})^T\right)\mathbf{x}(t) = \mathbf{x}(t)^T\Big(\sum_{i=1}^N \omega_i(\mathbf{p})\left(\mathbf{A}_i + \mathbf{A}_i^T\right)\Big)\mathbf{x}(t) \tag{2.9}$$

$$= \sum_{i=1}^N \omega_i(\mathbf{p})\underbrace{\mathbf{x}(t)^T\left(\mathbf{A}_i + \mathbf{A}_i^T\right)\mathbf{x}(t)}_{<0} < 0. \tag{2.10}$$

■

2.2 Projection-Based Model Order Reduction

2.2.1 Problem Formulation

In order to motivate the reduction of the parameter-dependent system $\mathbf{G}(\mathbf{p})$, let us first discuss model order reduction (MOR) of nonparametric systems. Such systems are obtained in this thesis when we compute the parametric system $\mathbf{G}(\mathbf{p})$ for a set of grid points $\mathbf{p}_i \in \mathcal{D}$ with $i \in \{1, \dots, N\}$. We denote $\mathbf{E}(\mathbf{p}_i)$ by $\mathbf{E}_i$ and similarly for the other matrices. Then, the i-th nonparametric system is given in the time domain by

$$\mathbf{G}_i(t) : \begin{cases} \mathbf{E}_i\dot{\mathbf{x}}_i(t) = \mathbf{A}_i\mathbf{x}_i(t) + \mathbf{B}_i\mathbf{u}(t) \\ \mathbf{y}_i(t) = \mathbf{C}_i\mathbf{x}_i(t). \end{cases} \tag{2.11}$$

The representation of the system in the frequency domain is

$$\mathbf{G}_i(s) = \mathbf{C}_i\left(s\mathbf{E}_i - \mathbf{A}_i\right)^{-1}\mathbf{B}_i. \tag{2.12}$$

If we refer to the system independent from its representation, it is denoted by $\mathbf{G}_i$.

Then, the goal of MOR is to find a low-dimensional system $\hat{\mathbf{G}}_i$ of order $q \ll n$ which approximates the high-order system $\mathbf{G}_i$. For this, projection matrices $\mathbf{V}_i := \mathbf{V}(\mathbf{p}_i) \in \mathbb{R}^{n\times q}$ and $\mathbf{W}_i := \mathbf{W}(\mathbf{p}_i) \in \mathbb{R}^{n\times q}$ which are referred to as *reduced order bases* (ROBs) are calculated, see for example [15]. The case $\mathbf{W}_i = \mathbf{V}_i$ is called *one-sided reduction* and the case $\mathbf{W}_i \neq \mathbf{V}_i$ is referred to as *two-sided reduction.* The ROBs span the right $\mathcal{V}_i = \text{span}(\mathbf{V}_i)$ and left subspace $\mathcal{W}_i = \text{span}(\mathbf{W}_i)$, respectively. Using the approximation $\mathbf{x}_i(t) \approx \mathbf{V}_i\hat{\mathbf{x}}_i(t)$ with the reduced state vector $\hat{\mathbf{x}}_i(t) \in \mathbb{R}^q$ and enforcing the Petrov-Galerkin condition

$$\mathbf{W}_i^T\Big(\mathbf{E}_i\mathbf{V}_i\dot{\hat{\mathbf{x}}}_i(t) - \mathbf{A}_i\mathbf{V}_i\hat{\mathbf{x}}_i(t) - \mathbf{B}_i\mathbf{u}(t)\Big) = \mathbf{0} \tag{2.13}$$

leads to the reduced order model

$$\hat{\mathbf{G}}_i : \begin{cases} \hat{\mathbf{E}}_i\dot{\hat{\mathbf{x}}}_i(t) = \hat{\mathbf{A}}_i\hat{\mathbf{x}}_i(t) + \hat{\mathbf{B}}_i\mathbf{u}(t) \\ \quad\hat{\mathbf{y}}_i(t) = \hat{\mathbf{C}}_i\hat{\mathbf{x}}_i(t), \end{cases} \tag{2.14}$$

where the reduced system matrices are given by

$$\hat{\mathbf{E}}_i = \mathbf{W}_i^T\mathbf{E}_i\mathbf{V}_i,\ \hat{\mathbf{A}}_i = \mathbf{W}_i^T\mathbf{A}_i\mathbf{V}_i,\ \hat{\mathbf{B}}_i = \mathbf{W}_i^T\mathbf{B}_i,\ \hat{\mathbf{C}}_i = \mathbf{C}_i\mathbf{V}_i. \tag{2.15}$$

To conclude, MOR aims to calculate the ROBs in such a way that the output of the reduced order model $\hat{\mathbf{G}}_i$ is a good approximation of the output of the original system $\mathbf{G}_i$ with $\hat{\mathbf{y}}_i(t) \approx \mathbf{y}_i(t)$.

2.2.2 Review of Model Order Reduction Methods

For calculating the ROBs, there are various projection-based reduction methods such as moment matching techniques, truncated balanced realization (TBR) or proper orthogonal decomposition (POD), see for example [15] and the references therein. In the following, two approaches are reviewed which are applied in this thesis.

Truncated Balanced Realization

A common method of MOR is TBR which was first derived in [122, 119]. For introducing this approach, consider first the energy $J_{\text{c},i}(\mathbf{x}_\text{e})$ which is the minimum energy required to drive the system from the zero state at time $t = -\infty$ to the state $\mathbf{x}_\text{e}$ at time $t = 0$. In a dual way, $J_{\text{o},i}(\mathbf{x}_0)$ is the energy which is obtained by observing the output due to the initial state $\mathbf{x}_0$ with zero input:

$$J_{\text{c},i}(\mathbf{x}_\text{e}) = \min_{\mathbf{x}_i(-\infty)=\mathbf{0},\ \mathbf{x}_i(0)=\mathbf{x}_\text{e}} \int_{-\infty}^{0} \mathbf{u}(t)^T\mathbf{u}(t)\text{d}t \tag{2.16}$$

$$J_{\text{o},i}(\mathbf{x}_0) = \int_0^\infty \mathbf{y}_i(t)^T\mathbf{y}_i(t)\text{d}t,\ \mathbf{x}_i(0) = \mathbf{x}_0,\ \mathbf{u}(t) = \mathbf{0},\ t \geq 0. \tag{2.17}$$

These two energies can also be determined by

$$J_{\mathrm{c},i}(\mathbf{x}_\mathrm{e}) = \mathbf{x}_\mathrm{e}^T \mathbf{P}_i^{-1} \mathbf{x}_\mathrm{e} \quad \text{and} \quad J_{\mathrm{o},i}(\mathbf{x}_0) = \mathbf{x}_0^T \mathbf{E}_i^T \mathbf{Q}_i \mathbf{E}_i \mathbf{x}_0, \tag{2.18}$$

where $\mathbf{P}_i, \mathbf{Q}_i \in \mathbb{S}_q^{++}$ solve for an asymptotically stable system the two dual generalized Lyapunov equations

$$\mathbf{A}_i \mathbf{P}_i \mathbf{E}_i^T + \mathbf{E}_i \mathbf{P}_i \mathbf{A}_i^T + \mathbf{B}_i \mathbf{B}_i^T = \mathbf{0} \tag{2.19}$$

$$\mathbf{A}_i^T \mathbf{Q}_i \mathbf{E}_i + \mathbf{E}_i^T \mathbf{Q}_i \mathbf{A}_i + \mathbf{C}_i^T \mathbf{C}_i = \mathbf{0}. \tag{2.20}$$

The two energies $J_{\mathrm{c},i}(\mathbf{x}_\mathrm{e})$ and $J_{\mathrm{o},i}(\mathbf{x}_0)$ measure the controllability of $\mathbf{x}_\mathrm{e}$ and observability of $\mathbf{x}_0$, respectively: The smaller the energy $J_{\mathrm{c},i}(\mathbf{x}_\mathrm{e})$, the easier the state $\mathbf{x}_\mathrm{e}$ can be reached. The larger the energy $J_{\mathrm{o},i}(\mathbf{x}_0)$, the easier the state $\mathbf{x}_0$ can be observed.

In the first step of TBR, a state representation of the high-order system is calculated so the state variables are equally controllable and observable. For this, matrices $\mathbf{P}_i$ and $\mathbf{E}_i^T \mathbf{Q}_i \mathbf{E}_i$, which are denoted as *Controllability* and *Observability Gramian*, are used. In the second step, the projection matrices $\mathbf{V}_i$ and $\mathbf{W}_i$ are calculated so the state variables, which are hard to reach and hard to observe at the same time, are eliminated. The beneficial properties of TBR are the stability of the reduced system and the existence of an a priori error bound. The basic procedure can only be applied for original systems with a moderate order n because the high-order Lyapunov equations (2.19) and (2.20) need to be solved. In order to reduce the computational effort of TBR, methods have been developed which approximatively solve these equations [135, 109, 54].

Moment Matching Techniques

Moment matching techniques approximate the high-order model by matching moments of the transfer function with regard to the Laplace variable s [85, 67]. These methods are based on a Taylor series expansion of the transfer function of the original $\mathbf{G}_i(s)$ and of the reduced system $\hat{\mathbf{G}}_i(s)$ with expansion points s_0 in the frequency space:

$$\mathbf{G}_i(s) = \sum_{j=0}^{\infty} \mathbf{M}_{i,j} (s - s_0)^j \tag{2.21}$$

$$\hat{\mathbf{G}}_i(s) = \sum_{j=0}^{\infty} \hat{\mathbf{M}}_{i,j} (s - s_0)^j, \tag{2.22}$$

where $\mathbf{M}_{i,j}$ and $\hat{\mathbf{M}}_{i,j}$ are the moments of the transfer function of the original and reduced system, respectively. The methods calculate Krylov subspaces spanned by the bases $\mathbf{V}_i$

and $\mathbf{W}_i$ such that $q_{s_0} \leq 2q$ moments of the reduced system match the ones of the high-order system:

$$\hat{\mathbf{M}}_{i,j} = \mathbf{M}_{i,j}, \quad \forall\, j \in \{1, \dots, q_{s_0}\}. \tag{2.23}$$

The methods can be extended for multiple expansion points. As the choice of the expansion points determines the accuracy of the reduced model, automatic selection procedures have been proposed, such as the Iterative Rational Krylov Algorithm (IRKA) [86] or the Stability-Preserving Adaptive Rational Krylov (SPARK) [126]. Moment matching techniques have the beneficial property that they can easily be used to reduce system $\mathbf{G}_i$ of very large order n. However, stability preservation is in general not guaranteed for the reduced system $\hat{\mathbf{G}}_i$.

2.3 Parametric Model Order Reduction

2.3.1 Problem Formulation

The previous section presented the reduction of nonparametric systems. If we have parameter-dependent systems, these systems could be reduced for every desired parameter value using MOR. However, this is computationally expensive as high-order operations have to be performed for every new parameter value. As a remedy, methods of parametric model order reduction (pMOR) have been developed. Their goal is to find a low-dimensional system

$$\tilde{\mathbf{G}}(\mathbf{p}) : \begin{cases} \tilde{\mathbf{E}}(\mathbf{p})\dot{\tilde{\mathbf{x}}}(t) = \tilde{\mathbf{A}}(\mathbf{p})\tilde{\mathbf{x}}(t) + \tilde{\mathbf{B}}(\mathbf{p})\mathbf{u}(t) \\ \qquad \tilde{\mathbf{y}}(t) = \tilde{\mathbf{C}}(\mathbf{p})\tilde{\mathbf{x}}(t) \end{cases} \tag{2.24}$$

of order $q \ll n$ which approximates the high-order system $\mathbf{G}(\mathbf{p})$ while preserving the parameter-dependency. We demand that the output of the reduced order model $\tilde{\mathbf{G}}(\mathbf{p})$ is a good approximation of the output of the original system $\mathbf{G}(\mathbf{p})$ with $\tilde{\mathbf{y}}(t) \approx \mathbf{y}(t)$ for the entire domain $\mathcal{D}$. Hence, if we want to obtain a reduced system for a new parameter vector $\mathbf{p} \in \mathcal{D}$, this is done without computing additional high-order solves just by inserting the new parameter into the low-order model $\tilde{\mathbf{G}}(\mathbf{p})$.

2.3.2 Review of Parametric Model Order Reduction Methods

This section discusses several approaches for pMOR. One can distinguish between global basis and local bases approaches [23]. The following review especially focuses on local bases approaches because they include the method of interest of this thesis.

Global Basis Approaches

A common basis is constructed by a single pair of matrices $\mathbf{V}$ and $\mathbf{W}$ which are supposed to contain information regarding the entire parameter domain $\mathcal{D}$. The resulting parametric reduced model $\tilde{\mathbf{G}}(\mathbf{p})$ consists of the system matrices

$$\tilde{\mathbf{E}}(\mathbf{p}) = \mathbf{W}^T\mathbf{E}(\mathbf{p})\mathbf{V},\ \tilde{\mathbf{A}}(\mathbf{p}) = \mathbf{W}^T\mathbf{A}(\mathbf{p})\mathbf{V},\ \tilde{\mathbf{B}}(\mathbf{p}) = \mathbf{W}^T\mathbf{B}(\mathbf{p}),\ \tilde{\mathbf{C}}(\mathbf{p}) = \mathbf{C}(\mathbf{p})\mathbf{V}. \quad (2.25)$$

As these matrix multiplications depend on the original order n, the procedure is expensive. The computational effort is reduced for affinely parameter-dependent systems. In this case, the parametric reduced model is decomposed into nonparametric matrices which can be precomputed. For the general case where the parametric dependencies are non-affine, a suitable approximate affine decomposition of the system matrices (2.25) can be provided by using, e.g., the Taylor series expansion or the Discrete Empirical Interpolation Method (DEIM) [42, 23]. Another disadvantage is that the order q of the parametric reduced model $\tilde{\mathbf{G}}(\mathbf{p})$ is often large, because the global basis needs to capture the parametric dependency by embedding information regarding the entire parameter space, and increases rapidly with the number of parameters. An advantage of global basis approaches is that rigorous error estimates have been proposed which allow an efficient greedy adaptive parameter sampling strategy.

There is a large variety of global basis approaches such as global proper orthogonal decomposition (POD) [36, 129], the reduced basis method [84, 91, 90], the concatenation of local bases for multiple parameter values [107, 131] or piecewise $\mathcal{H}_2$ tangential interpolation [19]. In the following, one approach is described in more detail because it is the first method suggested for pMOR.

Multiparameter Moment Matching

The method proposed in [161] approximates the high-order model by matching moments of the transfer function with regard to the Laplace variable s and to one parameter p. This method is based on a Taylor series expansion of the parametric transfer function of the original $\mathbf{G}(s, p)$ and of the reduced system $\tilde{\mathbf{G}}(s, p)$ with expansion points s_0 in the frequency space and p_0 in the parameter space:

$$\mathbf{G}(s,p) = \sum_{j=0}^{\infty}\sum_{k=0}^{j} \mathbf{M}_j^k (s-s_0)^j (p-p_0)^{j-k} \quad (2.26)$$

$$\tilde{\mathbf{G}}(s,p) = \sum_{j=0}^{\infty}\sum_{k=0}^{j} \tilde{\mathbf{M}}_j^k (s-s_0)^j (p-p_0)^{j-k}, \quad (2.27)$$

where $\mathbf{M}_j^k$ and $\tilde{\mathbf{M}}_j^k$ are the moments of the transfer function of the original and reduced system, respectively. Basically, these formulas are the generalization of the expansions from equation (2.21) and (2.22) to parametric systems with one parameter. The method calculates Krylov subspaces such that the reduced system shares some moments with the original one with respect to both the Laplace variable s and the parameter p. This method is extended in [88, 49, 60] to be applicable to the multiple parameter case. These approaches differ in details, such as if mixed moments are matched or not.

Local Bases Approaches

Local bases approaches rely on a set of nonparametric systems $\mathbf{G}_1, \ldots, \mathbf{G}_N$ which are obtained when the parameter-dependent system $\mathbf{G}(\mathbf{p})$ is computed for vectors $\mathbf{p}_1, \ldots, \mathbf{p}_N \in \mathcal{D}$. For each $i \in \{1, \ldots, N\}$, projection matrices $\mathbf{V}_i \in \mathbb{R}^{n \times q_i}$ and $\mathbf{W}_i \in \mathbb{R}^{n \times q_i}$ with size q_i are calculated for system $\mathbf{G}_i$ with any projection-based method, such as the ones described in Section 2.2.2. Then, the parametric reduced system $\tilde{\mathbf{G}}(\mathbf{p})$ is constructed by using only this local information. Local bases approaches have the advantage that the parameter-dependency can be non-affine and does not need to be analytically given as only local systems need to be available. In addition, some methods lead to a small reduced order as the ROBs only need to capture the parametric dependency regarding the local parameter space. In the following, three approaches are described for constructing the parametric reduced system $\tilde{\mathbf{G}}(\mathbf{p})$.

Interpolation of the Reduced Order Bases

One option for local bases approaches interpolates the ROBs of the local systems which need to have the same number of columns $q_1 = \ldots = q_N = q$. A few of the suggested methods firstly perform congruence transformations which adjust the ROBs and lead to a consistent set of coordinates $\tilde{\mathbf{V}}_1, \ldots, \tilde{\mathbf{V}}_N \in \mathbb{R}^{n \times q}$ and $\tilde{\mathbf{W}}_1, \ldots, \tilde{\mathbf{W}}_N \in \mathbb{R}^{n \times q}$ [30, 95]. Afterwards, the transformed ROBs are interpolated with

$$\tilde{\mathbf{V}}(\mathbf{p}) = \sum_{i=1}^{N} \omega_i(\mathbf{p}) \tilde{\mathbf{V}}_i \tag{2.28}$$

$$\tilde{\mathbf{W}}(\mathbf{p}) = \sum_{i=1}^{N} \omega_i(\mathbf{p}) \tilde{\mathbf{W}}_i, \tag{2.29}$$

where $\omega_1, \ldots, \omega_N : \mathcal{D} \to \mathbb{R}$ are cardinal basis functions (A.2). Alternatively, the matrix entries can be interpolated independently. Other approaches interpolate the ROBs on the tangent space to the Grassmann manifold of q-dimensional subspaces [13, 10, 8, 150, 38, 168]. For this, the ROBs are mapped onto the tangent space, the interpolation process is performed on the tangent space and the resulting ROBs are mapped back onto the original manifold. As these approaches preserve the full rank of the ROBs, they do not require the adjustment procedure described above. For both methods, the resulting parametric reduced model $\tilde{\mathbf{G}}(\mathbf{p})$ consists of the system matrices

$$\begin{aligned} \tilde{\mathbf{E}}(\mathbf{p}) &= \tilde{\mathbf{W}}(\mathbf{p})^T\mathbf{E}(\mathbf{p})\tilde{\mathbf{V}}(\mathbf{p}), & \tilde{\mathbf{A}}(\mathbf{p}) &= \tilde{\mathbf{W}}(\mathbf{p})^T\mathbf{A}(\mathbf{p})\tilde{\mathbf{V}}(\mathbf{p}), \\ \tilde{\mathbf{B}}(\mathbf{p}) &= \tilde{\mathbf{W}}(\mathbf{p})^T\mathbf{B}(\mathbf{p}), & \tilde{\mathbf{C}}(\mathbf{p}) &= \mathbf{C}(\mathbf{p})\tilde{\mathbf{V}}(\mathbf{p}). \end{aligned} \tag{2.30}$$

As these multiplications depend on the original order n, this procedure is expensive. The computational effort is reduced for the special case of affinely parameter-dependent systems [150]. Another disadvantage is that the local ROBs need to have the same size q. In addition, stability is not guaranteed for the interpolated system. The advantage of this method is that it leads to a parametric reduced system $\tilde{\mathbf{G}}(\mathbf{p})$ with small order q which is independent of the number of the local systems.

Interpolation of Transfer Functions

The method proposed in [20, 21] firstly calculates for each $i \in \{1, \dots, N\}$ the locally reduced system $\hat{\mathbf{G}}_i$ of order q_i with projection matrices $\mathbf{V}_i \in \mathbb{R}^{n\times q_i}$ and $\mathbf{W}_i \in \mathbb{R}^{n\times q_i}$ according to matrix multiplications (2.15). Then, we interpolate the transfer functions $\hat{\mathbf{G}}_i(s)$ for $i \in \{1, \dots, N\}$ with cardinal basis functions $\omega_1, \dots, \omega_N : \mathcal{D} \to \mathbb{R}$ in order to obtain the parameter-dependent reduced system

$$\tilde{\mathbf{G}}(s, \mathbf{p}) = \sum_{i=1}^{N} \omega_i(\mathbf{p})\hat{\mathbf{G}}_i(s) = \tilde{\mathbf{C}}(\mathbf{p})\Big[s\tilde{\mathbf{E}} - \tilde{\mathbf{A}}\Big]^{-1}\tilde{\mathbf{B}} \tag{2.31}$$

$$= \Big[\omega_1(\mathbf{p})\hat{\mathbf{C}}_1, \quad \dots, \quad \omega_N(\mathbf{p})\hat{\mathbf{C}}_N\Big] \begin{bmatrix} s\hat{\mathbf{E}}_1 - \hat{\mathbf{A}}_1 & & \\ & \ddots & \\ & & s\hat{\mathbf{E}}_N - \hat{\mathbf{A}}_N \end{bmatrix}^{-1} \begin{bmatrix} \hat{\mathbf{B}}_1 \\ \vdots \\ \hat{\mathbf{B}}_N \end{bmatrix}. \tag{2.32}$$

For the interpolation on high-dimensional parameter spaces, the authors suggest a sparse-grid-based approach. The reduced order of the interpolated system is $q = \sum_{i=1}^{N} q_i$ and hence, grows rapidly for an increasing number of grid points. This method also faces difficulties for systems which exhibit weakly damped modes. The main advantage of the procedure is that it is very efficient as only low-order systems are involved in the interpolation procedure. In addition, preservation of stability can be guaranteed for the interpolated system if the locally reduced systems are asymptotically stable. The latter can, e.g., be achieved if the local systems are reduced using TBR. Another advantage is that the reduced orders q_i of the local systems can be different.

Interpolation of System Matrices

Another option for local bases approaches is the interpolation of the matrices of the locally reduced systems, which need to have the same size $q_1 = \ldots = q_N = q$. The methods proposed in [6, 112, 128, 11, 137] firstly calculate for each $i \in \{1, \ldots, N\}$ the locally reduced system $\hat{\mathbf{G}}_i$ of order q with projection matrices $\mathbf{V}_i \in \mathbb{R}^{n \times q}$ and $\mathbf{W}_i \in \mathbb{R}^{n \times q}$ using matrix multiplications (2.15). Then, transformation matrices $\mathbf{T}_i$ and $\mathbf{M}_i$ are introduced which adjust the ROBs and lead to a consistent set of coordinates. A slightly different approach is suggested in [51, 143]. It concatenates the local ROBs to common projection matrices $\mathbf{V} = [\mathbf{V}_1, \ldots, \mathbf{V}_N]$ and $\mathbf{W} = [\mathbf{W}_1, \ldots, \mathbf{W}_N]$ which result in reduced systems $\hat{\mathbf{G}}_i$ of order Nq. As the reduced systems share a common basis, the adjustment procedure is not required. The resulting parametric reduced model $\tilde{\mathbf{G}}(\mathbf{p})$ consists—taking the adjustment into account—of the interpolated system matrices

$$\begin{aligned} \tilde{\mathbf{E}}(\mathbf{p}) &= \textstyle\sum_{i=1}^{N} \omega_i(\mathbf{p}) \mathbf{M}_i^T \hat{\mathbf{E}}_i \mathbf{T}_i, \\ \tilde{\mathbf{A}}(\mathbf{p}) &= \textstyle\sum_{i=1}^{N} \omega_i(\mathbf{p}) \mathbf{M}_i^T \hat{\mathbf{A}}_i \mathbf{T}_i, \\ \tilde{\mathbf{B}}(\mathbf{p}) &= \textstyle\sum_{i=1}^{N} \omega_i(\mathbf{p}) \mathbf{M}_i^T \hat{\mathbf{B}}_i, \\ \tilde{\mathbf{C}}(\mathbf{p}) &= \textstyle\sum_{i=1}^{N} \omega_i(\mathbf{p}) \hat{\mathbf{C}}_i \mathbf{T}_i, \end{aligned} \tag{2.33}$$

where $\omega_1, \ldots, \omega_N : \mathcal{D} \to \mathbb{R}$ are cardinal basis functions (A.2). Alternatively, the matrix entries can be interpolated independently. The approaches in [6, 51, 11] interpolate the system matrices on the tangent space to a matrix manifold. For this, the system matrices are mapped onto the tangent space, the interpolation process is performed on the tangent space and the resulting matrices are mapped back onto the original manifold. The disadvantage of the method is that stability is not guaranteed. The main advantage of the procedure is that it is very efficient as only low-order matrices are interpolated. In addition, it is well-adapted to systems with dominant eigenmodes as mode veering and crossing are recognized. The approach using the transformation also has the advantage that a parametric reduced system $\tilde{\mathbf{G}}(\mathbf{p})$ with small order q, which is independent of the number of the local systems, is obtained. However, the locally reduced systems need to have the same size q.

Comparison of Local Bases Approaches

In Table 2.1 it is listed if the reviewed local bases approaches in their general form fulfill (+) or do not fulfill (-) some important properties. The properties are stability preservation (StabPres), ability of interpolating locally reduced systems of different orders (DiffOrder), the independence of the order of the parametric reduced system from the number of grid points (RedOrderInd) and the operation based on low-order matrices which leads to a low computational effort (CompEffort). For all local bases approaches it is still an open question how the interpolation method leading to the most accurate reduced system can be chosen. In contrast to global basis approaches, these methods also lack an efficient error indicator which would allow a computationally cheap refinement strategy.

Table 2.1: Properties of different local bases approaches.

	StabPres	DiffOrder	RedOrderInd	CompEffort
Transfer functions	+	+	-	+
Reduced order bases	-	-	+	-
System matrices	-	-	+	+

To sum up, pMOR based on the interpolation of system matrices is a very promising approach because it operates on low-order matrices—and hence, has a low computational effort—and because the size of the interpolated system is independent of the number of grid points. However, there are still a lot of issues which need to be solved.

2.3.3 Thesis Contributions

It was pointed out in the previous section that pMOR based on the interpolation of system matrices has a high potential. Hence, this thesis focuses on this interpolation-based approach, develops solutions to the open questions and presents a self-contained method which can easily be applied by the user. In detail, the following contributions are made in this thesis:

1. The current methods of model order reduction by matrix interpolation have been presented as different methods. A general framework is proposed in Section 3. It combines and extends the existing methods to a construction kit which illustrates the necessary steps and presents the different options for each step.

2. This interpolation-based method of model order reduction requires the user to choose between many different design parameters. However, the user often does not have enough insight into the physics of the model in order to decide on the options leading to the most accurate reduced model. Hence, a new efficient error indicator is introduced in model order reduction and, based on this indicator, a Black-Box method is proposed in Section 4. It automatically selects the best interpolation method, performs the refinement in the parameter space iteratively and stops when the maximum predicted error of the interpolated system falls below a given error tolerance.

3. Model order reduction by the interpolation of reduced system matrices does not necessarily lead to asymptotically stable systems even if the local systems are asymptotically stable. Hence, a stability-preserving approach is presented in Section 5 which results in low-order optimization problems and hence, can efficiently be solved.

4. The existing methods of matrix interpolation suffer from the curse of dimensionality leading to costs that grow exponentially with the number of parameters. In order to extend model order reduction by matrix interpolation to multidimensional parameter spaces, a sparse-grid-based method is proposed in Section 6.

5. The current methods of matrix interpolation rely on a set of reduced systems which are of equal size. However, it is necessary for some applications to have differently-sized systems. For this, an interpolation procedure is proposed in Section 7 which resizes the system matrices to the same size and hence, provides a tool for a meaningful interpolation.

6. This interpolation-based method has rarely been applied to practical applications. For that reason, the suitability of this approach for control applications is demonstrated by means of an example at a test rig in Section 8.

Part II

A Black-Box Method for Parametric Model Order Reduction by Matrix Interpolation

3 The General Framework for Interpolating Reduced System Matrices

3.1 Motivation

It was pointed out in Section 2.3.2 that a few approaches for pMOR based on matrix interpolation were developed independently and presented as different methods. In this section, these methods are combined and extended to a general framework for the reduction of parameter-dependent LTI systems. The reader can use the framework as a construction kit for pMOR by matrix interpolation which illustrates the necessary steps and presents the different options for each step.

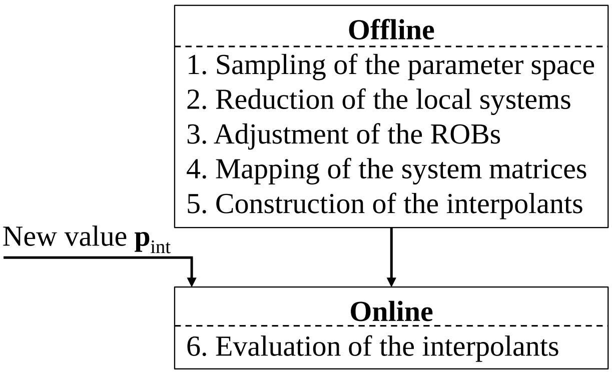

Offline
1. Sampling of the parameter space
2. Reduction of the local systems
3. Adjustment of the ROBs
4. Mapping of the system matrices
5. Construction of the interpolants
New value $\mathbf{p}_{\text{int}}$
Online
6. Evaluation of the interpolants

Figure 3.1: Structure of the general framework including two parts.

The procedure which is shown in Figure 3.1 consists of two parts which altogether comprise six steps. The first part is the *offline* or pre-processing phase which is performed only once. To begin with, it computes a set of nonparametric high-order systems for different parameter vectors. The resulting local systems are reduced by projection-based MOR and the reduced order bases (ROBs) are adjusted to the reference ROBs. Afterwards, the reduced system matrices are mapped onto the tangent space of a matrix manifold. Finally, the mapped matrices are used to construct the matrix interpolants with regard to an interpolation method. In the *online* phase, i.e. when a reduced order model is supposed to be calculated for the new parameter vector $\mathbf{p}_{\text{int}}$, the interpolants are simply evaluated and the resulting matrices are mapped back onto the manifold.

The framework, which was published in [76], is presented in Sections 3.2–3.7 and the options we have for each step are pointed out. An error analysis is given in Section 3.8 followed by the integration of two existing methods into the framework in Section 3.9.

3.2 Sampling of the Parameter Space

The parameter domain $\mathcal{D}$ is sampled for a set of vectors $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\} \subset \mathcal{D}$, where $N = |\mathcal{P}|$ is the number of grid points. The grid type, i.e. if the parameter vectors are located on a regular grid such as described in Appendix A.2 or on an irregular grid such as described in Appendix A.3, is determined by the kind of application and its constraints on the domain. The original system $\mathbf{G}(\mathbf{p})$ is computed for the parameter vectors in the set $\mathcal{P}$ which results in a set of high-order systems $\mathcal{G} = \{\mathbf{G}(\mathbf{p}_1), \ldots, \mathbf{G}(\mathbf{p}_N)\}$, where for example the i-th system $\mathbf{G}_i := \mathbf{G}(\mathbf{p}_i)$ with $i \in \{1, \ldots, N\}$ is

$$\mathbf{G}_i : \begin{cases} \mathbf{E}_i\dot{\mathbf{x}}_i(t) = \mathbf{A}_i\mathbf{x}_i(t) + \mathbf{B}_i\mathbf{u}(t) \\ \quad \mathbf{y}_i(t) = \mathbf{C}_i\mathbf{x}_i(t). \end{cases} \tag{3.1}$$

To sum up, the selection of the parameter vectors $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ is the option we have and the set of high-order systems $\mathcal{G} = \{\mathbf{G}_1, \ldots, \mathbf{G}_N\}$ is the output of this step.

3.3 Reduction of the Local Systems

Each local system of the set $\mathcal{G}$ is reduced individually to order $q \ll n$. For this, every projection-based reduction method such as described in Section 2.2 can be applied. As a result, we obtain for each $i \in \{1, \ldots, N\}$ the subspaces $\mathcal{V}_i$ and $\mathcal{W}_i$ which are spanned by the bases $\mathbf{V}_i \in \mathbb{R}^{n \times q}$ and $\mathbf{W}_i \in \mathbb{R}^{n \times q}$ with linearly independent columns, respectively. This delivers a set of low-order systems $\hat{\mathcal{G}} = \{\hat{\mathbf{G}}(\mathbf{p}_1), \ldots, \hat{\mathbf{G}}(\mathbf{p}_N)\}$, where for example the i-th system $\hat{\mathbf{G}}_i := \hat{\mathbf{G}}(\mathbf{p}_i)$ is

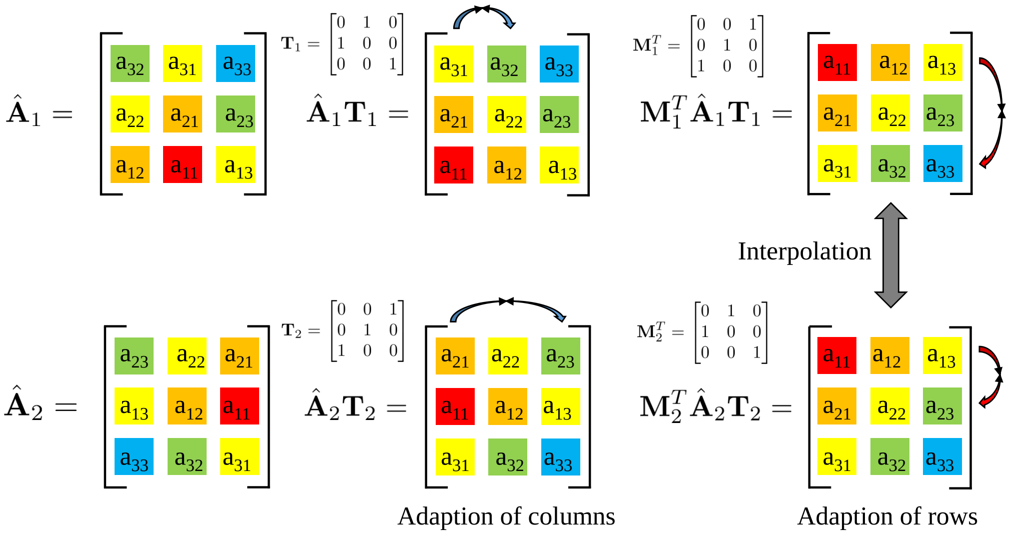
$$\hat{\mathbf{G}}_i : \begin{cases} \overbrace{\mathbf{W}_i^T\mathbf{E}_i\mathbf{V}_i}^{\hat{\mathbf{E}}_i}\,\dot{\hat{\mathbf{x}}}_i(t) = \overbrace{\mathbf{W}_i^T\mathbf{A}_i\mathbf{V}_i}^{\hat{\mathbf{A}}_i}\,\hat{\mathbf{x}}_i(t) + \overbrace{\mathbf{W}_i^T\mathbf{B}_i}^{\hat{\mathbf{B}}_i}\,\mathbf{u}(t) \\ \qquad\qquad \hat{\mathbf{y}}_i(t) = \underbrace{\mathbf{C}_i\mathbf{V}_i}_{\hat{\mathbf{C}}_i}\,\hat{\mathbf{x}}_i(t). \end{cases} \tag{3.2}$$

To sum up, the selection of the reduction method and of the reduced order q are the options we have. The set of low-order systems $\hat{\mathcal{G}} = \{\hat{\mathbf{G}}_1, \ldots, \hat{\mathbf{G}}_N\}$ and the bases $\mathbf{V}_1, \ldots, \mathbf{V}_N$, $\mathbf{W}_1, \ldots, \mathbf{W}_N$ are the outputs of this step.

3.4 Adjustment of the Reduced Order Bases

If we interpolate the system matrices of the reduced systems from the set $\hat{\mathcal{G}}$, we have to ensure that associated matrix entries are interpolated, see the following example.

Example: Consider two reduced systems $\dot{\hat{\mathbf{x}}}_1(t) = \hat{\mathbf{A}}_1\hat{\mathbf{x}}_1(t)$ and $\dot{\hat{\mathbf{x}}}_2(t) = \hat{\mathbf{A}}_2\hat{\mathbf{x}}_2(t)$ whose matrices $\hat{\mathbf{A}}_1, \hat{\mathbf{A}}_2$ are shown in the left column of Figure 3.2. The reader can verify that associated matrix entries, e.g. a_{11}, are not located on the same spot and hence, element-wise interpolation will fail. As a remedy, we adapt the columns of matrices $\hat{\mathbf{A}}_1, \hat{\mathbf{A}}_2$ by multiplication from the right with matrices $\mathbf{T}_1, \mathbf{T}_2$ and we adapt the rows by multiplication from the left with matrices $\mathbf{M}_1^T, \mathbf{M}_2^T$. Finally, associated matrix entries are located on the same position. Note that the absolute position is of no importance.



$\hat{\mathbf{A}}_1 = \begin{bmatrix} a_{32} & a_{31} & a_{33} \\ a_{22} & a_{21} & a_{23} \\ a_{12} & a_{11} & a_{13} \end{bmatrix}$
$\mathbf{T}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
$\hat{\mathbf{A}}_1\mathbf{T}_1 = \begin{bmatrix} a_{31} & a_{32} & a_{33} \\ a_{21} & a_{22} & a_{23} \\ a_{11} & a_{12} & a_{13} \end{bmatrix}$
$\mathbf{M}_1^T = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$
$\mathbf{M}_1^T\hat{\mathbf{A}}_1\mathbf{T}_1 = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$
Interpolation
$\hat{\mathbf{A}}_2 = \begin{bmatrix} a_{23} & a_{22} & a_{21} \\ a_{13} & a_{12} & a_{11} \\ a_{33} & a_{32} & a_{31} \end{bmatrix}$
$\mathbf{T}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$
$\hat{\mathbf{A}}_2\mathbf{T}_2 = \begin{bmatrix} a_{21} & a_{22} & a_{23} \\ a_{11} & a_{12} & a_{13} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$
$\mathbf{M}_2^T = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
$\mathbf{M}_2^T\hat{\mathbf{A}}_2\mathbf{T}_2 = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$
Adaption of columns
Adaption of rows

Figure 3.2: Graphic depiction of the adjustment procedure.

Hence, we perform for each system $\hat{\mathbf{G}}_i$ with $i \in \{1, \ldots, N\}$ the state transformation $\hat{\mathbf{x}}_i(t) = \mathbf{T}_i\tilde{\mathbf{x}}_i(t)$ with $\mathbf{T}_i \in \mathbb{R}^{q\times q}$ which linearly combines the columns of the corresponding system matrices. In addition, we multiply the system matrices with $\mathbf{M}_i^T \in \mathbb{R}^{q\times q}$ from the left which linearly combines their rows. Both operations leave the input-output behavior of system $\hat{\mathbf{G}}_i$ unchanged, but they can be used to ensure accurate matrix interpolation. In the following, the computation of $\mathbf{T}_i$ and $\mathbf{M}_i$ is explained.

3.4.1 Adjustment of the Right Reduced Order Bases

The reduced systems of the set $\hat{\mathcal{G}}$ have different subspaces $\mathcal{V}_1, \ldots, \mathcal{V}_N$ which are spanned by the bases $\mathbf{V}_1, \ldots, \mathbf{V}_N$, respectively. Hence, the state vectors $\hat{\mathbf{x}}_1(t), \ldots, \hat{\mathbf{x}}_N(t)$ do not have a common meaning. In order to introduce a common set of states, we perform for each $i \in \{1, \dots, N\}$ the state transformation $\hat{\mathbf{x}}_i(t) = \mathbf{T}_i \tilde{\mathbf{x}}_i(t)$ with $\mathbf{T}_i \in \mathbb{R}^{q \times q}$. This results in the new basis

$$\tilde{\mathbf{V}}_i = \mathbf{V}_i \mathbf{T}_i \in \mathbb{R}^{n \times q} \tag{3.3}$$

which of course spans the same subspace $\tilde{\mathcal{V}}_i = \operatorname{span}(\tilde{\mathbf{V}}_i) = \operatorname{span}(\mathbf{V}_i) = \mathcal{V}_i$. Hence, we aim to determine the bases $\tilde{\mathbf{V}}_1, \dots, \tilde{\mathbf{V}}_N$ which are adjusted to the basis $\mathbf{V}_0 \in \mathbb{R}^{n \times q}$ of a reference subspace in the sense that a quality function is minimized.

Reference Subspace

The reference subspace $\mathcal{V}_0$, which is spanned by the basis $\mathbf{V}_0$, should comprise the most important directions of the subspaces $\mathcal{V}_1, \dots, \mathcal{V}_N$. In the following, three possibilities for calculating the reference basis $\mathbf{V}_0$ are presented:

- Fixed [11]: One selects the basis $\mathbf{V}_{i_0}$ of the i_0-th reduced system:

  $$\mathbf{V}_0 = \mathbf{V}_{i_0}. \tag{3.4}$$

  This approach does not need any additional calculations. However, it requires a-priori knowledge in order to decide which subspace $\mathcal{V}_{i_0} = \operatorname{span}(\mathbf{V}_{i_0})$ captures the most important directions of all subspaces.

- Non-weighted SVD [128]: This approach sums up all bases $\mathbf{V}_1, \dots, \mathbf{V}_N$ and takes the q most important directions by calculating the economy version of the SVD. Then, the first q columns of $\mathbf{U}_\mathbf{V}$ are chosen as $\mathbf{V}_0$:

  $$\begin{aligned} \mathbf{U}_\mathbf{V} \mathbf{\Sigma}_\mathbf{V} \mathbf{Z}_\mathbf{V}^T &= \operatorname{svd}([\mathbf{V}_1, \dots, \mathbf{V}_N], \text{'econ'}) \\ \Rightarrow \mathbf{V}_0 &= \mathbf{U}_\mathbf{V}(:, 1:q). \end{aligned} \tag{3.5}$$

  This approach is more expensive. However, it automatically captures the most important directions of all subspaces provided that they are not too different.

- Weighted SVD [128]: This approach resembles method (3.5), but parameter-dependent weights $\omega_1(\mathbf{p}), \dots, \omega_N(\mathbf{p})$ are introduced for the bases $\mathbf{V}_1, \dots, \mathbf{V}_N$:

  $$\begin{aligned} \mathbf{U}_\mathbf{V} \mathbf{\Sigma}_\mathbf{V} \mathbf{Z}_\mathbf{V}^T &= \operatorname{svd}([\omega_1(\mathbf{p})\mathbf{V}_1, \dots, \omega_N(\mathbf{p})\mathbf{V}_N], \text{'econ'}) \\ \Rightarrow \mathbf{V}_0 &= \mathbf{U}_\mathbf{V}(:, 1:q). \end{aligned} \tag{3.6}$$

  This is the most expensive approach as the reference basis has to be calculated online for every new parameter vector $\mathbf{p}$.

The three approaches are ordered with increasing computational effort. On the other hand, we can expect the reduced model to become more accurate in the order of the presented approaches. The reason is the non-weighted SVD automatically captures the most important directions of all subspaces and the weighted SVD even allows to consider their influence on the desired interpolation point. The non-weighted SVD balances both tendencies and hence, is used for most examples in this thesis.

Objective Functions

For adjusting the bases $\mathbf{V}_1, \ldots, \mathbf{V}_N$ to the reference basis $\mathbf{V}_0$ we present in the following two quality functions and the corresponding optimal solution.

DS approach:
One objective function is for each $i \in \{1, \ldots, N\}$ the distance (DS) between the new basis $\tilde{\mathbf{V}}_i$ and the reference basis $\mathbf{V}_0$ which is measured here in Frobenius norm [81]:

$$J_{\mathrm{DS},\mathbf{V},i}(\mathbf{T}_i) = \|\tilde{\mathbf{V}}_i - \mathbf{V}_0\|_F = \|\mathbf{V}_i\mathbf{T}_i - \mathbf{V}_0\|_F. \tag{3.7}$$

The transformation matrix $\mathbf{T}_i \in \mathbb{R}^{q\times q}$ describes a permutation, rotation and distortion of the basis $\mathbf{V}_i$ in order to minimize the distance to the reference basis. The solution with minimum value of the cost function $J_{\mathrm{DS},\mathbf{V},i}$ and equivalently with minimum distance of the ROBs is obtained according to Appendix D by applying the left pseudoinverse [75]:

$$\mathbf{T}_i = \arg\min_{\mathbf{T}_i \in \mathbb{R}^{q\times q}} J_{\mathrm{DS},\mathbf{V},i}(\mathbf{T}_i) = \mathbf{V}_i^{(l)+}\mathbf{V}_0 \in \mathbb{R}^{q\times q}. \tag{3.8}$$

We calculate the pseudoinverse in (3.8) using the economy size SVD according to formula (D.3). The procedure for calculating $\mathbf{T}_1, \ldots, \mathbf{T}_N$ is shown in Algorithm 1.

Algorithm 1 Calculation of $\mathbf{T}_i$ using the DS approach

```
Input: $N$ matrices $\mathbf{V}_i$ and $\mathbf{V}_0$
Output: $N$ matrices $\mathbf{T}_i$
for $i = 1$ to $N$ do
    Compute $\mathbf{V}_i = \mathbf{U}_{\mathbf{V},i}\mathbf{\Sigma}_{\mathbf{V},i}\mathbf{Z}_{\mathbf{V},i}^T$        // SVD
    Compute $\mathbf{T}_i = \mathbf{Z}_{\mathbf{V},i}\mathbf{\Sigma}_{\mathbf{V},i}^{-1}\mathbf{U}_{\mathbf{V},i}^T\mathbf{V}_0$
end for
```

Then, the new basis $\tilde{\mathbf{V}}_i$, which we do not calculate explicitly, is found to be for the purpose of illustration:

$$\begin{aligned}\tilde{\mathbf{V}}_i &= \mathbf{V}_i\mathbf{T}_i = \mathbf{V}_i\mathbf{V}_i^{(l)+}\mathbf{V}_0 \\ &= \underbrace{\mathbf{V}_i(\mathbf{V}_i^T\mathbf{V}_i)^{-1}\mathbf{V}_i^T}_{\mathbf{P}_{\mathbf{V}_i\perp\mathbf{V}_i}}\mathbf{V}_0.\end{aligned} \tag{3.9}$$

One can verify that $\mathbf{P}_{\mathbf{V}_i\perp\mathbf{V}_i}$ is a projector which projects the columns of $\mathbf{V}_0$ orthogonally to the subspace $\mathcal{V}_i$ into the subspace $\mathcal{V}_i$. This is demonstrated in Figure 3.3 for the example of 1-dimensional subspaces.

For the case that $\mathbf{V}_i$ is an orthonormal basis with $\mathbf{V}_i^T\mathbf{V}_i = \mathbf{I}_q$, we want to preserve that property for the transformed basis $\tilde{\mathbf{V}}_i = \mathbf{V}_i\mathbf{T}_i$, which is known as the orthogonal Procrustes problem [81, 11]. Hence, this is in the following referred to as *PS approach*. For this, transformation matrix $\mathbf{T}_i \in \mathbb{O}_q$ has to be an orthogonal matrix. It describes a permutation and rotation of the basis $\mathbf{V}_i$ in order to minimize the distance to the reference basis with regard to the objective function (3.7), but it does not distort the basis. This is demonstrated in Figure 3.3. We rewrite the quality function (3.7) with $\mathbf{V}_0^T\mathbf{V}_0 = \mathbf{I}_q$ and with regard to the constraint $\mathbf{T}_i\mathbf{T}_i^T = \mathbf{I}_q$ and we obtain

$$\begin{aligned} J_{\mathrm{PS},\mathbf{V},i}(\mathbf{T}_i)^2 &= ||\mathbf{V}_i\mathbf{T}_i - \mathbf{V}_0||_F^2 \\ &= \mathrm{tr}\left(\mathbf{T}_i^T\mathbf{V}_i^T\mathbf{V}_i\mathbf{T}_i\right) - 2\mathrm{tr}\left(\mathbf{T}_i^T\mathbf{V}_i^T\mathbf{V}_0\right) + \mathrm{tr}\left(\mathbf{V}_0^T\mathbf{V}_0\right) \\ &= \mathrm{tr}\left(\mathbf{V}_i^T\mathbf{V}_i\mathbf{T}_i\mathbf{T}_i^T\right) - 2\mathrm{tr}\left(\mathbf{T}_i^T\mathbf{V}_i^T\mathbf{V}_0\right) + \mathrm{tr}\left(\mathbf{I}_q\right) \\ &= 2\mathrm{tr}\left(\mathbf{I}_q\right) - 2\mathrm{tr}\left(\mathbf{T}_i^T\mathbf{V}_i^T\mathbf{V}_0\right). \end{aligned} \tag{3.10}$$

Minimizing objective function (3.10) is equivalent to the maximization problem

$$\mathbf{T}_i = \arg\max_{\mathbf{T}_i\in\mathbb{O}_q} J^*_{\mathrm{PS},\mathbf{V},i}(\mathbf{T}_i) = \arg\max_{\mathbf{T}_i\in\mathbb{O}_q} \mathrm{tr}\left(\mathbf{T}_i^T\mathbf{V}_i^T\mathbf{V}_0\right). \tag{3.11}$$

After performing the SVD of $\mathbf{V}_i^T\mathbf{V}_0 = \mathbf{U}_{\mathbf{V},i}\mathbf{\Sigma}_{\mathbf{V},i}\mathbf{Z}_{\mathbf{V},i}^T$, we obtain

$$\begin{aligned}\mathbf{T}_i &= \arg\max_{\mathbf{T}_i\in\mathbb{O}_q} \mathrm{tr}\left(\mathbf{T}_i^T\mathbf{U}_{\mathbf{V},i}\mathbf{\Sigma}_{\mathbf{V},i}\mathbf{Z}_{\mathbf{V},i}^T\right) \\ &= \arg\max_{\mathbf{T}_i\in\mathbb{O}_q} \mathrm{tr}\left(\mathbf{Z}_{\mathbf{V},i}^T\mathbf{T}_i^T\mathbf{U}_{\mathbf{V},i}\mathbf{\Sigma}_{\mathbf{V},i}\right).\end{aligned} \tag{3.12}$$

This objective function reaches its maximum value, $J^*_{\mathrm{PS},\mathbf{V},i} = \mathrm{tr}\left(\mathbf{\Sigma}_{\mathbf{V},i}\right)$, for the case $\mathbf{Z}_{\mathbf{V},i}^T\mathbf{T}_i^T\mathbf{U}_{\mathbf{V},i} = \mathbf{I}_q$ or equivalently

$$\mathbf{T}_i = \mathbf{U}_{\mathbf{V},i}\mathbf{Z}_{\mathbf{V},i}^T \in \mathbb{O}_q. \tag{3.13}$$

To sum up, the procedure for calculating $\mathbf{T}_1, \ldots, \mathbf{T}_N$ is shown in Algorithm 2.

Algorithm 2 Calculation of $\mathbf{T}_i$ using the PS approach

```
Input: N matrices $\mathbf{V}_i$ and $\mathbf{V}_0$
Output: N matrices $\mathbf{T}_i$
for $i = 1$ to $N$ do
    Compute $\mathbf{V}_i^T\mathbf{V}_0 = \mathbf{U}_{\mathbf{V},i}\mathbf{\Sigma}_{\mathbf{V},i}\mathbf{Z}_{\mathbf{V},i}^T$        // SVD
    Compute $\mathbf{T}_i = \mathbf{U}_{\mathbf{V},i}\mathbf{Z}_{\mathbf{V},i}^T$
end for
```

MAC approach

The correlation between two vectors can be evaluated by the Modal Assurance Criterion (MAC), see [57]. The maximum value of the MAC is 1 which corresponds to the best correlation and the minimum value is 0. We define the MAC between the j-th vector of $\mathbf{V}_0$ and the l-th vector of $\tilde{\mathbf{V}}_i$ as follows:

$$\mathrm{MAC}(\mathbf{V}_0(:,j), \tilde{\mathbf{V}}_i(:,l)) = |\mathbf{V}_0(:,j)^T\tilde{\mathbf{V}}_i(:,l)|^2. \tag{3.14}$$

Then, the diagonal elements of the matrix $\mathbf{V}_0^T\tilde{\mathbf{V}}_i$ describe the square roots of the MACs between the corresponding vectors of $\tilde{\mathbf{V}}_i$ and $\mathbf{V}_0$ and the off-diagonal elements describe the square roots of the MACs between the non-corresponding vectors. Hence, we want the diagonal elements of $\mathbf{V}_0^T\tilde{\mathbf{V}}_i$ to have the maximum value 1 and the off-diagonal elements to have the minimum value 0 [128, 76]:

$$\mathbf{V}_0^T\tilde{\mathbf{V}}_i = \mathbf{V}_0^T\mathbf{V}_i\mathbf{T}_i \overset{!}{=} \mathbf{I}_q. \tag{3.15}$$

The transformation matrix $\mathbf{T}_i \in \mathbb{R}^{q\times q}$ describes a permutation, rotation and distortion of the basis $\mathbf{V}_i$ in order to fulfill the MAC criterion. The solution is

$$\mathbf{T}_i = (\mathbf{V}_0^T\mathbf{V}_i)^{-1} \in \mathbb{R}^{q\times q}. \tag{3.16}$$

The new basis $\tilde{\mathbf{V}}_i$ is found to be—again we do not calculate it explicitly—for the purpose of illustration with $\mathbf{V}_0^T\mathbf{V}_0 = \mathbf{I}_q$:

$$\begin{aligned}\tilde{\mathbf{V}}_i &= \mathbf{V}_i\mathbf{T}_i = \mathbf{V}_i(\mathbf{V}_0^T\mathbf{V}_i)^{-1}\\ &= \underbrace{\mathbf{V}_i(\mathbf{V}_0^T\mathbf{V}_i)^{-1}\mathbf{V}_0^T}_{\mathbf{P}_{\mathbf{V}_i\perp\mathbf{V}_0}}\mathbf{V}_0,\end{aligned} \tag{3.17}$$

where $\mathbf{P}_{\mathbf{V}_i\perp\mathbf{V}_0}$ is a projector which projects the columns of $\mathbf{V}_0$ orthogonally to the subspace spanned by $\mathbf{V}_0$ into the subspace $\mathcal{V}_i$. This is demonstrated in Figure 3.3 for the example of 1-dimensional subspaces.

Again, we consider the case that $\mathbf{V}_i$ is an orthonormal basis with $\mathbf{V}_i^T\mathbf{V}_i = \mathbf{I}_q$ and hence, transformation matrix $\mathbf{T}_i \in \mathbb{O}_q$ is desired to be orthogonal. For this, we formulate the following quality function for the MAC criterion (3.15):

$$J_{\mathrm{MAC},\mathbf{V},i}(\mathbf{T}_i) = \|\mathbf{V}_0^T\tilde{\mathbf{V}}_i - \mathbf{I}_q\|_F = \|\mathbf{V}_0^T\mathbf{V}_i\mathbf{T}_i - \mathbf{I}_q\|_F. \tag{3.18}$$

Of course, the minimum value of the quality function, $J_{\mathrm{MAC},\mathbf{V},i} = 0$, is obtained for solution (3.16). But now, we rewrite the cost function (3.18) with regard to the constraint $\mathbf{T}_i\mathbf{T}_i^T = \mathbf{I}_q$ and obtain

$$\begin{aligned} J_{\mathrm{MAC},\mathbf{V},i}(\mathbf{T}_i)^2 &= ||\mathbf{V}_0^T\mathbf{V}_i\mathbf{T}_i - \mathbf{I}_q||_F^2 \\ &= \mathrm{tr}\left(\mathbf{T}_i^T\mathbf{V}_i^T\mathbf{V}_0\mathbf{V}_0^T\mathbf{V}_i\mathbf{T}_i\right) - 2\mathrm{tr}\left(\mathbf{T}_i^T\mathbf{V}_i^T\mathbf{V}_0\right) + \mathrm{tr}\left(\mathbf{I}_q\right) \\ &= \mathrm{tr}\left(\mathbf{V}_i^T\mathbf{V}_0\mathbf{V}_0^T\mathbf{V}_i\mathbf{T}_i\mathbf{T}_i^T\right) - 2\mathrm{tr}\left(\mathbf{T}_i^T\mathbf{V}_i^T\mathbf{V}_0\right) + \mathrm{tr}\left(\mathbf{I}_q\right) \\ &= \mathrm{tr}\left(\mathbf{V}_i^T\mathbf{V}_0\mathbf{V}_0^T\mathbf{V}_i\right) - 2\mathrm{tr}\left(\mathbf{T}_i^T\mathbf{V}_i^T\mathbf{V}_0\right) + \mathrm{tr}\left(\mathbf{I}_q\right). \end{aligned} \tag{3.19}$$

Minimizing the objective function (3.19) is equivalent to the maximization problem (3.11) from the PS approach and hence, we obtain the same solution (3.13).

Comparison

The three different possibilities for adjusting the ROBs are shown in Table 3.1. The solutions of the DS and MAC approach differ for $\mathbf{T}_i \in \mathbb{R}^{q\times q}$ and coincide for $\mathbf{T}_i \in \mathbb{O}_q$, which we refer to as PS approach.

Table 3.1: Comparison of the approaches for the adjustment of the right ROBs.

	$\mathbf{T}_i \in \mathbb{R}^{q\times q}$	$\mathbf{T}_i \in \mathbb{O}_q$	
DS	$\mathbf{T}_i = \mathbf{V}_i^{(l)+}\mathbf{V}_0$	$\mathbf{T}_i = \mathbf{U}_{\mathbf{V},i}\mathbf{Z}_{\mathbf{V},i}^T$	PS
MAC	$\mathbf{T}_i = (\mathbf{V}_0^T\mathbf{V}_i)^{-1}$		

The three approaches are demonstrated in Figure 3.3 for the example of 1-dimensional subspaces $\mathcal{V}_i, \mathcal{V}_0$ which are spanned by unit vectors $\mathbf{v}, \mathbf{v}_0$, respectively. The adjusted basis $\tilde{\mathbf{v}}$ is obtained

- 1) for the MAC approach and $\mathbf{T}_i \in \mathbb{R}^{q\times q}$ according to formula (3.17) by projecting the reference basis $\mathbf{v}_0$ into the subspace $\mathcal{V}_i$ orthogonally to the subspace $\mathcal{V}_0$.
- 2) for the DS approach and $\mathbf{T}_i \in \mathbb{R}^{q\times q}$ according to formula (3.9) by projecting the reference basis $\mathbf{v}_0$ into the subspace $\mathcal{V}_i$ orthogonally to the subspace $\mathcal{V}_i$.

- ∘ 3) for the PS approach and $\mathbf{T} \in \mathbb{O}_q$ by changing the direction of basis $\mathbf{v}$. For this case, $\tilde{\mathbf{v}}$ is also a unit vector.

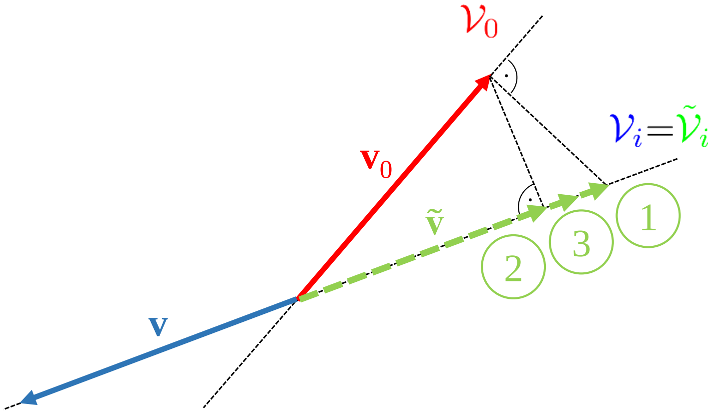


Figure 3.3: Comparison of the three approaches for adjusting the ROBs for the example of 1-dimensional subspaces: 1) MAC 2) DS 3) PS.

One can verify the difference between the approaches increases for a growing angle between the subspaces $\mathcal{V}_i, \mathcal{V}_0$. A special case is given in the following proposition.

Proposition 3.1. *If $\mathbf{V}_i, \mathbf{V}_0$ are orthonormal bases and span the same subspace, the MAC, DS and PS approaches are equivalent.*

The proof is given in Appendix B. This result can also be observed in Figure 3.3 for a diminishing angle between the subspaces $\mathcal{V}_i, \mathcal{V}_0$. For this case, all three approaches deliver the same solution.

3.4.2 Adjustment of the Left Reduced Order Bases

For the adjustment of the left ROBs, the concept of duality between the left and right subspace is exploited. Duality is a well-known notion in control theory [99]. For each $i \in \{1, \dots, N\}$, the system $\hat{\mathbf{G}}_i$ has a dual or adjoint system $\hat{\mathbf{G}}_i^T$ with state vector $\hat{\boldsymbol{\xi}}_i$:

$$\hat{\mathbf{G}}_i^T : \begin{cases} \hat{\mathbf{E}}_i^T \dot{\hat{\boldsymbol{\xi}}}_i(t) = \hat{\mathbf{A}}_i^T \hat{\boldsymbol{\xi}}_i + \hat{\mathbf{C}}_i^T \hat{\mathbf{y}}_i(t) \\ \mathbf{u}(t) = \hat{\mathbf{B}}_i^T \hat{\boldsymbol{\xi}}_i(t). \end{cases} \tag{3.20}$$

The right subspaces of the dual reduced systems are spanned by the columns of $\mathbf{W}_1, \dots, \mathbf{W}_N$. Therefore, the adjustment of the left ROBs of the systems $\hat{\mathbf{G}}_1, \dots, \hat{\mathbf{G}}_N$ is equivalent to the adjustment of the right ROBs of the dual systems $\hat{\mathbf{G}}_1^T, \dots, \hat{\mathbf{G}}_N^T$. The latter can be done in analogy to the explanations in Section 3.4.1 and we introduce the adjusted basis

$$\tilde{\mathbf{W}}_i = \mathbf{W}_i \mathbf{M}_i \in \mathbb{R}^{n \times q}. \tag{3.21}$$

The reference subspace spanned by $\mathbf{W}_0$ can be calculated in the same way as $\mathbf{V}_0$ just by replacing $\mathbf{V}_i$ with $\mathbf{W}_i$. For example, we obtain in analogy to formula (3.5)

$$\begin{aligned} \mathbf{U}_{\mathbf{W}} \mathbf{\Sigma}_{\mathbf{W}} \mathbf{Z}_{\mathbf{W}}^T &= \mathrm{svd}([\mathbf{W}_1, \dots, \mathbf{W}_N], \text{'econ'}) \\ \Rightarrow \mathbf{W}_0 &= \mathbf{U}_{\mathbf{W}}(:, 1:q). \end{aligned} \tag{3.22}$$

The adjustment of the left ROBs for accuracy reasons is done in the same way as proposed in Section 3.4.1 and we replace—due to duality—matrix $\mathbf{V}_i$ with $\mathbf{W}_i$ and $\mathbf{V}_0$ with $\mathbf{W}_0$. We obtain for the DS approach in analogy to the transformation matrix (3.8)

$$\mathbf{M}_i = \mathbf{W}_i^{(l)+} \mathbf{W}_0 \in \mathbb{R}^{n \times q}. \tag{3.23}$$

For the MAC approach we obtain in analogy to the transformation matrix (3.16)

$$\mathbf{M}_i = (\mathbf{W}_0^T \mathbf{W}_i)^{-1} \in \mathbb{R}^{n \times q}. \tag{3.24}$$

This is confirmed by the result from [128], where this choice of $\mathbf{M}_i$ was deduced by a projection-based approach and without the notion of duality. If we want the transformation matrix to be orthogonal for the PS approach, we perform the SVD of $\mathbf{W}_i^T \mathbf{W}_0 = \mathbf{U}_{\mathbf{W},i} \mathbf{\Sigma}_{\mathbf{W},i} \mathbf{Z}_{\mathbf{W},i}^T$ and we obtain in analogy to the transformation matrix (3.13)

$$\mathbf{M}_i = \mathbf{U}_{\mathbf{W},i} \mathbf{Z}_{\mathbf{W},i}^T \in \mathbb{O}_q. \tag{3.25}$$

The three approaches which are summarized in Table 3.2 adjust the ROBs to the reference ROB in order to achieve accurate matrix interpolation.

Table 3.2: Comparison of the approaches for the adjustment of the left ROBs.

	$\mathbf{M}_i \in \mathbb{R}^{q \times q}$	$\mathbf{M}_i \in \mathbb{O}_q$	
DS	$\mathbf{M}_i = \mathbf{W}_i^{(l)+} \mathbf{W}_0$	$\mathbf{M}_i = \mathbf{U}_{\mathbf{W},i} \mathbf{Z}_{\mathbf{W},i}^T$	PS
MAC	$\mathbf{M}_i = (\mathbf{W}_0^T \mathbf{W}_i)^{-1}$		

Another design objective for matrices $\mathbf{M}_i$ could be to reduce the computational effort. Then, the following formula can be used [11, 51, 76]:

$$\mathbf{M}_i = (\hat{\mathbf{E}}_i \mathbf{T}_i)^{-T}. \tag{3.26}$$

This choice delivers for each transformed matrix $\tilde{\mathbf{E}}_i = \mathbf{M}_i^T \hat{\mathbf{E}}_i \mathbf{T}_i = (\hat{\mathbf{E}}_i \mathbf{T}_i)^{-1} \hat{\mathbf{E}}_i \mathbf{T}_i = \mathbf{I}_q$ the identity matrix which we can disregard for the interpolation process. Hence, it reduces the number of interpolants which we will have to construct and evaluate. This approach implicitly adopts the formula from (3.24) by choosing for every system its own reference basis $\mathbf{W}_{0,i} = \mathbf{E}_i \mathbf{V}_i$ instead of a common reference basis for all systems:

$$\mathbf{M}_i = \hat{\mathbf{E}}_i^{-T} \mathbf{T}_i^{-T} = (\mathbf{V}_i^T \mathbf{E}_i^T \mathbf{W}_i)^{-1} \mathbf{T}_i^{-T} = (\mathbf{W}_{0,i}^T \mathbf{W}_i)^{-1} \mathbf{T}_i^{-T}. \tag{3.27}$$

Hence, it might lead to an inaccurate adjustment of the ROBs.

The outputs of this step are the matrices $\mathbf{T}_1, \ldots, \mathbf{T}_N$ and $\mathbf{M}_1, \ldots, \mathbf{M}_N$. Several approaches can be used for their calculation. The ones summarized in Table 3.1 and 3.2 adjust the ROBs for accuracy reasons. The selection with minimum interpolation error is problem-dependent. The approach (3.26) reduces the computational effort.

3.5 Mapping of the System Matrices

After adjusting the right and left ROBs, we finally obtain the set of compatible reduced systems $\tilde{\mathcal{G}} = \{\tilde{\mathbf{G}}_1, \ldots, \tilde{\mathbf{G}}_N\}$, where e.g. the i-th system is for $i \in \{1, \ldots, N\}$

$$\tilde{\mathbf{G}}_i : \begin{cases} \overbrace{\mathbf{M}_i^T \hat{\mathbf{E}}_i \mathbf{T}_i}^{\tilde{\mathbf{E}}_i} \dot{\tilde{\mathbf{x}}}_i(t) = \overbrace{\mathbf{M}_i^T \hat{\mathbf{A}}_i \mathbf{T}_i}^{\tilde{\mathbf{A}}_i} \tilde{\mathbf{x}}_i(t) + \overbrace{\mathbf{M}_i^T \hat{\mathbf{B}}_i}^{\tilde{\mathbf{B}}_i} \mathbf{u}(t) \\ \qquad\qquad \tilde{\mathbf{y}}_i(t) = \underbrace{\hat{\mathbf{C}}_i \mathbf{T}_i}_{\tilde{\mathbf{C}}_i} \tilde{\mathbf{x}}_i(t). \end{cases} \tag{3.28}$$

The system matrices $\tilde{\mathbf{E}}_i, \tilde{\mathbf{A}}_i, \tilde{\mathbf{B}}_i, \tilde{\mathbf{C}}_i$ are elements of a Riemannian manifold. Therefore, the interpolated matrices should as well be elements of the respective manifold. The concept of matrix manifolds was introduced for pMOR in [10, 6, 7, 11, 51]. The following matrix manifolds are of interest:

- Matrix $\tilde{\mathbf{B}}_i$ belongs to the manifold of real matrices $\mathbb{R}^{q \times r}$ and $\tilde{\mathbf{C}}_i$ to the manifold of real matrices $\mathbb{R}^{m \times q}$.
- In general, matrices $\tilde{\mathbf{E}}_i, \tilde{\mathbf{A}}_i$ can belong to the manifold of real $\mathbb{R}^{q \times q}$ or nonsingular matrices $\mathbb{GL}_q$. Hence, we can choose between $K_{\text{MA}} = 2$ candidate manifolds.

We choose a reference system $\tilde{\mathbf{G}}_{i_0}$ and map the matrices $\tilde{\mathbf{E}}_i, \tilde{\mathbf{A}}_i, \tilde{\mathbf{B}}_i, \tilde{\mathbf{C}}_i$ onto the tangent space of the respective matrix manifold by the logarithmic mapping with a formula from Table 3.3. For example, take matrix $\tilde{\mathbf{A}}_i$ of the i-th system $\tilde{\mathbf{G}}_i$ and let $\tilde{\mathbf{A}}_{i_0}$ belong to the reference system. Then, the mapped matrix is $\mathbf{\Gamma}_i^{\tilde{\mathbf{A}}} = \mathrm{Log}_{\tilde{\mathbf{A}}_{i_0}}(\tilde{\mathbf{A}}_i)$. In analogy, matrices $\tilde{\mathbf{E}}_i$, $\tilde{\mathbf{B}}_i$ and $\tilde{\mathbf{C}}_i$ are mapped to $\mathbf{\Gamma}_i^{\tilde{\mathbf{E}}}$, $\mathbf{\Gamma}_i^{\tilde{\mathbf{B}}}$ and $\mathbf{\Gamma}_i^{\tilde{\mathbf{C}}}$, respectively.

Table 3.3: Exponential and logarithmic mappings for matrix manifolds of interest.

	Real matrix $\tilde{\mathbf{X}}_i \in \mathbb{R}^{a\times b}$	Nonsingular matrix $\tilde{\mathbf{X}}_i \in \mathbb{GL}_a$
$\mathrm{Log}_{\tilde{\mathbf{X}}_{i_0}}(\tilde{\mathbf{X}}_i)$	$\tilde{\mathbf{X}}_i - \tilde{\mathbf{X}}_{i_0}$	$\log(\tilde{\mathbf{X}}_i\tilde{\mathbf{X}}_{i_0}^{-1})$
$\mathrm{Exp}_{\tilde{\mathbf{X}}_{i_0}}(\mathbf{\Gamma}^{\tilde{\mathbf{X}}})$	$\tilde{\mathbf{X}}_{i_0} + \mathbf{\Gamma}^{\tilde{\mathbf{X}}}$	$\exp(\mathbf{\Gamma}^{\tilde{\mathbf{X}}})\tilde{\mathbf{X}}_{i_0}$

Remark: The mapping for nonsingular matrices is not always possible. If e.g. $\tilde{\mathbf{A}}_i$ is not in the neighborhood of $\tilde{\mathbf{A}}_{i_0}$ and hence, the matrix $\tilde{\mathbf{A}}_i\tilde{\mathbf{A}}_{i_0}^{-1}$ has negative eigenvalues, then $\log(\tilde{\mathbf{A}}_i\tilde{\mathbf{A}}_{i_0}^{-1})$ may not be unique and delivers imaginary matrices. This follows from the following theorem.

Theorem 3.1 ([48]). *A real matrix has a unique and real logarithm if and only if it is nonsingular and each Jordan block belonging to a negative eigenvalue occurs an even number of times.*

As a side note, a group of matrices is presented to which the mapping onto the tangent space of nonsingular matrices can always be applied. Such matrices arise in strictly dissipative systems from Definition 2.2.

Corollary 3.1. *If $\tilde{\mathbf{A}}_i + \tilde{\mathbf{A}}_i^T < \mathbf{0}$, then the product $\tilde{\mathbf{A}}_i\tilde{\mathbf{A}}_{i_0}^{-1}$ only has complex and positive eigenvalues and hence, has a unique and real logarithm.*

Proof. It shall be investigated if the real eigenvalues λ_j to the real eigenvectors $\mathbf{v}_j$ of $\tilde{\mathbf{A}}_i\tilde{\mathbf{A}}_{i_0}^{-1}$ can be negative.

$$\begin{aligned}&(\tilde{\mathbf{A}}_i\tilde{\mathbf{A}}_{i_0}^{-1})\mathbf{v}_j = \lambda_j\mathbf{v}_j \Rightarrow \tilde{\mathbf{A}}_{i_0}^{-1}\mathbf{v}_j = \lambda_j\tilde{\mathbf{A}}_i^{-1}\mathbf{v}_j\\ \Rightarrow\ &\underbrace{\mathbf{v}_j^T\tilde{\mathbf{A}}_{i_0}^{-1}\mathbf{v}_j}_{<0} = \lambda_j\underbrace{\mathbf{v}_j^T\tilde{\mathbf{A}}_i^{-1}\mathbf{v}_j}_{<0} \Rightarrow \lambda_j > 0 \quad \forall\ j = \{1,\ldots,q\}.\end{aligned}$$

Hence, the eigenvalues of $\tilde{\mathbf{A}}_i\tilde{\mathbf{A}}_{i_0}^{-1}$ can only be complex or positive. Therefore, the logarithm is unique and real according to Theorem 3.1. ■

To sum up, we can choose between $K_{\mathrm{MA}} = 2$ candidate manifolds: the manifold of nonsingular λ_1^{MA} or real matrices λ_2^{MA}. The output of this step are the mapped matrices $\mathbf{\Gamma}_i^{\tilde{\mathbf{E}}}$, $\mathbf{\Gamma}_i^{\tilde{\mathbf{A}}}$, $\mathbf{\Gamma}_i^{\tilde{\mathbf{B}}}$ and $\mathbf{\Gamma}_i^{\tilde{\mathbf{C}}}$ for $i \in \{1,\ldots,N\}$.

3.6 Construction of the Interpolants

For the interpolation between the mapped matrices, an interpolant $z^{jk}_{\tilde{\mathbf{A}}} : \mathcal{D} \to \mathbb{R}$ is constructed for all q^2 matrix entries of the matrices $\boldsymbol{\Gamma}^{\tilde{\mathbf{A}}}_1, \ldots, \boldsymbol{\Gamma}^{\tilde{\mathbf{A}}}_N$. For example, the interpolant of the jk-th matrix entry is

$$z^{jk}_{\tilde{\mathbf{A}}}(\mathbf{p}) = \textstyle\sum_{i=1}^{N} \alpha^{jk}_{\tilde{\mathbf{A}},i} \omega^{jk}_{\tilde{\mathbf{A}},i}(\mathbf{p}) \tag{3.29}$$

with basis functions $\omega^{jk}_{\tilde{\mathbf{A}},1}, \ldots, \omega^{jk}_{\tilde{\mathbf{A}},N} : \mathcal{D} \to \mathbb{R}$ and coefficients $\alpha^{jk}_{\tilde{\mathbf{A}},1}, \ldots, \alpha^{jk}_{\tilde{\mathbf{A}},N} \in \mathbb{R}$. Then, we obtain the matrix of interpolants

$$\boldsymbol{\Gamma}^{\tilde{\mathbf{A}}}(\mathbf{p}) = \begin{bmatrix} z^{11}_{\tilde{\mathbf{A}}}(\mathbf{p}) & \ldots & z^{1q}_{\tilde{\mathbf{A}}}(\mathbf{p}) \\ \vdots & & \vdots \\ z^{q1}_{\tilde{\mathbf{A}}}(\mathbf{p}) & \ldots & z^{qq}_{\tilde{\mathbf{A}}}(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} \alpha^{11}_{\tilde{\mathbf{A}},i} \omega^{11}_{\tilde{\mathbf{A}},i}(\mathbf{p}) & \ldots & \sum_{i=1}^{N} \alpha^{1q}_{\tilde{\mathbf{A}},i} \omega^{1q}_{\tilde{\mathbf{A}},i}(\mathbf{p}) \\ \vdots & & \vdots \\ \sum_{i=1}^{N} \alpha^{q1}_{\tilde{\mathbf{A}},i} \omega^{q1}_{\tilde{\mathbf{A}},i}(\mathbf{p}) & \ldots & \sum_{i=1}^{N} \alpha^{qq}_{\tilde{\mathbf{A}},i} \omega^{qq}_{\tilde{\mathbf{A}},i}(\mathbf{p}) \end{bmatrix}. \tag{3.30}$$

If we use the same basis functions $\omega_1(\mathbf{p}), \ldots, \omega_N(\mathbf{p})$ for all matrix entries, we obtain the matrix interpolant

$$\boldsymbol{\Gamma}^{\tilde{\mathbf{A}}}(\mathbf{p}) = \begin{bmatrix} \sum_{i=1}^{N} \alpha^{11}_{\tilde{\mathbf{A}},i} \omega_i(\mathbf{p}) & \ldots & \sum_{i=1}^{N} \alpha^{1q}_{\tilde{\mathbf{A}},i} \omega_i(\mathbf{p}) \\ \vdots & & \vdots \\ \sum_{i=1}^{N} \alpha^{q1}_{\tilde{\mathbf{A}},i} \omega_i(\mathbf{p}) & \ldots & \sum_{i=1}^{N} \alpha^{qq}_{\tilde{\mathbf{A}},i} \omega_i(\mathbf{p}) \end{bmatrix} = \sum_{i=1}^{N} \omega_i(\mathbf{p}) \underbrace{\begin{bmatrix} \alpha^{11}_{\tilde{\mathbf{A}},i} & \ldots & \alpha^{1q}_{\tilde{\mathbf{A}},i} \\ \vdots & & \vdots \\ \alpha^{q1}_{\tilde{\mathbf{A}},i} & \ldots & \alpha^{qq}_{\tilde{\mathbf{A}},i} \end{bmatrix}}_{\boldsymbol{\alpha}^{\tilde{\mathbf{A}}}_i}, \tag{3.31}$$

where $\boldsymbol{\alpha}^{\tilde{\mathbf{A}}}_1, \ldots, \boldsymbol{\alpha}^{\tilde{\mathbf{A}}}_N$ are matrices containing the coefficients. In analogy, the matrix interpolants $\boldsymbol{\Gamma}^{\tilde{\mathbf{E}}}(\mathbf{p})$, $\boldsymbol{\Gamma}^{\tilde{\mathbf{B}}}(\mathbf{p})$ and $\boldsymbol{\Gamma}^{\tilde{\mathbf{C}}}(\mathbf{p})$ comprise q^2, qr and mq interpolants, respectively.

For constructing e.g. the interpolant (3.31), we firstly decide which basis functions $\omega_1(\mathbf{p}), \ldots, \omega_N(\mathbf{p})$ we use. Basically, we can use every interpolation method with corresponding basis functions. The most promising types of methods are polynomial interpolation (cf. Appendix A.2.1), piecewise polynomial interpolation (cf. Appendix A.2.2), interpolation with radial basis functions (RBF) (cf. Appendix A.3.1) or Kriging (cf. Appendix A.3.2). This results in a finite number K_{MI} of candidate interpolation methods and corresponding basis functions $\lambda^{\mathrm{MI}}_1, \ldots, \lambda^{\mathrm{MI}}_{K_{\mathrm{MI}}}$ and we choose one of them.

Secondly, we calculate the coefficients $\alpha^{jk}_{\tilde{\mathbf{A}},1}, \ldots, \alpha^{jk}_{\tilde{\mathbf{A}},N}$ of each interpolant (3.29). For example, if we use polynomial interpolation, we collect the data for the grid points $\{(\mathbf{p}_1, \gamma^{jk}_{\tilde{\mathbf{A}}}(\mathbf{p}_1)), \ldots, (\mathbf{p}_N, \gamma^{jk}_{\tilde{\mathbf{A}}}(\mathbf{p}_N))\}$ where $\gamma^{jk}_{\tilde{\mathbf{A}}}(\mathbf{p}_1), \ldots, \gamma^{jk}_{\tilde{\mathbf{A}}}(\mathbf{p}_N)$ are the jk-th entries of the mapped matrices $\boldsymbol{\Gamma}^{\tilde{\mathbf{A}}}_1, \ldots, \boldsymbol{\Gamma}^{\tilde{\mathbf{A}}}_N$. Then, we insert the data into the interpolant (3.29) and obtain a system of linear equations:

$$\underbrace{\begin{bmatrix} \omega_1(\mathbf{p}_1) & \omega_2(\mathbf{p}_1) & \dots & \omega_N(\mathbf{p}_1) \\ \omega_1(\mathbf{p}_2) & \omega_2(\mathbf{p}_2) & \dots & \omega_N(\mathbf{p}_2) \\ \vdots & \vdots & & \vdots \\ \omega_1(\mathbf{p}_N) & \omega_2(\mathbf{p}_N) & \dots & \omega_N(\mathbf{p}_N) \end{bmatrix}}_{\boldsymbol{\omega}} \underbrace{\begin{bmatrix} \alpha^{jk}_{\tilde{\mathbf{A}},1} \\ \alpha^{jk}_{\tilde{\mathbf{A}},2} \\ \vdots \\ \alpha^{jk}_{\tilde{\mathbf{A}},N} \end{bmatrix}}_{\boldsymbol{\alpha}^{jk}_{\tilde{\mathbf{A}}}} = \underbrace{\begin{bmatrix} \gamma^{jk}_{\tilde{\mathbf{A}}}(\mathbf{p}_1) \\ \gamma^{jk}_{\tilde{\mathbf{A}}}(\mathbf{p}_2) \\ \vdots \\ \gamma^{jk}_{\tilde{\mathbf{A}}}(\mathbf{p}_N) \end{bmatrix}}_{\gamma^{jk}_{\tilde{\mathbf{A}}}} . \tag{3.32}$$

We compute for the first interpolant $z^{11}_{\tilde{\mathbf{A}}}(\mathbf{p})$ the LU decomposition $\mathbf{P}\boldsymbol{\omega} = \mathbf{L}\mathbf{U}$ with lower and upper triangular matrix $\mathbf{L}$ and $\mathbf{U}$ and permutation matrix $\mathbf{P}$. These matrices are stored and reused in order to rapidly solve the system of linear equations (3.32) for the remaining interpolants by forward and back substitution [152, 3]. Hence, we obtain the matrices of coefficients $\boldsymbol{\alpha}^{\tilde{\mathbf{E}}}_i, \boldsymbol{\alpha}^{\tilde{\mathbf{A}}}_i, \boldsymbol{\alpha}^{\tilde{\mathbf{B}}}_i, \boldsymbol{\alpha}^{\tilde{\mathbf{C}}}_i$ with $i \in \{1, \dots, N\}$.

For the special case of cardinal basis functions (A.2) such as for linear or Lagrange polynomial functions we directly obtain the matrices containing the coefficients without solving a system of linear equations (3.32), see the following proposition.

Proposition 3.2. *For cardinal basis functions (A.2) the matrix interpolants consists of the superposition of the mapped matrices:*

$$\begin{aligned} \boldsymbol{\Gamma}^{\tilde{\mathbf{E}}}(\mathbf{p}) &= \textstyle\sum_{i=1}^{N} \omega_i(\mathbf{p}) \boldsymbol{\Gamma}^{\tilde{\mathbf{E}}}_i, \\ \boldsymbol{\Gamma}^{\tilde{\mathbf{A}}}(\mathbf{p}) &= \textstyle\sum_{i=1}^{N} \omega_i(\mathbf{p}) \boldsymbol{\Gamma}^{\tilde{\mathbf{A}}}_i, \\ \boldsymbol{\Gamma}^{\tilde{\mathbf{B}}}(\mathbf{p}) &= \textstyle\sum_{i=1}^{N} \omega_i(\mathbf{p}) \boldsymbol{\Gamma}^{\tilde{\mathbf{B}}}_i, \\ \boldsymbol{\Gamma}^{\tilde{\mathbf{C}}}(\mathbf{p}) &= \textstyle\sum_{i=1}^{N} \omega_i(\mathbf{p}) \boldsymbol{\Gamma}^{\tilde{\mathbf{C}}}_i. \end{aligned} \tag{3.33}$$

Proof. Calculate for example for the interpolant (3.29) the coefficients $\alpha^{jk}_{\tilde{\mathbf{A}},1}, \dots, \alpha^{jk}_{\tilde{\mathbf{A}},N}$ according to the system of linear equations (3.32). For cardinal basis functions with $\omega_i(\mathbf{p}_u) = \delta_{iu}$ for $i, u \in \{1, \dots, N\}$, it follows

$$\mathbf{I}_N \boldsymbol{\alpha}^{jk}_{\tilde{\mathbf{A}}} = \boldsymbol{\gamma}^{jk}_{\tilde{\mathbf{A}}}. \tag{3.34}$$

Hence, we obtain $\boldsymbol{\alpha}^{jk}_{\tilde{\mathbf{A}}} = \boldsymbol{\gamma}^{jk}_{\tilde{\mathbf{A}}}$ and consequently, the matrices containing the coefficients are the mapped matrices $\boldsymbol{\alpha}^{\tilde{\mathbf{A}}}_i = \boldsymbol{\Gamma}^{\tilde{\mathbf{A}}}_i$ with $i \in \{1, \dots, N\}$. The same holds true for the remaining matrix interpolants. ■

To sum up, we can choose between K_{MI} basis functions $\lambda^{\mathrm{MI}}_1, \dots, \lambda^{\mathrm{MI}}_{K_{\mathrm{MI}}}$ and we obtain the matrices interpolants $\boldsymbol{\Gamma}^{\tilde{\mathbf{E}}}(\mathbf{p})$, $\boldsymbol{\Gamma}^{\tilde{\mathbf{A}}}(\mathbf{p})$, $\boldsymbol{\Gamma}^{\tilde{\mathbf{B}}}(\mathbf{p})$, $\boldsymbol{\Gamma}^{\tilde{\mathbf{C}}}(\mathbf{p})$. The system which is constructed using these interpolants is referred to as $\tilde{\mathbf{G}}(\mathbf{p})$.

3.7 Evaluation of the Interpolants

The steps discussed so far are performed in the offline phase and have to be done only once. In the online phase, i.e. as soon as a reduced system is supposed to be calculated for a new parameter vector $\mathbf{p}_{\text{int}} \in \mathcal{D}$, we simply evaluate the matrix interpolants $\mathbf{\Gamma}^{\tilde{\mathbf{E}}}(\mathbf{p}_{\text{int}})$, $\mathbf{\Gamma}^{\tilde{\mathbf{A}}}(\mathbf{p}_{\text{int}})$, $\mathbf{\Gamma}^{\tilde{\mathbf{B}}}(\mathbf{p}_{\text{int}})$ and $\mathbf{\Gamma}^{\tilde{\mathbf{C}}}(\mathbf{p}_{\text{int}})$, see e.g. for (3.31)

$$\mathbf{\Gamma}^{\tilde{\mathbf{A}}}(\mathbf{p}_{\text{int}}) = \sum_{i=1}^{N} \omega_i(\mathbf{p}_{\text{int}}) \begin{bmatrix} \alpha_{\tilde{\mathbf{A}},i}^{11} & \cdots & \alpha_{\tilde{\mathbf{A}},i}^{1q} \\ \vdots & & \vdots \\ \alpha_{\tilde{\mathbf{A}},i}^{q1} & \cdots & \alpha_{\tilde{\mathbf{A}},i}^{qq} \end{bmatrix}. \tag{3.35}$$

This means the parameter value $\mathbf{p}_{\text{int}}$ is inserted into the basis functions and multiplied with the coefficients. Then, the resulting matrices are mapped back onto the respective manifold with exponential mappings $\text{Exp}_{\tilde{\mathbf{E}}_{i_0}}$,$\text{Exp}_{\tilde{\mathbf{A}}_{i_0}}$, $\text{Exp}_{\tilde{\mathbf{B}}_{i_0}}$, and $\text{Exp}_{\tilde{\mathbf{C}}_{i_0}}$ according to Table 3.3. The system matrices $\tilde{\mathbf{E}}(\mathbf{p}_{\text{int}}), \tilde{\mathbf{A}}(\mathbf{p}_{\text{int}}) \in \mathbb{R}^{q\times q}, \tilde{\mathbf{B}}(\mathbf{p}_{\text{int}}) \in \mathbb{R}^{q\times r}$, $\tilde{\mathbf{C}}(\mathbf{p}_{\text{int}}) \in \mathbb{R}^{m\times q}$ thus obtained are used to construct the interpolated reduced system

$$\tilde{\mathbf{G}}(\mathbf{p}_{\text{int}}) : \begin{cases} \tilde{\mathbf{E}}(\mathbf{p}_{\text{int}})\dot{\tilde{\mathbf{x}}}(t) = \tilde{\mathbf{A}}(\mathbf{p}_{\text{int}})\tilde{\mathbf{x}}(t) + \tilde{\mathbf{B}}(\mathbf{p}_{\text{int}})\mathbf{u}(t) \\ \qquad\quad \tilde{\mathbf{y}}(t) = \tilde{\mathbf{C}}(\mathbf{p}_{\text{int}})\tilde{\mathbf{x}}(t). \end{cases} \tag{3.36}$$

3.8 Error Analysis

In this section, the error types which appear in the context of the proposed method of pMOR are discussed. The following notations are introduced at the example of the $\mathcal{H}_p$-norm [15], but they also hold true for other norms. The total error $e_{\text{T}} : \mathcal{D} \to \mathbb{R}_0^+$ of the interpolated reduced system $\tilde{\mathbf{G}}(\mathbf{p})$ is referred to as the error compared to the original one $\mathbf{G}(\mathbf{p})$ for parameter $\mathbf{p}$:

$$e_{\text{T}}(\mathbf{p}) = \frac{||\mathbf{G}(\mathbf{p}) - \tilde{\mathbf{G}}(\mathbf{p})||_{\mathcal{H}_p}}{||\mathbf{G}(\mathbf{p})||_{\mathcal{H}_p}}. \tag{3.37}$$

Using the triangle inequality of a vector space yields an upper bound for the error:

$$\begin{aligned} e_{\text{T}}(\mathbf{p}) &= \frac{||\mathbf{G}(\mathbf{p}) - \tilde{\mathbf{G}}(\mathbf{p})||_{\mathcal{H}_p}}{||\mathbf{G}(\mathbf{p})||_{\mathcal{H}_p}} = \frac{||\mathbf{G}(\mathbf{p}) - \hat{\mathbf{G}}(\mathbf{p}) + \hat{\mathbf{G}}(\mathbf{p}) - \tilde{\mathbf{G}}(\mathbf{p})||_{\mathcal{H}_p}}{||\mathbf{G}(\mathbf{p})||_{\mathcal{H}_p}} \\ &\leq \underbrace{\frac{||\mathbf{G}(\mathbf{p}) - \hat{\mathbf{G}}(\mathbf{p})||_{\mathcal{H}_p}}{||\mathbf{G}(\mathbf{p})||_{\mathcal{H}_p}}}_{e_{\text{P}}(\mathbf{p})} + \underbrace{\frac{||\hat{\mathbf{G}}(\mathbf{p}) - \tilde{\mathbf{G}}(\mathbf{p})||_{\mathcal{H}_p}}{||\hat{\mathbf{G}}(\mathbf{p})||_{\mathcal{H}_p}}}_{e_{\text{I}}(\mathbf{p})} \cdot \underbrace{\frac{||\hat{\mathbf{G}}(\mathbf{p})||_{\mathcal{H}_p}}{||\mathbf{G}(\mathbf{p})||_{\mathcal{H}_p}}}_{V_{\text{P}}(\mathbf{p})}. \end{aligned} \tag{3.38}$$

The upper bound of the overall error can be decomposed into the error due to projection $e_{\mathrm{P}} : \mathcal{D} \to \mathbb{R}_0^+$ between the original system and the directly reduced one $\hat{\mathbf{G}}(\mathbf{p})$ obtained by projection-based MOR from Section 2.2 at parameter vector $\mathbf{p}$ and the error due to interpolation $e_{\mathrm{I}} : \mathcal{D} \to \mathbb{R}_0^+$ between the directly reduced system and the interpolated one scaled with $V_{\mathrm{P}} : \mathcal{D} \to \mathbb{R}^+$ due to projection-based MOR. On the other hand, the application of the reverse triangle inequality of the normed vector space delivers a lower bound for the overall error:

$$\begin{aligned} e_{\mathrm{T}}(\mathbf{p}) &= \frac{||\mathbf{G}(\mathbf{p}) - \tilde{\mathbf{G}}(\mathbf{p})||_{\mathcal{H}_p}}{||\mathbf{G}(\mathbf{p})||_{\mathcal{H}_p}} = \frac{||\mathbf{G}(\mathbf{p}) - \hat{\mathbf{G}}(\mathbf{p}) - \big(-\hat{\mathbf{G}}(\mathbf{p}) + \tilde{\mathbf{G}}(\mathbf{p})\big)||_{\mathcal{H}_p}}{||\mathbf{G}(\mathbf{p})||_{\mathcal{H}_p}} \\ &\geq \left| \underbrace{\frac{||\mathbf{G}(\mathbf{p}) - \hat{\mathbf{G}}(\mathbf{p})||_{\mathcal{H}_p}}{||\mathbf{G}(\mathbf{p})||_{\mathcal{H}_p}}}_{e_{\mathrm{P}}(\mathbf{p})} - \underbrace{\frac{||\hat{\mathbf{G}}(\mathbf{p}) - \tilde{\mathbf{G}}(\mathbf{p})||_{\mathcal{H}_p}}{||\hat{\mathbf{G}}(\mathbf{p})||_{\mathcal{H}_p}}}_{e_{\mathrm{I}}(\mathbf{p})} \cdot \underbrace{\frac{||\hat{\mathbf{G}}(\mathbf{p})||_{\mathcal{H}_p}}{||\mathbf{G}(\mathbf{p})||_{\mathcal{H}_p}}}_{V_{\mathrm{P}}(\mathbf{p})} \right|, \end{aligned} \tag{3.39}$$

where $|| - \hat{\mathbf{G}}(\mathbf{p}) + \tilde{\mathbf{G}}(\mathbf{p})||_{\mathcal{H}_p} = ||\hat{\mathbf{G}}(\mathbf{p}) - \tilde{\mathbf{G}}(\mathbf{p})||_{\mathcal{H}_p}$ was taken into account. Considering the upper (3.38) and lower bound (3.39), the overall error is located in the interval

$$e_{\mathrm{T}}(\mathbf{p}) \in [|e_{\mathrm{P}}(\mathbf{p}) - e_{\mathrm{I}}(\mathbf{p})V_{\mathrm{P}}(\mathbf{p})|, e_{\mathrm{P}}(\mathbf{p}) + e_{\mathrm{I}}(\mathbf{p})V_{\mathrm{P}}(\mathbf{p})]. \tag{3.40}$$

Of course, the bounds (3.40) are only of a more theoretical nature, but they allow us to understand that the overall error $e_{\mathrm{T}}(\mathbf{p})$ consists of two parts: the error $e_{\mathrm{P}}(\mathbf{p})$ due to projection-based MOR, the error $e_{\mathrm{I}}(\mathbf{p})$ due to interpolation and the correlation between the two. If both have a positive correlation, the overall error assumes its upper bound and if both have a negative correlation, it assumes its lower bound. Otherwise, it will have a value between the bounds. The bounds (3.40) also give an insight into three important special cases:

- Case 1: For accurate projection-based MOR at the grid points with $e_{\mathrm{P}}(\mathbf{p}) \to 0$ and $V_{\mathrm{P}}(\mathbf{p}) \to 1$ where $e_{\mathrm{P}}(\mathbf{p}) \ll e_{\mathrm{I}}(\mathbf{p})$ holds, the overall error can be computed very efficiently as only the interpolation error remains with $e_{\mathrm{T}}(\mathbf{p}) \to e_{\mathrm{I}}(\mathbf{p})$.
- Case 2: For accurate interpolation with $e_{\mathrm{I}}(\mathbf{p}) \to 0$ where $e_{\mathrm{I}}(\mathbf{p})V_{\mathrm{P}}(\mathbf{p}) \ll e_{\mathrm{P}}(\mathbf{p})$ holds, the overall error is dominated by the error due to projection-based MOR with $e_{\mathrm{T}}(\mathbf{p}) \to e_{\mathrm{P}}(\mathbf{p})$. This leads more importantly to the next case.
- Case 3: At the grid points the interpolation error vanishes with $e_{\mathrm{I}}(\mathbf{p}) = 0$ and the overall error is the one due to projection-based MOR with $e_{\mathrm{T}}(\mathbf{p}) = e_{\mathrm{P}}(\mathbf{p})$. Hence, the maximum error cannot be smaller than the error due to projection-based MOR at the grid points no matter how accurate the interpolation method is.

3.9 Integration of Existing Methods into the General Framework

3.9.1 Selection of the Degrees of Freedom for Existing Methods

It was pointed out in Section 2.3.2 that the approaches of pMOR by matrix interpolation from [112, 128] and [6, 11] were developed independently and presented as different methods. In fact, they can be integrated into the proposed general framework as special cases with a different selection of the degrees of freedom. The selections corresponding to the current approaches are shown in the left and right column of Table 3.4.

Table 3.4: Selection of the degrees of freedom for existing methods.

	Panzer et al. [128]	Amsallem et al. [11]
Reference ROB $\mathbf{V}_0$	non-/weighted SVD $*$	fixed
Adjust. right ROBs	$\mathbf{T}_i = (\mathbf{V}_0^T\mathbf{V}_i)^{-1}$ $*$	$\mathbf{T}_i = \mathbf{U}_{\mathbf{V},i}\mathbf{Z}_{\mathbf{V},i}^T$
Reference ROB $\mathbf{W}_0$	$\mathbf{W}_0 = \mathbf{V}_0$ $*$	$\mathbf{W}_{0,i} = \mathbf{E}_i\mathbf{V}_i$
Adjust. left ROBs	$\mathbf{M}_i = (\mathbf{W}_0^T\mathbf{W}_i)^{-1}$ $*$	$\mathbf{M}_i = (\mathbf{W}_{0,i}^T\mathbf{W}_i)^{-1}\mathbf{T}_i$
Interpol. manifold	real matrices	real/nonsingular matrices $*$
Interpol. method	cardinal basis functions	element-wise $*$

Besides these two special selections, there are many other possible combinations. Since the selection leading to the most accurate reduced model is problem-dependent for the adjustment of the ROBs and the interpolation method and manifold, a restriction to the two special choices is not useful. This is demonstrated in the next section by means of a numerical example.

3.9.2 Numerical Example

The example describes the motion of a 3D cantilever Timoshenko beam which is shown in Figure 3.4. The parameter of the beam is the length $L \in [0.8, 2]$m. The model input is the vertical force $F(t)$ which is applied at the tip of the beam and the model output is the vertical displacement at this point. A finite element model which is proposed in [125] comprises 100 nodes along the beam, each having six degrees of freedom: three translational displacements u, v, w and three rotational degrees of freedom α, β, γ with respect to the z_1-, z_2-, z_3-axis, respectively. This delivers a parameter-dependent SISO system of order $n = 1200$:

$$G(L) : \begin{cases} \mathbf{E}(L)\dot{\mathbf{x}}(t) = \mathbf{A}(L)\mathbf{x}(t) + \mathbf{b}F(t) \\ \qquad y(t) = \mathbf{c}^T\mathbf{x}(t), \end{cases} \tag{3.41}$$

where the state vector $\mathbf{x}(t) \in \mathbb{R}^{1200}$ comprises the six degrees of freedom for all elements and their derivatives. The system matrices $\mathbf{E}(L)$, $\mathbf{A}(L)$ have nonlinear parameter dependencies. For testing the system, the $\mathcal{H}_2$-norm is used [15]. The magnitude of frequency responses of the beam are shown for different parameter values in Figure 3.5.

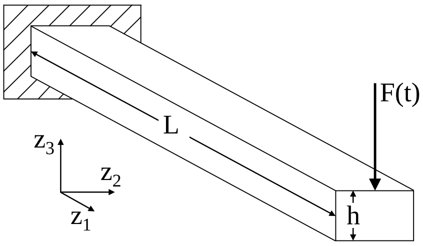


Figure 3.4: Depiction of the cantilever beam.

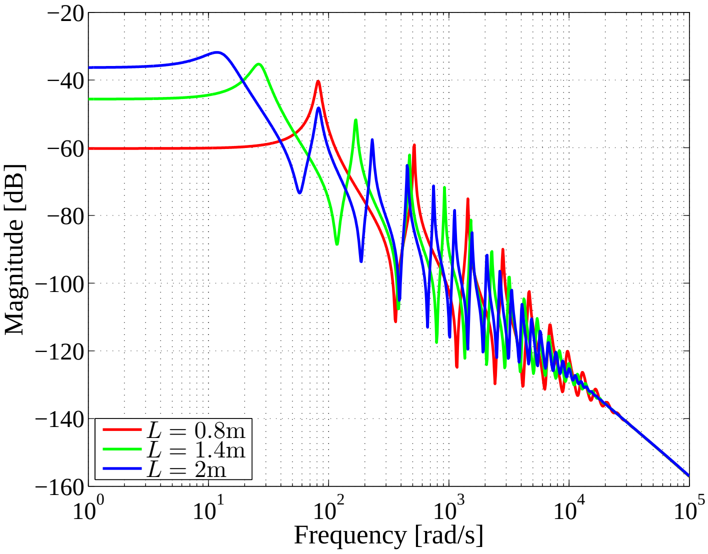


Figure 3.5: Plots of the magnitude of frequency responses for different parameter values.

The parameter domain is sampled for the values $L \in \{0.8, 1.2, 1.6, 2\}$m and the local models are reduced using a two-sided Krylov subspace method with the reduced order $q = 10$ and expansion points $s_0 = 0$. The remaining degrees of freedom for the general framework are selected as a combination of the two existing methods where the respective selection is labeled in Table 3.4 with $(*)$. For the calculation of $\mathbf{V}_0$ the non-weighted SVD is applied and the reduced system matrices are interpolated element-wise using cubic spline interpolation. This selection is compared to the latest methods from Panzer et al. [128] and Amsallem et al. [11]. The selection of the degrees of freedom for Panzer et al. is given in the left column of Table 3.4 with the non-weighted SVD for the calculation of $\mathbf{V}_0$ and Lagrange weighting functions (A.7) for interpolation. For Amsallem et al. the degrees of freedom are selected according to the right column of Table 3.4 where the reduced system matrices are interpolated element-wise using cubic spline interpolation. The reference ROB $\mathbf{V}_0$ is the ROB of the system at $L = 1.6$m. In Figure 3.6 the overall error e_{T} in $\mathcal{H}_2$-norm is shown for directly reduced order models (ROMs) using the above mentioned Krylov subspace method and for interpolated systems in the interval $L \in [0.8,\ 2]$m. At the grid points all considered methods share the overall error of the directly reduced models and between them an additional error due to interpolation occurs. In this example, the proposed selection delivers more accurate results than the two existing approaches. Hence, one can verify that it is not useful to restrict oneself to the two special cases. It will be explained in the next section how we can find the most accurate selection.

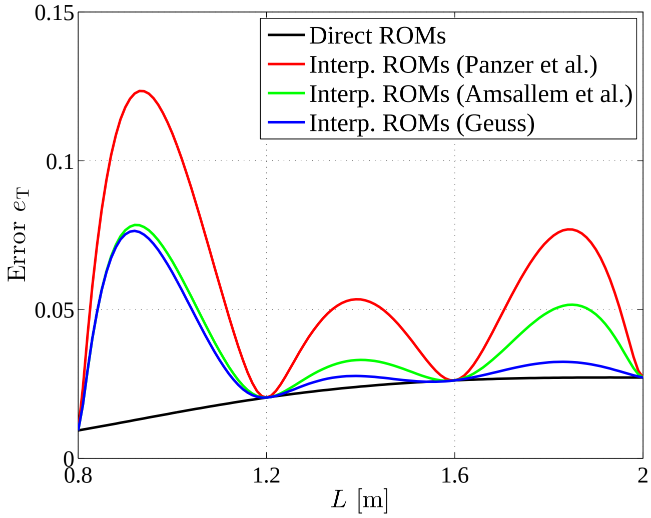


Figure 3.6: Overall error e_{T} in $\mathcal{H}_2$-norm for different interpolated ROMs at various lengths L for the cantilever beam.

4 A Black-Box Method for Interpolating Reduced System Matrices

4.1 Motivation

We want to approximate a parameter-dependent high-order system $\mathbf{G}(\mathbf{p})$ by a reduced system $\tilde{\mathbf{G}}(\mathbf{p})$ of order $q \ll n$ on the domain $\mathcal{D}$ using the framework from Section 3. It was pointed out that we have many options that we need to decide on. Firstly, we need to select the parameter vectors $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$. Secondly, the interpolation can take place on different matrix manifolds. Thirdly, there are various candidate interpolation methods. In most cases, the user does not have enough insight into the physics of the model in order to decide on the options leading to the most accurate reduced model. Hence, the user calls for an automatic approach which delivers a reduced model whose error is below a given tolerance e_{tol} using as few grid points as possible.

For selecting the interpolation method and manifold, current approaches such as [62] use a training grid for constructing the interpolated reduced system and an additional test grid for evaluating its accuracy. However, this is computationally expensive because of a large number of complex calculations and we lose a lot of grid points for training the model. For selecting the method efficiently, a heuristic measure is proposed in [51] for spline interpolation which selects the most appropriate matrix manifold. For the general case, it is still an open question how to find the interpolation method and manifold which result in the smallest error without performing high-order operations.

Refinement procedures for calculating the grid points $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ have been proposed for global POD and reduced basis methods which make use of efficient error estimators, see e.g. [84, 36, 91, 90, 129]. Unfortunately, these error estimators cannot be used for the framework from Section 3 as they disregard the effect of interpolation [23]. Therefore, current refinement schemes from [142, 38, 30] split the parameter domain into cells and the corners are the training points, which are used to construct the interpolated system. The center points of the cells or of the edges are the test points where the error of the interpolated systems compared to the high-order ones is calculated. If the errors are above a certain tolerance, the cells are divided into subcells. This iterative algorithm terminates when the error is sufficiently small for all cells. Unfortunately, these methods are expensive as they require high-order model solves at the test points and make inefficient use of the data as the test points cannot be used for training the model. Another approach, which is based on differential geometry, is proposed in [5]. The method divides the parameter domain in Delaunay triangles and calculates the geodesic distances of the subspaces at the grid points. Then, the next grid point is inserted in the middle of the edge with the largest geodesic distance. As the method does not provide an estimate for the model accuracy, it is an open question when to stop the refinement procedure.

In order to overcome the limitations of the reviewed methods, especially the high computational expense and the inefficient use of the data, we introduce in Section 4.2 a new error indicator in pMOR which is based on cross-validation. It only uses available data and allows us to efficiently choose the interpolation method minimizing an error measure (method selection, cf. Section 4.3), to efficiently find the next best grid points (model refinement, cf. Section 4.4) and to determine the accuracy of a parameter-dependent reduced system (model error prediction, cf. Section 4.5). These three methods are combined to a *Black-Box method.* This method automatically selects the best interpolation method, performs the refinement in the parameter space iteratively and stops when the maximum predicted error of the interpolated system falls below the given error tolerance e_{tol}. The procedure is shown in Figure 4.1. Numerical examples for the Black-Box method are given in Section 4.7.

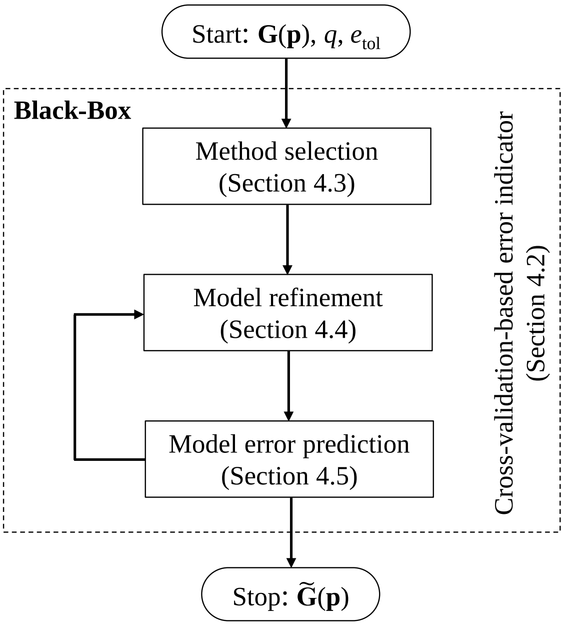

Start: $\mathbf{G}(\mathbf{p})$, q, e_{tol}
Black-Box
Method selection (Section 4.3)
Model refinement (Section 4.4)
Model error prediction (Section 4.5)
Cross-validation-based error indicator (Section 4.2)
Stop: $\widetilde{\mathbf{G}}(\mathbf{p})$

Figure 4.1: Structure of the proposed Black-Box method.

4.2 Cross-Validation-Based Error Indicator

4.2.1 Overview

In this section we introduce a new a posteriori error indicator for interpolation-based methods of pMOR such as the one proposed in Section 3. The only data we have is the set of original $\mathcal{G} = \{\mathbf{G}_1, \ldots, \mathbf{G}_N\}$ and of reduced systems $\tilde{\mathcal{G}} = \{\tilde{\mathbf{G}}_1, \ldots, \tilde{\mathbf{G}}_N\}$ at the grid points $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ where $N = |\mathcal{P}|$ is the number of points. We demand the calculation of the error indicator is based only on this available data. Therefore, we use an approach called cross-validation which is well-known in many fields of research [153, 4, 71]. In data mining it can be used to determine optimal parameters of an approximation model or to compare the accuracy of different models, see e.g. [25, 93]. In the design of experiments community it is employed to determine the accuracy of a surrogate model, see e.g. [18]. In this section we explain how we assign an error obtained by cross-validation to every grid point in $\mathcal{P}$ which in the following is referred to as *cross-validation-based error indicator*.

4.2.2 Assignment of the Error Indicator

Let us first describe the assignment procedure formally for the general case before we go into details for the different grid types. We take $k_{\mathrm{CV}} \leq N$ subsets $\mathcal{P}_1, \ldots, \mathcal{P}_{k_{\mathrm{CV}}}$, $\tilde{\mathcal{G}}_1, \ldots, \tilde{\mathcal{G}}_{k_{\mathrm{CV}}}$ out of the sets $\mathcal{P}$, $\tilde{\mathcal{G}}$ with the following property: for each $j \in \{1, \ldots, k_{\mathrm{CV}}\}$ we have a test set $\mathcal{P}_j^* \subseteq \mathcal{P}_j$ where $\mathcal{P}_1^* \cup \ldots \cup \mathcal{P}_{k_{\mathrm{CV}}}^* = \mathcal{P}$ and $\mathcal{P}_j^* \cap \mathcal{P}_u^* = \emptyset$ for $u \in \{1, \ldots, k_{\mathrm{CV}}\} \backslash j$. Hence, every grid point is a test point exactly once. Then, we construct for each j a parametric reduced system $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})$ with training sets $\mathcal{P} \backslash \mathcal{P}_j$ and $\tilde{\mathcal{G}} \backslash \tilde{\mathcal{G}}_j$ such as described in Section 3.6 and we evaluate each system $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})$ at the grid points from the corresponding test set $\mathcal{P}_j^*$ according to Section 3.7. Finally, we calculate the error indicator $e_{\mathrm{CV},j} : \mathcal{P}_j^* \to \mathbb{R}_0^+$ such as described in Section 4.2.3. This results—neglecting index j—in the set of error indicators $\mathcal{E}_{\mathrm{CV}} = \{e_{\mathrm{CV}}(\mathbf{p}_1), \ldots, e_{\mathrm{CV}}(\mathbf{p}_N)\}$. The procedure is described in Algorithm 3. How are the subsets $\mathcal{P}_j, \mathcal{P}_j^*, \tilde{\mathcal{G}}_j$ to be chosen? This depends on the grid type and is explained in the following for two types: the irregular grid such as described in Appendix A.3 and the regular grid such as described in Appendix A.2:

Irregular grid: We randomly partition the sets $\mathcal{P}$, $\tilde{\mathcal{G}}$ into k_{CV} folds $\mathcal{P}_1, \ldots, \mathcal{P}_{k_{\mathrm{CV}}}$, $\tilde{\mathcal{G}}_1, \ldots, \tilde{\mathcal{G}}_{k_{\mathrm{CV}}}$ of approximately equal size and have $\mathcal{P}_j^* = \mathcal{P}_j$ for all j, see e.g. [105, 17]. For the prominent case $k_{\mathrm{CV}} = N$, which is referred to as leave-one-out cross-validation, we have $|\mathcal{P}_j| = 1$. This means the test sets contain only one grid point. This is demonstrated in Figure 4.2. Leave-one-out cross-validation efficiently makes use of the

Algorithm 3 Assignment of the error indicator

```
Input: Set of reduced systems $\tilde{\mathcal{G}}$ and of grid points $\mathcal{P}$
Output: Set of error indicators $\mathcal{E}_{\mathrm{CV}}$
Initialize empty set $\mathcal{E}_{\mathrm{CV}} = \emptyset$
Take $k_{\mathrm{CV}}$ subsets $\mathcal{P}_1, \ldots, \mathcal{P}_{k_{\mathrm{CV}}}$, $\tilde{\mathcal{G}}_1, \ldots, \tilde{\mathcal{G}}_{k_{\mathrm{CV}}}$ from sets $\mathcal{P}$, $\tilde{\mathcal{G}}$
for $j = 1, \ldots, k_{\mathrm{CV}}$ do
    for $i = 1, \ldots, |\mathcal{P}_j^*|$ do
        Construct $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})$ with sets $\mathcal{P} \backslash \mathcal{P}_j$ and $\tilde{\mathcal{G}} \backslash \tilde{\mathcal{G}}_j$
        Evaluate $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p}_i)$ at grid point $\mathbf{p}_i \in \mathcal{P}_j^*$
        Calculate the error indicator $e_{\mathrm{CV},j}(\mathbf{p}_i)$ with formula (4.1), (4.4) or (4.11)
        Update the set $\mathcal{E}_{\mathrm{CV}} = \mathcal{E}_{\mathrm{CV}} \cup \{e_{\mathrm{CV},j}(\mathbf{p}_i)\}$
    end for
end for
```

data as it uses nearly all data points for training but it requires the construction of N parametric systems $\tilde{\mathbf{G}}_{\mathrm{CV},1}(\mathbf{p}), \ldots, \tilde{\mathbf{G}}_{\mathrm{CV},N}(\mathbf{p})$. Therefore, it is the method of choice for a moderate number of grid points. For a large number of grid points it is recommended to use, e.g., common values $k_{\mathrm{CV}} = 10$ or $k_{\mathrm{CV}} = 5$ because we have to construct only k_{CV} parameter-dependent systems $\tilde{\mathbf{G}}_{\mathrm{CV},1}(\mathbf{p}), \ldots, \tilde{\mathbf{G}}_{\mathrm{CV},k_{\mathrm{CV}}}(\mathbf{p})$ [93, 17].

Regular grid: The points of a regular grid with dimension d have a certain structure which must be preserved during the assignment procedure. The interior points belong to a d-dimensional structure, the boundary points belong to $d-1, \ldots, 1$-dimensional structures and the 0-dimensional structures are the corner points. Firstly, define the test sets $\mathcal{P}_1^* = \{\mathbf{p}_1\}, \ldots, \mathcal{P}_N^* = \{\mathbf{p}_N\}$ and, consequently, $k_{\mathrm{CV}} = N$ which means the test sets contain only one grid point. Then, let $\mathbf{p}_j$ with $j \in \{1, \ldots, N\}$ belong to a $l \in \{1, \ldots, d\}$-dimensional structure. Consider e.g. Figure 4.3 where an error indicator is assigned to an interior point $\mathbf{p}_j$ belonging to a $l = 2$-dimensional structure. We imagine a coordinate system to be shifted into this grid point and we leave 1-dimensional structures of points, which are lines spanned by the axes of the coordinate system, out. In Figure 4.4 the assignment is shown for a point $\mathbf{p}_j$ of a $l = 1$-dimensional boundary. Here, we leave all interior points of the 2-dimensional structure, the other edges, which are also 1-dimensional structures, and a 0-dimensional space, which is the point itself, out. For the general case, if we want to calculate the error indicator for the point $\mathbf{p}_j$, we imagine a coordinate system to be shifted into this grid point and we leave all grid points $\mathcal{P}_j$ and systems $\tilde{\mathcal{G}}_j$ out which lie on structures with dimension $l-1$ spanned by the axes and all points at other structures with dimensions greater or equal l. For 0-dimensional boundaries, which are corner points, this procedure leads to extrapolation. Hence, we use, e.g., the average of the error indicators of the neighbor points.

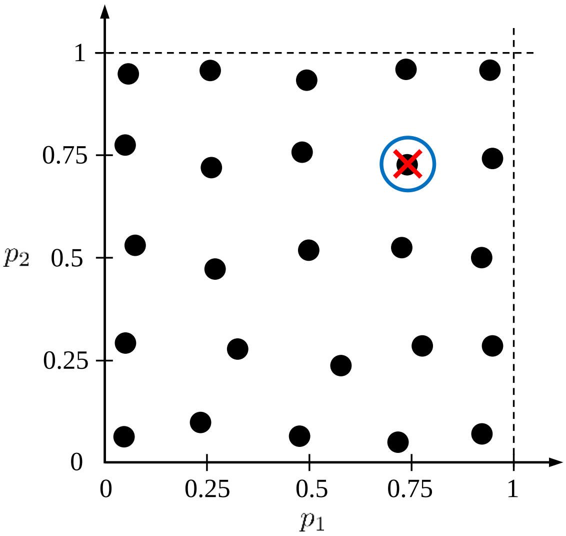


Figure 4.2: Assignment of the error indicator for an irregular grid with two parameters p_1, p_2. When the leave-one-out cross-validation error is to be calculated for the grid point indicated with a blue circle, only the test point itself is left out for building the interpolant. This point is denoted by a red cross.

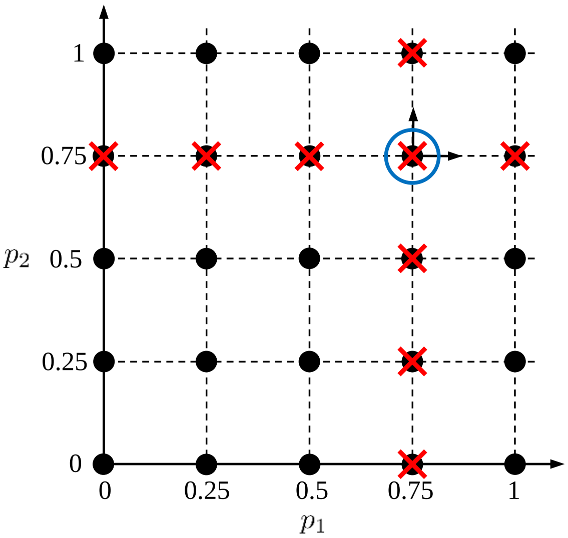


Figure 4.3: Assignment of the error indicator for interior points of a regular grid with two parameters p_1, p_2. When the cross-validation error is to be calculated for the grid point indicated with a blue circle, all points marked with a red cross which are on lines (1-dimensional spaces) including the test point are left out for building the interpolant.

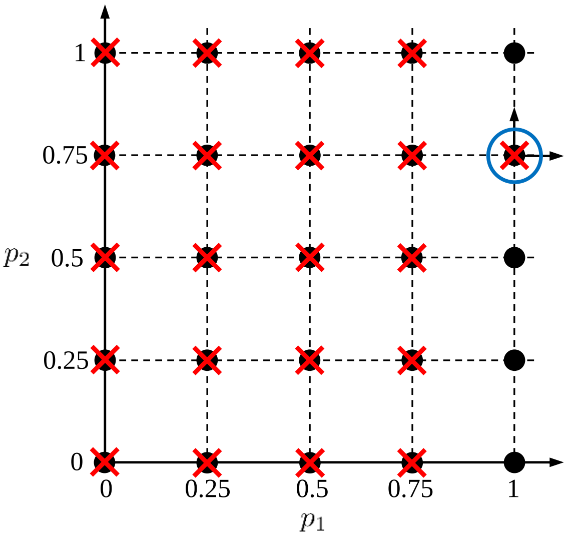


Figure 4.4: Assignment of the error indicator for points on a 1-dimensional boundary of a regular grid with two parameters p_1, p_2. When the cross-validation error is to be calculated for the grid point indicated with a blue circle, all red points outside the boundary and the test point itself (0-dimensional space) are left out for building the interpolant.

4.2.3 Error Computation

The error indicator $e_{\text{CV},j}(\mathbf{p})$ at grid point $\mathbf{p} \in \mathcal{P}_j^*$ is the error of the system obtained by cross-validation $\tilde{\mathbf{G}}_{\text{CV},j}(\mathbf{p})$ from Section 4.2.2 compared to a reference system $\mathbf{G}_\text{r}(\mathbf{p})$ of order n_r at the same grid point. The reference system can either be the reduced system $\mathbf{G}_\text{r}(\mathbf{p}) := \hat{\mathbf{G}}(\mathbf{p})$ with $n_\text{r} = q$ (cf. Section 4.3) or the original system $\mathbf{G}_\text{r}(\mathbf{p}) := \mathbf{G}(\mathbf{p})$ with $n_\text{r} = n$ (cf. Section 4.4). For the latter, we present formulas for the error computation which are split into a large-scale solve and a low-order solve. We will benefit from this in Section 4.4 as high-order operations will be done one time, the obtained values are then stored and reused in every iteration where we have a new system $\tilde{\mathbf{G}}_{\text{CV},j}(\mathbf{p})$. In this section, the calculation of the error indicator is explained for the norms of interest.

$\mathcal{L}_p$-norm of the output

The error indicator in $\mathcal{L}_p$-norm is the relative error between the output of the reference system $\mathbf{y}_\text{r}(t, \mathbf{p})$ and the reduced system obtained by cross-validation $\tilde{\mathbf{y}}_{\text{CV},j}(t, \mathbf{p})$:

$$e_{\text{CV},j}^{\mathcal{L}_p}(\mathbf{p}) = \frac{||\mathbf{y}_\text{r}(t, \mathbf{p}) - \tilde{\mathbf{y}}_{\text{CV},j}(t, \mathbf{p})||_{\mathcal{L}_p}}{||\mathbf{y}_\text{r}(t, \mathbf{p})||_{\mathcal{L}_p}}. \tag{4.1}$$

For approximating (4.1) we integrate system $\mathbf{G}_\text{r}(\mathbf{p})$ for K_L time steps with step size Δt with a problem-specific input signal and obtain $\mathbf{y}_\text{r}(\cdot, \mathbf{p}) = \{\mathbf{y}_\text{r}(t_0, \mathbf{p}), \ldots, \mathbf{y}_\text{r}(t_{K_\text{L}}, \mathbf{p})\}$, where for example the k-th value is $\mathbf{y}_\text{r}(t_k, \mathbf{p}) \in \mathbb{R}^m$. This has for sparse system matrices a computational complexity which is at most $\mathcal{O}(K_\text{L} n_\text{r}^3)$. Then, the $\mathcal{L}_p$-norm of the output of the reference system $\mathbf{G}_\text{r}(\mathbf{p})$ is approximately for $1 \leq p < \infty$ [15]

$$||\mathbf{y}_\text{r}(t, \mathbf{p})||_{\mathcal{L}_p} \approx \left(\Delta t \sum_{k=0}^{K_\text{L}} ||\mathbf{y}_\text{r}(t_k, \mathbf{p})||_p^p \right)^{1/p}. \tag{4.2}$$

The calculation of $\mathbf{y}_\text{r}(\cdot, \mathbf{p})$ and $||\mathbf{y}_\text{r}(t, \mathbf{p})||_{\mathcal{L}_p}$ has to be done only once per parameter vector $\mathbf{p}$ and the values are stored in order to reuse them. Every time the indicator $e_{\text{CV},j}^{\mathcal{L}_p}(\mathbf{p})$ is to be determined for a new reduced system obtained by cross-validation $\tilde{\mathbf{G}}_{\text{CV},j}(\mathbf{p})$, we calculate for the same time points the values $\tilde{\mathbf{y}}_{\text{CV},j}(\cdot, \mathbf{p}) = \{\tilde{\mathbf{y}}_{\text{CV},j}(t_0, \mathbf{p}), \ldots, \tilde{\mathbf{y}}_{\text{CV},j}(t_{K_\text{L}}, \mathbf{p})\}$, where for example the k-th value is $\tilde{\mathbf{y}}_{\text{CV},j}(t_k, \mathbf{p}) \in \mathbb{R}^m$. The computational complexity involving these dense reduced system matrices is only $\mathcal{O}(K_\text{L} q^3)$. Together with the stored values we finally obtain for the error indicator

$$e_{\text{CV},j}^{\mathcal{L}_p}(\mathbf{p}) \approx \frac{\left(\Delta t \sum_{k=0}^{K_\text{L}} ||\mathbf{y}_\text{r}(t_k, \mathbf{p}) - \tilde{\mathbf{y}}_{\text{CV},j}(t_k, \mathbf{p})||_p^p \right)^{1/p}}{||\mathbf{y}_\text{r}(t, \mathbf{p})||_{\mathcal{L}_p}}. \tag{4.3}$$

To sum up, when we calculate the error indicator $e^{\mathcal{L}_p}_{\mathrm{CV},j}(\mathbf{p})$ for system $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})$ at a new parameter $\mathbf{p}$, the complexity is at most $\mathcal{O}(K_{\mathrm{L}} n_{\mathrm{r}}^3)$. We introduce a set of stored values $\mathcal{S}$ which we update $\mathcal{S} = \mathcal{S} \cup \left\{(\mathbf{y}_{\mathrm{r}}(\cdot, \mathbf{p}), ||\mathbf{y}_{\mathrm{r}}(t, \mathbf{p})||_{\mathcal{L}_p})\right\}$ for each new parameter $\mathbf{p}$. For each additional calculation with a new system $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})$ at a previously used point, the complexity is only $\mathcal{O}(K_{\mathrm{L}} q^3)$ as the stored values $\mathcal{S}$ are reused.

$\mathcal{H}_\infty$-norm

The error indicator in $\mathcal{H}_\infty$-norm is the relative error between the reference system $\mathbf{G}_{\mathrm{r}}(\mathbf{p})$ and the reduced system obtained by cross-validation $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})$ [15]:

$$e^{\mathcal{H}_\infty}_{\mathrm{CV},j}(\mathbf{p}) = \frac{||\mathbf{G}_{\mathrm{r}}(\mathbf{p}) - \tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})||_{\mathcal{H}_\infty}}{||\mathbf{G}_{\mathrm{r}}(\mathbf{p})||_{\mathcal{H}_\infty}} = \frac{\sup\limits_{\omega \in \mathbb{R}} \sigma_{\max}(\mathbf{G}_{\mathrm{r}}(\mathrm{j}\omega, \mathbf{p}) - \tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathrm{j}\omega, \mathbf{p}))}{\sup\limits_{\omega \in \mathbb{R}} \sigma_{\max}(\mathbf{G}_{\mathrm{r}}(\mathrm{j}\omega, \mathbf{p}))} \tag{4.4}$$

where $\sigma_{\max}$ is the largest singular value and $\mathrm{j} = \sqrt{-1}$. For approximating (4.4) we choose K_{H} frequencies $\omega_1, \omega_2, \ldots, \omega_{K_{\mathrm{H}}}$ in the frequency range of interest and calculate $\mathbf{G}_{\mathrm{r}}(\cdot, \mathbf{p}) = \{\mathbf{G}_{\mathrm{r}}(\mathrm{j}\omega_1, \mathbf{p}), \ldots, \mathbf{G}_{\mathrm{r}}(\mathrm{j}\omega_{K_{\mathrm{H}}}, \mathbf{p})\}$, where for example $\mathbf{G}_{\mathrm{r}}(\mathrm{j}\omega_k, \mathbf{p}) \in \mathbb{C}^{m \times r}$ is the k-th value. This has for sparse system matrices a computational complexity which is at most $\mathcal{O}(K_{\mathrm{H}} n_{\mathrm{r}}^3)$. The $\mathcal{H}_\infty$-norm of the reference system $\mathbf{G}_{\mathrm{r}}(\mathbf{p})$ is approximately

$$||\mathbf{G}_{\mathrm{r}}(\mathbf{p})||_{\mathcal{H}_\infty} \approx \max_{1 \le k \le K_{\mathrm{H}}} \sigma_{\max}\{\mathbf{G}_{\mathrm{r}}(\mathrm{j}\omega_k, \mathbf{p})\}. \tag{4.5}$$

The calculation of $\mathbf{G}_{\mathrm{r}}(\cdot, \mathbf{p})$ and $||\mathbf{G}_{\mathrm{r}}(\mathbf{p})||_{\mathcal{H}_\infty}$ has to be done only once and the values are stored in the set $\mathcal{S}$ in order to reuse them. Every time the indicator $e^{\mathcal{H}_\infty}_{\mathrm{CV},j}(\mathbf{p})$ is determined for a new reduced system obtained by cross-validation $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})$, we calculate for the same set of frequencies the values $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\cdot, \mathbf{p}) = \{\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathrm{j}\omega_1, \mathbf{p}), \ldots, \tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathrm{j}\omega_{K_{\mathrm{H}}}, \mathbf{p})\}$, where for example the k-th value is $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathrm{j}\omega_k, \mathbf{p}) \in \mathbb{C}^{m \times r}$. The computational complexity involving these dense reduced system matrices is only $\mathcal{O}(K_{\mathrm{H}} q^3)$. Together with the stored values $\mathcal{S}$ we finally obtain for the error indicator

$$e^{\mathcal{H}_\infty}_{\mathrm{CV},j}(\mathbf{p}) \approx \frac{\max\limits_{1 \le k \le K_{\mathrm{H}}} \sigma_{\max}\{\mathbf{G}_{\mathrm{r}}(\mathrm{j}\omega_k, \mathbf{p}) - \tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathrm{j}\omega_k, \mathbf{p})\}}{||\mathbf{G}_{\mathrm{r}}(\mathbf{p})||_{\mathcal{H}_\infty}}. \tag{4.6}$$

To sum up, when we calculate the error indicator $e^{\mathcal{H}_\infty}_{\mathrm{CV},j}(\mathbf{p})$ for system $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})$ at a certain parameter value $\mathbf{p}$, the computational complexity is at most $\mathcal{O}(K_{\mathrm{H}} n_{\mathrm{r}}^3)$ and we update the set $\mathcal{S} = \mathcal{S} \cup \left\{(\mathbf{G}_{\mathrm{r}}(\cdot, \mathbf{p}), ||\mathbf{G}_{\mathrm{r}}(\mathbf{p})||_{\mathcal{H}_\infty})\right\}$. For each additional calculation with a new system $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})$ the complexity is only $\mathcal{O}(K_{\mathrm{H}} q^3)$ as the stored values $\mathcal{S}$ can be reused. For the case the reference system $\mathbf{G}_{\mathrm{r}}(\mathbf{p})$ is low-dimensional, this approximation method for calculating the error (4.4) is not required because we can simply apply methods such as [34, 31] at a reasonable expense.

$\mathcal{H}_2$-norm

For the calculation of the error in $\mathcal{H}_2$-norm between the reference system $\mathbf{G}_{\mathrm{r}}(\mathbf{p})$ with matrices $\mathbf{E}_{\mathrm{r}}(\mathbf{p}), \mathbf{A}_{\mathrm{r}}(\mathbf{p}), \mathbf{B}_{\mathrm{r}}(\mathbf{p}), \mathbf{C}_{\mathrm{r}}(\mathbf{p})$ and the reduced system obtained by cross-validation $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})$ with matrices $\tilde{\mathbf{E}}_{\mathrm{CV},j}(\mathbf{p}), \tilde{\mathbf{A}}_{\mathrm{CV},j}(\mathbf{p}), \tilde{\mathbf{B}}_{\mathrm{CV},j}(\mathbf{p}), \tilde{\mathbf{C}}_{\mathrm{CV},j}(\mathbf{p})$, we firstly consider the following error system

$$\mathbf{G}_{\mathrm{e},j}(\mathbf{p}) : \begin{cases} \overbrace{\begin{bmatrix} \mathbf{E}_{\mathrm{r}}(\mathbf{p}) & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{E}}_{\mathrm{CV},j}(\mathbf{p}) \end{bmatrix}}^{\mathbf{E}_{\mathrm{e},j}(\mathbf{p})} \begin{bmatrix} \dot{\mathbf{x}}_{\mathrm{r}} \\ \dot{\tilde{\mathbf{x}}}_{\mathrm{CV},j} \end{bmatrix} = \overbrace{\begin{bmatrix} \mathbf{A}_{\mathrm{r}}(\mathbf{p}) & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{A}}_{\mathrm{CV},j}(\mathbf{p}) \end{bmatrix}}^{\mathbf{A}_{\mathrm{e},j}(\mathbf{p})} \begin{bmatrix} \mathbf{x}_{\mathrm{r}} \\ \tilde{\mathbf{x}}_{\mathrm{CV},j} \end{bmatrix} + \overbrace{\begin{bmatrix} \mathbf{B}_{\mathrm{r}}(\mathbf{p}) \\ \tilde{\mathbf{B}}_{\mathrm{CV},j}(\mathbf{p}) \end{bmatrix}}^{\mathbf{B}_{\mathrm{e},j}(\mathbf{p})} \mathbf{u} \\ \mathbf{y}_{\mathrm{e},j} = \underbrace{\begin{bmatrix} \mathbf{C}_{\mathrm{r}}(\mathbf{p}) & -\tilde{\mathbf{C}}_{\mathrm{CV},j}(\mathbf{p}) \end{bmatrix}}_{\mathbf{C}_{\mathrm{e},j}(\mathbf{p})} \begin{bmatrix} \mathbf{x}_{\mathrm{r}} \\ \tilde{\mathbf{x}}_{\mathrm{CV},j} \end{bmatrix}. \end{cases} \tag{4.7}$$

Then, the error indicator can be calculated as follows [15]

$$e^{\mathcal{H}_2}_{\mathrm{CV},j}(\mathbf{p}) = \frac{||\mathbf{G}_{\mathrm{r}}(\mathbf{p}) - \tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})||_{\mathcal{H}_2}}{||\mathbf{G}_{\mathrm{r}}(\mathbf{p})||_{\mathcal{H}_2}} = \frac{\sqrt{\mathrm{tr}(\mathbf{C}_{\mathrm{e},j}(\mathbf{p})\mathbf{P}_{\mathrm{e},j}(\mathbf{p})\mathbf{C}_{\mathrm{e},j}(\mathbf{p})^T)}}{\sqrt{\mathrm{tr}(\mathbf{C}_{\mathrm{r}}(\mathbf{p})\mathbf{P}_{\mathrm{r}}(\mathbf{p})\mathbf{C}_{\mathrm{r}}(\mathbf{p})^T)}} \tag{4.8}$$

where $\mathbf{P}_{\mathrm{e},j}(\mathbf{p}) \in \mathbb{S}^{++}_{n+q}$ is the controllability Gramian of the error system $\mathbf{G}_{\mathrm{e},j}(\mathbf{p})$ and $\mathbf{P}_{\mathrm{r}}(\mathbf{p}) \in \mathbb{S}^{++}_{n}$ is the one of the reference system $\mathbf{G}_{\mathrm{r}}(\mathbf{p})$. These can be obtained as the solution to the generalized Lyapunov equation

$$\mathbf{A}_{\mathrm{e},j}(\mathbf{p})\mathbf{P}_{\mathrm{e},j}(\mathbf{p})\mathbf{E}_{\mathrm{e},j}(\mathbf{p})^T + \mathbf{E}_{\mathrm{e},j}(\mathbf{p})\mathbf{P}_{\mathrm{e},j}(\mathbf{p})\mathbf{A}_{\mathrm{e},j}(\mathbf{p})^T + \mathbf{B}_{\mathrm{e},j}(\mathbf{p})\mathbf{B}_{\mathrm{e},j}(\mathbf{p})^T = \mathbf{0} \tag{4.9}$$

and the partitioning

$$\mathbf{P}_{\mathrm{e},j}(\mathbf{p}) = \begin{bmatrix} \mathbf{P}_{\mathrm{r}}(\mathbf{p}) & \mathbf{X}_j(\mathbf{p}) \\ \mathbf{X}_j(\mathbf{p})^T & \tilde{\mathbf{P}}_{\mathrm{CV},j}(\mathbf{p}) \end{bmatrix} \tag{4.10}$$

where $\tilde{\mathbf{P}}_{\mathrm{CV},j}(\mathbf{p}) \in \mathbb{S}^{++}_{q}$ is the controllability Gramian of the system $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})$. We rewrite equation (4.8) using formula (4.10) as follows

$$e^{\mathcal{H}_2}_{\mathrm{CV},j}(\mathbf{p}) = \frac{\sqrt{||\mathbf{G}_{\mathrm{r}}(\mathbf{p})||^2_{\mathcal{H}_2} - 2\mathrm{tr}(\mathbf{C}_{\mathrm{r}}(\mathbf{p})\mathbf{X}_j(\mathbf{p})\tilde{\mathbf{C}}_{\mathrm{CV},j}(\mathbf{p})^T) + ||\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})||^2_{\mathcal{H}_2}}}{||\mathbf{G}_{\mathrm{r}}(\mathbf{p})||_{\mathcal{H}_2}}. \tag{4.11}$$

For calculating $||\mathbf{G}_\mathrm{r}(\mathbf{p})||_{\mathcal{H}_2}$ which is the $\mathcal{H}_2$-norm of the reference system $\mathbf{G}_\mathrm{r}(\mathbf{p})$, we have to solve the generalized Lyapunov equation of size n_r

$$\mathbf{A}_\mathrm{r}(\mathbf{p})\mathbf{P}_\mathrm{r}(\mathbf{p})\mathbf{E}_\mathrm{r}(\mathbf{p})^T + \mathbf{E}_\mathrm{r}(\mathbf{p})\mathbf{P}_\mathrm{r}(\mathbf{p})\mathbf{A}_\mathrm{r}(\mathbf{p})^T + \mathbf{B}_\mathrm{r}(\mathbf{p})\mathbf{B}_\mathrm{r}(\mathbf{p})^T = \mathbf{0}. \tag{4.12}$$

For this, one can use iterative methods such as the alternating directions implicit iteration method (ADI) [135, 109] or the rational Krylov subspace method (RKSM) [54] which compute a low-rank approximation $\mathbf{P}_\mathrm{a}(\mathbf{p})$ of $\mathbf{P}_\mathrm{r}(\mathbf{p})$ and have a complexity which is at most $\mathcal{O}(n_\mathrm{s}n_\mathrm{r}^3)$ where n_s is the number of iterations. Then, we obtain approximately

$$||\mathbf{G}_\mathrm{r}(\mathbf{p})||_{\mathcal{H}_2} \approx \sqrt{\mathrm{tr}(\mathbf{C}_\mathrm{r}(\mathbf{p})\mathbf{P}_\mathrm{a}(\mathbf{p})\mathbf{C}_\mathrm{r}(\mathbf{p})^T)}. \tag{4.13}$$

The calculation of $||\mathbf{G}_\mathrm{r}(\mathbf{p})||_{\mathcal{H}_2}$ has to be done only one time and the value is stored in the set $\mathcal{S}$ in order to reuse it. Every time the indicator $e_{\mathrm{CV},j}^{\mathcal{H}_2}(\mathbf{p})$ must be determined for a new reduced system obtained by cross-validation $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})$, we solve for the second term in the numerator of equation (4.11) the generalized sparse-dense Sylvester equation in order to obtain $\mathbf{X}_j(\mathbf{p}) \in \mathbb{R}^{n\times q}$

$$\mathbf{A}_\mathrm{r}(\mathbf{p})\mathbf{X}_j(\mathbf{p})\tilde{\mathbf{E}}_{\mathrm{CV},j}(\mathbf{p})^T + \mathbf{E}_\mathrm{r}(\mathbf{p})\mathbf{X}_j(\mathbf{p})\tilde{\mathbf{A}}_{\mathrm{CV},j}(\mathbf{p})^T + \mathbf{B}_\mathrm{r}(\mathbf{p})\tilde{\mathbf{B}}_{\mathrm{CV},j}(\mathbf{p})^T = \mathbf{0}. \tag{4.14}$$

The solution can be obtained e.g. by using the sparse-dense solver [24] which has a complexity which is at most $\mathcal{O}(qn_\mathrm{r}^3)$. In addition, we have to solve for the third term of the numerator the low-order generalized Lyapunov equation

$$\tilde{\mathbf{A}}_{\mathrm{CV},j}(\mathbf{p})\tilde{\mathbf{P}}_{\mathrm{CV},j}(\mathbf{p})\tilde{\mathbf{E}}_{\mathrm{CV},j}(\mathbf{p})^T + \tilde{\mathbf{E}}_{\mathrm{CV},j}(\mathbf{p})\tilde{\mathbf{P}}_{\mathrm{CV},j}(\mathbf{p})\tilde{\mathbf{A}}_{\mathrm{CV},j}(\mathbf{p})^T + \tilde{\mathbf{B}}_{\mathrm{CV},j}(\mathbf{p})\tilde{\mathbf{B}}_{\mathrm{CV},j}(\mathbf{p})^T = \mathbf{0}. \tag{4.15}$$

As this dense generalized Lyapunov equation is low-order, we can use direct methods [44, 69, 134] which have the complexity $\mathcal{O}(q^3)$. Then, the $\mathcal{H}_2$-norm of system $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})$ is

$$||\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})||_{\mathcal{H}_2} = \sqrt{\mathrm{tr}(\tilde{\mathbf{C}}_{\mathrm{CV},j}(\mathbf{p})\tilde{\mathbf{P}}_{\mathrm{CV},j}(\mathbf{p})\tilde{\mathbf{C}}_{\mathrm{CV},j}(\mathbf{p})^T)}. \tag{4.16}$$

Hence, the error indicator $e_{\mathrm{CV},j}^{\mathcal{H}_2}(\mathbf{p})$ is obtained when $\mathbf{X}_j(\mathbf{p})$, $||\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})||_{\mathcal{H}_2}$ together with $||\mathbf{G}_\mathrm{r}(\mathbf{p})||_{\mathcal{H}_2}$ from set $\mathcal{S}$ is inserted into equation (4.11). In analogy, this could be done with the observability Gramian.

To sum up, when we calculate the error indicator $e_{\mathrm{CV},j}^{\mathcal{H}_2}(\mathbf{p})$ for system $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})$ at a certain parameter value $\mathbf{p}$, the computational complexity is at most $\mathcal{O}(n_\mathrm{s}n_\mathrm{r}^3)$ and we update the set $\mathcal{S} = \mathcal{S} \cup \left\{||\mathbf{G}_\mathrm{r}(\mathbf{p})||_{\mathcal{H}_2}\right\}$. For each additional calculation with a new system $\tilde{\mathbf{G}}_{\mathrm{CV},j}(\mathbf{p})$ the complexity is at most $\mathcal{O}(qn_\mathrm{r}^3)$ with $q < n_\mathrm{s}$ as the stored values $\mathcal{S}$

can be reused. For the case the reference system $\mathbf{G}_\mathrm{r}(\mathbf{p})$ is low-dimensional, this proposed method for calculating (4.11) is not required because we can simply apply methods such as [44, 69, 134] at a reasonable expense.

4.2.4 Motivating Example

In this section a motivating example for promoting the understanding of the error indicator is given. It is an academic spring-mass-damper system [112] whose schematic is shown in Figure 4.5.

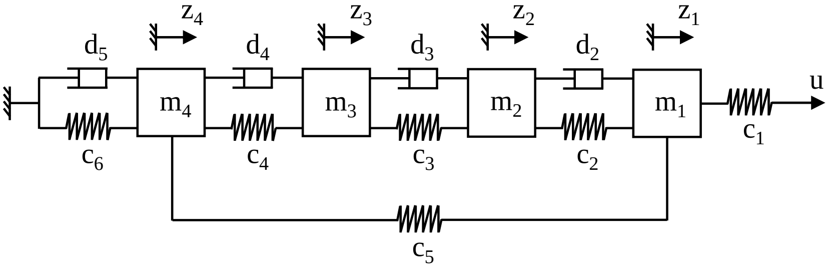


Figure 4.5: A schematic of the spring-mass-damper system.

Table 4.1: Mechanical properties of the spring-mass-damper system.

Springs [N/m]		Masses [kg]		Dampers [Ns/m]	
c_1	27	m_1	1	d_2	0.1
c_2	9	m_2	5	d_3	0.4
c_3	3	m_3	25	d_4	1.6
c_4	1	m_4	125	d_5	α
c_5	$1+2\alpha$				
c_6	$2+2\alpha$				

The system consists of 6 springs c_i, 4 masses m_i and 4 dampers d_i whose values are shown in Table 4.1. The stiffness and damping c_5 and d_5 and the stiffness c_6 vary within a certain given interval described by the parameter $\alpha \in [0,\ 1]$. The input is the displacement $u(t)$ acting on the spring c_1 and the output is the position of the mass m_4 which is $y(t) = z_4(t)$. Setting the state vector $\mathbf{x} = [z_1, \dot{z}_1, z_2, \dot{z}_2, z_3, \dot{z}_3, z_4, \dot{z}_4]^T$, the equations of translational momentum lead to a parameter-dependent SISO system of order $n = 8$:

$$G(\alpha) : \begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}(\alpha)\mathbf{x}(t) + \mathbf{b}u(t) \\ \quad y(t) = \mathbf{c}^T\mathbf{x}(t). \end{cases} \tag{4.17}$$

We have a set of five grid points $\mathcal{P} = \{0, 0.25, 0.5, 0.75, 1\}$ and reduce the local systems at the nodes individually to order $q = 4$ using IRKA. The ROBs are adjusted using the PS approach (3.13) and (3.25) to the reference ROBs which are calculated with the non-weighted SVD (3.5) and (3.22). Figure 4.6 shows the error e_{T} in $\mathcal{H}_2$-norm using linear and cubic spline interpolation on the manifold of nonsingular matrices for the square system matrices and the error due to projection e_{P} using IRKA. In addition, leave-one-out error indicators $e_{\mathrm{CV}}(\alpha)$ are shown for the interior points $\alpha \in \{0.25, 0.5, 0.75\}$.

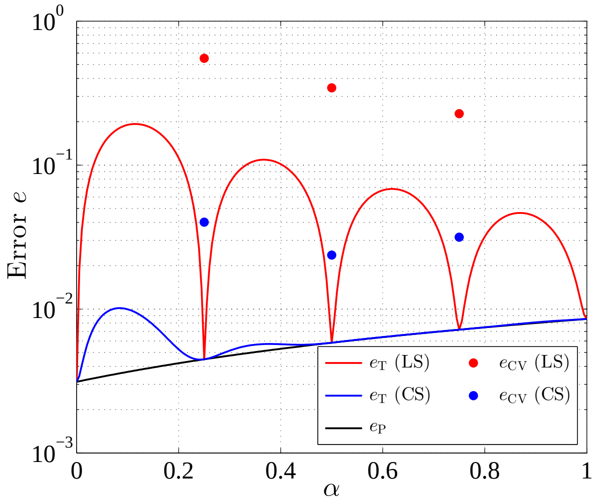


Figure 4.6: Plots of the error with linear (LS) and cubic spline (CS) interpolation and error indicators allocated to the interior grid points.

One can verify there is a correlation between the error indicator at a certain point and the true error in its surroundings: If the error indicator allocated to a grid point is large or small, then the true error in its surroundings is large or small. At the grid points the true error itself has only a part due to projection, see Section 3.8. This and the following observations motivate us to use the cross-validation-based error indicator for three applications of pMOR by matrix interpolation:

1. Calculating, e.g., the mean of the error indicators in Figure 4.6 hints that cubic spline interpolation has a better performance which is confirmed by the true error. This will be generalized in Section 4.3 where an approach for the selection of the interpolation method with minimum error in domain $\mathcal{D}$ is proposed. For this, an aggregated error measure such as the mean is calculated for the set of error indicators $\mathcal{E}_{\mathrm{CV}}$ of different candidate methods. Hence, the method delivering the best accuracy is the one with minimum error measure.

2. In Figure 4.6 one can see the error indicators are largest for linear and cubic spline interpolation at grid point $\alpha = 0.25$. The true error shows that the interpolation

error is largest around the same point. Hence, it makes sense to insert a new grid point in its surroundings. Keeping this observation in mind, we propose an adaptive refinement method in Section 4.4 where a surrogate model is constructed using the set of error indicators $\mathcal{E}_{\mathrm{CV}}$. This surrogate model can then be exploited to identify points for unobserved regions with a high error indicator.

3. One can also see in Figure 4.6 the error indicator overestimates the true error because of a smaller training data. Hence, a mapping function from the error indicator to the true error is determined in Section 4.5. This is used to calculate an upper bound for the accuracy of the reduced model. Then, the refinement procedure is stopped as soon as a given error tolerance is reached.

4.3 Selection of the Interpolation Method

4.3.1 Overview

Consider a given set of grid points $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ where $N = |\mathcal{P}|$ is the number of grid points and a corresponding set of reduced systems $\tilde{\mathcal{G}} = \{\tilde{\mathbf{G}}_1, \ldots, \tilde{\mathbf{G}}_N\}$. The interpolation procedure between the local systems in set $\tilde{\mathcal{G}}$, such as the method described in Section 3, allows the user to choose between K_{MI} interpolation methods $\lambda_1^{\mathrm{MI}}, \ldots, \lambda_{K_{\mathrm{MI}}}^{\mathrm{MI}}$ and K_{MA} candidate manifolds $\lambda_1^{\mathrm{MA}}, \ldots, \lambda_{K_{\mathrm{MA}}}^{\mathrm{MA}}$. This results in a set of candidate methods $\mathcal{K} = \{\lambda_1, \ldots, \lambda_{K_{\mathrm{M}}}\}$ where $K_{\mathrm{M}} = |\mathcal{K}| = K_{\mathrm{MI}} K_{\mathrm{MA}}$ is the number of combinations between interpolation methods and manifolds. We aim to find the optimal method λ^* which is the one minimizing an aggregated quality criterion $J(\lambda)$ on domain $\mathcal{D}$. Hence, the optimization problem of interest is

$$\lambda^* = \underset{\lambda \in \mathcal{K}}{\arg\min}\ J(\lambda). \tag{4.18}$$

In order to understand what the quality function looks like, recall Section 3.8 where it was shown that the overall error consists of a part due to projection and one due to interpolation. As the first one is the same for all candidate methods, only the latter can be influenced by the interpolation method. That is why the quality criterion is based on the error due to interpolation $e_{\mathrm{I}}(\mathbf{p}, \lambda)$ for $\lambda \in \mathcal{K}$. Then, one common quality criterion is the mean [148]

$$J_\mu(\lambda) = \frac{1}{|\mathcal{D}|} \int_{\mathcal{D}} e_{\mathrm{I}}(\mathbf{p}, \lambda) \mathrm{d}\mathcal{D}. \tag{4.19}$$

Another common aggregated error measure is the root mean square

$$J_{\mathrm{RMSE}}(\lambda) = \sqrt{\frac{1}{|\mathcal{D}|}\int_{\mathcal{D}} e_{\mathrm{I}}(\mathbf{p},\lambda)^2 \mathrm{d}\mathcal{D}}. \tag{4.20}$$

An estimation of the aggregated error measures could be obtained when the interpolation error $e_{\mathrm{I}}(\mathbf{p},\lambda)$ is calculated for a number N_{test} of test points such as in [62]

$$J_{\mu}(\lambda) \approx \frac{1}{N_{\mathrm{test}}}\sum_{i=1}^{N_{\mathrm{test}}} e_{\mathrm{I}}(\mathbf{p}_i,\lambda), \tag{4.21}$$

$$J_{\mathrm{RMSE}}(\lambda) \approx \sqrt{\frac{1}{N_{\mathrm{test}}}\sum_{i=1}^{N_{\mathrm{test}}} e_{\mathrm{I}}(\mathbf{p}_i,\lambda)^2}. \tag{4.22}$$

However, this is computationally expensive since complex calculations such as projection-based MOR from Section 2.2 have to be performed for the N_{test} additional test points. Hence, we use the cross-validation-based error indicator from Section 4.2 which is based only on available data $\mathcal{P}, \tilde{\mathcal{G}}$ and we calculate an efficient estimate $J_{\mu}^{\mathrm{CV}}(\lambda)$ for (4.19) or $J_{\mathrm{RMSE}}^{\mathrm{CV}}(\lambda)$ for (4.20). For every candidate interpolation method $\lambda \in \mathcal{K}$ we calculate such an estimate and select the one with minimum error measure. The procedure is demonstrated in Figure 4.7. It will be shown in numerical examples in Section 4.7 that this procedure reliably identifies the interpolation method with minimum error measure at a very low computational expense.

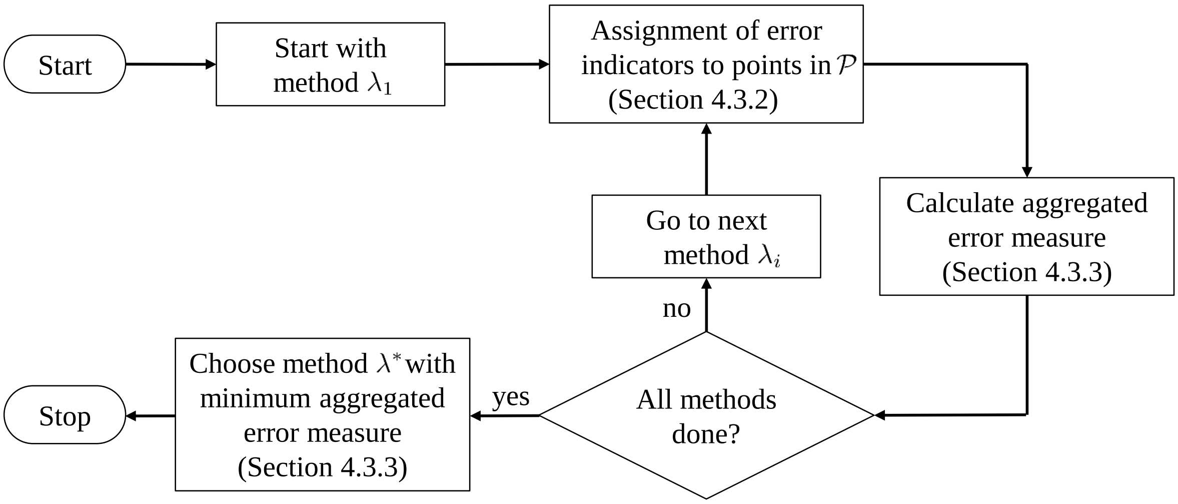

Start
Start with method λ_1
Assignment of error indicators to points in $\mathcal{P}$ (Section 4.3.2)
Calculate aggregated error measure (Section 4.3.3)
Go to next method λ_i
no
All methods done?
yes
Choose method λ^* with minimum aggregated error measure (Section 4.3.3)
Stop

Figure 4.7: Structure of the proposed selection method.

4.3.2 Assignment of the Error Indicator

We start the method by assigning the cross-validation-based error indicator $e_{\mathrm{CV}}(\mathbf{p}, \lambda)$ to all grid points $\mathbf{p} \in \mathcal{P}$ for every interpolation method $\lambda \in \mathcal{K}$ using Algorithm 3. As we want to measure the error due to interpolation $e_{\mathrm{I}}(\mathbf{p}, \lambda)$, we use the reduced systems from set $\tilde{\mathcal{G}}$ as reference systems for calculating the error norms in Section 4.2.3. Then, the error norms comprise only low-order models of dimension q. Hence, the calculation of the error indicators is cheap with complexity $\mathcal{O}(q^3)$. The procedure results in K_{M} sets of error indicators $\mathcal{E}_{\mathrm{CV}}(\lambda) = \{e_{\mathrm{CV}}(\mathbf{p}_1, \lambda), \dots, e_{\mathrm{CV}}(\mathbf{p}_N, \lambda)\}$ with $\lambda \in \mathcal{K}$.

4.3.3 Minimization of an Aggregated Error Measure

In order to compare the accuracy of the candidate methods, the error indicators in the sets $\mathcal{E}_{\mathrm{CV}}(\lambda)$ are used to calculate an estimate $J_{\mu}^{\mathrm{CV}}(\lambda)$ for (4.19) or $J_{\mathrm{RMSE}}^{\mathrm{CV}}(\lambda)$ for (4.20) for every method $\lambda \in \mathcal{K}$:

$$J_{\mu}^{\mathrm{CV}}(\lambda) = \frac{1}{N} \sum_{i=1}^{N} e_{\mathrm{CV}}(\mathbf{p}_i, \lambda), \tag{4.23}$$

$$J_{\mathrm{RMSE}}^{\mathrm{CV}}(\lambda) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} e_{\mathrm{CV}}(\mathbf{p}_i, \lambda)^2}. \tag{4.24}$$

The estimate (4.23) or (4.24) is inserted into the introductory optimization problem (4.18) and we minimize the following problem with regard to all methods $\lambda \in \mathcal{K}$:

$$\lambda^* = \arg \min_{\lambda \in \mathcal{K}} J^{\mathrm{CV}}(\lambda). \tag{4.25}$$

Solving the optimization problem (4.25) is trivial because the set $\mathcal{K}$ is finite and usually has only a few components. The entire procedure is summarized in Algorithm 4. As the required systems are low-dimensional, this approach offers an efficient way with complexity $\mathcal{O}(q^3)$ for selecting the most accurate candidate method.

Algorithm 4 Selection of the interpolation method and manifold

```
Input: Set of reduced systems $\tilde{\mathcal{G}}$, of grid points $\mathcal{P}$ and of candidate methods $\mathcal{K}$
Output: Interpolation method $\lambda^*$ with minimum error measure
for $j = 1, \dots, |\mathcal{K}|$ do
    Compute set $\mathcal{E}_{\mathrm{CV}}(\lambda_j)$ according to Algorithm 3 using sets $\mathcal{P}, \tilde{\mathcal{G}}$
    Calculate the aggregated error measure $J^{\mathrm{CV}}(\lambda_j)$ for method $\lambda_j$ using $\mathcal{E}_{\mathrm{CV}}(\lambda_j)$
end for
Minimize the aggregated error measure: $\lambda^* = \arg\min_{\lambda \in \mathcal{K}} J^{\mathrm{CV}}(\lambda)$
```

4.4 Model Refinement

4.4.1 Overview

Consider a given set of grid points $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ where $N = |\mathcal{P}|$ is the number of grid points. In addition, we have a corresponding set of high-dimensional systems $\mathcal{G} = \{\mathbf{G}_1, \ldots, \mathbf{G}_N\}$ and of reduced systems $\tilde{\mathcal{G}} = \{\tilde{\mathbf{G}}_1, \ldots, \tilde{\mathbf{G}}_N\}$ and the interpolation method which is obtained using Algorithm 4. In order to improve the accuracy of the resulting parameter-dependent reduced system $\tilde{\mathbf{G}}(\mathbf{p})$, we aim to insert a new grid point $\mathbf{p}^*$ in the domain $\mathcal{D}$ where the overall error $e_{\mathrm{T}}(\mathbf{p})$ is largest:

$$\mathbf{p}^* = \underset{\mathbf{p}\in\mathcal{D}}{\arg\max}\; e_{\mathrm{T}}(\mathbf{p}). \tag{4.26}$$

Common methods [142, 38, 30] compute the overall error $e_{\mathrm{T}}(\mathbf{p})$ for a large number of test points and take the test point with the largest error as new grid point. However, this procedure is computationally expensive as it requires evaluating the large-scale model very often. We replace the error $e_{\mathrm{T}}(\mathbf{p})$ by the cross-validation-based error indicator from Section 4.2. As this error indicator is only available at the grid points, a surrogate model $\tilde{e}_{\mathrm{CV}} : \mathcal{D} \to \mathbb{R}$ is constructed for the indicator which can cheaply be optimized on domain $\mathcal{D}$. Hence, we propose a refinement method which starts with an initial grid in Section 4.4.2 and then, iteratively selects new grid points in Sections 4.4.3–4.4.4 by maximizing the surrogate model $\tilde{e}_{\mathrm{CV}}(\mathbf{p})$. The adaptive refinement is shown in Figure 4.8. It stops as soon as the maximum predicted error falls below a given tolerance e_{tol}, see Section 4.5. We show the procedure is efficient as it requires only one high-order solve per new grid point instead of a large number like for current methods.

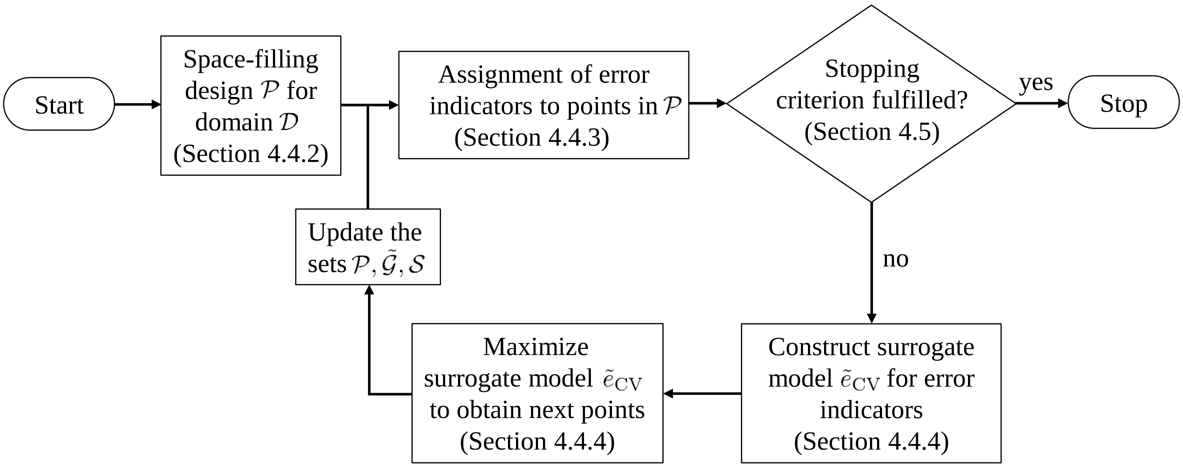

Start
Space-filling design $\mathcal{P}$ for domain $\mathcal{D}$ (Section 4.4.2)
Assignment of error indicators to points in $\mathcal{P}$ (Section 4.4.3)
Stopping criterion fulfilled? (Section 4.5)
yes
Stop
no
Update the sets $\mathcal{P}, \tilde{\mathcal{G}}, \mathcal{S}$
Construct surrogate model $\tilde{e}_{\mathrm{CV}}$ for error indicators (Section 4.4.4)
Maximize surrogate model $\tilde{e}_{\mathrm{CV}}$ to obtain next points (Section 4.4.4)

Figure 4.8: Structure of the proposed refinement method.

4.4.2 Selection of Initial Grid Points

We start with an initial set of grid points $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\} \subset \mathcal{D}$ with domain $\mathcal{D} \subseteq \mathbb{R}^d$ and a corresponding set of original systems $\mathcal{G} = \{\mathbf{G}_1, \dots, \mathbf{G}_N\}$. The grid type, i.e. if the parameter vectors are located on an irregular grid (cf. Appendix A.3) or on a regular grid (cf. Appendix A.2), is determined by the kind of application and its constraints on the domain $\mathcal{D}$. As we want the error indicators, which are allocated to the grid points, to have a reasonable approximation of the entire domain $\mathcal{D}$, we choose N points which have approximately the same distance from each other and capture all regions of the domain $\mathcal{D}$. In the case of a regular grid, we start, e.g., with a full grid. Then, the points are equidistantly distributed in the domain with the same number of points in every direction (cf. Figure A.2). Since the refinement procedure requires at least four grid points per dimension for a regular grid (cf. Section 4.4.4), we start with $N = 4^d$ grid points. For irregular grids, there are various methods for space-filling designs (cf. Figure A.4), such as Latin hypercube sampling [116], Halton or Sobol quasi-random sequence [92, 149]. A rule of thumb for the number of points is $N = 10d$ [98, 65].

After we have decided on the starting grid, we reduce the local systems in set $\mathcal{G}$ according to Sections 3.3–3.4 and we obtain the set of reduced systems $\tilde{\mathcal{G}} = \{\tilde{\mathbf{G}}_1, \dots, \tilde{\mathbf{G}}_N\}$. In addition, we perform for the systems in set $\mathcal{G}$ certain high-order operations depending on the error norm of interest which are described in Section 4.2.3 and we store the results in the set $\mathcal{S}$: If we want to measure the error indicators in the $\mathcal{L}_p$-norm of the output, we obtain the set of tuples $\mathcal{S} = \{(\mathbf{y}(\cdot, \mathbf{p}_1), ||\mathbf{y}(t, \mathbf{p}_1)||_{\mathcal{L}_p}), \dots, (\mathbf{y}(\cdot, \mathbf{p}_N), ||\mathbf{y}(t, \mathbf{p}_N)||_{\mathcal{L}_p})\}$, for the $\mathcal{H}_\infty$-norm we obtain $\mathcal{S} = \{(\mathbf{G}(\cdot, \mathbf{p}_1), ||\mathbf{G}_1||_{\mathcal{H}_\infty}), \dots, (\mathbf{G}(\cdot, \mathbf{p}_N), ||\mathbf{G}_N||_{\mathcal{H}_\infty})\}$ and for the $\mathcal{H}_2$-norm we obtain $\mathcal{S} = \{||\mathbf{G}_1||_{\mathcal{H}_2}, \dots, ||\mathbf{G}_N||_{\mathcal{H}_2}\}$.

4.4.3 Assignment of the Error Indicator

In every iteration such as described in Figure 4.8, error indicators are assigned to the grid points $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$ using Algorithm 3. As we want to measure the overall error $e_{\mathrm{T}}(\mathbf{p})$, we use the high-order systems from the set $\mathcal{G}$ as reference systems for calculating the error norms in Section 4.2.3. Unfortunately, high-order solves appear in the formulas for calculating the error indicators. In order to reduce the computational effort, ways were presented to split the resulting error norms in one part using low-order solves and another part using high-order solves. As we can reuse for the latter the data which was computed one time and stored in the set $\mathcal{S}$, only low-order solves have to be performed for subsequent iterations. Hence, the assignment procedure is cheap with complexity $\mathcal{O}(q^3)$ and we obtain a set of error indicators $\mathcal{E}_{\mathrm{CV}} = \{e_{\mathrm{CV}}(\mathbf{p}_1), \dots, e_{\mathrm{CV}}(\mathbf{p}_N)\}$.

4.4.4 Construction and Optimization of a Surrogate Model

We extract the data $\{(\mathbf{p}_1, e_{\mathrm{CV}}(\mathbf{p}_1)), \ldots, (\mathbf{p}_N, e_{\mathrm{CV}}(\mathbf{p}_N))\}$ from sets $\mathcal{P}, \mathcal{E}_{\mathrm{CV}}$. As we want to have the error indicator for the entire parameter domain $\mathcal{D}$, we construct a surrogate model $\tilde{e}_{\mathrm{CV}} : \mathcal{D} \to \mathbb{R}$ using this data. Then, the surrogate model is inserted into the introductory optimization problem (4.26) and we maximize the following problem:

$$\mathbf{p}^* = \underset{\mathbf{p} \in \mathcal{D}}{\arg\max}\ \tilde{e}_{\mathrm{CV}}(\mathbf{p}). \tag{4.27}$$

Optimization problem (4.27) can efficiently be solved because the evaluation of the surrogate model is computationally cheap. Instead of using just the global maximum, it is also possible to insert a set of new grid points $\mathcal{P}^* = \{\mathbf{p}_1^*, \ldots, \mathbf{p}_{N_{\mathrm{new}}}^*\}$ at local maxima of the surrogate model. The entire procedure is summarized in Algorithm 5. At the end of each iteration, we calculate the high-order values $\mathcal{S}^*$ according to Section 4.2.3 and the reduced systems $\tilde{\mathcal{G}}^*$ at the new grid points $\mathcal{P}^*$. Then, we update the sets $\mathcal{P} = \mathcal{P} \cup \mathcal{P}^*$, $\mathcal{S} = \mathcal{S} \cup \mathcal{S}^*$, $\tilde{\mathcal{G}} = \tilde{\mathcal{G}} \cup \tilde{\mathcal{G}}^*$.

Algorithm 5 Calculation of the next best grid points

```
Input: Set of reduced systems $\tilde{\mathcal{G}}$, of points $\mathcal{P}$, of stored values $\mathcal{S}$
Output: Next best grid points $\mathbf{p}_1^*, \ldots, \mathbf{p}_{N_{\mathrm{new}}}^*$
1: Compute set $\mathcal{E}_{\mathrm{CV}}$ according to Algorithm 3 using sets $\mathcal{P}, \tilde{\mathcal{G}}, \mathcal{S}$
2: Build surrogate model $\tilde{e}_{\mathrm{CV}}(\mathbf{p})$ using sets $\mathcal{P}, \mathcal{E}_{\mathrm{CV}}$
3: Solve the optimization problem
       $\{\mathbf{p}_1^*, \ldots, \mathbf{p}_{N_{\mathrm{new}}}^*\} = \arg\max_{\mathbf{p} \in \mathcal{D}} \tilde{e}_{\mathrm{CV}}(\mathbf{p}).$   (4.28)
```

What does the surrogate model $\tilde{e}_{\mathrm{CV}}(\mathbf{p})$ look like and how can it be optimized? This depends on the grid type and is explained in the following for two types: the irregular and the regular grid, both of which are described in more detail in Appendix A.

Surrogate Models for an Irregular Grid

There is a large variety of surrogate models and ways to exploit them for irregular grids [141, 97, 89, 66, 64]. We apply two promising, interpolation-based surrogate models:

Radial basis function model: We construct a surrogate $\tilde{e}_{\mathrm{CV}}(\mathbf{p})$ for the error indicator on the domain $\mathcal{D}$ by interpolating the data $\{(\mathbf{p}_1, e_{\mathrm{CV}}(\mathbf{p}_1)), \ldots, (\mathbf{p}_N, e_{\mathrm{CV}}(\mathbf{p}_N))\}$ using RBF according to Appendix A.3.1. In order to find the optimal basis function with respect to a quality criterion such as the root mean square error, one can use e.g.

cross-validation. The next grid point is then obtained by finding the maximum value of the surrogate $\tilde{e}_{\mathrm{CV}}(\mathbf{p})$. In order to avoid that new data points are inserted too closely to already sampled data, we introduce constraints for the parameter values around every grid point $||\mathbf{p}-\mathbf{p}_i||_2 \geq \epsilon_i$ for $i \in \{1,\ldots,N\}$. Let $d_{\min}(\mathbf{p}_i)$ be the distance from point $\mathbf{p}_i$ to its closest neighbor. The following strategies are proposed for the constraint ϵ_i:

- RB1: The common value $\epsilon_i = \epsilon = 0.5 \min\limits_{i\in\{1,\ldots,N\}} d_{\min}(\mathbf{p}_i)$ minimizes the minimum distance between any two sample points and hence, finely distributes points in the domain with high error indicator but might insert points too closely together.

- RB2: The common value $\epsilon_i = \epsilon = 0.5 \max\limits_{i\in\{1,\ldots,N\}} d_{\min}(\mathbf{p}_i)$ maximizes the minimum distance between any two sample points and hence, broadly distributes the grid points in the domain [96]. But it prevents from sampling very accurately in a certain region if necessary.

- RB3: The individual value $\epsilon_i = 0.5\ d_{\min}(\mathbf{p}_i)$ balances the two effects by broadly distributing where there are few grid points and allows fine sampling in areas where there are many grid points.

- RB4: The fixed value $\epsilon_i = c$ is useful for a multidimensional parameter space where the calculation of the distance between the points is expensive.

We can use one strategy for all iterations or switch in between. To sum up, we solve

$$\begin{aligned}\{\mathbf{p}_1^*,\ldots,\mathbf{p}_{N_{\mathrm{new}}}^*\} = \arg\max_{\mathbf{p}\in\mathcal{D}} \ & \tilde{e}_{\mathrm{CV}}(\mathbf{p}) \\ \text{s.t.}\ \ & ||\mathbf{p}-\mathbf{p}_i||_2 \geq \epsilon_i \quad \forall\ i \in \{1,\ldots,N\}.\end{aligned} \tag{4.29}$$

We use a multistart optimization method which runs a local gradient-based solver for problem (4.29) from various starting points in order to find the global maximum and multiple local maxima which give the new points $\mathcal{P}^* = \{\mathbf{p}_1^*,\ldots,\mathbf{p}_{N_{\mathrm{new}}}^*\}$ [52]. For this, take a large number of starting points and leave out all starting points which lie in the constraints in order to reduce the number of optimization procedures. With growing dimension d of the parameter space, it is useful to choose a growing number N_{new} of new grid points per iteration to decrease the computation time of the refinement process.

An example for the procedure with a 1-dimensional parameter space $p \in [0,\ 1]$ with $N = 7$ grid points is given in Figure 4.9. One can see the sampled data points $\{(p_1, e_{\mathrm{CV}}(p_1)),\ldots,(p_7, e_{\mathrm{CV}}(p_7))\}$, the RBF surrogate $\tilde{e}_{\mathrm{CV}}(p)$ and the constraints for strategy RB3. The next best grid point p^* is obtained at the maximum value of the surrogate which satisfies the constraints.

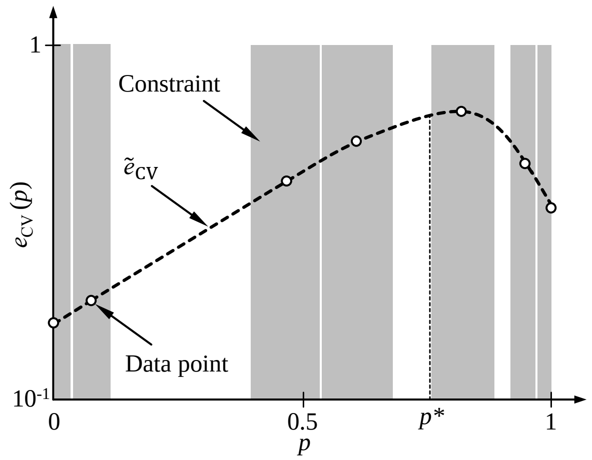


Figure 4.9: A graphic interpretation of the RBF surrogate model with constraints for strategy RB3 (grey surfaces).

Kriging model: We have the data $\{(\mathbf{p}_1, e_{\mathrm{CV}}(\mathbf{p}_1)), \ldots, (\mathbf{p}_N, e_{\mathrm{CV}}(\mathbf{p}_N))\}$, but the error indicator is unknown for the remaining points of the domain $\mathcal{D}$. The Kriging predictor models this uncertainty in such a way the indicator is assumed to be the realization of a normally distributed random variable $E_{\mathrm{CV}}(\mathbf{p}) \sim \mathcal{N}\left(\tilde{e}_{\mathrm{CV}}(\mathbf{p}), \tilde{\sigma}_{\mathrm{CV}}(\mathbf{p})^2\right)$ where $\tilde{e}_{\mathrm{CV}}(\mathbf{p})$ is the mean and $\tilde{\sigma}_{\mathrm{CV}}(\mathbf{p})^2$ the variance, see Appendix A.3.2. The hyperparameters of the Kriging predictor can be determined by using the maximum likelihood estimation. In Figure 4.10 the Kriging model is shown for a 1-dimensional space $p \in [0,\ 1]$. One can see the data points and the Kriging mean $\tilde{e}_{\mathrm{CV}}(\mathbf{p})$ as a surrogate for the error indicator. In addition, a confidence interval $\tilde{e}_{\mathrm{CV}}(\mathbf{p}) \pm 2\tilde{\sigma}_{\mathrm{CV}}(\mathbf{p})$ with about 95% probability is shown. The variance, which is a measure of uncertainty, is large where there are few sample points and it is zero at the grid points because there is no uncertainty.

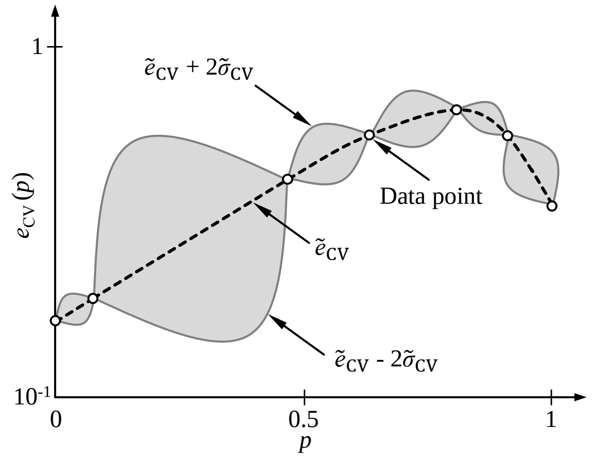


Figure 4.10: A graphic interpretation of the Kriging surrogate model and the confidence interval.

A review of optimization methods based on Kriging surrogates is given in [97, 66]. We use the method Efficient Global Optimization from [98] because it gives the amount of improvement which we can expect to achieve if we insert a grid point at a desired parameter value. Firstly, let us introduce the improvement function which is the random variable $I(\mathbf{p}) = \max(E_{\text{CV}}(\mathbf{p}) - e_{\text{CV}}^{\max}, 0)$. Hence, an improvement is obtained for parameter $\mathbf{p}$ when the realization of $E_{\text{CV}}(\mathbf{p})$ is higher than the so far highest value $e_{\text{CV}}^{\max} = \max(\mathcal{E}_{\text{CV}})$. This is the case for some probability given by the distribution $E_{\text{CV}}(\mathbf{p}) \sim \mathcal{N}\big(\tilde{e}_{\text{CV}}(\mathbf{p}), \tilde{\sigma}_{\text{CV}}(\mathbf{p})^2\big)$. We give an example for demonstrating the improvement function in Figure 4.11 where one can see the Kriging surrogate model from Figure 4.10 and the maximum error indicator $e_{\text{CV}}^{\max}$ given by the already sampled data. In addition, the probability distribution at two parameter values p_1, p_2 centered around the means $\tilde{e}_{\text{CV}}(p_1) < \tilde{e}_{\text{CV}}(p_2)$ is shown with different variances $\tilde{\sigma}_{\text{CV}}(p_1)^2 > \tilde{\sigma}_{\text{CV}}(p_2)^2$. An improvement occurs when the realization of the random distribution of the error indicator is higher than $e_{\text{CV}}^{\max}$. The probability for that improvement is given by the distribution, see an example in Figure 4.11. The probability for an improvement can be high for two reasons: the first one is shown for p_2 where the mean of the distribution is high and the second one is shown for p_1 where there is a high variance and hence, a broad distribution. Below $e_{\text{CV}}^{\max}$ there is no improvement.

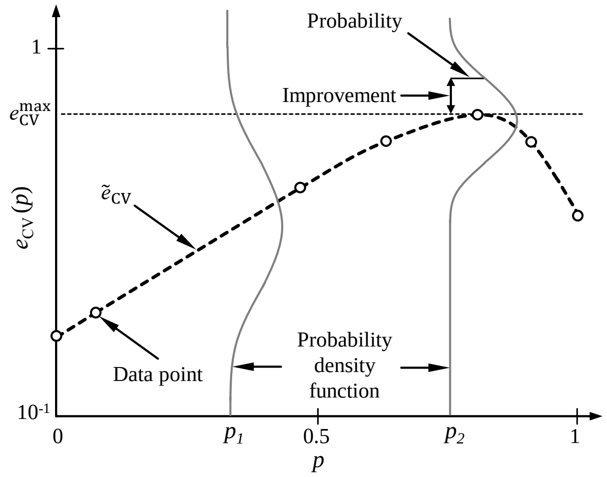


Figure 4.11: A graphic interpretation of the improvement function.

We can now calculate the expected improvement $EI : \mathcal{D} \to \mathbb{R}_0^+$ such as described in [118, 145, 98]. It is obtained by taking the expected value of the improvement function

$$\begin{aligned} EI(\mathbf{p}) = \mathbb{E}[I(\mathbf{p})] &= \mathbb{E}[\max(E_{\text{CV}}(\mathbf{p}) - e_{\text{CV}}^{\max}, 0)] \\ &= \int_{e_{\text{CV}}^{\max}}^{\infty} (E_{\text{CV}}(\mathbf{p}) - e_{\text{CV}}^{\max}) \phi_{\tilde{e}_{\text{CV}}(\mathbf{p})}^{\tilde{\sigma}_{\text{CV}}(\mathbf{p})}(E_{\text{CV}}(\mathbf{p})) \mathrm{d}E_{\text{CV}}(\mathbf{p}) \end{aligned} \tag{4.30}$$

where $\phi_{\tilde{e}_{\mathrm{CV}}(\mathbf{p})}^{\tilde{\sigma}_{\mathrm{CV}}(\mathbf{p})}(E_{\mathrm{CV}}(\mathbf{p}))$ is the probability density function corresponding to the distribution $E_{\mathrm{CV}}(\mathbf{p}) \sim \mathcal{N}\Big(\tilde{e}_{\mathrm{CV}}(\mathbf{p}), \tilde{\sigma}_{\mathrm{CV}}(\mathbf{p})^2\Big)$. The first term in the integral represents the improvement over the current highest error indicator and the second term represents the probability of that improvement. Performing some calculations such as described in [146], one obtains the formula for the expected improvement

$$EI(\mathbf{p}) = \begin{cases} (\tilde{e}_{\mathrm{CV}}(\mathbf{p}) - e_{\mathrm{CV}}^{\max})\,\Phi\left(\frac{\tilde{e}_{\mathrm{CV}}(\mathbf{p}) - e_{\mathrm{CV}}^{\max}}{\tilde{\sigma}_{\mathrm{CV}}(\mathbf{p})}\right) + \tilde{\sigma}_{\mathrm{CV}}(\mathbf{p})\phi\left(\frac{\tilde{e}_{\mathrm{CV}}(\mathbf{p}) - e_{\mathrm{CV}}^{\max}}{\tilde{\sigma}_{\mathrm{CV}}(\mathbf{p})}\right) & \text{if } \tilde{\sigma}_{\mathrm{CV}}(\mathbf{p}) > 0 \\ 0 & \text{if } \tilde{\sigma}_{\mathrm{CV}}(\mathbf{p}) = 0 \end{cases} \tag{4.31}$$

where Φ and ϕ are the cumulative distribution function and the probability density function of the standard normal distribution, respectively. The expected improvement can be large because of two reasons: firstly because the mean $\tilde{e}_{\mathrm{CV}}(\mathbf{p})$ is large and hence, because the error indicator is large, which is called exploitation, or secondly because $\tilde{\sigma}_{\mathrm{CV}}(\mathbf{p})$ is large because of a large uncertainty of the error indicator, which is called exploration. A sketch of the expected improvement with the Kriging surrogate model from Figure 4.10 is given in Figure 4.12 on a logarithmic scale. The expected improvement is very small in most parts of the domain. It is large where the mean is large, e.g. around p_2, or where the uncertainty due to a lack of sample points is large, e.g. around p_1. These two tendencies are balanced by the expected improvement function which finds a compromise between exploration and exploitation. In this example p^* with largest value of the expected improvement is the next grid point. At the already sampled grid points $\mathcal{P}$, where we have $\tilde{\sigma}_{\mathrm{CV}}(\mathbf{p}) = 0$, we can expect no improvement.

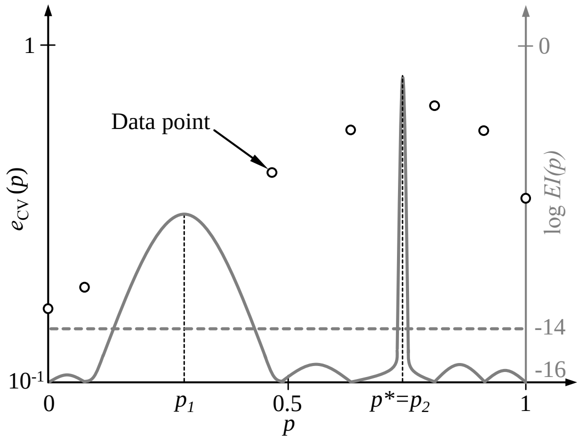


Figure 4.12: A graphic interpretation of the refinement procedure for an irregular grid exploiting the expected improvement.

As it can be seen in Figure 4.12, the expected improvement $EI(\mathbf{p})$ is a function with few peaks and in between it is very small, so it has a wide range in the order of magnitude. Therefore, we restrict it to $EI(\mathbf{p}) \geq 10^{-14}$ and replace it by its logarithm which can be searched by an optimizer more efficiently [65]. To sum up, we want to find the points with maximum value of the expected improvement function by solving the following optimization problem:

$$\{\mathbf{p}_1^*, \dots, \mathbf{p}_{N_{\text{new}}}^*\} = \underset{\mathbf{p}\in\mathcal{D}}{\arg\max} \log EI(\mathbf{p}). \tag{4.32}$$

We use a multistart method which runs a local gradient-based solver for problem (4.32) from multiple starting points of domain $\mathcal{D}$ to find the new points $\mathcal{P}^* = \{\mathbf{p}_1^*, \dots, \mathbf{p}_{N_{\text{new}}}^*\}$ which are local (and the global) maxima of the expected improvement function [52]. For this, we have a large number of starting points and leave out all starting points which do not lie in the basin of attraction of the local maxima in order to reduce the number of optimization procedures. These points have the property that the expected improvement is below $EI(\mathbf{p}) < 10^{-14}$.

Surrogate Model for a Regular Grid

For a regular grid of dimension d, we construct a surrogate $\tilde{e}_{\text{CV}}(\mathbf{p})$ for the error indicator on the domain $\mathcal{D}$ by interpolating the data $\{(\mathbf{p}_1, e_{\text{CV}}(\mathbf{p}_1)), \dots, (\mathbf{p}_N, e_{\text{CV}}(\mathbf{p}_N))\}$ using linear interpolation between the interior points and otherwise nearest neighbor interpolation. This is shown for a 1-dimensional parameter space $p \in [0,\ 1]$ with $N = 7$ grid points in Figure 4.13. One can see the data points $\{(p_1, e_{\text{CV}}(p_1)), \dots, (p_7, e_{\text{CV}}(p_7))\}$ and the linear surrogate $\tilde{e}_{\text{CV}}(p)$.

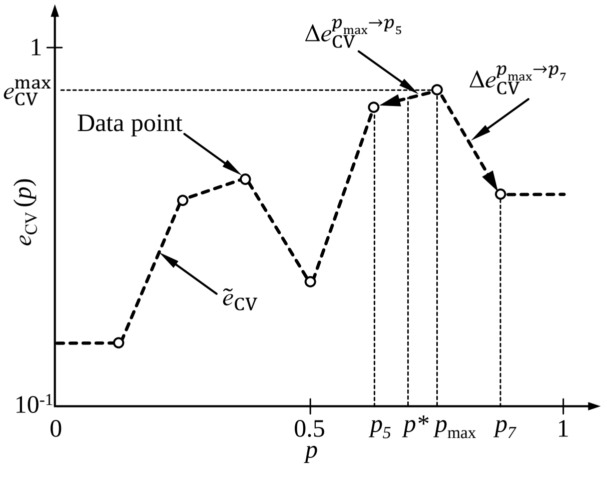


Figure 4.13: A graphic interpretation of the refinement procedure for a regular grid.

Then, we insert the surrogate $\tilde{e}_{\text{CV}}(\mathbf{p})$ into the optimization problem (4.27) which can easily be solved making use of the following property of a regular grid: Each point is indexed along the coordinate directions and has a fixed number of predefined neighbors which can be used to define a slope. Firstly, we calculate the maximum error indicator $e_{\text{CV}}^{\max} = \max(\mathcal{E}_{\text{CV}})$ with corresponding grid point $\mathbf{p}_{\max}$. Secondly, we calculate the slopes $\Delta e_{\text{CV}}^{\mathbf{p}_{\max}\to\mathbf{p}_j}$ from point $\mathbf{p}_{\max}$ to its N_{nb} neighbor points $\mathbf{p}_j$ with $j \in \{1, \dots, N_{\text{nb}}\}$:

$$\Delta e_{\text{CV}}^{\mathbf{p}_{\max}\to\mathbf{p}_j} = \frac{e_{\text{CV}}(\mathbf{p}_j) - e_{\text{CV}}^{\max}}{||\mathbf{p}_j - \mathbf{p}_{\max}||_2}. \tag{4.33}$$

Thirdly, we take the coordinate direction with the most negative slope and we insert a $d-1$-dimensional structure of new grid points $\mathcal{P}^*$ orthogonal to this direction, but on the opposite side of $\mathbf{p}_{\max}$ in the center between the points so that the regularity of the grid is preserved. For example, for a 2-dimensional grid we insert points lying on a line and for a 1-dimensional grid we insert one point. The latter case is demonstrated in Figure 4.13. One can observe the slopes from point $p_{\max}$ to its neighbor points p_5, p_7. As the latter slope is most negative, we insert a new point p^* in the center between grid points $p_5, p_{\max}$.

Remark: New points are sometimes inserted in the same direction many times, especially at the boundaries, although it is not necessary. In such a case, choose the second most negative slope and refine in the corresponding direction in the next iteration.

In contrast to current approaches [142, 38, 30] where the local structure is preserved using cells—and hence only local interpolation schemes such as linear interpolation can be applied—we preserve the global structure. For this, some points might be inserted in regions, where they seem to be unnecessary but they are needed to preserve the structure of the grid in order to apply global interpolation schemes such as higher order polynomials, see e.g. Figure 4.22, which saves a lot of grid points.

4.4.5 Properties

The proposed refinement strategy has the following properties:

1. The refinement strategy does not revisit previously sampled points. For a Kriging surrogate, the variance is zero at the grid points and hence, there is no expectation of improvement at an already sampled point. For a RBF surrogate, constraints around the grid points prohibit a re-sampling of a point. For a regular grid, we insert new grid points in the center between grid points and therefore, the approach will never sample at previously sampled points.

2. We insert grid points where the error indicator is large or where we can expect the error indicator to be large for the Kriging surrogate. As the cross-validation-based

error is a good indicator for the true error, we can expect to insert new points in regions where the true error is large.

Hence, we may expect the overall error to quickly fall with growing iterations and to converge to the maximum error due to projection, as the error due to interpolation vanishes and only the one due to projection remains, see Section 3.8:

$$\lim_{N\to\infty}\left(\max_{\mathbf{p}\in\mathcal{P}} e_{\mathrm{T}}(\mathbf{p})\right) = \max_{\mathbf{p}\in\mathcal{D}} e_{\mathrm{P}}(\mathbf{p}). \tag{4.34}$$

Global convergence is guaranteed for piecewise polynomial interpolation whereas polynomial interpolation may not lead to global convergence [138]. A well-known counterexample is the Runge's phenomenon. In that case, the predicted upper bound in Section 4.5 provides a tool to monitor the convergence.

The computational complexity for one iteration is shown in Table 4.2 for different norms where we need only N_{new} high-order solves. The sign $\ll$ indicates that the actual complexity which is dominated by the solution of a system of linear equations is between $\mathcal{O}(n)$ and $\mathcal{O}(n^3)$ depending on the sparsity of the system matrices. With this method, it is now possible for interpolation-based methods of pMOR to have the same computational complexity per iteration as global basis approaches based on error indicators such as global POD which use the $\mathcal{L}_p$-norm, see e.g. [129].

Table 4.2: Computational complexity per iteration for calculating the set of error indicators and the set of high-order values for different error norms.

	$\mathcal{L}_p$	$\mathcal{H}_\infty$	$\mathcal{H}_2$
Calculation of set $\mathcal{E}_{\mathrm{CV}}$	$NK_{\mathrm{L}}\mathcal{O}(q^3)$	$NK_{\mathrm{H}}\mathcal{O}(q^3)$	$\ll Nq\mathcal{O}(n^3)$
Calculation of set $\mathcal{S}^*$	$\ll N_{\mathrm{new}}K_{\mathrm{L}}\mathcal{O}(n^3)$	$\ll N_{\mathrm{new}}K_{\mathrm{H}}\mathcal{O}(n^3)$	$\ll N_{\mathrm{new}}n_{\mathrm{s}}\mathcal{O}(n^3)$

4.5 Model Error Prediction

4.5.1 Overview

The cross-validation error has so far been used as an indicator for the accuracy of the reduced system. In most cases this indicator is larger than the (unknown) true error because its computation relies on a smaller training set. Hence, we aim to predict the true error with a desired probability $(1-\alpha)\cdot 100\%$ for $\alpha \in [0,\ 1]$:

$$e_{\mathrm{T}}(\mathbf{p}) \lesssim e_{\mathrm{est}}^{1-\alpha}(\mathbf{p}). \tag{4.35}$$

As the error indicator correlates well with the true error, we firstly determine a mapping function from the error indicator to the error $\tilde{e}_{\mathrm{CV}}(\mathbf{p}) \mapsto e_{\mathrm{T}}(\mathbf{p})$. Secondly, we measure the uncertainty which results in the upper bound $e_{\mathrm{est}}^{1-\alpha}(\mathbf{p})$. In numerical examples in Section 4.7, we find the cross-validation-based error indicator to exhibit an approximately linear dependence on the true error on a logarithmic scale. This assumption especially holds true if the error has some orders of magnitude during the refinement process. Hence, we create an error pool in Section 4.5.2 and calculate a predicted upper bound assuming a linear error model in Section 4.5.3 which we use as a stopping criterion for the refinement procedure such as described in Figure 4.8. Then, the refinement process stops as soon as the maximum predicted error falls below a given tolerance.

4.5.2 Creation of an Error Pool

For the mapping we collect the error indicator and the true error assigned to the grid points in an error pool $\mathcal{E}$. According to Section 4.4.4 we insert N_{new} grid points from the set $\mathcal{P}^*$ in every iteration. At these points we evaluate the surrogate model to obtain the error indicators $\tilde{e}_{\mathrm{CV}}(\mathbf{p}_1), \dots, \tilde{e}_{\mathrm{CV}}(\mathbf{p}_{N_{\mathrm{new}}})$. In addition, we evaluate the reduced system $\tilde{\mathbf{G}}(\mathbf{p})$—before it is updated—at the new grid points and compute the overall errors $e_{\mathrm{T}}(\mathbf{p}_1), \dots, e_{\mathrm{T}}(\mathbf{p}_{\mathrm{new}})$ using set $\mathcal{S}^*$. Then, the error pool is updated in every iteration and grows for N_{new} entries to $\mathcal{E} = \mathcal{E} \cup \{(\tilde{e}_{\mathrm{CV}}(\mathbf{p}_1), e_{\mathrm{T}}(\mathbf{p}_1)), \dots, (\tilde{e}_{\mathrm{CV}}(\mathbf{p}_{N_{\mathrm{new}}}), e_{\mathrm{T}}(\mathbf{p}_{N_{\mathrm{new}}}))\}$. For the initial set of grid points we go one step back, leave some points and reduced systems out and assign error indicators according to Section 4.2.2. Then, we construct a surrogate model according to Section 4.4.4 and evaluate it at the left-out points where we also compute the overall error with set $\mathcal{S}$.

4.5.3 Predicted Upper Bound

With the error pool $\mathcal{E} = \{(\tilde{e}_{\mathrm{CV}}(\mathbf{p}_1), e_{\mathrm{T}}(\mathbf{p}_1)), \dots, (\tilde{e}_{\mathrm{CV}}(\mathbf{p}_N), e_{\mathrm{T}}(\mathbf{p}_N))$, we calculate a mapping from the error indicator to the true error on a logarithmic scale. For this, we assume a linear trend on domain $\mathcal{D}$ with added random noise [162, 148]

$$y_{\mathrm{reg}}(\mathbf{p}) = \mathbf{x}_{\mathrm{reg}}^T(\mathbf{p})\boldsymbol{\beta} + \epsilon \tag{4.36}$$

with the predictor $\mathbf{x}_{\mathrm{reg}}^T(\mathbf{p}) = [1 \;\; \log(\tilde{e}_{\mathrm{CV}}(\mathbf{p}))]$, response $y_{\mathrm{reg}}(\mathbf{p}) = \log\left(e_{\mathrm{T}}(\mathbf{p})\right)$ and vector $\boldsymbol{\beta} = [\beta_0 \;\; \beta_1]^T$ with intercept β_0 and slope β_1. The random noise $\epsilon \sim \mathcal{N}\left(0, \sigma^2\right)$ is assumed to be normally distributed with zero mean and variance σ^2 and it is independent for each observation. Therefore, the response $y_{\mathrm{reg}}(\mathbf{p}) \sim \mathcal{N}\left(\mathbf{x}_{\mathrm{reg}}^T(\mathbf{p})\boldsymbol{\beta}, \sigma^2\right)$ is also random with variance σ^2 and mean $y_{\mathrm{mean}}(\mathbf{p}) = \mathbf{x}_{\mathrm{reg}}^T(\mathbf{p})\boldsymbol{\beta}$. Since the parameters $\boldsymbol{\beta}$ and σ^2 of the error model are unknown, we firstly aim to construct unbiased estimates $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ which give the best fit to the data $\mathcal{E}$. For this, we insert the logarithm of the observed data points of the set $\mathcal{E}$ into the error model (4.36) and obtain the following estimation problem, where each row corresponds to one data sample:

$$\underbrace{\begin{bmatrix} \log\left(e_{\mathrm{T}}(\mathbf{p}_1)\right) \\ \log\left(e_{\mathrm{T}}(\mathbf{p}_2)\right) \\ \vdots \\ \log\left(e_{\mathrm{T}}(\mathbf{p}_N)\right) \end{bmatrix}}_{\mathbf{Y}_{\mathrm{reg}} \in \mathbb{R}^N} = \underbrace{\begin{bmatrix} 1 & \log\left(\tilde{e}_{\mathrm{CV}}(\mathbf{p}_1)\right) \\ 1 & \log\left(\tilde{e}_{\mathrm{CV}}(\mathbf{p}_2)\right) \\ \vdots & \vdots \\ 1 & \log\left(\tilde{e}_{\mathrm{CV}}(\mathbf{p}_N)\right) \end{bmatrix}}_{\mathbf{X}_{\mathrm{reg}} \in \mathbb{R}^{N \times 2}} \underbrace{\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}}_{\boldsymbol{\beta}} + \underbrace{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}}_{\boldsymbol{\epsilon} \in \mathbb{R}^N}. \tag{4.37}$$

According to the Gauss-Markov-Theorem the best linear unbiased estimator $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ is given by the least squares solution

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}_{\mathrm{reg}}^T \mathbf{X}_{\mathrm{reg}})^{-1} \mathbf{X}_{\mathrm{reg}}^T \mathbf{Y}_{\mathrm{reg}}, \tag{4.38}$$

where $\hat{\boldsymbol{\beta}}$ is a random variable with normal distribution $\hat{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}, \sigma^2 (\mathbf{X}_{\mathrm{reg}}^T \mathbf{X}_{\mathrm{reg}})^{-1})$. An unbiased estimator $\hat{\sigma}^2$ for σ^2 is

$$\hat{\sigma}^2 = \frac{(\mathbf{Y}_{\mathrm{reg}} - \mathbf{X}_{\mathrm{reg}} \hat{\boldsymbol{\beta}})^T (\mathbf{Y}_{\mathrm{reg}} - \mathbf{X}_{\mathrm{reg}} \hat{\boldsymbol{\beta}})}{N-2} \tag{4.39}$$

which is also a random variable. Secondly, we are interested in the prediction error which is the error between a future random observation $y_{\mathrm{reg}}(\mathbf{p})$ from model (4.36) and its predicted mean $\hat{y}_{\mathrm{mean}}(\mathbf{p}) = \mathbf{x}_{\mathrm{reg}}^T(\mathbf{p}) \hat{\boldsymbol{\beta}}$

$$y_{\mathrm{reg}}(\mathbf{p}) - \hat{y}_{\mathrm{mean}}(\mathbf{p}) = \mathbf{x}_{\mathrm{reg}}^T(\mathbf{p})(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) + \epsilon, \tag{4.40}$$

where the difference is drawn from the normal distribution

$$y_{\mathrm{reg}}(\mathbf{p}) - \hat{y}_{\mathrm{mean}}(\mathbf{p}) \sim \mathcal{N}(0, \hat{\sigma}^2 (1 + \mathbf{x}_{\mathrm{reg}}^T(\mathbf{p}) (\mathbf{X}_{\mathrm{reg}}^T \mathbf{X}_{\mathrm{reg}})^{-1} \mathbf{x}_{\mathrm{reg}}(\mathbf{p}))). \tag{4.41}$$

Finally, we calculate a prediction $y_{\mathrm{est}}^{1-\alpha}(\mathbf{p})$ so that a future random observation $y_{\mathrm{reg}}(\mathbf{p})$ is lower than that value with $(1-\alpha) \cdot 100\%$ probability where $\alpha \in [0,\ 1]$

$$y_{\mathrm{reg}}(\mathbf{p}) \lesssim y_{\mathrm{est}}^{1-\alpha}(\mathbf{p}). \tag{4.42}$$

The predicted value is the predicted mean superposed with the uncertainty from (4.41), where t_{N-2}^{α} is the cumulative distribution function of the t-distribution with $N-2$

degrees of freedom and $(1-\alpha)\cdot 100\%$ probability

$$y_{\text{est}}^{1-\alpha}(\mathbf{p}) = \mathbf{x}_{\text{reg}}^{T}(\mathbf{p})\hat{\boldsymbol{\beta}} + t_{N-2}^{\alpha}\hat{\sigma}\sqrt{1+\mathbf{x}_{\text{reg}}^{T}(\mathbf{p})(\mathbf{X}_{\text{reg}}^{T}\mathbf{X}_{\text{reg}})^{-1}\mathbf{x}_{\text{reg}}(\mathbf{p})}. \tag{4.43}$$

Hence, the predicted value considers three parts for the uncertainty: firstly the random error, secondly the error of the estimator $\hat{\boldsymbol{\beta}}$ and thirdly the error of the estimator $\hat{\sigma}$ which is taken into account using the t-distribution. A prediction for the true error $e_{\text{T}}(\mathbf{p}) \lesssim e_{\text{est}}^{1-\alpha}(\mathbf{p})$ with $(1-\alpha)\cdot 100\%$ probability can then be obtained as

$$e_{\text{est}}^{1-\alpha}(\mathbf{p}) = 10^{\mathbf{x}_{\text{reg}}^{T}(\mathbf{p})\hat{\boldsymbol{\beta}} + t_{N-2}^{\alpha}\hat{\sigma}\sqrt{1+\mathbf{x}_{\text{reg}}^{T}(\mathbf{p})(\mathbf{X}_{\text{reg}}^{T}\mathbf{X}_{\text{reg}})^{-1}\mathbf{x}_{\text{reg}}(\mathbf{p})}}. \tag{4.44}$$

As an example for a 1-dimensional space $p \in [0,\ 1]$, in Figure 4.14 one can see the prediction $e_{\text{est}}^{1-\alpha}(\mathbf{p})$ with $(1-\alpha)\cdot 100\%$ probability which is demonstrated for the value $\tilde{e}_{\text{CV}}(p_1)$ with the corresponding probability density function. The underlying data follows a linear trend on a logarithmic scale.

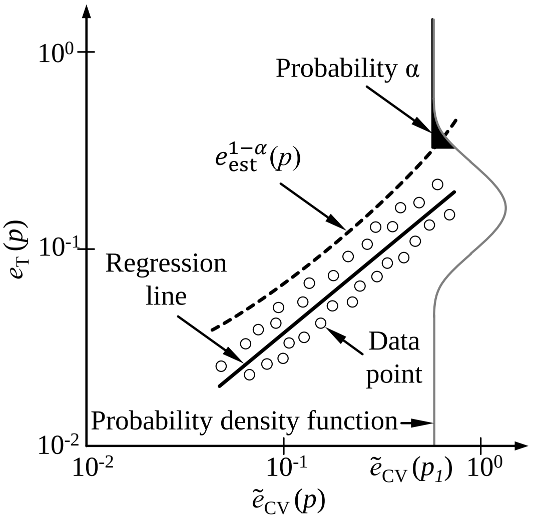

10^0
10^{-1}
10^{-2}
$e_{\text{T}}(p)$
Probability α
$e_{\text{est}}^{1-\alpha}(p)$
Regression line
Data point
Probability density function
10^{-2}
10^{-1}
$\tilde{e}_{\text{CV}}(p_1)$
10^0
$\tilde{e}_{\text{CV}}(p)$

Figure 4.14: A graphic interpretation of the linear error model and the predicted value.

In each iteration, we are interested in an upper bound $e_{\text{est,max}}^{1-\alpha}$ for the true maximum error with probability $(1-\alpha)\cdot 100\%$ which we refer to as *predicted upper bound*:

$$\max_{\mathbf{p}\in\mathcal{D}} e_{\text{T}}(\mathbf{p}) \lesssim e_{\text{est,max}}^{1-\alpha}. \tag{4.45}$$

For this, we insert the maximum error indicator $e_{\text{CV}}^{\max} = \max(\mathcal{E}_{\text{CV}})$ into (4.44) and finally, we obtain with $\mathbf{x}_{\text{reg,max}}^{T} = [1 \;\; \log(e_{\text{CV}}^{\max})]$

$$e_{\text{est,max}}^{1-\alpha} = 10^{\mathbf{x}_{\text{reg,max}}^{T}\hat{\boldsymbol{\beta}} + t_{N-2}^{\alpha}\hat{\sigma}\sqrt{1+\mathbf{x}_{\text{reg,max}}^{T}(\mathbf{X}_{\text{reg}}^{T}\mathbf{X}_{\text{reg}})^{-1}\mathbf{x}_{\text{reg,max}}}}. \tag{4.46}$$

The whole procedure is shown in Algorithm 6.

Algorithm 6 Model error prediction

```
Input: Error indicators $\mathcal{E}_{\mathrm{CV}}$, error pool $\mathcal{E}$ and probability $\alpha$
Output: Predicted upper bound $e^{1-\alpha}_{\mathrm{est,max}}$
Compute $\mathbf{X}_{\mathrm{reg}}, \mathbf{Y}_{\mathrm{reg}}$ and unbiased estimators $\hat{\boldsymbol{\beta}}, \hat{\sigma}^2$ using set $\mathcal{E}$
Compute $e^{\max}_{\mathrm{CV}} = \max(\mathcal{E}_{\mathrm{CV}})$
Calculate with $\mathbf{x}^T_{\mathrm{reg,max}} = [1 \ \ \log(e^{\max}_{\mathrm{CV}})]$:
    $e^{1-\alpha}_{\mathrm{est,max}} = 10^{\mathbf{x}^T_{\mathrm{reg,max}}\hat{\boldsymbol{\beta}} + t^{\alpha}_{N-2}\hat{\sigma}\sqrt{1+\mathbf{x}^T_{\mathrm{reg,max}}(\mathbf{X}^T_{\mathrm{reg}}\mathbf{X}_{\mathrm{reg}})^{-1}\mathbf{x}_{\mathrm{reg,max}}}}.$    (4.47)
```

4.5.4 Properties and Application

The predicted upper bound $e^{1-\alpha}_{\mathrm{est,max}}$ leads to a notion of probabilistic rigor, i.e. the error is bounded with the desired probability $(1-\alpha)\cdot 100\%$ assuming the error model holds true [53]. Looking at its exponent, we can expect the upper bound to fall for growing iterations for the following reasons:

1. The maximum error indicator $e^{\max}_{\mathrm{CV}}$ is expected to fall for the proposed refinement method as a new point is inserted where the indicator is maximum and hence, the left term $\mathbf{x}^T_{\mathrm{reg,max}}\hat{\boldsymbol{\beta}}$ decreases.

2. With each iteration, the error pool grows and we can expect the estimators $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}$ to become more accurate and hence, the right term decreases.

We use the predicted upper bound to monitor the convergence of the refinement method from Section 4.4. In addition, we apply it as a stopping criterion and we stop the procedure when the predicted upper bound falls below a given tolerance:

$$e^{1-\alpha}_{\mathrm{est,max}} \leq e_{\mathrm{tol}}. \tag{4.48}$$

4.6 Algorithm

In the previous sections, the algorithms for method selection, model refinement and error prediction have been presented. These steps are combined to the Black-Box method which we have already seen in Figure 4.1 in the introductory section. The method is given in Algorithm 7. We start with a space-filling design of grid points and identify the most accurate interpolation method. Afterwards, new grid points are determined iteratively and the refinement procedure stops when the predicted upper bound falls below a given tolerance or when the maximum number of iterations is reached. Finally, the parameter-dependent reduced system is constructed.

Algorithm 7 Black-Box method

```
Input: Original system $\mathbf{G}(\mathbf{p})$, number of starting points $N$, error tolerance $e_{\text{tol}}$,
       maximum number of iterations $n_{\text{iter}}^{\max}$, reduced order $q$, set $\mathcal{K}$, probability $\alpha$
Output: Reduced order model $\tilde{\mathbf{G}}(\mathbf{p})$
Choose a space-filling design with points $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$
Calculate set $\mathcal{S}$ of high-order values at points of the set $\mathcal{P}$
Calculate set of low-order systems $\tilde{\mathcal{G}}$ at points of the set $\mathcal{P}$
Calculate most accurate interpolation method for set $\mathcal{K}$ using Algorithm 4
Calculate starting error pool $\mathcal{E}$
Initialize $n_{\text{iter}} = 0$
while $n_{\text{iter}} < n_{\text{iter}}^{\max}$ do
    Calculate error indicators $\mathcal{E}_{\text{CV}}$ at points $\mathcal{P}$ using data $\mathcal{S}$ with Algorithm 3
    Calculate the predicted upper bound $e_{\text{est,max}}^{1-\alpha}$ with Algorithm 6.
    if $e_{\text{est,max}}^{1-\alpha} \leq e_{\text{tol}}$ then
        Break
    end if
    Calculate surrogate model $\tilde{e}_{\text{CV}}(\mathbf{p})$ with error indicators $\mathcal{E}_{\text{CV}}$
    Find set of new grid points $\mathcal{P}^*$ using Algorithm 5
    Calculate set $\mathcal{S}^*$ of high-order values at grid points $\mathcal{P}^*$
    Calculate set of low-order systems $\tilde{\mathcal{G}}^*$ at grid points $\mathcal{P}^*$
    Update error pool $\mathcal{E} = \mathcal{E} \cup \{(\tilde{e}_{\text{CV}}(\mathbf{p}_1), e_{\text{T}}(\mathbf{p}_1)), \dots, (\tilde{e}_{\text{CV}}(\mathbf{p}_{|\mathcal{P}^*|}), e_{\text{T}}(\mathbf{p}_{|\mathcal{P}^*|}))\}$
    Update the sets $\mathcal{P} = \mathcal{P} \cup \mathcal{P}^*$, $\mathcal{S} = \mathcal{S} \cup \mathcal{S}^*$, $\tilde{\mathcal{G}} = \tilde{\mathcal{G}} \cup \tilde{\mathcal{G}}^*$
    Set $n_{\text{iter}} = n_{\text{iter}} + 1$
end while
Build interpolant $\tilde{\mathbf{G}}(\mathbf{p})$ using the reduced systems $\tilde{\mathcal{G}}$ at grid points $\mathcal{P}$
```

The complexity of the Black-Box method is summarized in Table 4.3, where N is the final number of grid points. The method selection approach is computationally cheap as it only relies on low-order solves. The error prediction method is cheap for a moderate number of grid points. The complexity is dominated by the refinement procedure. This requires N high-order solves, which is however much less than for current approaches.

Table 4.3: Computational complexity of the Black-Box method.

	Method selection	Model refinement	Error prediction
Complexity	$\mathcal{O}(q^3)$	$\ll N\mathcal{O}(n^3)$	$\mathcal{O}(N)$

4.7 Numerical Examples

4.7.1 Cantilever Beam ($n = 1200$)

The first example is the model of a cantilever beam with order $n = 1200$ which was introduced in Section 3.9.2. Parameters of the beam are the length $L \in [1,\ 2]$m and the height $h \in [0.01,\ 0.02]$m. The system is driven with the force $F(t) = 10$N in the time interval $t \in [0,\ 0.5]$s with initial state vector $\mathbf{x}_0 = \mathbf{0}$. The integration is done using the implicit Euler method with the step size $\Delta t = 10^{-4}$s. The accuracy of the output signal will be measured with the $\mathcal{L}_2$-norm. Exemplary plots of the output for different parameter values are shown in Figure 4.15.

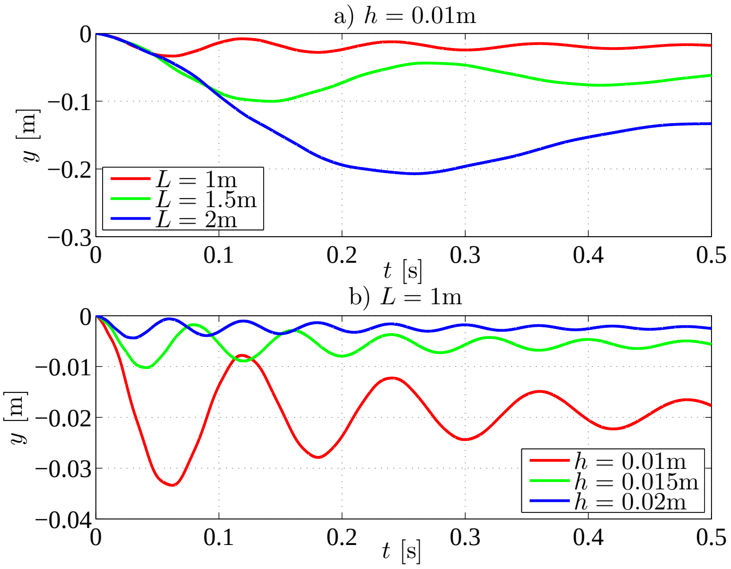


Figure 4.15: Plots of the output of the beam for different parameter values.

The parameter domain is normalized to $\mathcal{D} = [0,\ 1] \times [0,\ 1]$ for the sake of an easier presentation and we start with the grid $\mathcal{P} = \{0, 1/3, 2/3, 1\} \times \{0, 1/3, 2/3, 1\}$ which is shown in Figure 4.16(a). Then, we obtain $N = 16$ high-order systems which we reduce to order $q = 18$ using the two-sided Krylov subspace method with expansion points $s_0 = 0$ according to Section 2.2.2. The adjustment of the ROBs is done with the PS approach (3.13) and (3.25). The reference ROBs are calculated with the non-weighted SVD approach (3.5) and (3.22).

Method Selection

We want to find the most accurate interpolation method for gridded data interpolation or in other words for a regular grid. For this, we choose between linear (LIN) and

Lagrange (LAG) interpolation and between the real (RE) or nonsingular (NO) matrix manifold for the square matrices which give a set of four candidate methods. Using Algorithm 3, we assign error indicators to the grid points in set $\mathcal{P}$ which we use to calculate the estimated error measures J_μ^{CV} and $J_{\mathrm{RMSE}}^{\mathrm{CV}}$ for every combination with formula (4.23) and (4.24). The true error measures J_μ and J_{RMSE} are approximated using the test grid $\mathcal{P}_{\mathrm{test}} = \{0, 1/40, \ldots, 1\} \times \{0, 1/40, \ldots, 1\}$ with formula (4.21) and (4.22), respectively. Of course, this is very expensive and only done here for the reason of comparison. The estimated and the true error measures are shown in Table 4.4.

Table 4.4: Comparison between the mean and the root mean square of the error indicators at the grid points and of the errors of a test grid for different interpolation methods ordered from small (1) to large (4).

	LIN (RE)	LIN (NO)	LAG (RE)	LAG (NO)
J_μ^{CV}	0.2255 (4)	0.1078 (3)	0.0945 (2)	0.0339 (1)
J_μ	0.0678 (4)	0.0211 (3)	0.0066 (2)	0.0019 (1)
$J_{\mathrm{RMSE}}^{\mathrm{CV}}$	0.2370 (4)	0.1154 (3)	0.1093 (2)	0.0364 (1)
J_{RMSE}	0.0731 (4)	0.0269 (3)	0.0079 (2)	0.0021 (1)

One can verify the estimated error measures deliver the correct order of the candidate methods, both for the mean and for the root mean square error. The estimated error measures are larger than the true ones because they rely on a smaller training data. The candidate method with Lagrange interpolation and the nonsingular matrix manifold is identified as the method with smallest error measure, both for the mean and for the root mean square error which is confirmed by the true error measures. We will use this method for the refinement procedure in the next section.

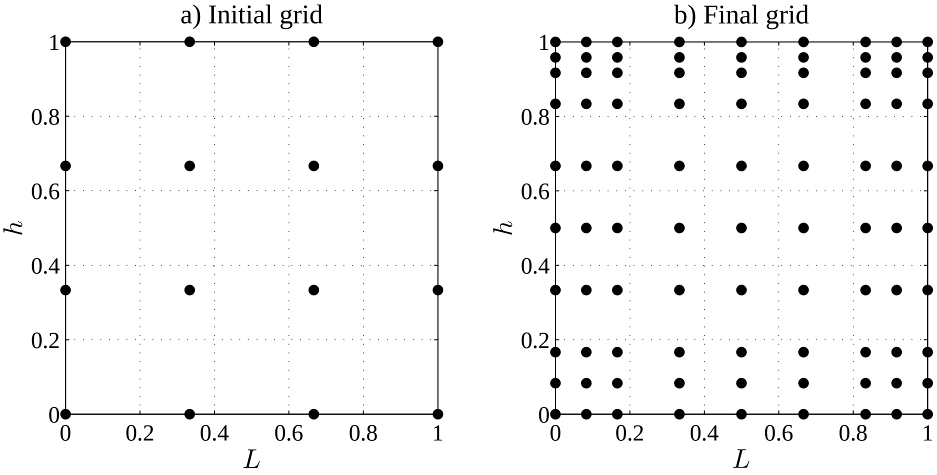


Figure 4.16: Grid for a) iteration $n_{\mathrm{iter}} = 0$ and b) iteration $n_{\mathrm{iter}} = 11$.

Model Refinement and Error Prediction

We perform the refinement procedure from Algorithm 7 with $n_{\text{iter}}^{\max} = 11$. Then, we obtain an error pool which grows in every iteration and we calculate the predicted error with 99% probability as a function of the error indicator in every iteration. The respective values are plotted in Figure 4.17 for iterations $n_{\text{iter}} \in \{0, 3, 7, 11\}$. Iteration $n_{\text{iter}} = 0$ describes the starting pool. One can verify there is an approximately linear behavior on a logarithmic scale between the error indicator and the error which becomes more accurate with growing error pool. In Figure 4.18 one can see the maximum error indicator and the maximum error for a growing number of grid points. The latter is obtained in every iteration using again the test grid $\mathcal{P}_{\text{test}}$. The true error converges to the minimum value which is the maximum error due to projection according to formula (4.34). The final grid is shown in Figure 4.16(b). One can see that the procedure inserts more points at the boundaries and preserves the regular structure of the grid. For comparison, the error obtained by full grids such as in Figure A.2 with $N \in \{4^2, \ldots, 9^2\}$

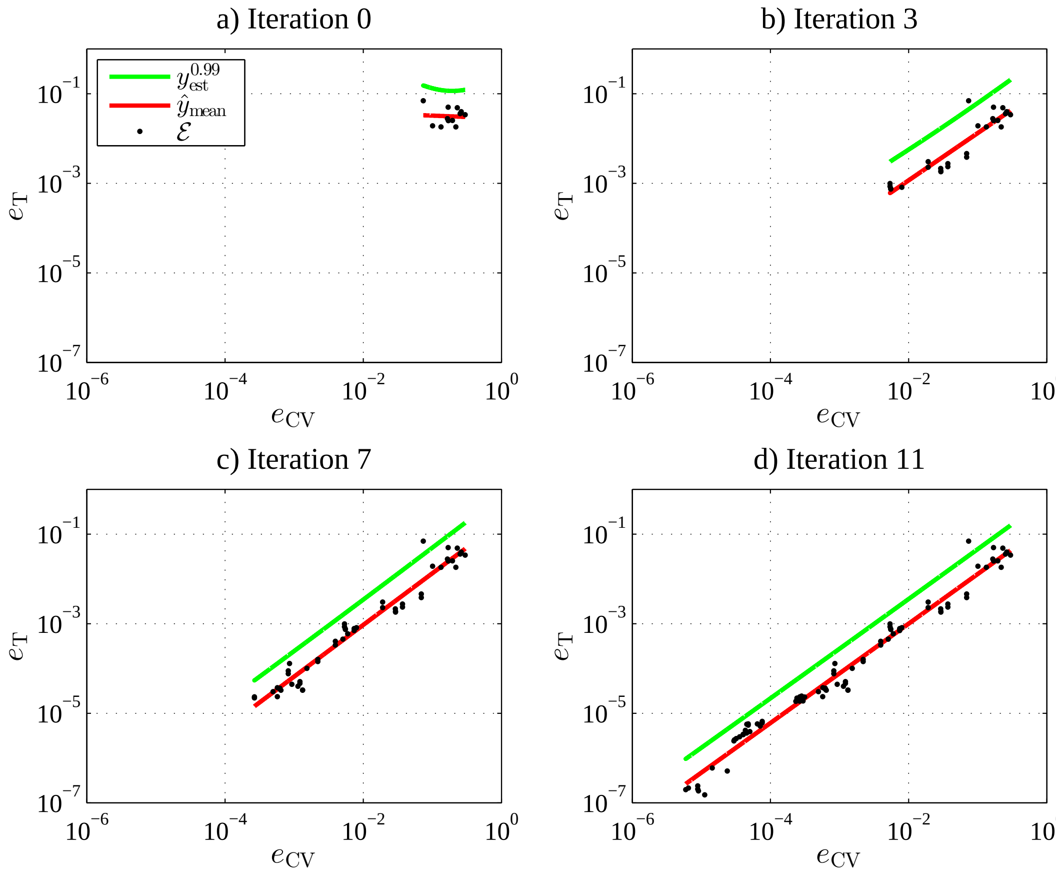


Figure 4.17: Error pool, regression line and the predicted error with 99% probability for different iterations for the beam.

points is shown and one can verify that the proposed method leads to smaller errors. In addition, one can see the predicted upper bound with 99% probability provides a tight bound for the error and it becomes more accurate for a growing error pool. Hence, it can be used as a stopping criterion for the refinement procedure.

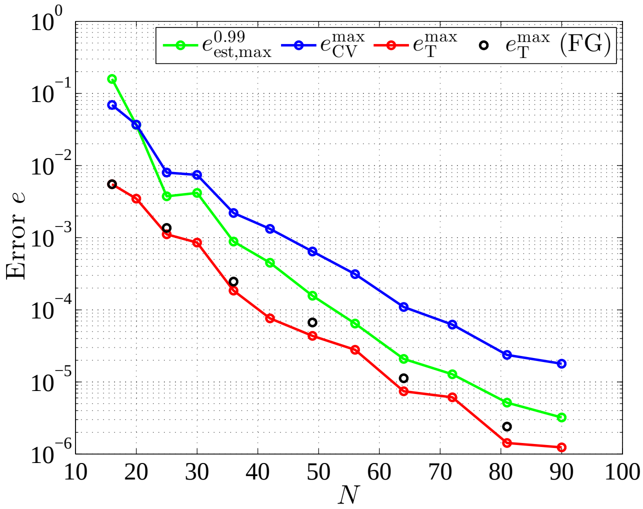


Figure 4.18: Predicted upper bound, maximum error indicator and maximum error obtained by the proposed method and by full grids (FG) for the beam.

4.7.2 Microthruster Unit ($n = 4257$)

The next system is a microthruster unit. It is taken from the Oberwolfach Model Reduction Benchmark Collection [2, 60]. The aim of a microthruster is to produce propulsion or to generate pressure by igniting a solid fuel. The fuel is ignited by passing an electric current which generates heat through a polysilicon resistor embedded in a dielectric membrane. The model we consider is shown in Figure 4.19. It is a device with a single heat source, where the generated heat dissipates through the device to the surroundings. The heat generation rate is zero outside of the resistor. The temperature distribution in the microthruster unit can be expressed by the heat diffusion equation

$$\nabla(\kappa\nabla T) + Q - \rho c_p \frac{\partial T}{\partial t} = 0 \tag{4.49}$$

with thermal conductivity κ, density ρ, specific heat capacity c_p, heat source Q and temperature T. The boundary is divided into three regions: top, bottom and side. For each of them, we apply the convection boundary condition describing the heat exchange

$$Q_i^{\text{B}} = h_i(T - T_{\text{air}}) \;\; \forall \;\; i \in \{1, 2, 3\} \tag{4.50}$$

where T_{air} is the temperature of the surroundings which is assumed to be zero. The two film coefficients $h_1,\ h_2$ vary in the interval $h_i \in [1,\ 10^3]\text{Wm}^{-2}\text{K}^{-1}$ and the third one is $h_3 = 1\text{Wm}^{-2}\text{K}^{-1}$. The discretization of (4.49) and (4.50) using the finite element method leads to a parameter-dependent state-space system of order $n = 4257$

$$\begin{aligned}\mathbf{E}\dot{\mathbf{T}}(t) &= \overbrace{(\mathbf{A}_0 - h_1\mathbf{A}_1 - h_2\mathbf{A}_2)}^{\mathbf{A}(h_1,h_2)}\mathbf{T}(t) + \mathbf{b}Q(t) \\ y(t) &= \mathbf{c}^T\mathbf{T}(t),\end{aligned} \tag{4.51}$$

where $\mathbf{T}(t) \in \mathbb{R}^{4257}$ is the state vector containing the temperatures at the discretization nodes. The input is the heat source $Q(t)$ and the output is the temperature in the center of the heater. For testing the system, we use the $\mathcal{H}_\infty$-norm which is approximated using an error grid at 50 frequencies which are logarithmically equally spaced in the interval $10^{-1}, \ldots, 10^8$. In Figure 4.20, the magnitude of frequency responses is shown for different parameter values. There is a strong change for small parameter values which the refinement procedure will have to discover. In the following, we perform the Black-Box method using firstly a regular grid and secondly an irregular grid.

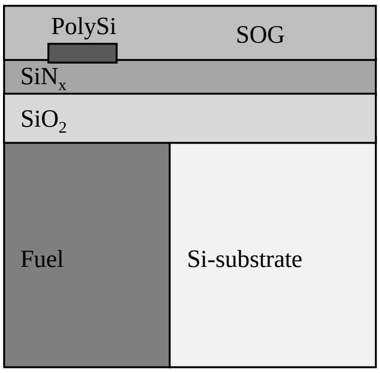


Figure 4.19: Two-dimensional asymmetrical model of the microthruster unit. The axis of the symmetry is on the left side.

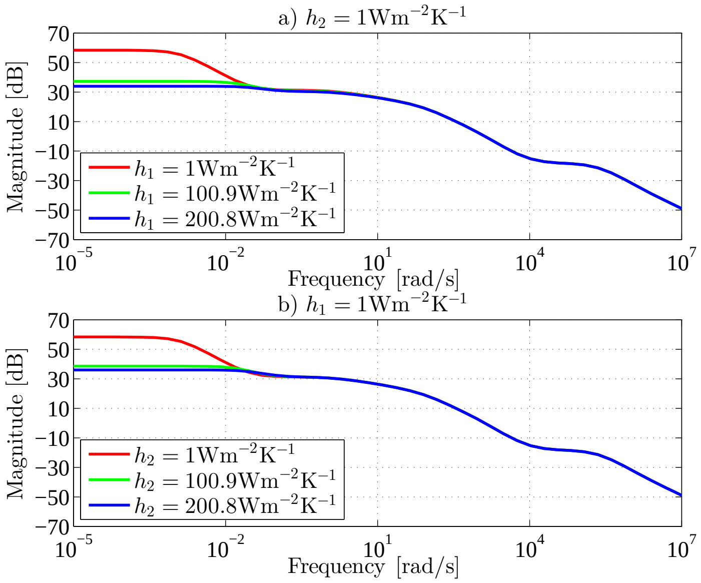


Figure 4.20: Plots of the magnitude of frequency responses of the microthruster unit for different parameter values.

Method Selection (Regular Grid)

Firstly, we consider a regular grid for the Black-Box method. For this, the parameter domain is normalized to $\mathcal{D} = [0,\ 1] \times [0,\ 1]$ and we start with the grid $\mathcal{P} = \{0, 1/3, 2/3, 1\} \times \{0, 1/3, 2/3, 1\}$ which is shown in Figure 4.21(a). This delivers $N = 16$ high-order systems which we reduce to order $q = 30$ using the one-sided Krylov subspace method with expansion points $s_0 = 40$. The adjustment of the ROBs is done with the PS approach (3.13) and (3.25). The reference ROBs are calculated with the non-weighted SVD approach (3.5) and (3.22). For the interpolation process, we consider linear (LIN) and Lagrange (LAG) interpolation and the real (RE) or nonsingular (NO) matrix manifold for the square matrices which give a set with four candidate methods. The estimated error measures J_μ^{CV} and $J_{\text{RMSE}}^{\text{CV}}$ are calculated with formula (4.23) and (4.24) and the true ones J_μ and J_{RMSE} are approximated using the test grid $\mathcal{P}_{\text{test}}^{\text{selec}} = \{0, 1/40, \ldots, 1\} \times \{0, 1/40, \ldots, 1\}$ with formula (4.21) and (4.22), respectively. The estimated and the true error measures are shown in Table 4.5.

Table 4.5: Comparison between the mean and the root mean square of the error indicators at the grid points and of the errors of a test grid for different interpolation methods ordered from small (1) to large (4).

	LIN (RE)	LIN (NO)	LAG (RE)	LAG (NO)
J_μ^{CV}	0.0057 (2)	0.2863 (4)	0.0003 (1)	0.1398 (3)
J_μ	0.0015 (2)	0.0802 (4)	0.0002 (1)	0.0348 (3)
$J_{\text{RMSE}}^{\text{CV}}$	0.0066 (2)	0.5853 (4)	0.0003 (1)	0.2446 (3)
J_{RMSE}	0.0019 (2)	0.3481 (4)	0.0006 (1)	0.1727 (3)

One can verify the estimated error measures predict the correct order of the candidate methods, both for the mean and for the root mean square error. The candidate method with Lagrange interpolation and the real matrix manifold is identified as method with smallest error measure which is confirmed by the true error measures. We will use this method for the refinement procedure.

Model Refinement and Error Prediction (Regular Grid)

We perform the refinement procedure from Algorithm 7 with $n_{\text{iter}}^{\text{max}} = 5$. In Figure 4.22 one can see the linear surrogate of the error indicator compared to the true error for iterations $n_{\text{iter}} \in \{0, 3\}$. For $n_{\text{iter}} = 0$ the linear surrogate is still quite rough as there are few grid points and the error indicators are assigned to these points. One can verify the error indicator captures the behavior of the true error, i.e. it is large where the true error

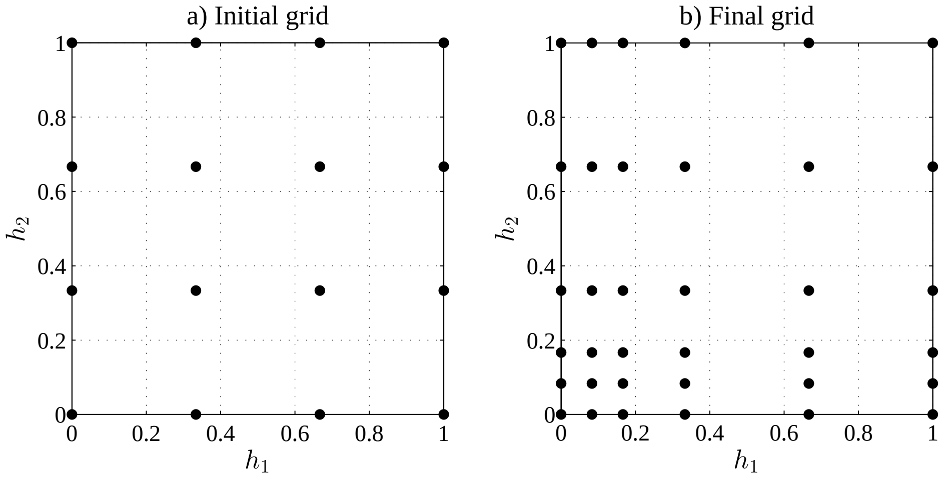


Figure 4.21: Grid for a) iteration $n_{\text{iter}} = 0$ and b) iteration $n_{\text{iter}} = 4$.

is large. As we have a regular grid of dimension $d = 2$, we insert a line of new points such as described in Section 4.4.4. For some new points it seems they are inserted in regions where the error is low. But these points are necessary in order to preserve the regular grid so we can use Lagrange interpolation. In Figure 4.23 one can observe the maximum error indicator and the true error for a growing number of grid points. The latter is assessed in every iteration using the test grid $\mathcal{P}_{\text{test}}^{\text{refine}} = \{0, 0.01, \ldots, 0.1, 0.145, \ldots, 1\} \times \{0, 0.01, \ldots, 0.1, 0.145, \ldots, 1\}$. The maximum error indicator is smaller than the true error in the first two iterations. The reason is there is a large parameter change around the origin of the domain and this region is so far not explored enough. This means there are not enough grid points so far where an error indicator could be assigned to, see Figure 4.22(a). Starting with iteration $n_{\text{iter}} = 2$, there are enough grid points in the domain in order to capture the error behavior. One can also see the true error converges to the minimum value which is defined by the error due to projection according to formula (4.34) at iteration $n_{\text{iter}} = 4$. The corresponding grid is shown in Figure 4.21(b). The reader can verify the procedure especially refines around the origin where the largest parameter change occurs, which is in agreement to Figure 4.20. For comparison, the errors obtained by different full grids with $N \in \{4^2, 5^2, 6^2\}$ points are shown and one can observe the proposed method leads to smaller errors. In addition, the predicted upper bound with 99% probability is plotted which underestimates the error in the first two iterations. The reason is that it depends on the maximum error indicator which is small in the first two iterations such as described above. Then, it provides a tight bound for the true error.

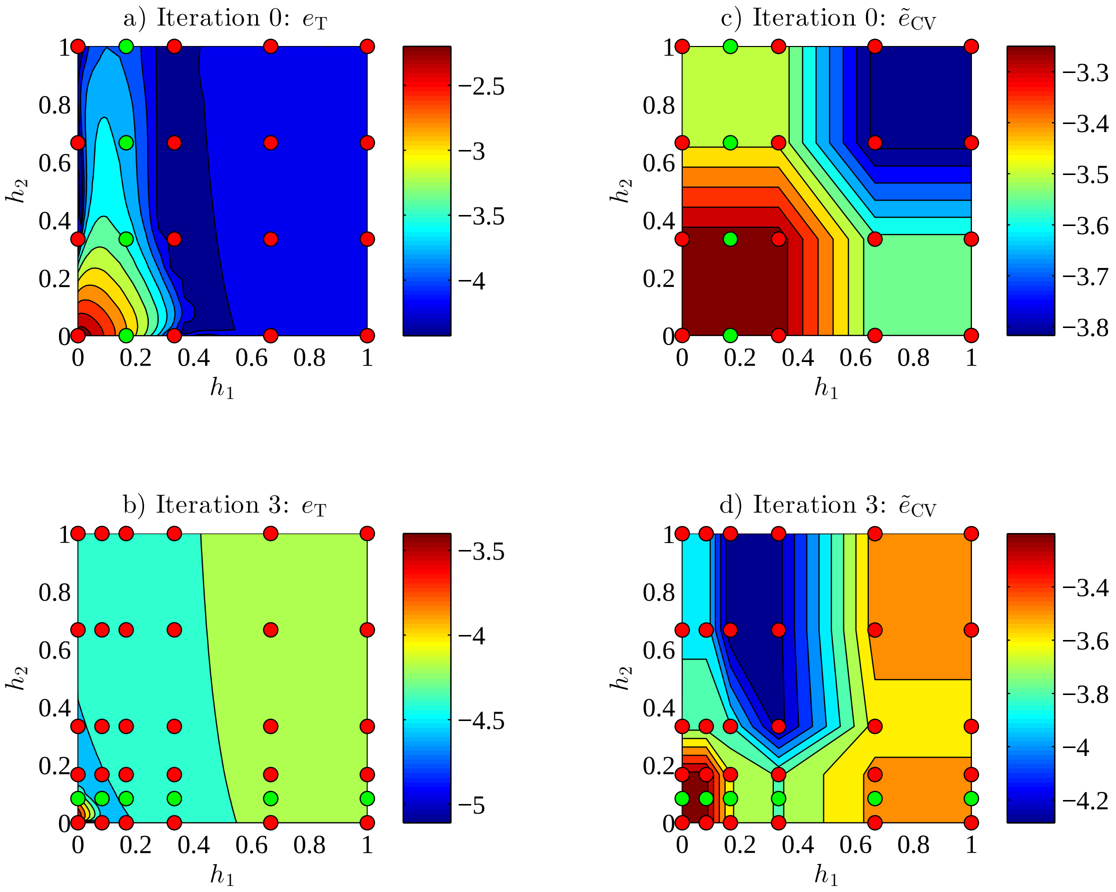


Figure 4.22: True error in a)-b) and surrogate of the error indicator in c)-d) for different iterations for the microthruster unit on a logarithmic scale. Current grid points are shown in red and the next grid points in green.

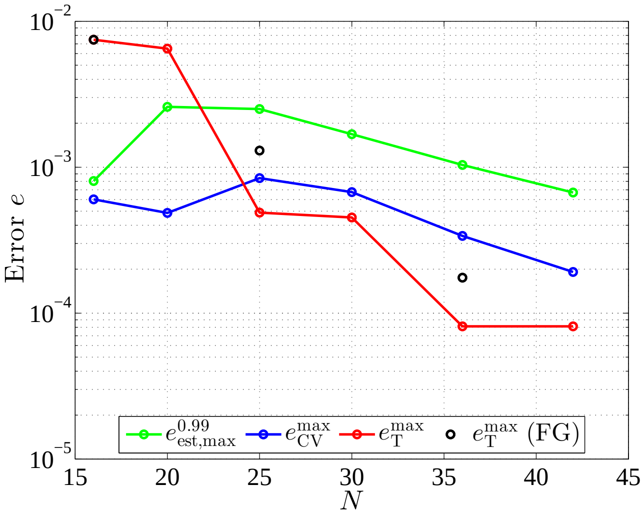


Figure 4.23: Predicted upper bound, maximum error indicator and maximum error obtained by the proposed method and by full grids (FG) for the microthruster unit.

Method Selection (Irregular Grid)

In the next two sections, we consider an irregular grid for the Black-Box method. For this, we have an initial grid with $N = 15$ points obtained by Latin hypercube sampling which is shown in Figure 4.24(a). We reduce the corresponding systems as explained above. As we have an irregular grid, we interpolate the reduced systems using radial basis functions, see Appendix A.3.1. We choose between linear (LH), cubic (CB) and multiquadratic with parameter $c = 0.1$ (MQ) basis functions and between the real (RE) or nonsingular (NO) matrix manifold for the square matrices. This gives a set of six candidate methods. For selecting the most accurate method, we apply Algorithm 4 where we use leave-one-out cross-validation for assigning the error indicators. The estimated error measures using the error indicators J_μ^{CV} and $J_{\mathrm{RMSE}}^{\mathrm{CV}}$ are calculated with formula (4.23) and (4.24). The true error measures J_μ and J_{RMSE} are approximated using the test grid $\mathcal{P}_{\mathrm{test}}^{\mathrm{selec}}$ with formula (4.21) and (4.22). The estimated and the true error measures are shown in Table 4.6.

Table 4.6: Comparison between the mean and the root mean square of the error indicators at the grid points and of the errors of a test grid for different interpolation methods ordered from small (1) to large (6).

	LH (RE)	LH (NO)	CB (RE)	CB (NO)	MQ (RE)	MQ (NO)
J_μ^{CV}	0.0423 (5)	0.0451 (6)	0.0013 (1)	0.0134 (2)	0.0339 (3)	0.0364 (4)
J_μ	0.0192 (5)	0.0221 (6)	0.0013 (1)	0.0082 (2)	0.0149 (3)	0.0177 (4)
$J_{\mathrm{RMSE}}^{\mathrm{CV}}$	0.0808 (6)	0.0755 (5)	0.0021 (1)	0.0281 (2)	0.0648 (4)	0.0613 (3)
J_{RMSE}	0.0523 (5)	0.0530 (6)	0.0186 (1)	0.0377 (2)	0.0433 (3)	0.0478 (4)

The candidate method with the cubic basis function and the real matrix manifold is identified as the method with smallest error measure, both for the mean and for the root mean square error which is confirmed by the true error measures. We will use this method for the refinement procedure in the next section. One can also see the estimated error measures predict the correct order of the candidate methods for the mean error. For the root mean square error, the order of the interpolation methods is obtained correctly, the order of the manifold is mixed up as the errors of the different manifolds are very similar.

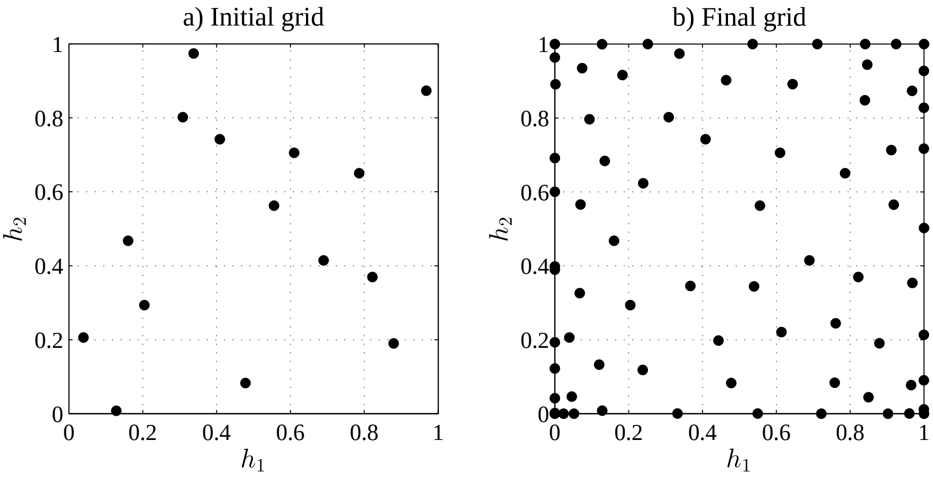


Figure 4.24: Grid for a) iteration $n_{\text{iter}} = 0$ and b) iteration $n_{\text{iter}} = 30$ using the Kriging surrogate.

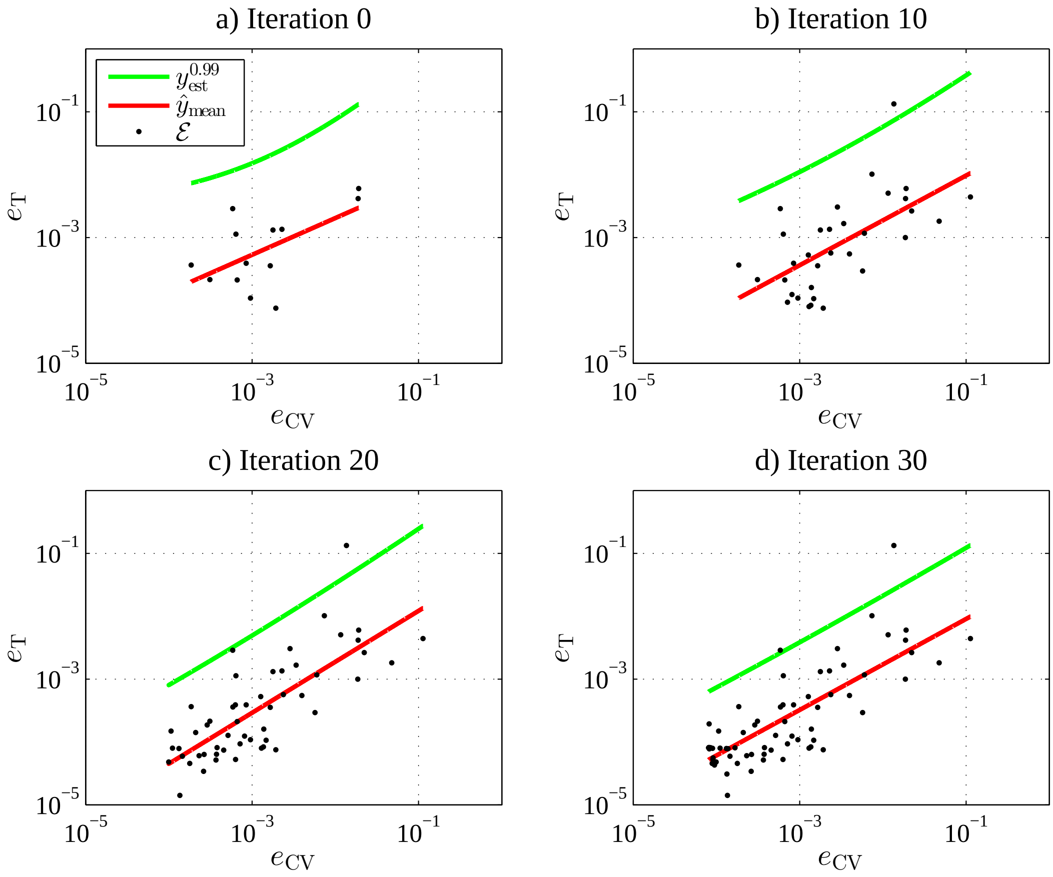


Figure 4.25: Error pool, regression line and the predicted error with 99% probability for different iterations with the RBF surrogate for the microthruster unit.

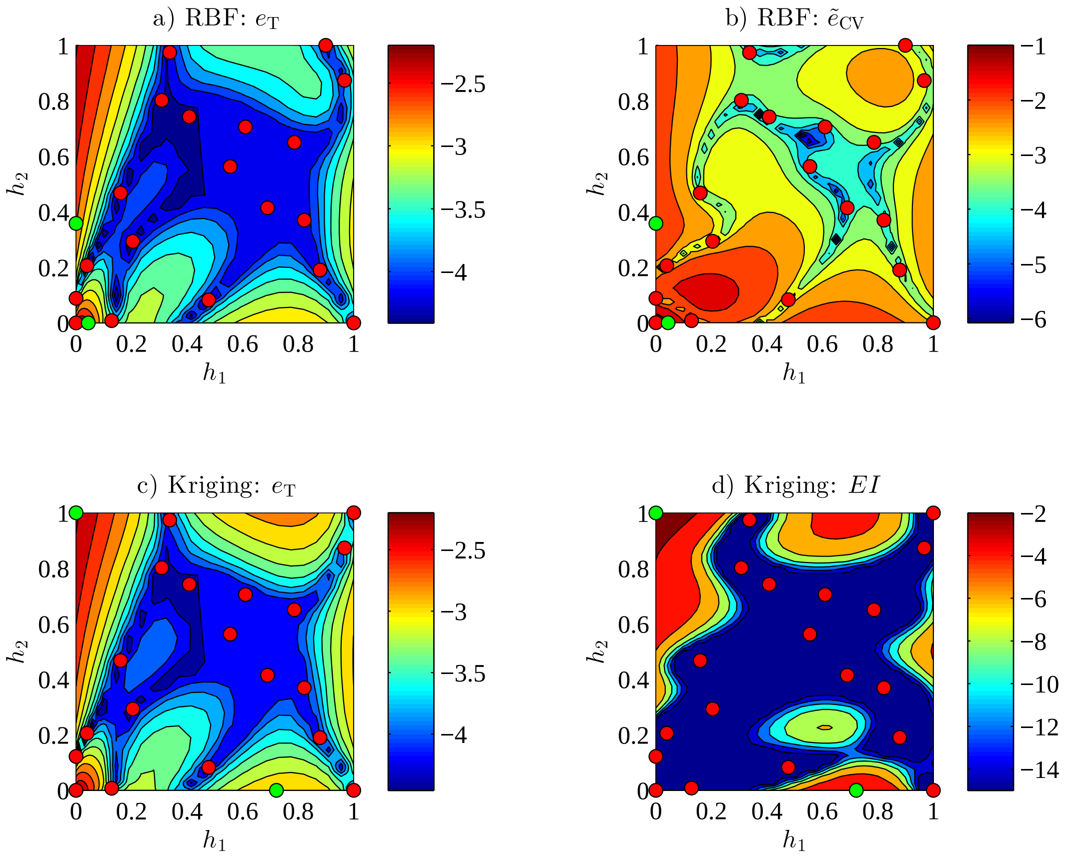


Figure 4.26: True error (left) and the corresponding surrogate model of the error indicator (right) for the RBF surrogate (top) and Kriging surrogate (bottom) for iteration $n_{\mathrm{iter}} = 2$ on a logarithmic scale. Current grid points are shown in red and the next $N_{\mathrm{new}} = 2$ grid points in green.

Model Refinement and Error Prediction (Irregular Grid)

We use the refinement method from Algorithm 7 with $n_{\mathrm{iter}}^{\max} = 30$ and $N_{\mathrm{new}} = 2$. Firstly, we apply a RBF surrogate for every iteration and use strategy RB3 for defining the constraints. In Figure 4.25 the error pool and the predicted error with 99% probability as a function of the error indicator are shown for iterations $n_{\mathrm{iter}} \in \{0, 10, 20, 30\}$. One can observe an approximately linear behavior on a logarithmic scale between the error indicator and the true error and the predicted error becomes more accurate when the error pool grows. Secondly, we apply a Kriging surrogate which is calculated using DACE [113] for all iterations and exploit it by maximizing the expected improvement. For both surrogate models, the error indicator and the true error is shown in Figure 4.26 for iteration $n_{\mathrm{iter}} = 2$. The latter is assessed in every iteration using the test grid $\mathcal{P}_{\mathrm{test}}^{\mathrm{refine}}$. The reader can verify the surrogate models identify regions with large error and insert points there. In addition, we can observe the expected improvement function is—such as described in Section 4.4.4—mostly very small with the restriction $EI(\mathbf{p}) \geq 10^{-14}$. In Figure 4.27 and 4.28 the maximum error indicator and the maximum error are plotted with a growing number of grid points for the RBF and Kriging surrogate, respectively. For both approaches the maximum error decreases. The Kriging surrogate, whose final grid is shown in Figure 4.24(b), reaches a smaller error because it balances exploitation and exploration of the domain. This means that new points are inserted at the origin—where the error indicator is large—and where we have a large uncertainty concerning the error indicator. For comparison, the error obtained by 100 random procedures is shown. For this, $N_{\text{new}} = 2$ randomly chosen points are inserted in every iteration. One can verify the proposed methods lead to smaller errors. One can also see the predicted upper bound with 99% probability. It grows in the first two iterations as it depends on the maximum error indicator which also grows. There is a large parameter-dependency around the origin and we have in this region so far too few grid points that we assign an error indicator to, see the initial grid in Figure 4.24(a). Starting with iteration $n_{\text{iter}} = 2$, there are enough grid points in the domain in order to capture the error behavior. Then, we have a tight bound for the error, especially for the case of the RBF surrogate as it focuses on diminishing the error indicator, whereas the Kriging surrogate also explores the domain. This is why we find the maximum error indicator and hence, the predicted upper bound to be larger for the Kriging surrogate than for the RBF surrogate.

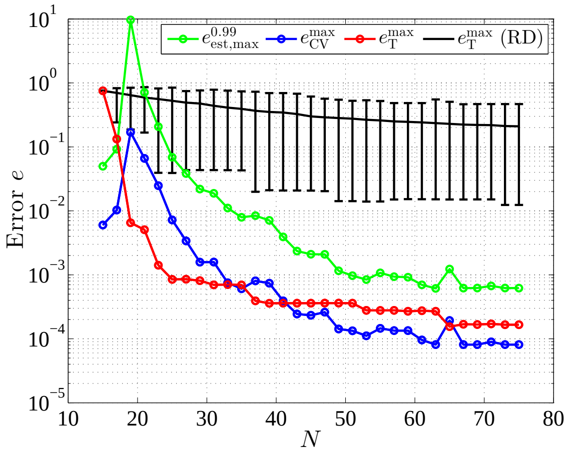


Figure 4.27: Predicted upper bound, maximum error indicator and maximum error obtained by the proposed method and by 100 random procedures (RD) (shown is the mean with the range of obtained values per iteration) using the RBF surrogate for the microthruster unit.

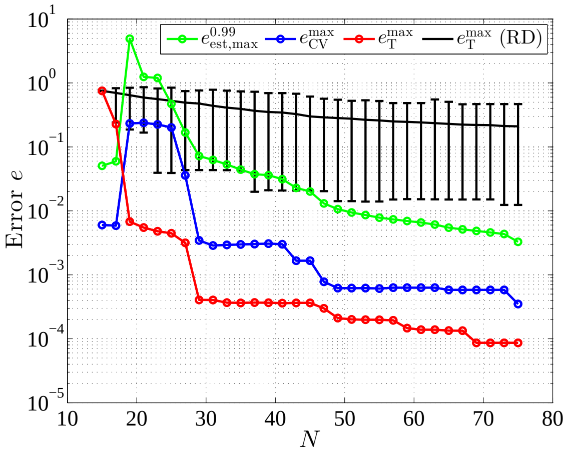


Figure 4.28: Predicted upper bound, maximum error indicator and maximum error obtained by the proposed method and by 100 random procedures (RD) (shown is the mean with the range of obtained values per iteration) using the Kriging surrogate for the microthruster unit.

4.7.3 Silicon Nitride Membrane ($\mathbf{n} = 60020$)

The next example is a silicon nitride membrane with integrated heater and sensing element which can be a part of a gas sensor, a microthruster unit or an optical filter [22, 1]. The system is shown in Figure 4.29.

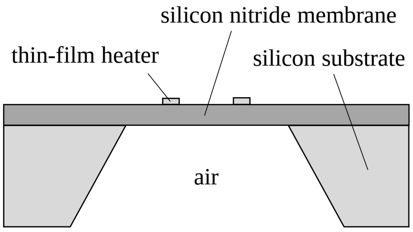


Figure 4.29: A silicon nitride membrane with integrated heater and sensing element.

The governing equation is the heat diffusion equation (4.49) with initial condition $T_0 = 273\text{K}$ with specific heat capacity $c_p = 400\text{Jkg}^{-1}\text{K}^{-1}$, density $\rho = 3000\text{kgm}^{-3}$ and thermal conductivity which varies in $\kappa \in [2,\ 5]\text{Wm}^{-1}\text{K}^{-1}$. The Dirichlet boundary condition is $T = 273\text{K}$ at the bottom of the computational domain. The condition at the top of the membrane is

$$Q^{\mathrm{B}} = h(T - T_{\mathrm{air}}), \tag{4.52}$$

where h is the heat transfer coefficient between the membrane and the ambient air with temperature T_{air} which is assumed to be zero. This parameter varies in $h \in [10,\ 12]\text{Wm}^{-2}\text{K}^{-1}$. Finite element discretization of the heat diffusion equation (4.49) and the convection boundary condition (4.52) leads to the parameter-dependent state-space system of order $n = 60020$

$$\begin{aligned} \mathbf{E}\dot{\mathbf{T}}(t) &= \overbrace{(\mathbf{A}_0 + \kappa\mathbf{A}_1 + h\mathbf{A}_2)}^{\mathbf{A}(\kappa,h)}\mathbf{T}(t) + \mathbf{b}Q(t) \\ y(t) &= \mathbf{c}^T\mathbf{T}(t), \end{aligned} \tag{4.53}$$

where $\mathbf{T}(t) \in \mathbb{R}^{60020}$ is the state vector containing the temperatures at the discretization nodes. Input of the system is the heat generation rate $Q(t)$ and output is the temperature change within the heater. As a test signal, the system is driven with the heating power of $Q(t) = 2.49\text{mW}$ in time interval $t \in [0, 0.02]\text{s}$ and the input is zero in $t \in]0.02, 0.04]\text{s}$. The integration is done using the implicit Euler method with the step size $\Delta t = 10^{-4}\text{s}$ and the accuracy of the output signal is measured with the $\mathcal{L}_2$-norm. The parameter domain is normalized to $\mathcal{D} = [0,\ 1] \times [0,\ 1]$ and we use an irregular grid. Therefore, we start with an initial grid obtained by Latin hypercube sampling with

$N = 6$ points. We obtain six high-order systems which we reduce to order $q = 6$ using the one-sided Krylov subspace method with expansion points $s_0 = 0$. The adjustment of the ROBs is done with the PS approach (3.13) and (3.25). The reference ROBs are chosen according to the non-weighted SVD approach (3.5) and (3.22).

Method Selection

For the interpolation process we choose between linear (LH), cubic (CB) and multiquadratic with parameter $c = 0.1$ (MQ) basis functions and between the real (RE) and nonsingular (NO) matrix manifold for the square matrices. This delivers a set of six candidate methods. For selecting the most accurate method we apply Algorithm 4 where we use leave-one-out cross-validation for assigning the error indicators. The estimated error measures using the error indicators J_{μ}^{CV} and $J_{\text{RMSE}}^{\text{CV}}$ are calculated with formula (4.23) and (4.24). The true error measures J_{μ} and J_{RMSE} are approximated using the test grid $\mathcal{P}_{\text{test}} = \{0, 1/40, \dots, 1\} \times \{0, 1/40, \dots, 1\}$ with formula (4.21) and (4.22), respectively. The estimated and the true error measures are shown in Table 4.7. The candidate method with the cubic basis function and the real matrix manifold is identified as the method with smallest error measure, both for the mean and for the root mean square error which is confirmed by the true error measures. We use this method for the refinement procedure in the next section. One can also see that the estimated error measures predict the correct order of the interpolation methods both for the mean and the root mean square error using only a grid with $N = 6$ points. The order of the manifold is mixed up as the errors of the different manifolds are very similar.

Table 4.7: Comparison between the mean and the root mean square of the error indicators at the grid points and of the real errors of a test grid for different interpolation methods ordered from small (1) to large (6).

	LH (RE)	LH (NO)	CB (RE)	CB (NO)	MQ (RE)	MQ (NO)
J_{μ}^{CV}	0.0694 (5)	0.0764 (6)	0.0036 (1)	0.0230 (2)	0.0591 (3)	0.0664 (4)
J_{μ}	0.0221 (6)	0.0206 (5)	0.0006 (1)	0.0035 (2)	0.0177 (4)	0.0161 (3)
$J_{\text{RMSE}}^{\text{CV}}$	0.0871 (5)	0.0906 (6)	0.0042 (1)	0.0260 (2)	0.0751 (3)	0.0795 (4)
J_{RMSE}	0.0340 (6)	0.0322 (5)	0.0009 (1)	0.0052 (2)	0.0266 (4)	0.0252 (3)

Model Refinement and Error Prediction

We perform the refinement procedure from Algorithm 7 with $n_{\text{iter}}^{\text{max}} = 24$ and $N_{\text{new}} = 2$. Firstly, we apply a RBF surrogate and use strategy RB3 for defining the constraints and secondly, we use the Kriging surrogate. In Figure (4.30) and (4.31) one can see the maximum error indicator and the maximum true error for a growing number of grid points for the RBF and Kriging surrogate, respectively. The true error is assessed in every iteration using the test grid $\mathcal{P}_{\text{test}} = \{0, 1/19, \dots, 1\} \times \{0, 1/19, \dots, 1\}$. For both approaches the maximum error decreases and the Kriging surrogate reaches a smaller value as it leads to a better exploration of the parameter domain. For comparison, the errors obtained by 100 random procedures are shown. For this, in every iteration $N_{\text{new}} = 2$ randomly chosen points are inserted. One can verify the proposed methods lead to smaller errors. In addition, one can see the predicted upper bound with 99% probability which provides a tight bound for the error for both the RBF and the Kriging surrogate. It can be used as a stopping criterion for the refinement.

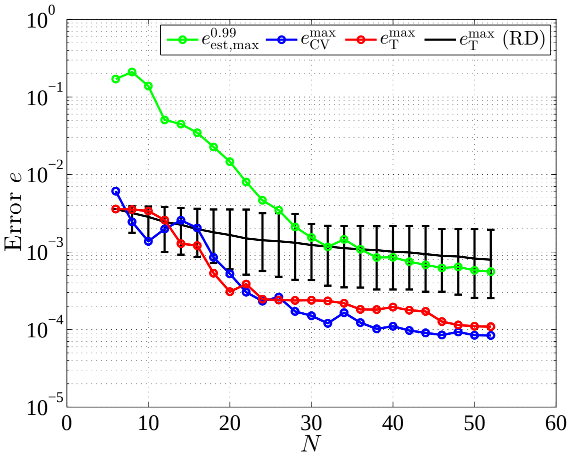


Figure 4.30: Predicted upper bound, maximum error indicator and maximum error obtained by the proposed method and by 100 random procedures (RD) (shown is the mean with the range of obtained values per iteration) using the RBF surrogate for the silicon nitride membrane.

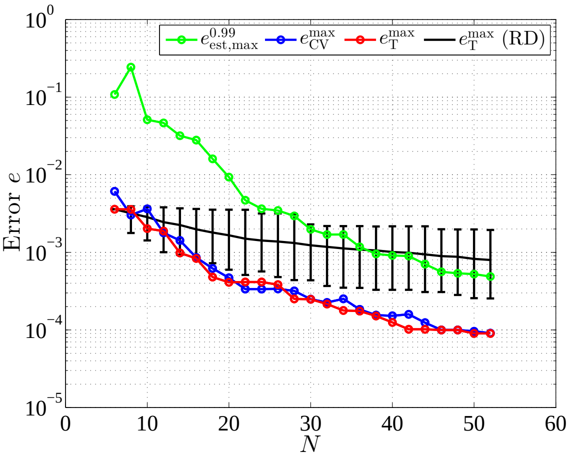


Figure 4.31: Predicted upper bound, maximum error indicator and maximum error obtained by the proposed method and by 100 random procedures (RD) (shown is the mean with the range of obtained values per iteration) using the Kriging surrogate for the silicon nitride membrane.

Part III

Further Results for Interpolating Reduced System Matrices

5 Stability-Preserving Interpolation of Reduced System Matrices

5.1 Motivation

Asymptotic stability according to Definition 2.1 is an important issue in signal and system theory which will be examined in this section for pMOR by matrix interpolation. It is well-known that a system obtained by the interpolation of local system matrices is not necessarily asymptotically stable even if the involved models fulfill this property. This is demonstrated in the following example.

Example: Consider the two asymptotically stable systems $\dot{\mathbf{x}}_1(t) = \mathbf{A}_1\mathbf{x}_1(t)$ and $\dot{\mathbf{x}}_2(t) = \mathbf{A}_2\mathbf{x}_2(t)$ with matrices

$$\mathbf{A}_1 = \begin{bmatrix} -1 & 4 \\ 0 & -1 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} -1 & 0 \\ 4 & -1 \end{bmatrix}. \tag{5.1}$$

For example, the interpolated matrix

$$\mathbf{A}_{\text{int}} = 0.5\mathbf{A}_1 + 0.5\mathbf{A}_2 = \begin{bmatrix} -1 & 2 \\ 2 & -1 \end{bmatrix} \tag{5.2}$$

has the spectrum {-3,1} and the resulting system $\dot{\mathbf{x}}(t) = \mathbf{A}_{\text{int}}\mathbf{x}(t)$ is, therefore, unstable according to criterion S3 of Theorem 2.1.

If we think back to the general framework from Section 3, we now have to realize that it does not necessarily lead to asymptotically stable interpolated systems even if the locally reduced systems are asymptotically stable. Some stability-preserving approaches are suggested in the literature, most of them being restricted to systems showing certain conditions concerning the structure of the large-scale model. In [11], stable interpolation on matrix manifolds is proposed for systems of ODE in second-order form. Stability-preserving matrix interpolation for dissipative systems is suggested in [55], for Port-Hamiltonian systems in [78] and for passive systems in [61, 58]. All these methods are efficient because they can make use of Corollary 2.2 as the structure of the considered high-order systems meets the associated requirements. For the general class of parameter-dependent LTI systems, some stability-preserving methods are proposed in [28, 27, 55]. The approaches have in common that they modify one of the two ROBs of the local systems in a way that they can make use of Corollary 2.2 to enforce stability for the interpolated systems. The approach from [55] requires the solution of high-order generalized Lyapunov equations, which is computationally very expensive. The method from [28, 27] determines an optimal stabilizing solution with regard to a quality function which is found by solving high-order optimization problems. For reducing the computational effort, the authors also propose low-order but suboptimal optimization problems.

In the following, a procedure for stability-preserving pMOR for the general class of parameter-dependent, linear time-invariant systems is presented, where all steps of the (stability-preserving) matrix interpolation are exclusively performed in low-dimensional vectors and matrices. The method starts with optimization steps similar to [28, 27]. However, it searches for the modified reduced order bases in the available low-order subspaces and hence, the optimization steps can be done very efficiently.

Preliminary studies for the proposed method were published in [77]. The final stability-preserving method is described in Sections 5.2–5.6. In Section 5.7, the method is integrated into the general framework from Section 3 and compared to the preliminary results and to an existing approach in the literature. In Section 5.8, numerical examples are given which demonstrate the effectiveness of the method.

5.2 Calculation of a Set of Asymptotically Stable Reduced Systems

The proposed method starts with a set of high-order systems $\mathcal{G} = \{\mathbf{G}_1, \ldots, \mathbf{G}_N\}$. These systems are reduced individually to the order $q \ll n$ with projection matrices $\mathbf{V}_1, \ldots, \mathbf{V}_N$ and $\mathbf{W}_1, \ldots, \mathbf{W}_N$ according to formula (3.2), and we assume that the reduced systems $\hat{\mathcal{G}} = \{\hat{\mathbf{G}}_1, \ldots, \hat{\mathbf{G}}_N\}$ are asymptotically stable. In order to ensure this stability, two general approaches are possible: (i) We can use stability-preserving reduction methods such as TBR or extended versions of Krylov subspace methods (For the latter, the Stability-Preserving, Adaptive Rational Krylov (SPARK) from [126] guarantees stability and the Iterative Rational Krylov Algorithm (IRKA) from [86] delivers asymptotically stable systems as long as it converges). (ii) For MOR methods which do not implicitly guarantee stability, one can easily check if the reduced systems are asymptotically stable by examining their eigenvalues according to criterion S3 from Theorem 2.1. If a system is then found to be unstable, it can be stabilized via post-processing, for example by modifying the left or right ROBs [29, 12].

To sum up, the result of this step is a set of N asymptotically stable reduced systems $\hat{\mathcal{G}} = \{\hat{\mathbf{G}}_1, \ldots, \hat{\mathbf{G}}_N\}$ where e.g. $\hat{\mathbf{G}}_i$ is

$$\hat{\mathbf{G}}_i : \begin{cases} \hat{\mathbf{E}}_i \dot{\hat{\mathbf{x}}}_i(t) = \hat{\mathbf{A}}_i \hat{\mathbf{x}}_i(t) + \hat{\mathbf{B}}_i \mathbf{u}(t) \\ \hat{\mathbf{y}}_i(t) = \hat{\mathbf{C}}_i \hat{\mathbf{x}}_i(t) \end{cases} \tag{5.3}$$

with eigenvalues of the pencil $(\hat{\mathbf{A}}_i, \hat{\mathbf{E}}_i)$ lying in the open left half of the complex plane.

5.3 Adjustment of the Right Reduced Order Bases

The reduced systems of the set $\hat{\mathcal{G}}$ have different bases $\mathbf{V}_1, \ldots, \mathbf{V}_N$ and they need to be adjusted to the basis of the reference subspace $\mathbf{V}_0$ which can be calculated such as described in Section 3.4.1. For this, the right ROBs are transformed with matrices $\mathbf{T}_i \in \mathbb{R}^{q\times q}$ to $\tilde{\mathbf{V}}_i = \mathbf{V}_i\mathbf{T}_i$ so that a quality function is minimized. A possible quality function is again the MAC between the local basis $\tilde{\mathbf{V}}_i$ and the reference basis $\mathbf{V}_0$ which can be written according to (3.18):

$$J_{\mathrm{MAC},\mathbf{V},i}(\mathbf{T}_i) = \|\mathbf{V}_0^T \tilde{\mathbf{V}}_i - \mathbf{I}_q\|_F = \|\mathbf{V}_0^T \mathbf{V}_i \mathbf{T}_i - \mathbf{I}_q\|_F. \tag{5.4}$$

The solution with the minimum value of the cost function, which is zero, is again

$$\mathbf{T}_i = \arg \min_{\mathbf{T}_i \in \mathbb{R}^{q\times q}} J_{\mathrm{MAC},\mathbf{V},i}(\mathbf{T}_i) = (\mathbf{V}_0^T \mathbf{V}_i)^{-1}. \tag{5.5}$$

A second possible quality function is like (3.7) the distance (DS) between the local basis $\tilde{\mathbf{V}}_i$ and the reference basis $\mathbf{V}_0$ which is measured in the Frobenius norm:

$$J_{\mathrm{DS},\mathbf{V},i}(\mathbf{T}_i) = \|\tilde{\mathbf{V}}_i - \mathbf{V}_0\|_F = \|\mathbf{V}_i\mathbf{T}_i - \mathbf{V}_0\|_F. \tag{5.6}$$

The solution with minimum value of the cost function is again obtained by using the pseudoinverse

$$\mathbf{T}_i = \arg \min_{\mathbf{T}_i \in \mathbb{R}^{q\times q}} J_{\mathrm{DS},\mathbf{V},i}(\mathbf{T}_i) = \mathbf{V}_i^{(l)+} \mathbf{V}_0. \tag{5.7}$$

To sum up, the matrices $\mathbf{T}_1, \ldots, \mathbf{T}_N$ are calculated such as described in the framework from Section 3.4.1 in order to adjust the right ROBs for accuracy reasons.

5.4 Stabilizing Procedure Using Semidefinite Programming

In this section, a stabilization method is described for the interpolation of reduced matrices. The demands that we impose on the method are—besides stability preservation—accuracy and the operation on low-order systems. The main idea is the following: Starting from the transformed bases $\tilde{\mathbf{W}}_i = \mathbf{W}_i\mathbf{M}_i$ for $i \in \{1, \dots, N\}$ with $\mathbf{M}_i = (\mathbf{W}_0^T\mathbf{W}_i)^{-1}$ using the MAC approach or with $\mathbf{M}_i = \mathbf{W}_i^{(l)+}\mathbf{W}_0$ using the DS approach, we minimally modify the new left ROBs $\tilde{\mathbf{W}}_i$ until stability is guaranteed.

5.4.1 Feasibility Problem

An optimization problem is formulated in which a convex objective function $J_{\mathbf{W}}$ is minimized subject to the constraint that the interpolated system $\tilde{\mathbf{G}}(\mathbf{p})$ is asymptotically stable. According to criterion S1 from Theorem 2.1, this constraint is equivalent to the existence of a Lyapunov function $V(\tilde{\mathbf{x}}(t), \mathbf{p})$. Hence, we obtain the optimization problem

$$\min J_{\mathbf{W}} \quad \text{s.t.} \quad \exists\ V(\tilde{\mathbf{x}}(t), \mathbf{p})\ \forall\ \mathbf{p} \in \mathcal{D}. \tag{5.8}$$

According to Corollary 2.2, the existence of a Lyapunov function is guaranteed for the superposition of strictly dissipative systems when nonnegative weighting functions are applied. Hence, the optimization problem (5.8) is rewritten as N problems, one for each grid point. We introduce for each $i \in \{1, \dots, N\}$ the basis $\tilde{\mathbf{W}}_i$ as optimization variable and we aim to calculate this variable so that the system $\tilde{\mathbf{G}}_i$ is strictly dissipative and the objective function $J_{\mathbf{W},i}$ reaches its minimum value:

$$\begin{aligned} &\min_{\tilde{\mathbf{W}}_i \in \mathbb{R}^{n\times q}} J_{\mathbf{W},i}(\tilde{\mathbf{W}}_i) \\ &\text{s.t. } \tilde{\mathbf{W}}_i^T\mathbf{A}_i\tilde{\mathbf{V}}_i + (\tilde{\mathbf{W}}_i^T\mathbf{A}_i\tilde{\mathbf{V}}_i)^T < \mathbf{0} \\ &\qquad\qquad\qquad\qquad \tilde{\mathbf{W}}_i^T\mathbf{E}_i\tilde{\mathbf{V}}_i > \mathbf{0}. \end{aligned} \tag{5.9}$$

Unfortunately, the matrices $\tilde{\mathbf{W}}_i$ are of high-order and hence, the optimization problems are intractable. As a remedy, the new bases $\tilde{\mathbf{W}}_i$ are searched in the available subspaces $\mathcal{W}_i = \text{span}\{\mathbf{W}_i\}$. For this, the matrices $\mathbf{M}_i \in \mathbb{R}^{q\times q}$ are introduced as new variables which provide the new bases $\tilde{\mathbf{W}}_i = \mathbf{W}_i\mathbf{M}_i$ and the optimization problems

$$\min_{\mathbf{M}_i \in \mathbb{R}^{q\times q}} J_{\mathbf{W},i}(\mathbf{M}_i) \tag{5.10}$$

$$\text{s.t. } \mathbf{M}_i^T\hat{\mathbf{A}}_i\mathbf{T}_i + (\mathbf{M}_i^T\hat{\mathbf{A}}_i\mathbf{T}_i)^T < \mathbf{0} \tag{5.11}$$

$$\mathbf{M}_i^T\hat{\mathbf{E}}_i\mathbf{T}_i > \mathbf{0} \tag{5.12}$$

which only contain low-order matrices. For solving these problems efficiently using interior point methods such as described e.g. in [159], they are transformed into semidefinite programs. These kinds of optimization problems deal with symmetric positive or negative (semi-)definite matrices as variables. To get to this form, the change of variable

$$\mathbf{M}_i = \mathbf{P}_i \hat{\mathbf{E}}_i \mathbf{T}_i \tag{5.13}$$

is introduced, where the new variables $\mathbf{P}_i \in \mathbb{S}_q$ are symmetric matrices [155]. Then, formula (5.13) is inserted into (5.12) which gives the necessary constraint on the definiteness of the optimization variables

$$\begin{aligned} \mathbf{M}_i^T \hat{\mathbf{E}}_i \mathbf{T}_i = \mathbf{T}_i^T \hat{\mathbf{E}}_i^T \mathbf{P}_i \hat{\mathbf{E}}_i \mathbf{T}_i = \left(\hat{\mathbf{E}}_i \mathbf{T}_i\right)^T \mathbf{P}_i \left(\hat{\mathbf{E}}_i \mathbf{T}_i\right) > \mathbf{0} \\ \Rightarrow \mathbf{P}_i > \mathbf{0}. \end{aligned} \tag{5.14}$$

For constraint (5.11) it follows

$$\begin{aligned} \mathbf{M}_i^T \hat{\mathbf{A}}_i \mathbf{T}_i + (\mathbf{M}_i^T \hat{\mathbf{A}}_i \mathbf{T}_i)^T &= \mathbf{T}_i^T \hat{\mathbf{E}}_i^T \mathbf{P}_i \hat{\mathbf{A}}_i \mathbf{T}_i + \mathbf{T}_i^T \hat{\mathbf{A}}_i^T \mathbf{P}_i \hat{\mathbf{E}}_i \mathbf{T}_i \\ &= \mathbf{T}_i^T \left(\hat{\mathbf{E}}_i^T \mathbf{P}_i \hat{\mathbf{A}}_i + \hat{\mathbf{A}}_i^T \mathbf{P}_i \hat{\mathbf{E}}_i\right) \mathbf{T}_i < \mathbf{0} \\ &\Rightarrow \hat{\mathbf{E}}_i^T \mathbf{P}_i \hat{\mathbf{A}}_i + \hat{\mathbf{A}}_i^T \mathbf{P}_i \hat{\mathbf{E}}_i < \mathbf{0}. \end{aligned} \tag{5.15}$$

Finally, the new optimization problems are

$$\begin{aligned} &\min_{\mathbf{P}_i \in \mathbb{S}_q} J_{\mathbf{W},i}(\mathbf{P}_i) \\ \text{s.t.} \quad &\hat{\mathbf{E}}_i^T \mathbf{P}_i \hat{\mathbf{A}}_i + \hat{\mathbf{A}}_i^T \mathbf{P}_i \hat{\mathbf{E}}_i < \mathbf{0} \\ &\mathbf{P}_i > \mathbf{0}. \end{aligned} \tag{5.16}$$

For assessing the solvability of problems (5.16), the theory of convex optimization tells us that only the constraints have to be examined and the quality function can be set to zero which is referred to as *feasibility problem* [82]. In this case, the constraints are linear matrix inequalities (LMIs) which are equivalent to the generalized Lyapunov inequality [32]. The problem (5.16) admits a solution because the eigenvalues of the pencil $(\hat{\mathbf{A}}_i, \hat{\mathbf{E}}_i)$ lie in the open left half of the complex plane as a result of Section 5.2.

5.4.2 Objective Functions

In the previous section the feasibility problem was introduced which guarantees stable matrix interpolation for all solutions satisfying the constraints. In this section, amongst these solutions the optimal stabilizing one is determined with regard to an objective function. The objective function, which needs to be minimized, is chosen in such a way that we obtain an accurate matrix interpolation. For this, we can use the objective functions described in Section 5.3 which adjust for each $i \in \{1, \dots, N\}$ the basis $\mathbf{W}_i$ to the basis of the reference subspace $\mathbf{W}_0$ by transforming it to $\tilde{\mathbf{W}}_i = \mathbf{W}_i\mathbf{M}_i$ with $\mathbf{M}_i \in \mathbb{R}^{q \times q}$. The basis of the reference subspace $\mathbf{W}_0$ can be calculated according to Section 3.4.2. Again, one can use the MAC criterion as an objective function:

$$J_{\mathrm{MAC},\mathbf{W},i}(\mathbf{M}_i) = \|\mathbf{W}_0^T \mathbf{W}_i \mathbf{M}_i - \mathbf{I}_q\|_F. \tag{5.17}$$

After inserting the new choice for $\mathbf{M}_i$ from (5.13), it follows

$$J_{\mathrm{MAC},\mathbf{W},i}(\mathbf{P}_i) = \|\mathbf{W}_0^T \mathbf{W}_i \mathbf{P}_i \hat{\mathbf{E}}_i \mathbf{T}_i - \mathbf{I}_q\|_F. \tag{5.18}$$

The matrices $\mathbf{X}_i = \mathbf{W}_0^T \mathbf{W}_i \in \mathbb{R}^{q \times q}$ and $\mathbf{Y}_i = \hat{\mathbf{E}}_i \mathbf{T}_i \in \mathbb{R}^{q \times q}$ can be precomputed and hence, the objective function which only comprises low-order matrices is obtained:

$$J_{\mathrm{MAC},\mathbf{W},i}(\mathbf{P}_i) = \|\mathbf{X}_i \mathbf{P}_i \mathbf{Y}_i - \mathbf{I}_q\|_F. \tag{5.19}$$

This objective function is convex as the composition of the Frobenius norm, which is a convex function, with the affine map $\mathbf{P}_i \mapsto \mathbf{X}_i \mathbf{P}_i \mathbf{Y}_i - \mathbf{I}_q$ is convex [33].

The second possible objective function is again the DS approach which describes the distance between the ROBs:

$$J_{\mathrm{DS},\mathbf{W},i}(\mathbf{M}_i) = \|\mathbf{W}_i \mathbf{M}_i - \mathbf{W}_0\|_F. \tag{5.20}$$

After inserting the new choice for $\mathbf{M}_i$ from (5.13), it follows

$$J_{\mathrm{DS},\mathbf{W},i}(\mathbf{P}_i) = \|\mathbf{W}_i \mathbf{P}_i \hat{\mathbf{E}}_i \mathbf{T}_i - \mathbf{W}_0\|_F. \tag{5.21}$$

Unfortunately, this objective function is of large order since the matrices $\mathbf{W}_i, \mathbf{W}_0$ have the size $n \times q$. As a remedy, we take the square of the Frobenius norm, rewrite it using the trace and obtain

$$\begin{aligned} J_{\mathrm{DS},\mathbf{W},i}(\mathbf{P}_i)^2 &= \|\mathbf{W}_i \mathbf{P}_i \hat{\mathbf{E}}_i \mathbf{T}_i - \mathbf{W}_0\|_F^2 \\ &= \mathrm{tr}(\mathbf{T}_i^T \hat{\mathbf{E}}_i^T \mathbf{P}_i \mathbf{W}_i^T \mathbf{W}_i \mathbf{P}_i \hat{\mathbf{E}}_i \mathbf{T}_i) - 2\mathrm{tr}(\mathbf{W}_0^T \mathbf{W}_i \mathbf{P}_i \hat{\mathbf{E}}_i \mathbf{T}_i) + \mathrm{tr}(\mathbf{W}_0^T \mathbf{W}_0) \\ &= \|(\mathbf{W}_i^T \mathbf{W}_i)^{1/2} \mathbf{P}_i \hat{\mathbf{E}}_i \mathbf{T}_i\|_F^2 - 2\mathrm{tr}(\mathbf{W}_0^T \mathbf{W}_i \mathbf{P}_i \hat{\mathbf{E}}_i \mathbf{T}_i) + \mathrm{tr}(\mathbf{W}_0^T \mathbf{W}_0). \end{aligned} \tag{5.22}$$

Again, the low-order matrices $\mathbf{X}_i$, $\mathbf{Y}_i$ and $\mathbf{Z}_i = (\mathbf{W}_i^T \mathbf{W}_i)^{1/2} \in \mathbb{R}^{q \times q}$ can be precomputed. When the third, constant term is left-out, the final objective function is obtained:

$$J_{\mathrm{DS},\mathbf{W},i}(\mathbf{P}_i) = \|\mathbf{Z}_i \mathbf{P}_i \mathbf{Y}_i\|_F^2 - 2\mathrm{tr}(\mathbf{X}_i \mathbf{P}_i \mathbf{Y}_i). \tag{5.23}$$

The first term is convex because the composition of the square which is a convex, nondecreasing function for nonnegative arguments and the convex Frobenius norm is convex. The whole quality function is convex as the sum of the first convex function and the trace which is a linear function is also convex [33].

5.4.3 Algorithm

The proposed method modifies the left ROBs for an accurate interpolation while making the local reduced models' representations strictly dissipative and hence, guaranteeing stability of any interpolated reduced model. For this, the objective function (5.19) or (5.23) is minimized for every grid point subject to the constraint (5.14) and (5.15). As the objective functions are convex and the constraints are linear, a convex optimization problem is obtained which is referred to as STability Algorithm Based on Linear matrix inequalitiEs (STABLE) and which is given in Algorithm 8. As STABLE is a low-order optimization problem, it can easily be implemented and solved e.g. in MATLAB using convex optimization solver packages which rely on semidefinite programming such as CVX [83, 82] or YALMIP [108]. Then, STABLE delivers the matrices $\mathbf{M}_1, \ldots, \mathbf{M}_N$.

Algorithm 8 STABLE

```
Input: $N$ matrices $\hat{\mathbf{E}}_i, \hat{\mathbf{A}}_i, \mathbf{T}_i, \mathbf{W}_i$, reference basis $\mathbf{W}_0$
Output: $N$ matrices $\mathbf{M}_i$
for $i = 1$ to $N$ do
    Compute $\mathbf{X}_i = \mathbf{W}_0^T \mathbf{W}_i$, $\mathbf{Y}_i = \hat{\mathbf{E}}_i \mathbf{T}_i$, $\mathbf{Z}_i = (\mathbf{W}_i^T \mathbf{W}_i)^{1/2}$
    Solve the convex optimization problem:
        $\min_{\mathbf{P}_i \in \mathbb{S}_q} \Big( \|\mathbf{Z}_i \mathbf{P}_i \mathbf{Y}_i\|_F^2 - 2\mathrm{tr}(\mathbf{X}_i \mathbf{P}_i \mathbf{Y}_i) \Big)$
        or
        $\min_{\mathbf{P}_i \in \mathbb{S}_q} \|\mathbf{X}_i \mathbf{P}_i \mathbf{Y}_i - \mathbf{I}_q\|_F$
        s.t. $\hat{\mathbf{E}}_i^T \mathbf{P}_i \hat{\mathbf{A}}_i + \hat{\mathbf{A}}_i^T \mathbf{P}_i \hat{\mathbf{E}}_i < \mathbf{0}$
             $\mathbf{P}_i > \mathbf{0}$
    Compute $\mathbf{M}_i = \mathbf{P}_i \hat{\mathbf{E}}_i \mathbf{T}_i$
end for
```

5.5 A Dual Stabilizing Procedure

In Section 3.4.2 the notion of duality between the right and left ROBs was introduced. Hence, in contrast to STABLE we can as well calculate $\mathbf{M}_i$ for accuracy reasons with $\mathbf{M}_i = (\mathbf{W}_0^T \mathbf{W}_i)^{-1}$ for the MAC approach or $\mathbf{M}_i = \mathbf{W}_i^{(l)+} \mathbf{W}_0$ for the DS approach with $i \in \{1, \dots, N\}$. Then, the stabilizing procedure minimally modifies the right ROBs $\tilde{\mathbf{V}}_1, \dots, \tilde{\mathbf{V}}_N$ compared to the approach of the framework from Section 3.4.1. For this, in analogy to the previous section we consider the transformation matrices $\mathbf{T}_i$ as variables for the N low-order optimization problems

$$\begin{aligned} \min_{\mathbf{T}_i \in \mathbb{R}^{q \times q}} & J_{\mathbf{V},i}(\mathbf{T}_i) \\ \text{s.t. } & \mathbf{M}_i^T \hat{\mathbf{A}}_i \mathbf{T}_i + (\mathbf{M}_i^T \hat{\mathbf{A}}_i \mathbf{T}_i)^T < \mathbf{0} \\ & \mathbf{M}_i^T \hat{\mathbf{E}}_i \mathbf{T}_i > \mathbf{0} \end{aligned} \tag{5.24}$$

where $J_{\mathbf{V},i}$ is a convex quality function which is minimized. These optimization problems are again supposed to be transformed into semidefinite programs. For this, we introduce the change of variable

$$\mathbf{T}_i = \mathbf{P}_i \hat{\mathbf{E}}_i^T \mathbf{M}_i \tag{5.25}$$

where the new variables $\mathbf{P}_i \in \mathbb{S}_q$ are symmetric matrices [155] which lead to the following optimization problems

$$\begin{aligned} \min_{\mathbf{P}_i \in \mathbb{S}_q} & J_{\mathbf{V},i}(\mathbf{P}_i) \\ \text{s.t. } & \hat{\mathbf{A}}_i \mathbf{P}_i \hat{\mathbf{E}}_i^T + \hat{\mathbf{E}}_i \mathbf{P}_i \hat{\mathbf{A}}_i^T < \mathbf{0} \\ & \mathbf{P}_i > \mathbf{0}. \end{aligned} \tag{5.26}$$

In this case the constraints are again LMIs which are now equivalent to the dual generalized Lyapunov inequality [32]. Amongst all solutions satisfying the constraints, the optimal stabilizing solution, which is the one minimizing a quality function for the sake of accurate matrix interpolation, is supposed to be determined. As an objective function one can again use the MAC approach

$$J_{\mathrm{MAC},\mathbf{V},i}(\mathbf{T}_i) = \|\mathbf{V}_0^T \mathbf{V}_i \mathbf{T}_i - \mathbf{I}_q\|_F \tag{5.27}$$

which can be written with (5.25) to

$$J_{\mathrm{MAC},\mathbf{V},i}(\mathbf{P}_i) = \|\mathbf{V}_0^T \mathbf{V}_i \mathbf{P}_i \hat{\mathbf{E}}_i^T \mathbf{M}_i - \mathbf{I}_q\|_F. \tag{5.28}$$

One can also take the DS approach with

$$J_{\mathrm{DS},\mathbf{V},i}(\mathbf{T}_i) = \|\mathbf{V}_i\mathbf{T}_i - \mathbf{V}_0\|_F \tag{5.29}$$

which can be rewritten with formula (5.25) to

$$J_{\mathrm{DS},\mathbf{V},i}(\mathbf{P}_i) = \|(\mathbf{V}_i^T\mathbf{V}_i)^{1/2}\mathbf{P}_i\hat{\mathbf{E}}_i^T\mathbf{M}_i\|_F^2 - 2\mathrm{tr}(\mathbf{V}_0^T\mathbf{V}_i\mathbf{P}_i\hat{\mathbf{E}}_i^T\mathbf{M}_i). \tag{5.30}$$

With $\mathbf{X}_i = \mathbf{V}_0^T\mathbf{V}_i$, $\mathbf{Y}_i = \hat{\mathbf{E}}_i^T\mathbf{M}_i$, $\mathbf{Z}_i = (\mathbf{V}_i^T\mathbf{V}_i)^{1/2}$ one obtains the dual stability-preserving method d-STABLE which is given in Algorithm 9 and which is also a convex, low-order optimization problem.

Algorithm 9 d-STABLE

```
Input: N matrices $\hat{\mathbf{E}}_i, \hat{\mathbf{A}}_i, \mathbf{M}_i, \mathbf{V}_i$, reference basis $\mathbf{V}_0$
Output: N matrices $\mathbf{T}_i$
for $i = 1$ to $N$ do
    Compute $\mathbf{X}_i = \mathbf{V}_0^T\mathbf{V}_i$, $\mathbf{Y}_i = \hat{\mathbf{E}}_i^T\mathbf{M}_i$, $\mathbf{Z}_i = (\mathbf{V}_i^T\mathbf{V}_i)^{1/2}$
    Solve the convex optimization problem:
        $\min_{\mathbf{P}_i \in \mathcal{S}_q} \left(\|\mathbf{Z}_i\mathbf{P}_i\mathbf{Y}_i\|_F^2 - 2\mathrm{tr}(\mathbf{X}_i\mathbf{P}_i\mathbf{Y}_i)\right)$
        or
        $\min_{\mathbf{P}_i \in \mathcal{S}_q} \|\mathbf{X}_i\mathbf{P}_i\mathbf{Y}_i - \mathbf{I}_q\|_F$
        s.t. $\hat{\mathbf{A}}_i\mathbf{P}_i\hat{\mathbf{E}}_i^T + \hat{\mathbf{E}}_i\mathbf{P}_i\hat{\mathbf{A}}_i^T < \mathbf{0}$
             $\mathbf{P}_i > \mathbf{0}$
    Compute $\mathbf{T}_i = \mathbf{P}_i\hat{\mathbf{E}}_i^T\mathbf{M}_i$
end for
```

5.6 Online Interpolation Process

Performing STABLE or d-STABLE for every grid point gives strictly dissipative reduced systems with $\tilde{\mathbf{E}}_i > \mathbf{0}$ and $\tilde{\mathbf{A}}_i + \tilde{\mathbf{A}}_i^T < \mathbf{0}$. According to Corollary 2.1 they have the Lyapunov function $V(\tilde{\mathbf{x}}_i(t), \mathbf{p}_i) = \tilde{\mathbf{x}}_i(t)^T\tilde{\mathbf{E}}_i\tilde{\mathbf{x}}_i(t)$. The resulting systems are

$$\tilde{\mathbf{G}}_i : \begin{cases} \overbrace{\mathbf{M}_i^T\hat{\mathbf{E}}_i\mathbf{T}_i}^{\tilde{\mathbf{E}}_i}\,\dot{\tilde{\mathbf{x}}}_i(t) = \overbrace{\mathbf{M}_i^T\hat{\mathbf{A}}_i\mathbf{T}_i}^{\tilde{\mathbf{A}}_i}\,\tilde{\mathbf{x}}_i(t) + \overbrace{\mathbf{M}_i^T\hat{\mathbf{B}}_i}^{\tilde{\mathbf{B}}_i}\,\mathbf{u}(t) \\ \tilde{\mathbf{y}}_i(t) = \underbrace{\hat{\mathbf{C}}_i\mathbf{T}_i}_{\tilde{\mathbf{C}}_i}\,\tilde{\mathbf{x}}_i(t). \end{cases} \tag{5.31}$$

In the online phase, i.e. when a reduced order model for the new parameter vector $\mathbf{p}_{\text{int}}$ is supposed to be calculated, this is simply done by the interpolation of the precomputed and stored matrices. According to Corollary 2.2 the interpolation has to take place with nonnegative weighting functions $\omega_i(\mathbf{p}_{\text{int}}) \in \mathbb{R}_0^+$ and at least one positive $\omega_i(\mathbf{p}_{\text{int}}) \in \mathbb{R}^+$. One example for such weighting functions are linear basis functions. Then, the new reduced system for the parameter vector $\mathbf{p}_{\text{int}}$ is given by

$$\tilde{\mathbf{G}}(\mathbf{p}_{\text{int}}) : \begin{cases} \tilde{\mathbf{E}}(\mathbf{p}_{\text{int}})\dot{\tilde{\mathbf{x}}}(t) = \tilde{\mathbf{A}}(\mathbf{p}_{\text{int}})\tilde{\mathbf{x}}(t) + \tilde{\mathbf{B}}(\mathbf{p}_{\text{int}})\mathbf{u}(t) \\ \tilde{\mathbf{y}}(t) = \tilde{\mathbf{C}}(\mathbf{p}_{\text{int}})\tilde{\mathbf{x}}(t), \end{cases} \tag{5.32}$$

where

$$\begin{aligned} \tilde{\mathbf{E}}(\mathbf{p}_{\text{int}}) &= \textstyle\sum_{i=1}^{N} \omega_i(\mathbf{p}_{\text{int}})\tilde{\mathbf{E}}_i, \\ \tilde{\mathbf{A}}(\mathbf{p}_{\text{int}}) &= \textstyle\sum_{i=1}^{N} \omega_i(\mathbf{p}_{\text{int}})\tilde{\mathbf{A}}_i, \\ \tilde{\mathbf{B}}(\mathbf{p}_{\text{int}}) &= \textstyle\sum_{i=1}^{N} \omega_i(\mathbf{p}_{\text{int}})\tilde{\mathbf{B}}_i, \\ \tilde{\mathbf{C}}(\mathbf{p}_{\text{int}}) &= \textstyle\sum_{i=1}^{N} \omega_i(\mathbf{p}_{\text{int}})\tilde{\mathbf{C}}_i. \end{aligned} \tag{5.33}$$

Following Corollary 2.2, the interpolated system possesses the Lyapunov function $V(\tilde{\mathbf{x}}(t), \mathbf{p}_{\text{int}}) = \tilde{\mathbf{x}}(t)^T \left(\sum_{i=1}^{N} \omega_i(\mathbf{p}_{\text{int}})\tilde{\mathbf{E}}_i \right) \tilde{\mathbf{x}}(t)$. The procedure implicitly interpolates the Lyapunov functions and hence, the interpolated system is asymptotically stable for the entire domain $\mathcal{D}$.

5.7 Classification of the Stability-Preserving Method

5.7.1 Integration into the General Framework

The proposed method can be formulated in the general framework from Section 3 and hence, the Black-Box method from Section 4 and extends it to stability preservation without any limiting conditions concerning the structure of the high-dimensional system. The stability-preserving procedure is shown in Figure 5.1. In the first step the parameter space is sampled and in the second step the local systems are reduced assuming a set of asymptotically stable reduced system. In the third step the ROBs are adjusted to a reference basis. Here, we have two possibilities: we can either use STABLE or d-STABLE. For STABLE, we first adjust the right ROBs with matrices $\mathbf{T}_i$ like the general framework and the algorithm determines matrices $\mathbf{M}_i$ which minimally modify the left ROBs compared to the general framework so that the stability constraints are satisfied. For d-STABLE, this is the other way round and we adjust the left ROBs with matrices $\mathbf{M}_i$ like the general framework and the algorithm determines matrices $\mathbf{T}_i$

which minimally modify the right ROBs compared to the general framework. For the interpolation procedure, the proposed method has a limited number of options as it is restricted to use the manifold of real matrices and to apply nonnegative weighting functions such as linear ones. In the online phase, the weighting functions are again evaluated and multiplied with the system matrices.

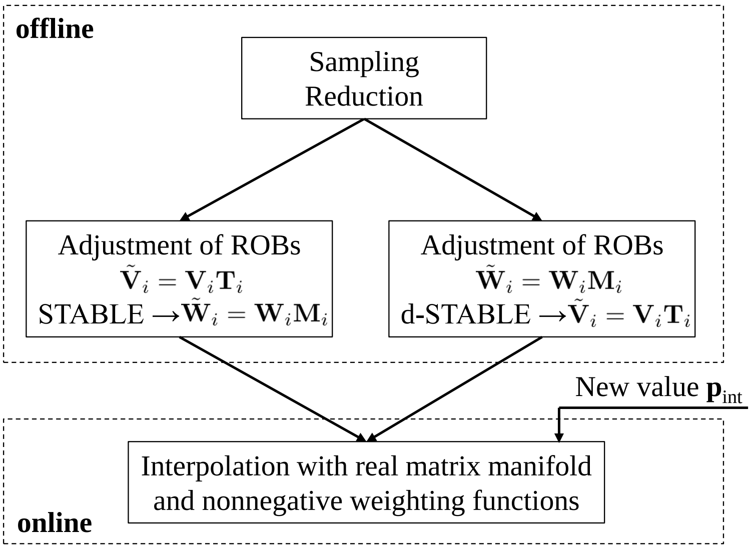

offline
Sampling
Reduction
Adjustment of ROBs
$\tilde{\mathbf{V}}_i = \mathbf{V}_i\mathbf{T}_i$
STABLE →$\tilde{\mathbf{W}}_i = \mathbf{W}_i\mathbf{M}_i$
Adjustment of ROBs
$\tilde{\mathbf{W}}_i = \mathbf{W}_i\mathbf{M}_i$
d-STABLE →$\tilde{\mathbf{V}}_i = \mathbf{V}_i\mathbf{T}_i$
New value $\mathbf{p}_{\text{int}}$
Interpolation with real matrix manifold
and nonnegative weighting functions
online

Figure 5.1: Structure of the stability-preserving method.

So far, it is still an open question if STABLE or d-STABLE leads to more accurate interpolated systems on domain $\mathcal{D}$ in the sense of an aggregated error measure such as the mean (4.19) or the root mean square (4.20). The best choice is problem-dependent and it can be determined using Algorithm 4. For this, cross-validation-based error indicators are calculated for both approaches in order to determine an estimate for the error measure. Then, the approach with the minimum error measure is chosen. This is demonstrated by means of an example in Section 5.8.2.

5.7.2 Comparison to the Literature

In the following, the proposed method is compared to preliminary results and to an existing approach in the literature.

Comparison to Preliminary Results

A first approach for stability preservation was published in [77]. In principle, this method resembles the explanations in Section 5.4.1, i.e. the locally reduced systems are transformed to a strictly dissipative form and hence, stability preservation is guaranteed for all solutions satisfying the Lyapunov inequality (5.14)–(5.15). However, it has so far been an open question which solution we should choose as—depending on this choice and on the examined system—the interpolation can become very inaccurate. The new method determines amongst these solutions automatically the optimal stabilizing one such as described in Section 5.4.2 and hence, we can expect an accurate interpolation. Due to this automatic selection procedure, the new method can easily be integrated into the Black-Box method. By contrast, the method in [77] requires the user to interfere which prohibits an automatic procedure. The new method is computationally more expensive than the one in [77] as LMIs need to be solved. Nevertheless, this approach is still very cheap as the optimization problems only contain low-order matrices.

Comparison to an Existing Approach

A method is proposed in [55] which guarantees stable interpolation of reduced system matrices for the special case of strictly dissipative high-order systems. It applies for each $i \in \{1, \dots, N\}$ the one-sided reduction $\mathbf{W}_i = \mathbf{V}_i$, adjusts the right ROB with $\mathbf{T}_i = (\mathbf{V}_0^T \mathbf{V}_i)^{-1}$ and chooses for the left ROB the matrices $\mathbf{M}_i = \mathbf{T}_i$.

Proposition 5.1. *STABLE comprises the method from [55] as a special case for strictly dissipative systems and extends it to the general case of LTI systems.*

Proof. For high-order system $\mathbf{G}_i$ with $\mathbf{E}_i > \mathbf{0}$ and $\mathbf{A}_i + \mathbf{A}_i^T < \mathbf{0}$, perform a one-sided reduction method with $\mathbf{W}_i = \mathbf{V}_i$ which leads to a reduced system fulfilling $\hat{\mathbf{E}}_i > \mathbf{0}$ and $\hat{\mathbf{A}}_i + \hat{\mathbf{A}}_i^T < \mathbf{0}$ [41]. Take for the adjustment of the right ROBs, just like in [55], the matrix $\mathbf{T}_i = (\mathbf{V}_0^T \mathbf{V}_i)^{-1}$ which corresponds to the MAC approach (5.5). Hence, we are in the branch using STABLE in Figure 5.1. For adjusting the left ROBs, we consider the objective function of the MAC approach (5.18)

$$J_{\mathrm{MAC},\mathbf{W},i}(\mathbf{P}_i) = \|\mathbf{W}_0^T \mathbf{W}_i \mathbf{P}_i \hat{\mathbf{E}}_i \mathbf{T}_i - \mathbf{I}_q\|_F = \|\mathbf{V}_0^T \mathbf{V}_i \mathbf{P}_i \hat{\mathbf{E}}_i \mathbf{T}_i - \mathbf{I}_q\|_F = \|\mathbf{T}_i^{-1} \mathbf{P}_i \hat{\mathbf{E}}_i \mathbf{T}_i - \mathbf{I}_q\|_F$$

which reaches its minimum value, $J_{\mathrm{MAC},\mathbf{W},i} = 0$, for $\mathbf{P}_i = \hat{\mathbf{E}}^{-1}$. The same holds true for the objective function of the DS approach (5.21). The optimal solution belongs to the feasible set because it fulfills the first constraint (5.14) owing to

$$\mathbf{P}_i = \hat{\mathbf{E}}_i^{-1} > \mathbf{0}$$

and the second constraint (5.15) because of

$$\hat{\mathbf{E}}_i^T \mathbf{P}_i \hat{\mathbf{A}}_i + \hat{\mathbf{A}}_i^T \mathbf{P}_i \hat{\mathbf{E}}_i = \hat{\mathbf{E}}_i^T \hat{\mathbf{E}}_i^{-1} \hat{\mathbf{A}}_i + \hat{\mathbf{A}}_i^T \hat{\mathbf{E}}_i^{-1} \hat{\mathbf{E}}_i = \hat{\mathbf{A}}_i + \hat{\mathbf{A}}_i^T < \mathbf{0}.$$

Hence, STABLE gives $\mathbf{M}_i = \mathbf{P}_i \hat{\mathbf{E}}_i \mathbf{T}_i = \hat{\mathbf{E}}_i^{-1} \hat{\mathbf{E}}_i \mathbf{T}_i = \mathbf{T}_i$ which is the result from [55]. ■

5.8 Numerical Examples

5.8.1 Academic Example

Consider a high-order system $\mathbf{E}(p)\dot{\mathbf{x}}(t) = \mathbf{A}(p)\mathbf{x}(t)$ with $\mathbf{x}(t) \in \mathbb{R}^n$. The domain $p \in \mathcal{D}$ with $\mathcal{D} = [0,\ 1]$ is sampled for two parameter values $p_1 = 0, p_2 = 1$. The resulting systems $\mathbf{E}_i\dot{\mathbf{x}}_i(t) = \mathbf{A}_i\mathbf{x}_i(t)$ with $i \in \{1,2\}$ are reduced to order $q = 2 \ll n$ using projection matrices $\mathbf{V}_i = \mathbf{V}_0 \in \mathbb{R}^{n\times 2}$ and $\mathbf{W}_i = \mathbf{W}_0 \in \mathbb{R}^{n\times 2}$ where the columns are an orthonormal basis. This is supposed to result according to formula (3.2) in the asymptotically stable systems $\hat{\mathbf{E}}_i\dot{\hat{\mathbf{x}}}_i(t) = \hat{\mathbf{A}}_i\hat{\mathbf{x}}_i(t)$ with $\hat{\mathbf{E}}_i = \mathbf{I}_2$ and

$$\hat{\mathbf{A}}_1 = \begin{bmatrix} -2 & 5 \\ 0 & -2 \end{bmatrix}, \quad \hat{\mathbf{A}}_2 = \begin{bmatrix} -2 & 0 \\ 5 & -2 \end{bmatrix}. \tag{5.34}$$

Firstly, it will be shown what happens without using STABLE. The respective values are labeled with WS. The transformation matrices are found to be $\mathbf{T}_i^{\mathrm{WS}} = \mathbf{I}_2, \mathbf{M}_i^{\mathrm{WS}} = \mathbf{I}_2$ according to formulas (3.16) or (3.8) and (3.24) or (3.23) which minimize the objective functions to $J_{\mathbf{V},i}^{\mathrm{WS}} = 0, J_{\mathbf{W},i}^{\mathrm{WS}} = 0$. Then, the adjusted ROBs are $\tilde{\mathbf{V}}_i^{\mathrm{WS}} = \mathbf{V}_0, \tilde{\mathbf{W}}_i^{\mathrm{WS}} = \mathbf{W}_0$ and the transformed systems are $\dot{\tilde{\mathbf{x}}}_i(t) = \hat{\mathbf{A}}_i\tilde{\mathbf{x}}_i(t)$ according to (3.28). The interpolated system is $\dot{\tilde{\mathbf{x}}}(t) = \tilde{\mathbf{A}}^{\mathrm{WS}}(p)\tilde{\mathbf{x}}(t)$ with $\tilde{\mathbf{A}}^{\mathrm{WS}}(p) = \sum_{i=1}^{2}\omega_i(p)\hat{\mathbf{A}}_i$ where $\omega_i(p)$ are linear weighting functions. Figure 5.2 shows the maximum value of the real part of the eigenvalues $\lambda_{\max}$ of $\tilde{\mathbf{A}}^{\mathrm{WS}}(p)$ which is nonnegative for $p \in [0.2, 0.8]$. Hence, unstable systems are obtained in this interval according to stability criterion S3 from Theorem 2.1.

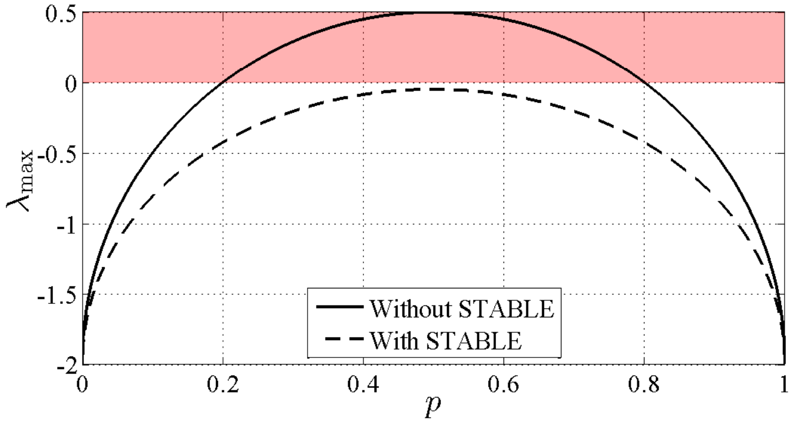


Figure 5.2: Maximum real part of the eigenvalues $\lambda_{\max}$ of the interpolated system without and with STABLE. The unstable region is shown in red.

Secondly, the effect of the proposed stability-preserving method is demonstrated. For adjusting the right ROBs, we obtain the transformation matrices $\mathbf{T}_i = \mathbf{I}_2$ according to (5.5) or (5.7) which is equivalent to the approach without using STABLE. For adjusting the left ROBs, we use STABLE which is demonstrated graphically below. To begin with, we introduce the variables of the algorithm

$$\mathbf{P}_i = \begin{bmatrix} p_{11,i} & p_{12,i} \\ p_{12,i} & p_{22,i} \end{bmatrix} \in \mathbb{S}_2. \tag{5.35}$$

Then, the first constraint (5.14) is described for the two systems by the convex cones

$$\mathcal{P}_i = \{\mathbf{P}_i : \ \mathbf{P}_i > \mathbf{0}\}. \tag{5.36}$$

The notion of convex cones is described in Definition C.2. The intersection of the convex cone of positive definite matrices with the linear space from the second constraint (5.15) provides for the two systems the convex cones

$$\begin{aligned} \mathcal{C}_1 &= \{\mathbf{P}_1 : \ \mathbf{P}_1 > \mathbf{0}, \ \mathbf{P}_1\hat{\mathbf{A}}_1 + \hat{\mathbf{A}}_1^T\mathbf{P}_1 < \mathbf{0}\}, \\ \mathcal{C}_2 &= \{\mathbf{P}_2 : \ \mathbf{P}_2 > \mathbf{0}, \ \mathbf{P}_2\hat{\mathbf{A}}_2 + \hat{\mathbf{A}}_2^T\mathbf{P}_2 < \mathbf{0}\}. \end{aligned} \tag{5.37}$$

The convex cones $\mathcal{C}_1, \mathcal{C}_2$ are non-empty as the systems $\hat{\mathbf{E}}_i\dot{\hat{\mathbf{x}}}_i(t) = \hat{\mathbf{A}}_i\hat{\mathbf{x}}_i(t)$ are asymptotically stable [32, 115] and they comprise all stability-preserving solutions $\mathbf{P}_i$.

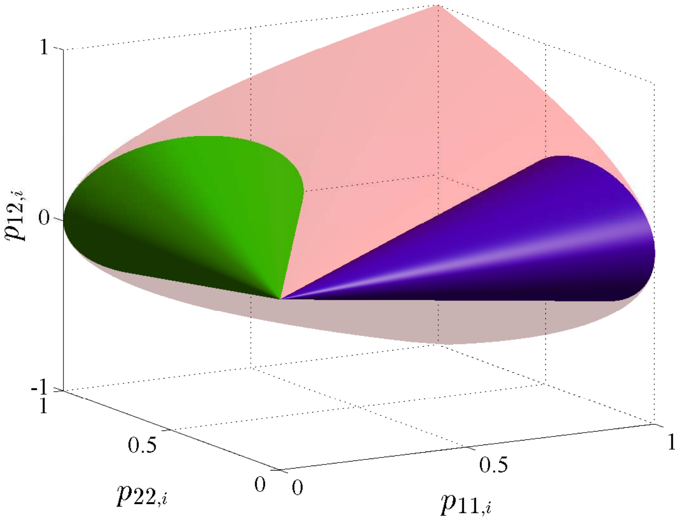


Figure 5.3: Convex cones $\mathcal{C}_1$ (green), $\mathcal{C}_2$ (blue) and $\mathcal{P}_1, \mathcal{P}_2$ (red).

The convex cones $\mathcal{P}_1, \mathcal{P}_2, \mathcal{C}_1, \mathcal{C}_2$ are illustrated in Figure 5.3 and the axes show the entries of $\mathbf{P}_i$. It can be seen that the solutions $\mathbf{M}_i^{\mathrm{WS}} = \mathbf{P}_i^{\mathrm{WS}} = \mathbf{I}_2$ from the procedure above without using STABLE are not included in the convex cones $\mathcal{C}_1, \mathcal{C}_2$ and hence, do not guarantee stable matrix interpolation. It can also be seen that the convex cones $\mathcal{C}_1, \mathcal{C}_2$ do not intersect which shows that the proposed stability-preserving method does not require a common Lyapunov function for the two systems such as known in the field of switched systems [111, 110, 74]. In order to find the optimal solutions $\mathbf{P}_i^*$, consider the objective function (5.19) and (5.22)

$$\begin{aligned} J_{\mathbf{W},i} = J_{\mathrm{MAC},\mathbf{W},i} = \|\mathbf{P}_i - \mathbf{I}_2\|_F^2 &= \mathrm{tr}(\mathbf{P}_i\mathbf{P}_i) - 2\mathrm{tr}(\mathbf{P}_i) + \mathrm{tr}(\mathbf{I}_2) = J_{\mathrm{DS},\mathbf{W},i} \\ &= p_{11,i}^2 + 2p_{12,i}^2 + p_{22,i}^2 - 2p_{11,i} - 2p_{22,i} + 2 \\ &= (p_{11,i} - 1)^2 + (p_{22,i} - 1)^2 + 2p_{12,i}^2. \end{aligned} \tag{5.38}$$

Then, STABLE minimizes the objective functions $J_{\mathbf{W},i}$ so that the optimal solutions $\mathbf{P}_i^*$ are elements of the respective convex cone $\mathcal{C}_i$:

$$\mathbf{P}_1^* = \begin{bmatrix} 0.7446 & 0 \\ 0 & 1.1635 \end{bmatrix}, \quad \mathbf{P}_2^* = \begin{bmatrix} 1.1635 & 0 \\ 0 & 0.7446 \end{bmatrix}. \tag{5.39}$$

A graphical interpretation of the optimization problem is given in Figure 5.4. The axes show the entries $p_{11,i}, p_{22,i}$ of the matrices $\mathbf{P}_i$ with $p_{12,i} = 0$. The objective functions $J_{\mathbf{W},i}$ from equation (5.38) describe circles in the plane $p_{12,i} = 0$ with midpoint $(p_{11,i}, p_{22,i}) = (1, 1)$ and radius $\sqrt{J_{\mathbf{W},i}}$. The convex cones $\mathcal{C}_1, \mathcal{C}_2$ are shown as triangles which are the intersection of the convex cones from Figure 5.3 with the plane $p_{12,i} = 0$. STABLE searches for the solutions $(p_{11,i}^*, p_{22,i}^*)$ with minimum objective functions $J_{\mathbf{W},i}$ so that the solutions lie in the convex cones $\mathcal{C}_1, \mathcal{C}_2$. This is the case for the solutions (5.39) which are the points where the circle with radius $\sqrt{J_{\mathbf{W},i}} = 0.303$ touches the cones.

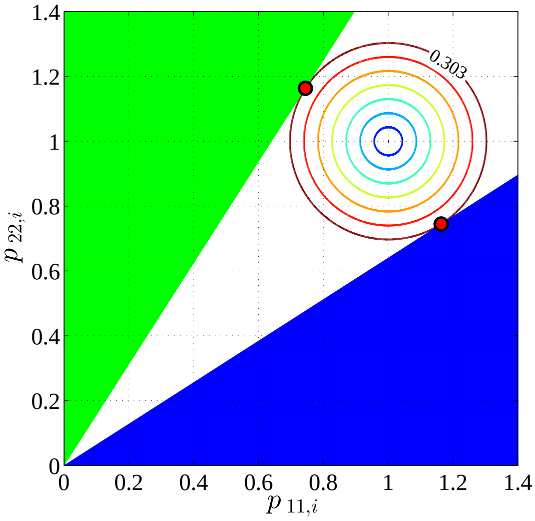


Figure 5.4: Convex cones $\mathcal{C}_1$ (green), $\mathcal{C}_2$ (blue) in the plane $p_{12,i} = 0$ and optimal $(p_{11,i}^*, p_{22,i}^*)$ in red with minimum cost function $J_{\mathbf{W},i} = 0.303^2$.

This results in transformation matrices $\mathbf{M}_i = \mathbf{P}_i^*$ which lead to reduced systems $\mathbf{P}_i^*\dot{\tilde{\mathbf{x}}}_i(t) = \mathbf{P}_i^*\hat{\mathbf{A}}_i\tilde{\mathbf{x}}_i(t)$. Then, the interpolated reduced system is $\tilde{\mathbf{E}}(p)\dot{\tilde{\mathbf{x}}}(t) = \tilde{\mathbf{A}}(p)\tilde{\mathbf{x}}(t)$ with $\tilde{\mathbf{E}}(p) = \sum_{i=1}^{2}\omega_i(p)\mathbf{P}_i^*$ and $\tilde{\mathbf{A}}(p) = \sum_{i=1}^{2}\omega_i(p)\mathbf{P}_i^*\hat{\mathbf{A}}_i$ with linear $\omega_1(p), \omega_2(p)$. The interpolated system possesses the Lyapunov function $V(\tilde{\mathbf{x}}(t), p) = \tilde{\mathbf{x}}(t)^T\Big(\sum\limits_{i=1}^{2}\omega_i(p)\mathbf{P}_i^*\Big)\tilde{\mathbf{x}}(t)$ and hence, is asymptotically stable according to criterion S1 from Theorem 2.1 for the whole domain $\mathcal{D}$. This is confirmed in Figure 5.2 where the maximum value of the real part of the eigenvalues λ_{max} of the pencil $(\tilde{\mathbf{A}}(p), \tilde{\mathbf{E}}(p))$ is negative for domain $\mathcal{D}$.

The price paid is a slight increase of the minimum value of the objective functions $J_{\mathbf{W},i}(\mathbf{P}_i^*)$ because the left ROBs are distorted to $\tilde{\mathbf{W}}_i = \mathbf{W}_0\mathbf{P}_i^*$ which is shown in Table 5.1. Due to duality, it would also be possible to use d-STABLE which distorts the right ROBs as little as necessary and leaves the left ROBs unchanged.

Table 5.1: Comparison of the adjusted ROBs and of the respective minimum values of the objective functions for matrix interpolation with and without STABLE.

	Without STABLE	With STABLE
ROBs	$\tilde{\mathbf{V}}_i^{\text{WS}} = \mathbf{V}_0$ $\tilde{\mathbf{W}}_i^{\text{WS}} = \mathbf{W}_0$	$\tilde{\mathbf{V}}_i = \mathbf{V}_0$ $\tilde{\mathbf{W}}_i = \mathbf{W}_0\mathbf{P}_i^*$
Objective functions	$J_{\mathbf{V},i}^{\text{WS}}(\mathbf{I}_2) = 0$ $J_{\mathbf{W},i}^{\text{WS}}(\mathbf{I}_2) = 0$	$J_{\mathbf{V},i}(\mathbf{I}_2) = 0$ $J_{\mathbf{W},i}(\mathbf{P}_i^*) = 0.303^2$

5.8.2 Anemometer ($n = 29008$)

The next example is an anemometer which is a flow sensing device [1, 121]. It consists of a heater and two temperature sensors which are located before and after the heater. If there is no flow, the heat dissipates symmetrically into the fluid. This symmetry is disturbed if a flow is applied to the fluid which leads to convection of the temperature field. Then, the fluid velocity can be determined from the difference between the temperatures measured with the sensors. The governing equation is given by the convection-diffusion PDE

$$\rho c_p \frac{\partial T}{\partial t} = \nabla(\kappa\nabla T) - \rho c_p v \nabla T + Q, \tag{5.40}$$

where ρ denotes the mass density, $c_p = 0.5\text{Jkg}^{-1}\text{K}^{-1}$ is the specific heat, T is the temperature and Q the heat flow into the system caused by the heater. The thermal conductivity varies in $\kappa \in [1,\ 2]\text{Wm}^{-1}\text{K}^{-1}$ and the fluid velocity in $v \in [0.1,\ 2]\text{ms}^{-1}$. Finite element discretization of the convection-diffusion equation (5.40) with Dirichlet boundary conditions, which are set to zero, leads to the parameter-dependent state-space system of order $n = 29008$

$$\begin{aligned} \mathbf{E}\dot{\mathbf{T}}(t) &= \overbrace{(\mathbf{A}_0 + \kappa\mathbf{A}_1 + \upsilon\mathbf{A}_2)}^{\mathbf{A}(\kappa,\upsilon)}\mathbf{T}(t) + \mathbf{b}Q(t) \\ y(t) &= \mathbf{c}^T\mathbf{T}(t), \end{aligned} \tag{5.41}$$

where $\mathbf{T}(t) \in \mathbb{R}^{29008}$ is the state vector containing the temperatures at the discretization nodes. The input is the heat source $Q(t)$ and the output is the temperature difference between the sensors. In the following, the parameter domain is normalized to $\mathcal{D} = [0,\ 1] \times [0,\ 1]$. The domain is sampled for the set of grid points $\mathcal{P} = \{0, 1/5, \ldots, 1\} \times \{0, 1/5, \ldots, 1\}$ which result in $N = 36$ high-order systems. They are reduced to order $q = 10$ using IRKA and deliver a set of asymptotically stable reduced systems. The reference bases $\mathbf{V}_0, \mathbf{W}_0$ are calculated with the non-weighted SVD approach (3.5) and (3.22).

Firstly, it will be demonstrated what happens if we do not use the stability-preserving method. The transformation matrices for adjusting the ROBs are calculated using the MAC approach and DS approach according to formulas (3.16) or (3.8) and (3.24) or (3.23) such as in the generalized framework. The transformed systems from (3.28) are interpolated using linear weighting functions. In Figure 5.5 the overall error e_T of the interpolated reduced system is plotted for both quality functions using the test grid $\mathcal{P}_\mathrm{test} = \{0, 1/80, \ldots, 1\} \times \{0, 1/80, \ldots, 1\}$. For testing the system, the $\mathcal{H}_\infty$-norm is approximated according to formula (4.6) using an error grid with 50 frequencies which are equally spaced in the interval $[10^1, \ldots, 10^6]$. For the regions with unstable interpolated systems the $\mathcal{H}_\infty$-norm equals infinity.

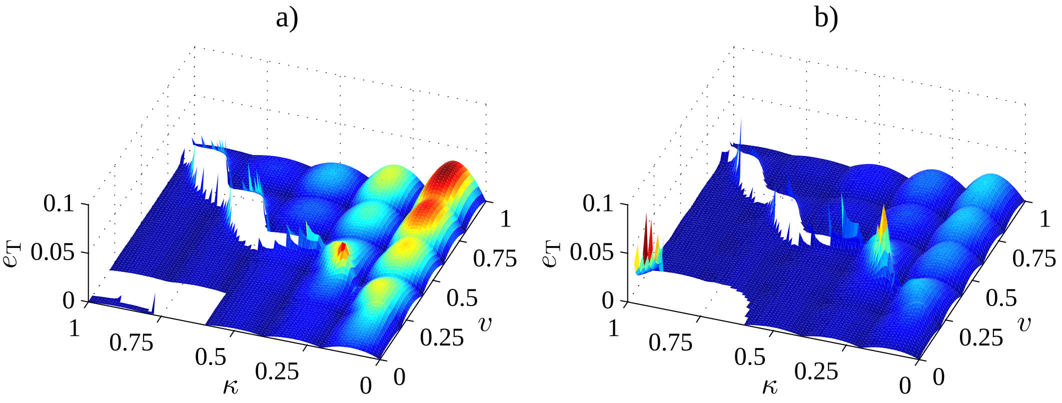


Figure 5.5: Error e_T measured in $\mathcal{H}_\infty$-norm without the stability-preserving method and a) MAC approach b) DS approach.

Secondly, the proposed stability-preserving method is applied. For specifying and solving the convex programs STABLE or d-STABLE, the package CVX is used [83, 82]. The results are obtained using solver SeDuMi [154] and verified with SDPT3 [158]. The error e_{T} of the interpolated reduced system is shown in Figure 5.6 for both STABLE and d-STABLE with the MAC approach and the DS approach. The reader can verify that the proposed methods preserve the stability of the interpolated system for domain $\mathcal{D}$. In addition, the methods deliver nearly as accurate results as the method without stability preservation in regions where the latter is asymptotically stable.

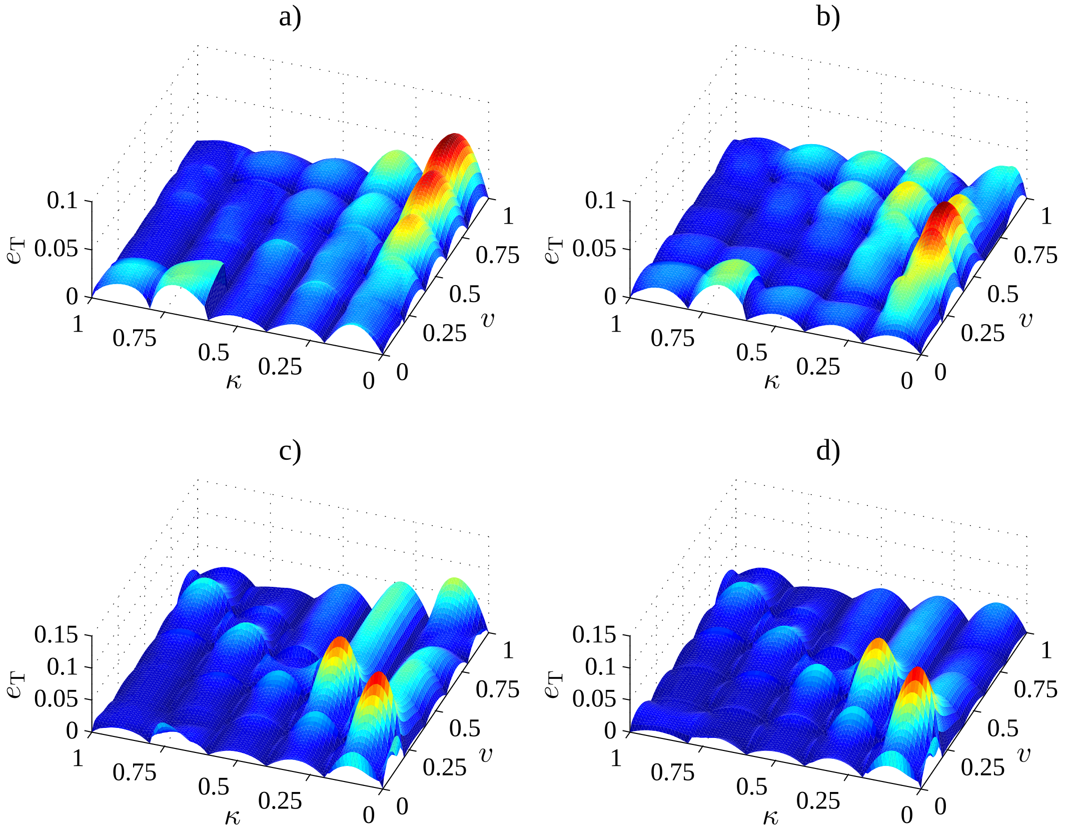


Figure 5.6: Error e_{T} measured in $\mathcal{H}_\infty$-norm using a) STABLE (MAC) b) STABLE (DS) c) d-STABLE (MAC) d) d-STABLE (DS).

In order to decide which of the four combinations to use, we apply Algorithm 4. For this, cross-validation-based error indicators are assigned to the grid points. In Table 5.2 one can see the mean of the indicators J_μ^{CV} with formula (4.23). The true error measure J_μ is approximated using the test grid $\mathcal{P}_{\mathrm{test}}$ with formula (4.21). In this example, the method selection algorithm identifies STABLE with the MAC approach as the best combination which is confirmed by the true error.

Table 5.2: Comparison between the estimated mean using the error indicator and the true mean for different interpolation methods ordered from small (1) to large (4).

	STABLE (MAC)	STABLE (DS)	d-STABLE (MAC)	d-STABLE (DS)
J_μ^{CV}	0.0561 (1)	0.0628 (2)	0.0926 (4)	0.0817 (3)
J_μ	0.0156 (1)	0.0170 (2)	0.0206 (4)	0.0187 (3)

6 Sparse-Grid-Based Interpolation for High-Dimensional Parameter Spaces

6.1 Motivation

If we have many parameters, i.e., the dimension d of $\mathcal{D} \subseteq \mathbb{R}^d$ is large, the interpolation with the pMOR approach from Section 3 can become expensive. The reason is that common grid-based approaches suffer from the *curse of dimensionality*, leading to costs that grow exponentially with the number of parameters [37]. To overcome the curse of dimensionality, we need to find a basis of the interpolant where the relevant information is represented by as few basis functions as possible. Hence, we need to identify and omit the basis functions from a full grid which contribute only little to the quality of the interpolation. In order to find such a basis, we employ interpolation methods based on sparse grids, see Appendix A.4, which allow us to have a grid-based approach in (moderately) high-dimensional settings and which hierarchically set new grid points based on a sparse tensor product construction with asymptotic error decay. A d-dimensional sparse grid has $\mathcal{O}(h^{-1}\log(h^{-1})^{d-1})$ grid points. Hence, it has significantly fewer points than a corresponding full grid with mesh width h in each direction which has $\mathcal{O}(h^{-d})$ points. In return, the accuracy deteriorates only slightly, e.g. from $\mathcal{O}(h^2)$ to $\mathcal{O}(h^2\log(h^{-1})^{d-1})$ for linear basis functions. This beneficial cost-benefit ratio enables us to achieve a similar approximation quality while the sparse grid needs much fewer points in higher dimensions than common grid-based methods. In addition, the sparse grid defines where to insert new grid points and provides error estimates which can be used for developing a refinement method.

The beneficial combination of pMOR and sparse grids has already been exploited in [20, 21] for the interpolation of transfer functions and in [130, 39] for the interpolation of the output of interest derived from the solution of parametrized PDEs. Here, the pMOR approach based on matrix interpolation from Section 3 is combined with sparse grids in order to extend it to multidimensional parameter spaces. The resulting method was published in [72]. It is divided into two parts: The first, offline part is the construction of the interpolants which is described in Section 6.2. It automatically calculates the grid points and the interpolants according to an adaptivity criterion. The second, online part is the evaluation of the interpolants which is described in Section 6.3. A comparison to the general framework concerning the computational effort is given in Section 6.4. In Section 6.5, the method is demonstrated using two examples.

6.2 Sparse-Grid-Based, Iterative Construction of the Interpolants

6.2.1 Introduction

Before the algorithm is explained in detail in Section 6.2.2, a prerequisite and the main principle for the application of sparse grids to matrix interpolation are discussed.

Prerequisite

According to Appendix A.4, sparse grid interpolation ensures—compared to a full-grid-based approach—an optimal cost-approximation ratio, which means that optimal grid points are chosen with respect to the interpolation error in the $\mathcal{L}_2$-norm. Besides, it ensures that an asymptotic error decay is obtained when increasing the number of levels provided that the parameter dependencies are smooth. This condition holds true for the matrix coefficients in the considered applications as the parameter dependencies are continuous and free of kinks in the domain $\mathcal{D}$.

Main principle

We construct a sparse grid interpolant $z^j(\mathbf{p}) = \sum_{i=1}^{N} \alpha_i^j \phi_i^j(\mathbf{p})$ for every matrix coefficient j which leads to matrices consisting of interpolants such as (3.30). In order to keep the computational effort of the proposed method low, we use the same sparse grid, i.e. the same set of grid points and the same set of basis functions $\phi_1(\mathbf{p}), \dots, \phi_N(\mathbf{p})$ for all matrix coefficients, but each coefficient has its own hierarchical surpluses $\alpha_1^j, \dots, \alpha_N^j$. This enables us to reuse local systems which have already been reduced. In addition, the interpolants of all matrix entries can be computed in parallel.

6.2.2 Algorithm

The algorithm for the construction of the interpolants performs steps 1–5 of the general framework from Section 3 iteratively in order to determine the optimal level ℓ of the sparse grid. Level ℓ and hence, the grid points of the sparse grid are automatically determined by an adaptivity criterion based on hierarchical surpluses. The iterative procedure is shown in Figure 6.1. Inputs of the algorithm are the grid type, dimension d, order q of the reduced systems, maximum level l_{max} and the desired relative and absolute error tolerance e_{rel} and e_{abs}, respectively.

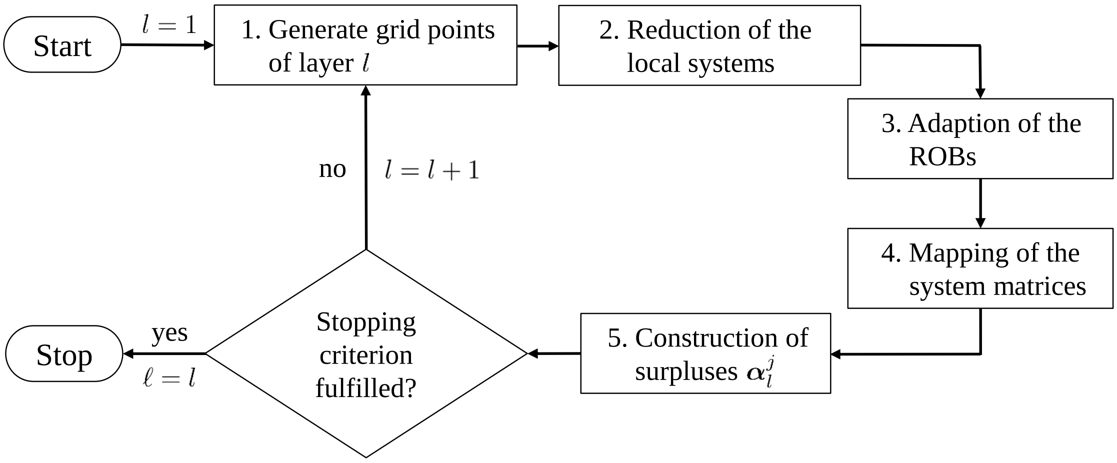

Start
$l = 1$
1. Generate grid points of layer l
2. Reduction of the local systems
3. Adaption of the ROBs
4. Mapping of the system matrices
5. Construction of surpluses $\boldsymbol{\alpha}_l^j$
Stopping criterion fulfilled?
no
$l = l + 1$
yes
$\ell = l$
Stop

Figure 6.1: Sparse-grid-based, iterative construction of the interpolants.

Firstly, the user chooses a grid type and the sparse grid points of the d-dimensional sparse grid of level l are generated beginning from $l = 1$ such as described in Appendix A.4. These become the parameter vectors $\mathbf{p}_1, \ldots, \mathbf{p}_{N_l}$ at which the high-order systems $\mathbf{G}(\mathbf{p}_1), \ldots, \mathbf{G}(\mathbf{p}_{N_l})$ are computed. Afterwards, these systems are reduced to order q according to Section 3.3 and transformed in order to adjust the ROBs such as described in Section 3.4. As reference ROB we use the ROB of system $\hat{\mathbf{G}}(\mathbf{p}_{i_0})$ at $\mathbf{p}_{i_0} = (0.5, \ldots, 0.5)$ because its corresponding grid point already appears in the first level and lies in the center of the domain. Afterwards, the system matrices are projected onto the tangent space to obtain matrices $\boldsymbol{\Gamma}_i^{\tilde{\mathbf{E}}}$, $\boldsymbol{\Gamma}_i^{\tilde{\mathbf{A}}}$, $\boldsymbol{\Gamma}_i^{\tilde{\mathbf{B}}}$, $\boldsymbol{\Gamma}_i^{\tilde{\mathbf{C}}}$ with $i \in \{1, \ldots, N_l\}$ according to Section 3.5. The mapped matrices are reshaped into vectors $\boldsymbol{\gamma}(\mathbf{p}_1), \ldots, \boldsymbol{\gamma}(\mathbf{p}_{N_l})$ of length $K = 2q^2 + qr + mq$ where $\gamma^j(\mathbf{p}_i)$ is the j-th entry of $\boldsymbol{\gamma}(\mathbf{p}_i)$ with $j \in \{1, \ldots, K\}$. Then, for each component j a sparse grid interpolant $z^j : \mathcal{D} \to \mathbb{R}$ is constructed:

$$\hat{\boldsymbol{\gamma}}(\mathbf{p}) = [z^1(\mathbf{p}), \ldots, z^K(\mathbf{p})]^T. \tag{6.1}$$

The vector $\hat{\boldsymbol{\gamma}}(\mathbf{p})$ is an approximation of $\boldsymbol{\gamma}(\mathbf{p})$ which is the vector containing the elements of the transformed and mapped system matrices of a directly reduced system obtained by MOR from Section 2.2 at parameter value $\mathbf{p}$. The interpolants

$$z^j(\mathbf{p}) = z^j(\mathbf{p}) + \sum_{i=1}^{N_l} \alpha^j_{l,i} \phi_{l,i}(\mathbf{p}) \tag{6.2}$$

grow in every iteration. In (6.2), the basis functions $\phi_{l,1}(\mathbf{p}), \ldots, \phi_{l,N_l}(\mathbf{p})$ of level l are equal for every matrix entry j and the best choice for the basis functions clearly depends on the application. The surpluses $\boldsymbol{\alpha}_l^j = (\alpha^j_{l,1}, \ldots, \alpha^j_{l,N_l}) \in \mathbb{R}^{N_l}$ are computed for level l

with the procedure described in Section A.4. Due to the hierarchical structure, the surpluses $\boldsymbol{\alpha}_l^j \to \mathbf{0}$ converge to zero as the level $l \to \infty$ tends to infinity. Hence, they are appropriate for being used as error indicators to find the optimal number of sparse grid levels and hence, of grid points concerning a desired accuracy. As adaptivity criteria we employ error estimates presented in [103, 101]. An absolute error estimate is proposed therein for the interpolation error $||\gamma^j(\mathbf{p}) - z^j(\mathbf{p})||_{\mathcal{L}_2}$ for every j where $\gamma^j(\mathbf{p})$ is the j-th entry of $\boldsymbol{\gamma}(\mathbf{p})$. The error estimate is the maximum of the absolute values of all hierarchical surpluses of the current level l

$$e^j_{\text{abs,est}} = \max(|\boldsymbol{\alpha}_l^j|). \tag{6.3}$$

For the relative error estimate, the absolute one is divided by the difference between $\gamma^j_{\text{max}} = \max\{\gamma^j(\mathbf{p}_1), \dots, \gamma^j(\mathbf{p}_{\sum_{i=1}^l N_i})\}$ and $\gamma^j_{\text{min}} = \min\{\gamma^j(\mathbf{p}_1), \dots, \gamma^j(\mathbf{p}_{\sum_{i=1}^l N_i})\}$

$$e^j_{\text{rel,est}} = \frac{e^j_{\text{abs,est}}}{\gamma^j_{\text{max}} - \gamma^j_{\text{min}}} = \frac{\max(|\boldsymbol{\alpha}_l^j|)}{\gamma^j_{\text{max}} - \gamma^j_{\text{min}}}. \tag{6.4}$$

These criteria are reasonable because the hierarchical surplus $\alpha^j_{l,i}$ measures the contribution of the corresponding basis function $\phi_{l,i}(\mathbf{p})$ to the interpolant (6.2).

In this iterative procedure, level number l is increased until either the absolute (6.3) or the relative (6.4) error estimates are less than the desired tolerances e_{rel} and e_{abs} for every coefficient $j \in \{1, \dots, K\}$ determining the final level ℓ of the sparse grid. One can use the maximum level l_{max} as an additional stopping criterion.

Remark: Note that the error estimates (6.3) and (6.4) hold true for the interpolation error $||\gamma^j(\mathbf{p}) - z^j(\mathbf{p})||_{\mathcal{L}_2}$ of the separate K matrix entries of the parameter-dependent reduced system. However, we are interested in the error of the reduced system, e.g. $||\mathbf{G}(\mathbf{p}) - \tilde{\mathbf{G}}(\mathbf{p})||_{\mathcal{H}_p}$, such as presented in Section 3.8 and to the best of the author's knowledge it is so far not possible to deduce the error of the system from the error of its matrix entries. Nevertheless, the error indicators can be applied to monitor the decay of the error of the matrix entries and the procedure stops as soon as a new level of grid points has little impact on the interpolants (6.2).

To sum up, the procedure is given in Algorithm 10. It delivers the vector of interpolants (6.1). As all component functions $z^1(\mathbf{p}), \dots, z^K(\mathbf{p})$ share the same basis function, it is enough to save all hierarchical surpluses $\boldsymbol{\alpha}^j = (\alpha^j_1, \dots, \alpha^j_N)$ with $N = \sum_{l=1}^{\ell} N_l$ of the sparse grid with level ℓ for all j matrix entries.

Algorithm 10 Sparse-grid-based, iterative construction of interpolants

```
Input: $d$, $q$, $e_{\text{rel}}$, $e_{\text{abs}}$, $l_{max}$
Output: Surpluses $\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^K$, grid points $\mathcal{P}$
 1: Initialize empty set $\mathcal{P} = \emptyset$, empty $\boldsymbol{\alpha}^j = [\,]$, $\gamma^j_{\text{min}} = \infty$,$\gamma^j_{\text{max}} = -\infty$ for $j \in \{1, \ldots, K\}$
 2: for $l = 1, \ldots, l_{max}$ do
 3:     Generate points $\mathcal{P}_l = \{\mathbf{p}_1, \ldots, \mathbf{p}_{N_l}\}$ of $d$-dimensional sparse grid of level $l$
 4:     Update the set $\mathcal{P} = \mathcal{P} \cup \mathcal{P}_l$
 5:     Compute systems $\mathbf{G}(\mathbf{p}_1), \ldots, \mathbf{G}(\mathbf{p}_{N_l})$ of order $n$
 6:     Reduce and transform systems to $\tilde{\mathbf{G}}(\mathbf{p}_1), \ldots, \tilde{\mathbf{G}}(\mathbf{p}_{N_l})$ of order $q \ll n$
 7:     Map system matrices to $\boldsymbol{\Gamma}_i^{\tilde{\mathbf{E}}}$, $\boldsymbol{\Gamma}_i^{\tilde{\mathbf{A}}}$, $\boldsymbol{\Gamma}_i^{\tilde{\mathbf{B}}}$, $\boldsymbol{\Gamma}_i^{\tilde{\mathbf{C}}}$ for $i \in \{1, \ldots, N_l\}$
 8:     Reshape $\boldsymbol{\Gamma}_i^{\tilde{\mathbf{E}}}$, $\boldsymbol{\Gamma}_i^{\tilde{\mathbf{A}}}$, $\boldsymbol{\Gamma}_i^{\tilde{\mathbf{B}}}$, $\boldsymbol{\Gamma}_i^{\tilde{\mathbf{C}}}$ into vector $\boldsymbol{\gamma}(\mathbf{p}_i)$ of length $K$ for $i \in \{1, \ldots, N_l\}$
 9:     for $j = 1, \ldots, K$ do
10:         Construct surpluses $\boldsymbol{\alpha}^j_l = (\alpha^j_1, \ldots, \alpha^j_{N_l})$ with $\mathcal{P}$, $\gamma^j(\mathbf{p}_1), \ldots, \gamma^j(\mathbf{p}_{N_l})$ and $\boldsymbol{\alpha}^j$
11:         Build $\boldsymbol{\alpha}^j = [\boldsymbol{\alpha}^j, \boldsymbol{\alpha}^j_l]$
12:         Compute $\gamma^j_{\text{min}} = \min\{\gamma^j_{\text{min}}, \gamma^j(\mathbf{p}_1), \ldots, \gamma^j(\mathbf{p}_{N_l})\}$
13:         Compute $\gamma^j_{\text{max}} = \max\{\gamma^j_{\text{max}}, \gamma^j(\mathbf{p}_1), \ldots, \gamma^j(\mathbf{p}_{N_l})\}$
14:     end for
15:     if $\max(|\boldsymbol{\alpha}^j_l|) < \max\{e_{\text{rel}} \cdot (\gamma^j_{\text{max}} - \gamma^j_{\text{min}}), e_{\text{abs}}\} \quad \forall\, j \in \{1, \ldots, K\}$ then
16:         Break
17:     end if
18: end for
```

6.2.3 Implementation

Sparse grids are well-suited to deal with multiple model outputs and hence, allow to construct the interpolants of all matrix elements (6.1) at the same time. Therefore, we can use available sparse grid software packages, e.g. the Sparse Grid Interpolation Toolbox [103, 102] for MATLAB, and can easily implement Algorithm 10 as follows:

```
Options = spset('NumberOfOutputs', K, 'GridType', Type, 'RelTol', ...
    RelTol, 'AbsTol', AbsTol, 'MaxDepth', l_max);
z = spvals(@GetReducedMatrices, d, Range, Options);
```

The user feeds the toolbox with the number of matrix elements K, the grid type, the error tolerances $e_{\text{rel}}, e_{\text{abs}}$, the maximum level l_{max}, the domain $\mathcal{D}$ of the parameter space which is specified with `Range` and the dimension of the parameter space d. The function `GetReducedMatrices` includes all operations which are related to pMOR by matrix interpolation and which are given in lines 5–8 of Algorithm 10. Then, the software will automatically compute the interpolant of every matrix element.

6.3 Evaluation of the Interpolants

In the online phase, i.e. when a reduced system is supposed to be calculated for the desired parameter vector $\mathbf{p}_{\text{int}} \in \mathcal{D}$, the vector of interpolants $\hat{\boldsymbol{\gamma}}(\mathbf{p})$ from (6.1) is evaluated for vector $\mathbf{p}_{\text{int}}$. Hence, for each component $j \in \{1, \dots, K\}$ the corresponding sparse grid interpolant z^j is evaluated:

$$\hat{\boldsymbol{\gamma}}(\mathbf{p}_{\text{int}}) = \begin{bmatrix} z^1(\mathbf{p}_{\text{int}}) \\ \vdots \\ z^K(\mathbf{p}_{\text{int}}) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N \alpha_i^1 \phi_i(\mathbf{p}_{\text{int}}) \\ \vdots \\ \sum_{i=1}^N \alpha_i^K \phi_i(\mathbf{p}_{\text{int}}) \end{bmatrix}. \tag{6.5}$$

In the Sparse Grid Interpolation Toolbox [103, 102] the evaluation of the interpolants can be conducted using:

```
    for i=1:K
        z{1}.selectOutput = i;
        interpolant_eval(i) = spinterp(z{1}, p_int);
    end
```

However, there is a more efficient way of implementing this step. As we use the same basis function $\phi_i(\mathbf{p}_{\text{int}})$ for all coefficients, we can simplify formula (6.5) to

$$\hat{\boldsymbol{\gamma}}(\mathbf{p}_{\text{int}}) = \sum_{i=1}^{N} \phi_i(\mathbf{p}_{\text{int}}) \begin{bmatrix} \alpha_i^1 \\ \vdots \\ \alpha_i^K \end{bmatrix} \tag{6.6}$$

where every basis function $\phi_i(\mathbf{p}_{\text{int}})$ is evaluated only once. Then, the vector $\hat{\boldsymbol{\gamma}}(\mathbf{p}_{\text{int}})$ is rearranged into the matrix form. The first q^2 coefficients are arranged into the matrix

$$\boldsymbol{\Gamma}^{\tilde{\mathbf{E}}}(\mathbf{p}_{\text{int}}) = \sum_{i=1}^{N} \phi_i(\mathbf{p}_{\text{int}}) \underbrace{\begin{bmatrix} \alpha_{\tilde{\mathbf{E}},i}^{11} & \dots & \alpha_{\tilde{\mathbf{E}},i}^{1q} \\ \vdots & & \vdots \\ \alpha_{\tilde{\mathbf{E}},i}^{q1} & \dots & \alpha_{\tilde{\mathbf{E}},i}^{qq} \end{bmatrix}}_{\boldsymbol{\alpha}_i^{\tilde{\mathbf{E}}} \in \mathbb{R}^{q \times q}}. \tag{6.7}$$

The matrices $\boldsymbol{\alpha}_i^{\tilde{\mathbf{E}}}$ comprise the hierarchical surpluses and converge to the zero matrix with increasing i which represents a growing number of layers. In analogy, the following q^2 entries of $\hat{\boldsymbol{\gamma}}(\mathbf{p}_{\text{int}})$ are arranged into $\boldsymbol{\Gamma}^{\tilde{\mathbf{A}}}(\mathbf{p}_{\text{int}})$, the next qr entries of $\hat{\boldsymbol{\gamma}}(\mathbf{p}_{\text{int}})$ into $\boldsymbol{\Gamma}^{\tilde{\mathbf{B}}}(\mathbf{p}_{\text{int}})$ and the last mq entries into $\boldsymbol{\Gamma}^{\tilde{\mathbf{C}}}(\mathbf{p}_{\text{int}})$ with matrices $\boldsymbol{\alpha}_i^{\tilde{\mathbf{A}}} \in \mathbb{R}^{q \times q}$, $\boldsymbol{\alpha}_i^{\tilde{\mathbf{B}}} \in \mathbb{R}^{q \times r}$ and $\boldsymbol{\alpha}_i^{\tilde{\mathbf{C}}} \in \mathbb{R}^{m \times q}$, respectively.

Finally, the interpolated matrices are brought back to their manifold using exponential mappings $\mathrm{Exp}_{\tilde{\mathbf{E}}_{i_0}}, \mathrm{Exp}_{\tilde{\mathbf{A}}_{i_0}}, \mathrm{Exp}_{\tilde{\mathbf{B}}_{i_0}}$ and $\mathrm{Exp}_{\tilde{\mathbf{C}}_{i_0}}$ to obtain matrices $\tilde{\mathbf{E}}(\mathbf{p}_{\mathrm{int}})$, $\tilde{\mathbf{A}}(\mathbf{p}_{\mathrm{int}})$, $\tilde{\mathbf{B}}(\mathbf{p}_{\mathrm{int}})$ and $\tilde{\mathbf{C}}(\mathbf{p}_{\mathrm{int}})$, respectively. These matrices are then used to construct the interpolated reduced system $\tilde{\mathbf{G}}(\mathbf{p}_{\mathrm{int}})$ for the parameter $\mathbf{p}_{\mathrm{int}}$ according to formula (3.36).

6.4 Comparison to the General Framework

The proposed approach performs the steps 1–5 of the general framework from Section 3 iteratively, see Figure 6.1. As explained before, the framework offers many options for the different steps. The suggested method chooses in step 1 the grid points to be a sparse grid space. In step 5 hierarchical basis functions are used. As a result, the computation time of the offline and online phase is reduced considerably compared to classical grid-based methods which enables the reduction of parameter-dependent systems with high-dimensional parameter spaces.

6.4.1 Offline Part

The interpolants are formally equivalent for the sparse-grid-based approach (6.2) and the general framework (3.29) from Section 3. However, their calculation is much faster for the sparse-grid-based method because of three reasons: Firstly, the number of involved grid points N is considerably smaller and hence, less local high-order systems have to be reduced using projection-based MOR. Secondly, as less grid points are involved, less coefficients for the interpolants have to be calculated. Thirdly, due to the hierarchical basis it is not necessary to expensively solve a system of equations (3.32) for calculating these coefficients.

6.4.2 Online Part

Comparing formula (6.7) to formula (3.35) one can see that the evaluation of interpolants is formally equivalent for the proposed sparse-grid-based approach and the general framework from Section 3. However, the evaluation is much faster for the sparse-grid-based method as the number of involved grid points N is considerably smaller.

6.5 Numerical Examples

Two examples are given in the following. We use the Sparse Grid Interpolation Toolbox [103, 102] for constructing and evaluating the interpolants.

6.5.1 Spring-Mass-Damper System

The first, rather academic example is the spring-mass-damper system which was introduced in Section 4.2.4. In this section, the mechanical properties, which are given in Table 6.1, depend on two parameters $\alpha \in [0,\ 1]$ and $\beta \in [0,\ 1]$.

Table 6.1: Mechanical properties of the spring-mass-damper system.

Springs [N/m]		Masses [kg]		Dampers [Ns/m]	
c_1	27	m_1	1	d_2	0.1
c_2	9	m_2	$5-4\beta$	d_3	0.4
c_3	3	m_3	$5+20\beta$	d_4	1.6
c_4	$1+\beta$	m_4	125	d_5	α
c_5	$1+2\alpha$				
c_6	$2+2\alpha$				

We compare the sparse-grid-based approach to the framework from Section 3 using a full grid. For the proposed sparse-grid-based approach—respective values are labeled with SG—we use Algorithm 10 with a Clenshaw-Curtis grid from Figure A.5(c). The iterative procedure with fixed level number $\ell = l_{max} = 2$ delivers a sparse grid with $N = 13$ points which is shown in Figure 6.2. Then, a set of high-order systems $G_1^{\mathrm{SG}}, \dots, G_{13}^{\mathrm{SG}}$ is computed. The full grid—respective values are labeled with FG—with the same mesh width $\Delta\alpha = \Delta\beta = 0.25$, which is depicted in Figure 6.3, possesses $N = 25$ grid points and delivers a set of high-order systems $G_1^{\mathrm{FG}}, \dots, G_{25}^{\mathrm{FG}}$.

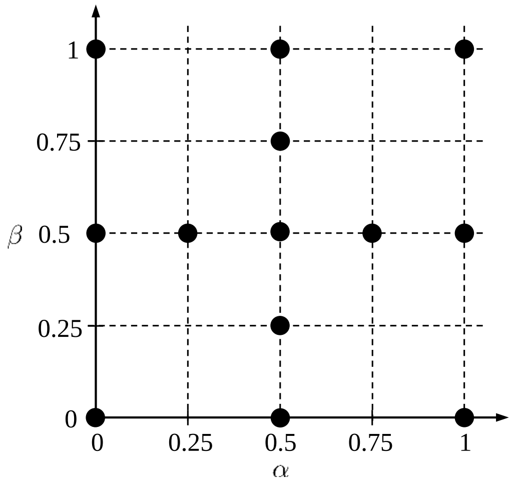


Figure 6.2: Clenshaw-Curtis grid with $\ell = 2$ for the proposed approach.

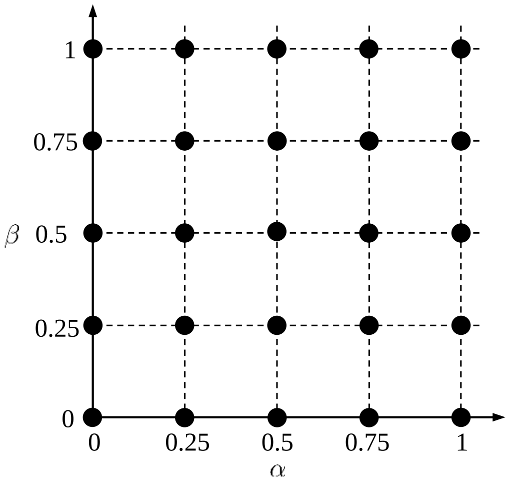


Figure 6.3: Full grid for the conventional approach.

The following steps are equivalent for the two approaches. The local systems are reduced to order $q = 4$ with the two-sided Krylov subspace method with expansion points $s_0 = 0$. The adjustment of the right ROBs is done with the PS approach (3.13) and of the left ROBs with the MAC approach (3.26) in order to reduce the number of interpolants. This results in the set of systems $\tilde{G}_1^{\text{SG}}, \ldots, \tilde{G}_{13}^{\text{SG}}$ and $\tilde{G}_1^{\text{FG}}, \ldots, \tilde{G}_{25}^{\text{FG}}$ according to (3.28) with $\tilde{\mathbf{E}}_i = \mathbf{I}_4$. We choose for the system matrices $\tilde{\mathbf{A}}_i$ the manifold of nonsingular matrices and for $\tilde{\mathbf{B}}_i$ and $\tilde{\mathbf{C}}_i$ the manifold of real matrices.

For the construction of the interpolants we choose for the sparse-grid-based method linear hierarchical basis functions such as demonstrated in Figure A.5 and for the conventional approach we use bilinear basis functions.

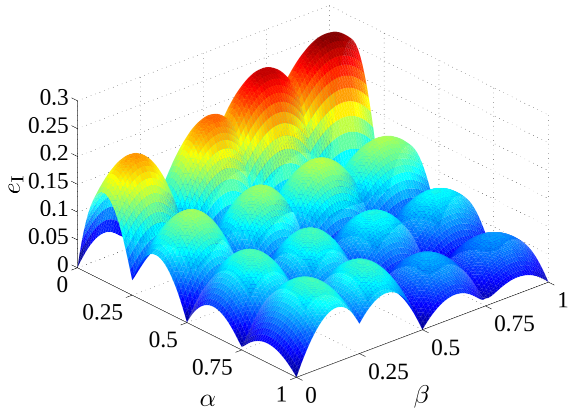


Figure 6.4: Error e_{I} in $\mathcal{H}_2$-norm of the spring-mass-damper system with the sparse-grid-based approach.

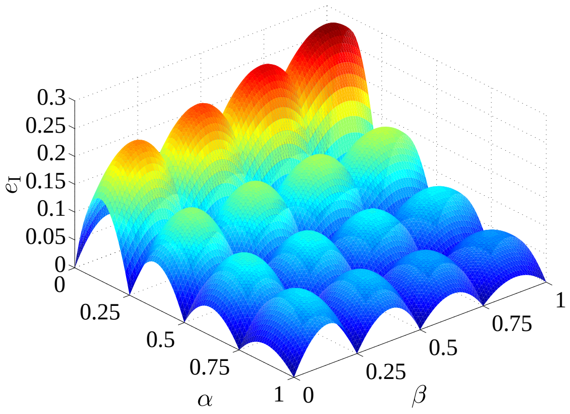


Figure 6.5: Error e_{I} in $\mathcal{H}_2$-norm of the spring-mass-damper system with the full-grid-based approach.

In Figure 6.5 the error e_{I} in $\mathcal{H}_2$-norm of the interpolated systems is shown for the full-grid-based approach and the error e_{I} for the sparse-grid-based method is shown in Figure 6.4. For both approaches the grid points can be identified as points where the interpolation error vanishes. One can verify that both error plots have a similar shape although the sparse grid has 48% less grid points compared to the full grid. The reason is that the sparse grid omits basis functions which contribute only little to the interpolation and captures the ones which represent relevant information.

6.5.2 Cantilever Beam

The second example is a state-space model of a cantilever beam with order $n = 2400$ which was introduced in Section 3.9.2. This time, the system has 7 geometry and material parameters which are shown in Table 6.2 with their range of values. All parameters have a significant impact on the transfer behavior of the system within their range. The parameter vector is defined as $\mathbf{p} = (L, t, h, \rho, E, d_1, d_2)$.

Table 6.2: Parameters of the beam.

Parameter [Unit]	Min $\widehat{=}$ 0	Max $\widehat{=}$ 1
Length L [m]	1	2
Thickness t [m]	0.01	0.05
Height h [m]	0.01	0.05
Density ρ [kg/m^3]	$6{\cdot}10^3$	$10{\cdot}10^3$
Young's modulus E [N/m^2]	$1.5{\cdot}10^{11}$	$2.5{\cdot}10^{11}$
Rayleigh damping factor d_1 [1/s]	$5{\cdot}10^{-6}$	$15{\cdot}10^{-6}$
Rayleigh damping factor d_2 [s]	6	10

In order to reduce the high-order system, Algorithm 10 is used with the Chebyshev sparse grid from Figure A.5(d) and polynomial basis functions [102]. The local systems are reduced to order $q = 10$ using the two-sided Krylov subspace method with expansion points $s_0 = 0$. The adjustment of the right ROBs is done with the PS approach (3.13) and of the left ROBs with the MAC approach (3.26). The latter choice reduces the number of interpolants leading to reduced systems according to (3.28) with $\tilde{\mathbf{E}}_i = \mathbf{I}_{10}$. For matrices $\tilde{\mathbf{A}}_i$ the manifold of nonsingular matrices and for matrices $\tilde{\mathbf{B}}_i$ and $\tilde{\mathbf{C}}_i$ the manifold of real matrices are chosen.

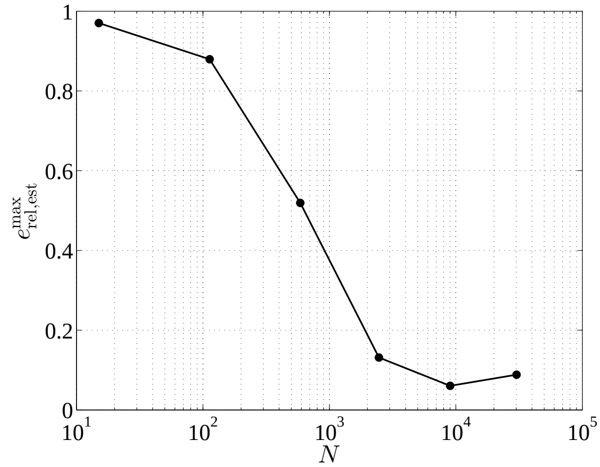


Figure 6.6: Maximum relative error estimate $e_{\text{rel,est}}^{\max}$ of the beam with sparse grids of level $\ell \in \{1, \dots, 6\}$.

The maximum relative error estimate $e_{\text{rel,est}}^{\max} = \max\{e_{\text{rel,est}}^{1}, \dots, e_{\text{rel,est}}^{120}\}$ which is the maximum value for the estimates (6.4) of all 120 matrix entries and which is obtained in every iteration of the hierarchical procedure is shown in Figure 6.6. We use the sparse grid with level $\ell = 4$ with $e_{\text{rel,est}}^{\max} = 0.1315$ as the relative estimate cannot be further decreased for a reasonable number of grid points. This delivers $N = 2465$ points for a 7-dimensional sparse grid. The calculation of the offline phase lasted 367.5s on an

AMD Athlon II X2 255 Duo CPU with 3.1GHz. The corresponding full grid would have $(2^4+1)^7 > 4 \cdot 10^8$ grid points and is computationally infeasible.

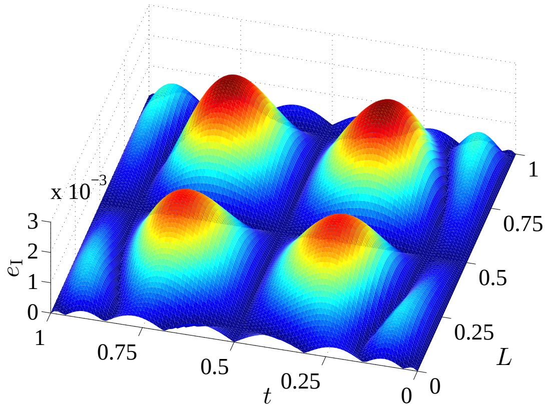


Figure 6.7: Error e_I in $\mathcal{H}_2$-norm of the beam at $\mathbf{p}_\mathrm{int} = (L, t, 0.5, 0.5, 0.5, 0.5, 0.5)$.

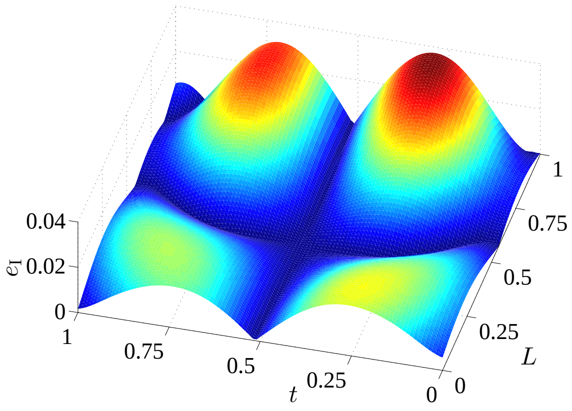


Figure 6.8: Error e_I in $\mathcal{H}_2$-norm of the beam at $\mathbf{p}_\mathrm{int} = (L, t, 0.3, 0.3, 0.3, 0.3, 0.3)$.

In Figure 6.7, the error e_I in $\mathcal{H}_2$-norm of the interpolated systems is shown for parameter vectors $\mathbf{p}_\mathrm{int} = (L, t, 0.5, 0.5, 0.5, 0.5, 0.5)$. One can verify that the error vanishes at the grid points and between the grid points there is an error due to interpolation. In Figure 6.8, the error is plotted for parameter vectors $\mathbf{p}_\mathrm{int} = (L, t, 0.3, 0.3, 0.3, 0.3, 0.3)$ where no grid points lie in the shown (L, t)-layer. Therefore, the error is larger for these parameter vectors and the maximum relative error of the interpolated systems is $e_\mathrm{I}^\mathrm{max} = 0.038$. The maximum error compared to the original system is found to be $e_\mathrm{T}^\mathrm{max} = 0.045$. This example shows that the sparse-grid-based method allows to accurately reduce a parameter-dependent high-order system with a relatively small number of grid points whereas a full-grid-based approach would be infeasible.

7 Interpolation of Differently-Sized Reduced Models using Pseudoinverses

7.1 Motivation

So far, we have examined pMOR by matrix interpolation where we have a set of locally reduced systems which all possess the same reduced order. However, there are some cases where it is important that we have the possibility to allow different local orders:

- For most parameter-dependent systems the dynamics are harder or easier to approximate in different regions of the parameter space [23]. If we reduce these systems at different parameter values e.g. with TBR using the same error bound, we might obtain differently-sized reduced models.
- If a reduced system shall be more accurate for some regions in the parameter domain than for others, we reduce the original system at the corresponding parameter values to larger reduced orders than at the remaining grid points.

The pMOR approaches which are based on the interpolation of system matrices such as the methods in the literature, which are reviewed in Section 2.3.2, or the general framework from Section 3 require a common reduced order of the local systems. The reason why it is not intuitive for these approaches to have different orders is demonstrated by means of the following example.

Example: Consider two high-dimensional systems G_1, G_2 which we reduce to order $q_1 = 2, q_2 = 3$ and we obtain the reduced systems $\dot{\hat{\mathbf{x}}}_1 = \hat{\mathbf{A}}_1\hat{\mathbf{x}}_1$, $\dot{\hat{\mathbf{x}}}_2 = \hat{\mathbf{A}}_2\hat{\mathbf{x}}_2$ with

$$\hat{\mathbf{A}}_1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad \hat{\mathbf{A}}_2 = \begin{bmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \\ 11 & 12 & 13 \end{bmatrix} \tag{7.1}$$

and $\mathbf{V}_1, \mathbf{W}_1 \in \mathbb{R}^{n\times 2}$ and $\mathbf{V}_2, \mathbf{W}_2 \in \mathbb{R}^{n\times 3}$. At first glance we cannot interpolate matrices $\hat{\mathbf{A}}_1, \hat{\mathbf{A}}_2$ because some matrix entries do not have an interpolation partner. The first idea would be—considering that we do not want to touch the high-order systems G_1, G_2 anymore—to reduce the second system further from order $q_2 = 3$ to $q_2^* = 2$. But what is the best way for the reduction? In addition, we loose accuracy for the second system. A second possibility would be to insert information into the reduced matrix $\hat{\mathbf{A}}_1$ which is redundant but which is important for the interpolation with matrix $\hat{\mathbf{A}}_2$. But what is the best way for doing that?

We will find answers to these question in this section. For this, we extend the current methods of pMOR by matrix interpolation to the general case where the local systems have different reduced orders. The method was published in [75]. Here, the proposed resizing procedure is described in Sections 7.3 and 7.4 and demonstrated using a numerical example in Section 7.5.

7.2 Calculation of a Set of Differently-Sized Reduced Systems

The parameter space is sampled for N vectors $\mathbf{p}_1, \dots, \mathbf{p}_N \in \mathcal{D}$ and the original system $\mathbf{G}(\mathbf{p})$ is computed at these grid points which give a set of N high-order systems $\mathcal{G} = \{\mathbf{G}_1, \dots, \mathbf{G}_N\}$. Every local system $\mathbf{G}_i$ with $i \in \{1, \dots, N\}$ is reduced to an individual reduced order $q_i \ll n$ applying a projection-based reduction method according to Section 2.2. This provides subspaces $\mathcal{V}_i$ and $\mathcal{W}_i$ which are spanned by the bases $\mathbf{V}_i \in \mathbb{R}^{n \times q_i}$ and $\mathbf{W}_i \in \mathbb{R}^{n \times q_i}$ with linearly independent columns. This leads to a set of N reduced systems $\hat{\mathcal{G}} = \{\hat{\mathbf{G}}_1, \dots, \hat{\mathbf{G}}_N\}$, where e.g. the i-th system $\hat{\mathbf{G}}_i$ is

$$\hat{\mathbf{G}}_i : \begin{cases} \hat{\mathbf{E}}_i \dot{\hat{\mathbf{x}}}_i(t) = \hat{\mathbf{A}}_i \hat{\mathbf{x}}_i(t) + \hat{\mathbf{B}}_i \mathbf{u}(t) \\ \quad \hat{\mathbf{y}}_i(t) = \hat{\mathbf{C}}_i \hat{\mathbf{x}}_i(t) \end{cases} \tag{7.2}$$

with matrices $\hat{\mathbf{E}}_i, \hat{\mathbf{A}}_i \in \mathbb{R}^{q_i \times q_i}, \hat{\mathbf{B}}_i \in \mathbb{R}^{q_i \times r}$ and $\hat{\mathbf{C}}_i \in \mathbb{R}^{m \times q_i}$ and bases $\mathbf{V}_i \in \mathbb{R}^{n \times q_i}$ and $\mathbf{W}_i \in \mathbb{R}^{n \times q_i}$ of different sizes $q_1, \dots, q_N$.

7.3 Resizing Procedure Using Pseudoinverses

In this section a procedure for interpolating the system matrices $\hat{\mathbf{E}}_i, \hat{\mathbf{A}}_i, \hat{\mathbf{B}}_i, \hat{\mathbf{C}}_i$ with different sizes is presented. The procedure is formally equivalent to the framework from Section 3.4 but extends it to the general case where the transformation matrices $\mathbf{T}_i, \mathbf{M}_i$ are rectangular instead of square. These matrices offer us the possibility to resize the system matrices to the reference size q_0. The resizing procedure is depicted graphically for the case that q_i is smaller than the reference size q_0 in Figure 7.1 and for the case that q_i is larger than the reference size q_0 in Figure 7.2.

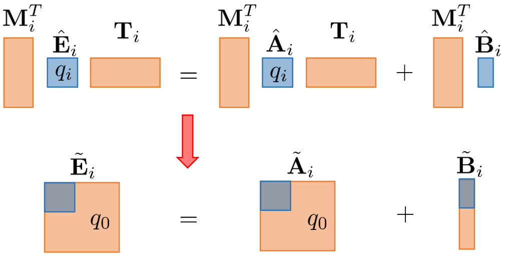


Figure 7.1: Resizing procedure for $q_i < q_0$.

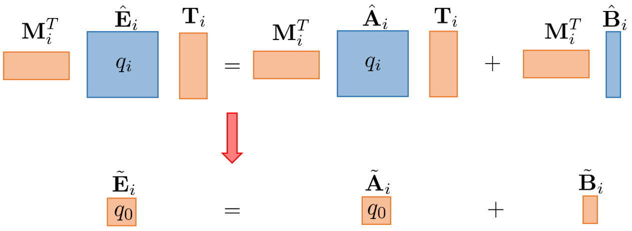


Figure 7.2: Resizing procedure for $q_i > q_0$.

As the system matrices have the same size q_0 after the resizing procedure, they can be reasonably interpolated. In the following, it will be described how the transformation matrices $\mathbf{T}_i, \mathbf{M}_i$ can be calculated with regard to an objective function and it will be pointed out why the pseudoinverse such as described in Appendix D is a crucial component of the proposed method.

7.3.1 Adjustment of the Right Reduced Order Bases

In analogy to Section 3.4.1, the basis $\mathbf{V}_i \in \mathbb{R}^{n \times q_i}$ has to be adjusted to the reference basis $\mathbf{V}_0 \in \mathbb{R}^{n \times q_0}$. Let the latter for now be given as we need more information in order to calculate it. Its calculation will be explained in Section 7.3.4. Then, the resizing and adjusting procedure of the right ROBs is done with

$$\tilde{\mathbf{V}}_i = \mathbf{V}_i \mathbf{T}_i \in \mathbb{R}^{n \times q_0}, \tag{7.3}$$

where $\mathbf{T}_i \in \mathbb{R}^{q_i \times q_0}$ is a rectangular matrix and the new basis $\tilde{\mathbf{V}}_i$ has the size of the reference basis $\mathbf{V}_0$. For adjusting the ROBs we discuss in the following the two objective functions such as described in Section 3.4.1:

DS approach:

In analogy to Section 3.4.1 we use as an objective function the distance of the bases $\tilde{\mathbf{V}}_i, \mathbf{V}_0$ measured with the Frobenius norm which needs to be minimized:

$$\mathbf{T}_i = \arg \min_{\tilde{\mathbf{V}}_i \in \mathbb{R}^{n \times q_0}} \|\tilde{\mathbf{V}}_i - \mathbf{V}_0\|_F = \arg \min_{\mathbf{T}_i \in \mathbb{R}^{q_i \times q_0}} \|\mathbf{V}_i \mathbf{T}_i - \mathbf{V}_0\|_F \tag{7.4}$$

which is similar to the optimization problem (3.7). It is the generalization for different sizes of the reduced bases. We apply the left pseudoinverse (D.1) and obtain the unique best solution

$$\mathbf{T}_i = \mathbf{V}_i^{(l)+} \mathbf{V}_0 \in \mathbb{R}^{q_i \times q_0} \tag{7.5}$$

which can be implemented using Algorithm 1. Although we do not calculate the new basis $\tilde{\mathbf{V}}_i$ explicitly, it is given—for the purpose of illustration—by

$$\begin{aligned}\tilde{\mathbf{V}}_i &= \mathbf{V}_i\mathbf{T}_i = \mathbf{V}_i\mathbf{V}_i^{(l)+}\mathbf{V}_0 \\ &= \underbrace{\mathbf{V}_i(\mathbf{V}_i^T\mathbf{V}_i)^{-1}\mathbf{V}_i^T}_{\mathbf{P}_{\mathbf{V}_i\perp\mathbf{V}_i}}\mathbf{V}_0\end{aligned} \tag{7.6}$$

where $\mathbf{P}_{\mathbf{V}_i\perp\mathbf{V}_i}$ is a projector which projects the columns of $\mathbf{V}_0$ orthogonally to the subspace $\mathcal{V}_i$ into the subspace $\mathcal{V}_i$. One can see that the calculation of the transformation matrix (7.5) and of the new ROB (7.6) is formally the same for all reduced orders $q_i, q_0 \in \{1, 2, \dots, n\}$. However, we have to distinguish between three cases in order to understand the underlying mechanism:

- Case 1: $q_i = q_0$
  This is the case which we already know from Section 3.4.1 where the transformation matrix $\mathbf{T}_i \in \mathbb{R}^{q_0 \times q_0}$ is square. As the bases $\mathbf{V}_i, \mathbf{V}_0$ have the same number of columns q_0, we find the new basis $\tilde{\mathbf{V}}_i$ simply by projecting the columns of $\mathbf{V}_0$ orthogonally to the subspace $\mathcal{V}_i$ into the subspace $\mathcal{V}_i$ according to (7.6). As a result, the basis $\tilde{\mathbf{V}}_i$ also comprises q_0 linearly independent columns and spans the subspace $\tilde{\mathcal{V}}_i = \text{span}(\tilde{\mathbf{V}}_i) = \text{span}(\mathbf{V}_i) = \mathcal{V}_i$. The procedure for adjusting the ROBs is demonstrated for the example $q_i = q_0 = 2$ in Figure 7.3.

- Case 2: $q_i > q_0$
  For this case, the transformation matrix $\mathbf{T}_i \in \mathbb{R}^{q_i \times q_0}$ has more rows than columns. As the basis $\mathbf{V}_i$ comprises more basis vectors than $\mathbf{V}_0$, the procedure (7.6) leaves $q_i - q_0$ columns of $\mathbf{V}_i$ out and adjusts the remaining q_0 basis vectors to $\mathbf{V}_0$ by projecting the columns of $\mathbf{V}_0$ again orthogonally to the subspace $\mathcal{V}_i$ into the subspace $\mathcal{V}_i$. This leads to the smaller basis $\tilde{\mathbf{V}}_i$ with q_0 linearly independent columns where $\text{span}(\tilde{\mathbf{V}}_i) \subset \text{span}(\mathbf{V}_i)$ holds true and one obtains a new reduced system of order $q_0 < q_i$ with right subspace $\tilde{\mathcal{V}}_i$ spanned by the columns of $\tilde{\mathbf{V}}_i$. The procedure for resizing and adjusting the ROBs is demonstrated for the example $q_i = 2, q_0 = 1$ in Figure 7.4.

- Case 3: $q_i < q_0$
  For this case, the transformation matrix $\mathbf{T}_i \in \mathbb{R}^{q_i \times q_0}$ has more columns than rows. As the basis $\mathbf{V}_i$ comprises less basis vectors than $\mathbf{V}_0$, the procedure (7.6) adds $q_0 - q_i$ columns to $\mathbf{V}_i$ and adjusts all basis vectors to $\mathbf{V}_0$ by projecting the columns of $\mathbf{V}_0$ orthogonally to the subspace $\mathcal{V}_i$ into the subspace $\mathcal{V}_i$. This

leads to the larger basis $\tilde{\mathbf{V}}_i$ with $q_0 - q_i$ linear dependent columns where $\tilde{\mathcal{V}}_i = \text{span}(\tilde{\mathbf{V}}_i) = \text{span}(\mathbf{V}_i) = \mathcal{V}_i$ holds true and one obtains a new reduced system of order $q_0 > q_i$. The procedure for resizing and adjusting the ROBs is demonstrated for the example $q_i = 1, q_0 = 2$ in Figure 7.5.

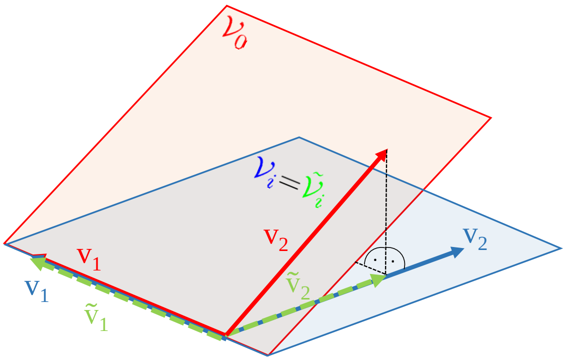


Figure 7.3: A graphic depiction for adjusting the ROBs for $q_i = q_0$.

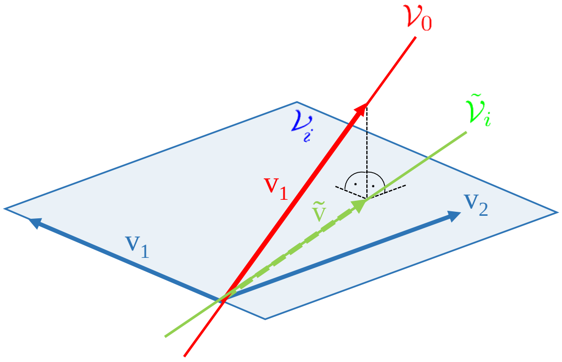


Figure 7.4: A graphic depiction for resizing and adjusting the ROBs for $q_i > q_0$.

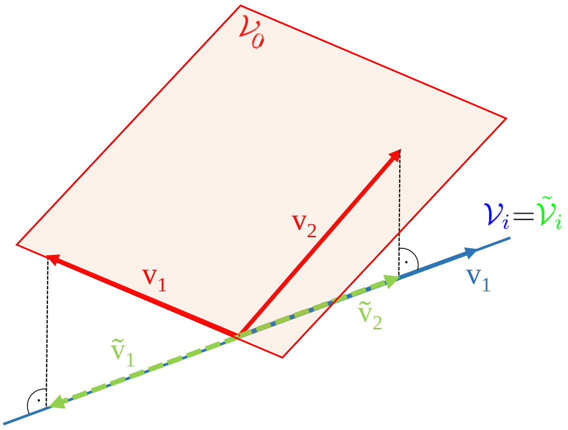


Figure 7.5: A graphic depiction for resizing and adjusting the ROBs for $q_i < q_0$.

MAC approach:

In analogy to Section 3.4.1 we use as an objective function the MAC criterion which measures the correlation of the vectors of the bases $\tilde{\mathbf{V}}_i, \mathbf{V}_0$. For an optimal criterion, we again demand

$$\text{MAC}(\mathbf{V}_0, \tilde{\mathbf{V}}_i) = \mathbf{V}_0^T \tilde{\mathbf{V}}_i = \mathbf{V}_0^T \mathbf{V}_i \mathbf{T}_i \overset{!}{=} \mathbf{I}_{q_0}. \tag{7.7}$$

Then, there exist three cases to solve equation (7.7) which depend on the reduced order q_i of the local system compared to the size q_0 of the reference subspace:

- Case 1: $q_i > q_0$
  For this case, the matrix $\mathbf{V}_0^T\mathbf{V}_i \in \mathbb{R}^{q_0 \times q_i}$ has more columns than rows and has a full row rank. This results in an underdetermined system of equations (7.7) with the following property.
  Proposition 7.1. *If $q_i > q_0$ holds true, a solution for matrix equation (7.7) with minimum $\|\mathbf{T}_i\|_F$ is obtained when the right pseudoinverse (D.2) is used:*

$$\mathbf{T}_i = (\mathbf{V}_0^T\mathbf{V}_i)^{(r)+} \in \mathbb{R}^{q_i \times q_0}. \tag{7.8}$$

Proof. The Penrose condition (D.5) is fulfilled:

$$\underbrace{(\mathbf{V}_0^T\mathbf{V}_i)(\mathbf{V}_0^T\mathbf{V}_i)^{(r)+}}_{\mathbf{I}_{q_0}}\mathbf{I}_{q_0} = \mathbf{I}_{q_0}. \tag{7.9}$$

■

Then, the new right ROB is given for the purpose of illustration by

$$\begin{aligned}\tilde{\mathbf{V}}_i &= \mathbf{V}_i\mathbf{T}_i = \mathbf{V}_i(\mathbf{V}_0^T\mathbf{V}_i)^{(r)+} \\ &= \mathbf{V}_i(\mathbf{V}_0^T\mathbf{V}_i)^T\left[\mathbf{V}_0^T\mathbf{V}_i(\mathbf{V}_0^T\mathbf{V}_i)^T\right]^{-1} \\ &= \mathbf{V}_i\mathbf{V}_i^T\mathbf{V}_0\left[\mathbf{V}_0^T\mathbf{V}_i\mathbf{V}_i^T\mathbf{V}_0\right]^{-1}.\end{aligned} \tag{7.10}$$

With orthogonal $\mathbf{V}_0$ and hence, $\mathbf{V}_0^T\mathbf{V}_0 = \mathbf{I}_{q_0}$ we obtain

$$\tilde{\mathbf{V}}_i = \mathbf{V}_i\mathbf{V}_i^T\mathbf{V}_0\left[\mathbf{V}_0^T\mathbf{V}_i\mathbf{V}_i^T\mathbf{V}_0\right]^{-1}\mathbf{V}_0^T\mathbf{V}_0. \tag{7.11}$$

How can the new ROBs be interpreted? Let us define $\mathbf{V}_i^* = \mathbf{V}_i\mathbf{V}_i^T\mathbf{V}_0 \in \mathbb{R}^{n \times q_0}$ and we obtain

$$\tilde{\mathbf{V}}_i = \underbrace{\mathbf{V}_i^*(\mathbf{V}_0^T\mathbf{V}_i^*)^{-1}\mathbf{V}_0^T}_{\mathbf{P}_{\mathbf{V}_i^*\perp\mathbf{V}_0}}\mathbf{V}_0 \in \mathbb{R}^{n \times q_0}, \tag{7.12}$$

where $\mathbf{P}_{\mathbf{V}_i^*\perp\mathbf{V}_0}$ is a projector which projects the columns of $\mathbf{V}_0$ orthogonally to the subspace $\mathcal{V}_0$ spanned by $\mathbf{V}_0$ into the subspace $\mathcal{V}_i^*$ spanned by $\mathbf{V}_i^*$. This leads to the smaller basis $\tilde{\mathbf{V}}_i$ with q_0 linearly independent columns where $\operatorname{span}(\tilde{\mathbf{V}}_i) \subset \operatorname{span}(\mathbf{V}_i)$ holds true. We can now answer the question which was raised in the introductory section: What is the best way to reduce the original system $\mathbf{G}_i$ via the system $\hat{\mathbf{G}}_i$ of order q_i to the system $\tilde{\mathbf{G}}_i$ with reference order $q_0 < q_i$ with regard to the quality criterion (7.7)? This is shown in the following proposition which is, of course, implicitly done by formula (7.8) without using the high-order system matrices.

Proposition 7.2. *Formula (7.8) is equivalent to reducing system* $\mathbf{G}_i$ *to* $\tilde{\mathbf{G}}_i$ *with the subspace spanned by* $\mathbf{V}_i^* = \mathbf{V}_i\mathbf{V}_i^T\mathbf{V}_0$ *and using* $\mathbf{T}_i^* = (\mathbf{V}_0^T\mathbf{V}_i^*)^{-1}$ *from (3.16).*

Proof. Calculate $\mathbf{T}_i$ according to (3.16)

$$\mathbf{T}_i^* = (\mathbf{V}_0^T\mathbf{V}_i^*)^{-1} = (\mathbf{V}_0^T\mathbf{V}_i\mathbf{V}_i^T\mathbf{V}_0)^{-1}. \tag{7.13}$$

Determine the new right ROB

$$\tilde{\mathbf{V}}_i = \mathbf{V}_i^*\mathbf{T}_i^* = \mathbf{V}_i\mathbf{V}_i^T\mathbf{V}_0\left(\mathbf{V}_0^T\mathbf{V}_i\mathbf{V}_i^T\mathbf{V}_0\right)^{-1} \tag{7.14}$$

which is the result from (7.10). ■

- Case 2: $q_i < q_0$

  For this case, matrix $\mathbf{V}_0^T\mathbf{V}_i \in \mathbb{R}^{q_0\times q_i}$ has more rows than columns and has a full column rank. This results in an overdetermined system of equations (7.7) which in general does not have a solution. Hence, we formulate the least squares optimization problem

$$\mathbf{T}_i = \arg\min_{\mathbf{T}_i\in\mathbb{R}^{q_i\times q_0}} \|\mathbf{V}_0^T\mathbf{V}_i\mathbf{T}_i - \mathbf{I}_{q_0}\|_F, \tag{7.15}$$

where the unique best approximate solution according to (D.6) is obtained by applying the left pseudoinverse

$$\mathbf{T}_i = (\mathbf{V}_0^T\mathbf{V}_i)^{(l)+} \in \mathbb{R}^{q_i\times q_0}. \tag{7.16}$$

The new right ROB is with $\mathbf{V}_0^T\mathbf{V}_0 = \mathbf{I}_{q_0}$ given by

$$\begin{aligned}\tilde{\mathbf{V}}_i &= \mathbf{V}_i\mathbf{T}_i = \mathbf{V}_i(\mathbf{V}_0^T\mathbf{V}_i)^{(l)+}\\ &= \mathbf{V}_i\left[(\mathbf{V}_0^T\mathbf{V}_i)^T\mathbf{V}_0^T\mathbf{V}_i\right]^{-1}(\mathbf{V}_0^T\mathbf{V}_i)^T\\ &= \mathbf{V}_i\left[(\mathbf{V}_i^T\mathbf{V}_0\mathbf{V}_0^T)\mathbf{V}_i\right]^{-1}(\mathbf{V}_i^T\mathbf{V}_0\mathbf{V}_0^T)\mathbf{V}_0.\end{aligned} \tag{7.17}$$

How can the new ROB be interpreted? Let us define $\mathbf{V}_{0,i}^* = \mathbf{V}_0\mathbf{V}_0^T\mathbf{V}_i \in \mathbb{R}^{n\times q_i}$ in order to obtain

$$\tilde{\mathbf{V}}_i = \underbrace{\mathbf{V}_i(\mathbf{V}_{0,i}^{*T}\mathbf{V}_i)^{-1}\mathbf{V}_{0,i}^{*T}}_{\mathbf{P}_{\mathbf{V}_i\perp\mathbf{V}_{0,i}^*}}\mathbf{V}_0 \in \mathbb{R}^{n\times q_0}, \tag{7.18}$$

where $\mathbf{P}_{\mathbf{V}_i\perp\mathbf{V}_{0,i}^*}$ is a projector which projects the columns of $\mathbf{V}_0$ orthogonally to the subspace spanned by $\mathbf{V}_{0,i}^*$ into the subspace $\mathcal{V}_i$. This leads to the larger basis $\tilde{\mathbf{V}}_i$

with $q_0 - q_i$ linear dependent columns with $\tilde{\mathcal{V}}_i = \operatorname{span}(\tilde{\mathbf{V}}_i) = \operatorname{span}(\mathbf{V}_i) = \mathcal{V}_i$.

- Case 3: $q_i = q_0$

  For this case, the matrix $\mathbf{V}_0^T\mathbf{V}_i \in \mathbb{R}^{q_0 \times q_i}$ is square and nonsingular. This results in a system of equations (7.7) which we already know and which has the unique solution $\mathbf{T}_i = (\mathbf{V}_0^T\mathbf{V}_i)^{-1}$ from (3.16). This, indeed, is a special case of the above presented cases:

  Proposition 7.3. *If $q_i = q_0$ holds true, the choice $\mathbf{T}_i = (\mathbf{V}_0^T\mathbf{V}_i)^{-1}$ from (3.16) results from (7.8) and (7.16).*

  Proof. With $q_i = q_0$ we obtain for formula (7.8)

$$\begin{aligned}\mathbf{T}_i = (\mathbf{V}_0^T\mathbf{V}_i)^{(r)+} &= (\mathbf{V}_0^T\mathbf{V}_i)^T \left[\mathbf{V}_0^T\mathbf{V}_i(\mathbf{V}_0^T\mathbf{V}_i)^T\right]^{-1} \\ &= (\mathbf{V}_0^T\mathbf{V}_i)^T(\mathbf{V}_0\mathbf{V}_i)^{-T}(\mathbf{V}_0^T\mathbf{V}_i)^{-1} \\ &= (\mathbf{V}_0^T\mathbf{V}_i)^{-1}.\end{aligned} \tag{7.19}$$

  In analogy, this can be done for formula (7.16). ■

  Then, the new right ROB is again obtained by projecting the columns of $\mathbf{V}_0$ into subspace $\mathcal{V}_i$ orthogonally to the subspace spanned by $\mathbf{V}_0$

$$\begin{aligned}\tilde{\mathbf{V}}_i = \mathbf{V}_i\mathbf{T}_i &= \mathbf{V}_i(\mathbf{V}_0^T\mathbf{V}_i)^{-1} \\ &= \underbrace{\mathbf{V}_i(\mathbf{V}_0^T\mathbf{V}_i)^{-1}\mathbf{V}_0^T}_{\mathbf{P}_{\mathbf{V}_i \perp \mathbf{V}_0}}\mathbf{V}_0.\end{aligned} \tag{7.20}$$

The three formulas (7.8), (7.16) and (3.16) for calculating $\mathbf{T}_i$ using the MAC approach are formally different. But, as we determine the pseudoinverse using the economy size SVD from (D.3), we can propose one procedure for all three cases in Algorithm 11. Note that $\mathbf{\Sigma}_i^+ = \mathbf{\Sigma}_i^{-1}$ holds true since $\mathbf{V}_0^T\mathbf{V}_i$ has full row rank for $q_i > q_0$ and accordingly, full column rank for $q_i \leq q_0$. Hence, there is no singular value at zero.

Algorithm 11 Calculation of $\mathbf{T}_i$ using the MAC approach

```
Input: $N$ matrices $\mathbf{V}_i$ and reference matrix $\mathbf{V}_0$
Output: $N$ matrices $\mathbf{T}_i$
for $i = 1$ to $N$ do
    Compute $\mathbf{V}_0^T\mathbf{V}_i = \mathbf{U}_{\mathbf{V},i}\mathbf{\Sigma}_{\mathbf{V},i}\mathbf{Z}_{\mathbf{V},i}^T$        // SVD
    Compute $\mathbf{T}_i = \mathbf{Z}_{\mathbf{V},i}\mathbf{\Sigma}_{\mathbf{V},i}^{-1}\mathbf{U}_{\mathbf{V},i}^T$
end for
```

To sum up, there are two approaches for the resizing procedure and the adjustment of the right ROBs: the MAC and DS approach. The choice leading to the most accurate reduced model is problem-dependent. This can e.g. be observed by means of the example in Section 7.5. The formulas for calculating matrix $\mathbf{T}_i$ are given in Table 7.1.

Table 7.1: Formulas for the calculation of matrix $\mathbf{T}_i$.

	MAC approach	DS approach
$q_i < q_0$	$\mathbf{T}_i = (\mathbf{V}_0^T\mathbf{V}_i)^{(l)+}$	$\mathbf{T}_i = \mathbf{V}_i^{(l)+}\mathbf{V}_0$
$q_i = q_0$	$\mathbf{T}_i = (\mathbf{V}_0^T\mathbf{V}_i)^{-1}$	
$q_i > q_0$	$\mathbf{T}_i = (\mathbf{V}_0^T\mathbf{V}_i)^{(r)+}$	

Relation between the two approaches

The MAC and DS approach for adjusting the ROBs have in common that the basis of the reference subspace $\mathbf{V}_0$ is projected in a way which results in the new right ROBs $\tilde{\mathbf{V}}_i$. However, they differ in the kind of projection. A relation between the two approaches is presented in the following proposition.

Proposition 7.4. *Both approaches are equivalent for $q_0 = n$ and $\mathbf{V}_0 \in \mathbb{O}_n$.*

Proof. If the reference basis $\mathbf{V}_0 \in \mathbb{O}_n$ is chosen to be orthogonal with $q_0 = n$, it holds true that $\mathbf{V}_0\mathbf{V}_0^T = \mathbf{I}_{q_0}$. Then, it follows for the case $q_i < q_0$ with formula (7.17)

$$\begin{aligned}\tilde{\mathbf{V}}_i &= \mathbf{V}_i\left[(\mathbf{V}_0^T\mathbf{V}_i)^T\mathbf{V}_0^T\mathbf{V}_i\right]^{-1}(\mathbf{V}_0^T\mathbf{V}_i)^T\\ &= \mathbf{V}_i\left[\mathbf{V}_i^T\mathbf{V}_0\mathbf{V}_0^T\mathbf{V}_i\right]^{-1}\mathbf{V}_i^T\mathbf{V}_0\\ &= \mathbf{V}_i(\mathbf{V}_i^T\mathbf{V}_i)^{-1}\mathbf{V}_i^T\mathbf{V}_0\end{aligned} \tag{7.21}$$

which is the result from (7.6). For the (unrealistic) case $q_0 = q_i = n$ with $\mathbf{V}_i \in \mathbb{R}^{n\times n}$ formulas (7.21) and (7.17) additionally simplify to $\tilde{\mathbf{V}}_i = \mathbf{V}_0$. Here, the case $q_i > q_0$ cannot occur as $q_0 = n$ is the maximum order. ■

7.3.2 Adjustment of the Left Reduced Order Bases

After the adjustment of the right ROBs, the left ROBs $\mathbf{W}_i \in \mathbb{R}^{n\times q_i}$ are adjusted with respect to a reference subspace spanned by $\mathbf{W}_0 \in \mathbb{R}^{n\times q_0}$ which we again for now assume to be given. The adjustment of the left ROBs is done in analogy to Section 3.4.2 with

$$\tilde{\mathbf{W}}_i = \mathbf{W}_i\mathbf{M}_i \in \mathbb{R}^{n\times q_0}, \tag{7.22}$$

where—due to duality—matrices $\mathbf{M}_i \in \mathbb{R}^{q_i \times q_0}$ can be obtained using the explanations for the right ROBs in Section 7.3.1. The formulas for calculating matrices $\mathbf{M}_i$ are shown in Table 7.2.

Table 7.2: Formulas for the calculation of matrix $\mathbf{M}_i$.

	MAC approach	DS approach
$q_i < q_0$	$\mathbf{M}_i = (\mathbf{W}_0^T \mathbf{W}_i)^{(l)+}$	$\mathbf{M}_i = \mathbf{W}_i^{(l)+} \mathbf{W}_0$
$q_i = q_0$	$\mathbf{M}_i = (\mathbf{W}_0^T \mathbf{W}_i)^{-1}$	
$q_i > q_0$	$\mathbf{M}_i = (\mathbf{W}_0^T \mathbf{W}_i)^{(r)+}$	

We can use Algorithm 1 for implementing the DS approach and Algorithm 11 for the MAC approach by replacing $\mathbf{V}_i$ with $\mathbf{W}_i$ and $\mathbf{V}_0$ with $\mathbf{W}_0$.

7.3.3 Selection of the Interpolation Manifold

After adjusting the right and left ROBs, we finally obtain the set of compatible reduced systems $\tilde{\mathcal{G}} = \{\tilde{\mathbf{G}}_1, \dots, \tilde{\mathbf{G}}_N\}$, where e.g. the i-th system is given by

$$\tilde{\mathbf{G}}_i : \begin{cases} \tilde{\mathbf{E}}_i \dot{\tilde{\mathbf{x}}}_i(t) = \tilde{\mathbf{A}}_i \tilde{\mathbf{x}}_i(t) + \tilde{\mathbf{B}}_i \mathbf{u}(t) \\ \quad \tilde{\mathbf{y}}_i(t) = \tilde{\mathbf{C}}_i \tilde{\mathbf{x}}_i(t) \end{cases} \tag{7.23}$$

and where $\tilde{\mathbf{x}}_i(t) \in \mathbb{R}^{q_0}$ is the reduced state vector and

$$\begin{aligned} \tilde{\mathbf{E}}_i &= \mathbf{M}_i^T \hat{\mathbf{E}}_i \mathbf{T}_i \in \mathbb{R}^{q_0 \times q_0}, \\ \tilde{\mathbf{A}}_i &= \mathbf{M}_i^T \hat{\mathbf{A}}_i \mathbf{T}_i \in \mathbb{R}^{q_0 \times q_0}, \\ \tilde{\mathbf{B}}_i &= \mathbf{M}_i^T \hat{\mathbf{B}}_i \in \mathbb{R}^{q_0 \times r}, \\ \tilde{\mathbf{C}}_i &= \hat{\mathbf{C}}_i \mathbf{T}_i \in \mathbb{R}^{m \times q_0} \end{aligned} \tag{7.24}$$

are the system matrices which now all have the same size q_0. At this point it is advisable for the reader to shortly look back to Figures 7.1 and 7.2 where the resizing procedure is illustrated graphically.

As we know from Section 3.5, the system matrices are mapped onto the tangent space of a certain matrix manifold. Here, the matrices $\tilde{\mathbf{B}}_i$ belong to the manifold of real matrices $\mathbb{R}^{q_0 \times r}$ and the matrices $\tilde{\mathbf{C}}_i$ to the manifold of real matrices $\mathbb{R}^{m \times q_0}$. For the manifold of the matrices $\tilde{\mathbf{E}}_i$ and $\tilde{\mathbf{A}}_i$ we have to distinguish between two cases depending on the reduced sizes q_i, q_0:

- Case 1: $q_i < q_0$

  Consider the following proposition.

  Proposition 7.5. *The matrices $\mathbf{T}_i$ introduce a common kernel of dimension $q_0 - q_i$ for matrices $\tilde{\mathbf{E}}_i$ and $\tilde{\mathbf{A}}_i$.*

  Proof. Matrix $\mathbf{T}_i \in \mathbb{R}^{q_i \times q_0}$ has a kernel of dimension $q_0 - q_i$. Then, for all vectors $\mathbf{v} \in \ker(\mathbf{T}_i)$ we obtain

$$\begin{aligned}\tilde{\mathbf{E}}_i\mathbf{v} = \mathbf{M}_i^T\hat{\mathbf{E}}_i\mathbf{T}_i\mathbf{v} = \mathbf{M}_i^T\hat{\mathbf{E}}_i\mathbf{0} = \mathbf{0},\\ \tilde{\mathbf{A}}_i\mathbf{v} = \mathbf{M}_i^T\hat{\mathbf{A}}_i\mathbf{T}_i\mathbf{v} = \mathbf{M}_i^T\hat{\mathbf{A}}_i\mathbf{0} = \mathbf{0}.\end{aligned} \tag{7.25}$$

■

  Hence, the matrices $\tilde{\mathbf{E}}_i$ and $\tilde{\mathbf{A}}_i$ are singular. Therefore, they have to be interpolated on the manifold of real matrices.

- Case 2: $q_i \geq q_0$

  In general, the matrices $\tilde{\mathbf{E}}_i$ and $\tilde{\mathbf{A}}_i$ are nonsingular. Hence, we can choose between the manifold of real and nonsingular matrices. However, they can also be singular because either the high-order system matrices $\mathbf{E}_i$ and $\mathbf{A}_i$ are singular or the reduction step is performed improperly. In this case, we are restricted to use the manifold of nonsingular matrices.

7.3.4 Calculation of the Reference Subspaces

So far, the reference subspace which is spanned by $\mathbf{V}_0 \in \mathbb{R}^{n \times q_0}$ was assumed to be given. Now we have enough insight into the proposed approach so that we can motivate the calculation of $\mathbf{V}_0$ and hence, the determination of the reference size q_0. The reference basis should span a subspace which comprises the directions which describe the most important dynamics of all reduced models. For this, we sum up all $\mathbf{V}_i \in \mathbb{R}^{n \times q_i}$ for $i \in \{1, \dots, N\}$ and calculate the economy version of the SVD according to (3.5):

$$\begin{aligned}\mathbf{U}_\mathbf{V}\mathbf{\Sigma}_\mathbf{V}\mathbf{Z}_\mathbf{V}^T = \mathrm{svd}([\mathbf{V}_1, \dots, \mathbf{V}_N], \text{'econ'})\\ \Rightarrow \mathbf{V}_0 = \mathbf{U}_\mathbf{V}(:, 1 : q_0).\end{aligned} \tag{7.26}$$

Then, the first q_0 columns of $\mathbf{U}_\mathbf{V}$ are chosen as reference basis $\mathbf{V}_0$. Another possibility is a weighted SVD approach such as (3.6). In contrast to the case of identical reduced orders from Section 3.4.1, we now have the option to select every order $q_0 \in \{1, \dots, N\}$. In the following, two strategies for determining the reference size q_0 are proposed:

- **Strategy 1:**
  The first strategy aims to capture all important directions of the subspaces. For this, we define a tolerance for the singular values in (7.26) and we take the size q_0 for which the singular values are above that tolerance. This strategy leads to the case $q_i < q_0$ and it results in singular matrices $\tilde{\mathbf{E}}_i$ and $\tilde{\mathbf{A}}_i$ according to Section 7.3.3. Hence, a restriction of this strategy is that we do not have the option to decide between the interpolation manifold of real and nonsingular matrices.

- **Strategy 2:**
  It was demonstrated in Section 4.7 that the selection of the most accurate interpolation manifold is problem-dependent. Therefore, the second strategy aims to obtain nonsingular matrices $\tilde{\mathbf{E}}_i$ and $\tilde{\mathbf{A}}_i$ so that we have the option to decide between the interpolation manifold of real and nonsingular matrices which is the case $q_i \geq q_0$ in Section 7.3.3. For this, we select the smallest reduced order $q_0 = \min\{q_1, \ldots, q_N\}$. A disadvantage of this strategy is that the local systems, which have a reduced order larger than q_0, are reduced to the smaller reference size and hence, we loose accuracy at the grid points.

Due to duality, the reference basis $\mathbf{W}_0$ can be calculated with formula (7.26) only by replacing $\mathbf{V}_i$ with $\mathbf{W}_i$. The size of $\mathbf{W}_0 \in \mathbb{R}^{n \times q_0}$ is determined by the same q_0 from above because different sizes of $\mathbf{V}_0$ and $\mathbf{W}_0$ would lead to under- or overdetermined systems of linear equations (7.23).

7.4 Interpolation Process

We construct the interpolants with the mapped matrices such as described in Section 3.6. As soon as a new reduced system for parameter vector $\mathbf{p}_{\text{int}} \in \mathcal{D}$ is supposed to be calculated, we evaluate the interpolants according to Section 3.7. The resulting matrices are mapped back to the original space and we obtain the interpolated reduced system

$$\tilde{\mathbf{G}}(\mathbf{p}_{\text{int}}) : \begin{cases} \tilde{\mathbf{E}}(\mathbf{p}_{\text{int}})\dot{\tilde{\mathbf{x}}}(t) = \tilde{\mathbf{A}}(\mathbf{p}_{\text{int}})\tilde{\mathbf{x}}(t) + \tilde{\mathbf{B}}(\mathbf{p}_{\text{int}})\mathbf{u}(t) \\ \tilde{\mathbf{y}}(t) = \tilde{\mathbf{C}}(\mathbf{p}_{\text{int}})\tilde{\mathbf{x}}(t) \end{cases} \tag{7.27}$$

where the matrices $\tilde{\mathbf{E}}(\mathbf{p}_{\text{int}}), \tilde{\mathbf{A}}(\mathbf{p}_{\text{int}}) \in \mathbb{R}^{q_0 \times q_0}, \tilde{\mathbf{B}}(\mathbf{p}_{\text{int}}) \in \mathbb{R}^{q_0 \times r}$, $\tilde{\mathbf{C}}(\mathbf{p}_{\text{int}}) \in \mathbb{R}^{m \times q_0}$ have the reference size q_0.

Remark: It was pointed out in Section 7.3.3 that the local system matrices $\tilde{\mathbf{E}}_i, \tilde{\mathbf{A}}_i$ share a common kernel for the case $q_i < q_0$ and hence, are singular. However, the interpolation generally leads to nonsingular matrices $\tilde{\mathbf{E}}(\mathbf{p}_{\text{int}})$, $\tilde{\mathbf{A}}(\mathbf{p}_{\text{int}})$ because every local system adds a part to the dynamics of the interpolated system.

7.5 Numerical Example

The considered example is a model of a cantilever beam with order $n = 1200$ which was introduced in Section 3.9.2. The parameter of the beam is the length $L \in [0.8,\ 1.4]$m. We compute $N = 4$ high-order systems $G_1, \ldots, G_4$ for lengths $L_1 = 0.8$m, $L_2 = 1$m, $L_3 = 1.2$m and $L_4 = 1.4$m. These local systems are reduced using the one-sided Krylov subspace method with expansion points $s_0 = 0$ and reduced orders $q_1 = 14$, $q_2 = 12$, $q_3 = 12$, and $q_4 = 10$ which result in the reduced systems $\hat{G}_1, \ldots, \hat{G}_4$. We choose decreasing reduced orders so that the reduced systems are more accurate for small lengths. For determining the reference size q_0, we compare the two strategies presented in Section 7.3.4. For strategy 1 we choose q_0 to capture all relevant singular values with formula (7.26) up to the relative tolerance of 10^{-1} resulting in $q_0 = 14$. For strategy 2 we choose the smallest reduced order $q_0 = q_4 = 10$ and we use the manifold of nonsingular matrices for interpolating $\tilde{\mathbf{E}}_i, \tilde{\mathbf{A}}_i$. For both strategies we apply linear interpolation. In Figure 7.6, the overall error e_{T} in $\mathcal{H}_2$-norm between the interpolated and the original systems for the two strategies and the MAC and DS approach is shown.

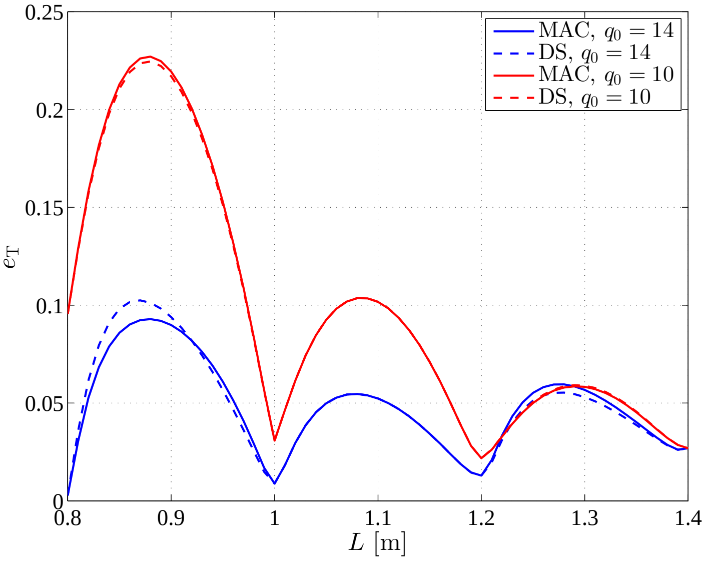


Figure 7.6: Error e_{T} in $\mathcal{H}_2$-norm at various lengths L for the cantilever beam.

The reader can verify that the proposed methods enable a meaningful interpolation of differently-sized reduced models. In this example, strategy 1 performs better than strategy 2 except for a small parameter interval. It can also be seen that there is only a small difference between the MAC and DS approach. For the plots of strategy 2 there is a loss of accuracy at the grid points L_1, L_2, L_3 as the systems are approximated by reduced systems of the smaller order $q = 10$ although systems $\hat{G}_1, \hat{G}_2, \hat{G}_3$ have order $q_1 = 14$, $q_2 = 12$, $q_3 = 12$, respectively. The error increases with growing discrepancy $q_0 - q_i$. As the system $\hat{G}_4$ already possesses order $q_4 = 10$, there is no loss of accuracy.

8 Controller Design by Interpolating Reduced System Matrices

8.1 Motivation

In control applications it is fundamental that a controller fulfills the real-time requirement. However, if a controller is to be designed for a high-dimensional plant model, it might be too complex compared to the power of the processor. As a remedy, one can either employ a more powerful, but also more expensive, hardware or reduce the complexity of the controller. For the sake of cost reduction one might choose the latter.

In order to design low-order controllers that function in real-time, basically two types of methods have been proposed [167, 124, 45, 165]. The first type is based on designing a controller for the high-order plant model and on reducing the controller afterwards e.g. with a $\mathcal{H}_\infty$ controller reduction [80, 123, 167], frequency-weighted balancing related controller reduction [56, 14, 144, 160] or controller reduction based on the rational Krylov method [87]. The second type reduces the high-order plant model and constructs a controller for the reduced model, see e.g. [79, 94]. As the latter approaches do not explicitly consider the effects which the reduction has on the closed-loop stability or performance, it is necessary to check if the desired closed-loop behavior is achieved [14, 167] or even to use an iterative reduction and design procedure [124, 16, 164]. For the control of high-dimensional systems which depend on parameters only few publications can be found. For example in [9], the combination of pMOR and control was proposed in order to optimize aircraft trajectories. Based on this, we propose a more general procedure for the control of parameter-dependent high-order systems.

We reduce the high-dimensional system using the general framework from Section 3 which enables the application of common methods of control unit design and ensures a fast calculation of the controller. The procedure was published in [73]. In this thesis, the proposed method is described in Section 8.2. Numerical and experimental results are presented for a practical example on a test rig for a gantry crane operating with different loads in Section 8.3, where flatness-based tracking control and model predictive control (MPC) is applied to the high-order system.

8.2 Controller Design Using a Database of Reduced Models

Assume a high-dimensional parameter-dependent LTI system $\mathbf{G}(\mathbf{p})$ with domain $\mathbf{p} \in \mathcal{D}$. A controller is to be designed for the system for a given parameter vector using only low-order operations. Therefore, the combination of pMOR by matrix interpolation such as described in Section 3 and common methods of control unit design is proposed. The steps of the procedure are depicted in Figure 8.1. It consists of an offline part that is performed only one time and an online part that is performed every time a controller is supposed to be calculated for a desired parameter $\mathbf{p}_{\text{int}}$.

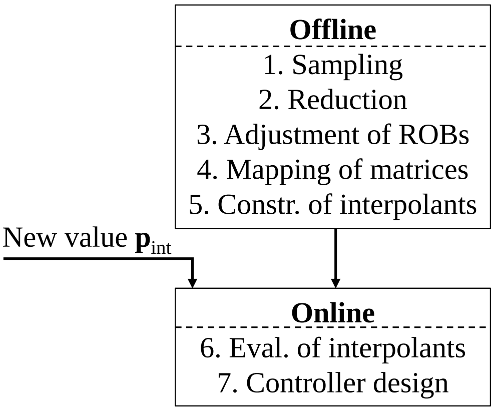


Figure 8.1: Structure including two phases of the controller design.

In the offline phase, steps 1–5 of the general framework are performed such as described in Section 3. To begin with, a set of high-order systems is obtained by inserting parameter values $\mathbf{p}_1, \ldots, \mathbf{p}_N$ into system $\mathbf{G}(\mathbf{p})$. Subsequently, the high-order systems are reduced according to Section 3.3 which results in a database of reduced systems. For these systems the ROBs are adjusted such as proposed in Section 3.4. Finally, the system matrices are mapped onto a matrix manifold such as described in Section 3.5 and the interpolants are constructed according to Section 3.6.

In the online phase, the interpolants are evaluated according to Section 3.7. Then, the matrices are mapped back from the tangent space and they are used to construct the interpolated reduced system (3.36) which is considered as the control plant for parameter $\mathbf{p}_{\text{int}}$. Since it is low-order, it can be used to design a controller or observer using common methods of control unit design, see e.g. [99, 167].

Remark: Note that in general controllability and observability is not guaranteed for a system obtained by matrix interpolation even if the locally reduced systems are controllable and observable. But if the grid points $\mathbf{p}_1, \ldots, \mathbf{p}_N$ have a reasonable distance, we can expect controllability and observability for the interpolated system.

In order to highlight the advantage of the proposed approach, it is compared to a local reduction of the high-order system for a new parameter vector $\mathbf{p}_{\text{int}}$ with projection-based MOR such as described in Section 2.2. The comparison showing the necessary steps of the two approaches for the offline and online part is given in Table 8.1. For a local reduction of the system at a desired parameter vector, the online phase includes computationally expensive MOR followed by the controller design. The advantage of the proposed method is that computational effort is shifted from the online to the offline phase and only operations using low-order matrices are performed in the online phase, which enables a fast controller design for a new parameter value.

Table 8.1: Comparison of the structure of local MOR and pMOR applied to controller design for the system at a desired parameter value.

Method	Local MOR	pMOR
Offline	-	Sampling Reduction Adjustment of ROBs Mapping of matrices Construction of interpolants
Online	Local reduction Controller design	Evaluation of interpolants Controller design

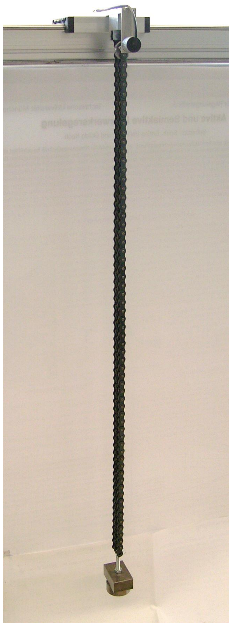


Figure 8.2: Experimental setting of the hanging chain.

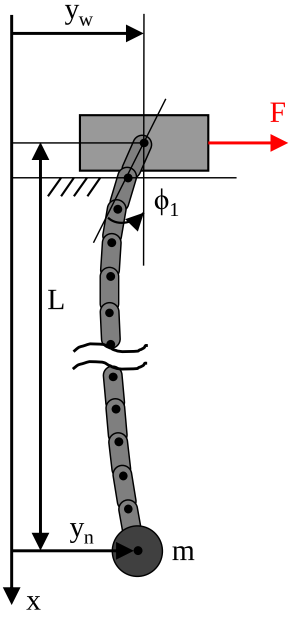


Figure 8.3: Schematic depiction of the hanging chain.

8.3 Control of the Hanging Chain

8.3.1 Experimental Setting

In the following, a hanging chain on a test rig is used as a practical example for demonstrating the proposed controller design process. The experimental setting is shown in Figure 8.2 and a schematic depiction of the system is given in Figure 8.3. The system consists of a car and a chain that is attached to the car. The chain has 62 links, the length $L = 1.2\text{m}$ and a load with mass $m \in [0,\ 2]\text{kg}$ which is located at the tip of the chain y_n. The input of the system is the force F which is applied to the car. Sensors are available for the position of the car y_w and for the angle of the first chain link ϕ_1. The real-time target machine has a 3.60Ghz processor and is operated with a sampling interval of 1ms.

8.3.2 Modeling

The hanging chain is modeled as a multibody system in order to demonstrate the proposed method for a high-order practical example in a lab.[1] The car and the load are assumed to be mass points and the chain links are considered as rigid bars with mass and moment of inertia. Then, the state vector is $\mathbf{x} = (y_\text{w}, \dot{y}_\text{w}, \boldsymbol{\phi}, \dot{\boldsymbol{\phi}}) \in \mathbb{R}^{126}$, where $y_\text{w}, \dot{y}_\text{w}$ are the position and velocity of the car and $\boldsymbol{\phi} = (\phi_1, \dots, \phi_{62})$ is the vector of angles of the chain links. The output $\mathbf{y} = (y_\text{w}, \phi_1, y_\text{n})$ consists of the two measured signals $\mathbf{y}_\text{m} = (y_\text{w}, \phi_1)$ and the position y_n of the load. The mass of the load $m \in [0,\ 2]\text{kg}$ is the parameter of this system. The equations of translational and angular momentum of the car and the chain links deliver the nonlinear model

$$\begin{aligned} \dot{\mathbf{x}}(t) &= f(\mathbf{x}(t), m, F(t)) \\ \mathbf{y}(t) &= h(\mathbf{x}(t), m, F(t)) \end{aligned} \tag{8.1}$$

which is subsequently linearized around the steady-state $\mathbf{x}_0 = (y_\text{w}, 0, \mathbf{0}, \mathbf{0})$ and $F_0 = 0$. This provides the high-order parameter-dependent LTI system[2] of order $n = 126$

$$\mathbf{G}(m) : \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}(m)\mathbf{x}(t) + \mathbf{b}(m)F(t) \\ \mathbf{y}(t) = \mathbf{C}(m)\mathbf{x}(t) \end{cases} \tag{8.2}$$

[1] It shall be noted that for this system there are also other possibilities of modeling. Based on the PDE of a rope, one can also use a modal description, a discretized model or an infinite-dimensional model for constructing the control law, see e.g. [136, 70, 139, 157].

[2] For simplifying the notation, the values $\mathbf{x}, \mathbf{y}, F$ denote the deviation from the steady-state for the remainder of this section.

with $\mathbf{A}(m) \in \mathbb{R}^{126\times126}, \mathbf{b}(m) \in \mathbb{R}^{126\times1}, \mathbf{C}(m) \in \mathbb{R}^{3\times126}$. In order to fulfill the linearity assumption, the constraint on the force $|F| \leq 300\text{N}$ is taken into account.

We apply the proposed method combining controller design with pMOR by matrix interpolation from Section 3. In the offline phase, we choose $N = 9$ uniformly distributed grid points in the parameter space $m \in [0,\ 2]\text{kg}$ with mesh width $\Delta m = 0.25\text{kg}$. Hence, the high-order system $\mathbf{G}(m)$ is computed for the parameter values $m_1 = 0, \dots, m_9 = 2\text{kg}$, which provides nonparametric high-order systems $\mathbf{G}_1, \dots, \mathbf{G}_9$. These systems are reduced to order $q = 12$ using TBR which result in the reduced systems $\hat{\mathbf{G}}_1, \dots, \hat{\mathbf{G}}_9$. The adjustment of the right ROBs is done with the MAC approach (3.16) and of the left ROBs with the MAC approach (3.26) with reference ROBs $\mathbf{V}_0 = \mathbf{V}_1$ and $\mathbf{W}_0 = \mathbf{W}_1$. This leads to reduced systems $\tilde{\mathbf{G}}_1, \dots, \tilde{\mathbf{G}}_9$ according to formula (3.28) with system matrices $\tilde{\mathbf{E}}_i = \mathbf{I}_{12}$ and $\tilde{\mathbf{A}}_i \in \mathbb{R}^{12\times12}, \tilde{\mathbf{b}}_i \in \mathbb{R}^{12\times1}, \tilde{\mathbf{C}}_i \in \mathbb{R}^{3\times12}$ for $i \in \{1, \dots, 9\}$.

As soon as a controller is to be constructed for a new system with parameter value m_{int}, the corresponding system $\tilde{\mathbf{G}}(m_{\text{int}})$ is obtained by the interpolation of the stored low-order matrices $\tilde{\mathbf{A}}_i, \tilde{\mathbf{b}}_i, \tilde{\mathbf{C}}_i$ on the manifold of real matrices with $i \in \{1, \dots, 9\}$:

$$\tilde{\mathbf{G}}(m_{\text{int}}) : \begin{cases} \dot{\tilde{\mathbf{x}}}(t) = \tilde{\mathbf{A}}(m_{\text{int}})\tilde{\mathbf{x}}(t) + \tilde{\mathbf{b}}(m_{\text{int}})F(t) \\ \tilde{\mathbf{y}}(t) = \tilde{\mathbf{C}}(m_{\text{int}})\tilde{\mathbf{x}}(t) \end{cases} \tag{8.3}$$

with

$$\begin{aligned} \tilde{\mathbf{A}}(m_{\text{int}}) &= \textstyle\sum_{i=1}^{N} \omega_i(m_{\text{int}})\tilde{\mathbf{A}}_i, \\ \tilde{\mathbf{b}}(m_{\text{int}}) &= \textstyle\sum_{i=1}^{N} \omega_i(m_{\text{int}})\tilde{\mathbf{b}}_i, \\ \tilde{\mathbf{C}}(m_{\text{int}}) &= \textstyle\sum_{i=1}^{N} \omega_i(m_{\text{int}})\tilde{\mathbf{C}}_i \end{aligned} \tag{8.4}$$

with $\tilde{\mathbf{y}} = (\tilde{y}_{\text{w}}, \tilde{\phi}_1, \tilde{y}_{\text{n}}) \approx \mathbf{y}$ and $\tilde{\mathbf{A}}(m_{\text{int}}) \in \mathbb{R}^{12\times12}, \tilde{\mathbf{b}}(m_{\text{int}}) \in \mathbb{R}^{12\times1}, \tilde{\mathbf{C}}(m_{\text{int}}) \in \mathbb{R}^{3\times12}$. In this example, linear weighting functions $\omega_i(m) = \max(1 - \frac{|m-m_i|}{\Delta m}, 0)$ for $m \in [m_1,\ m_9]$ with $i \in \{1, \dots, 9\}$ are used. They are shown in Figure 8.4.

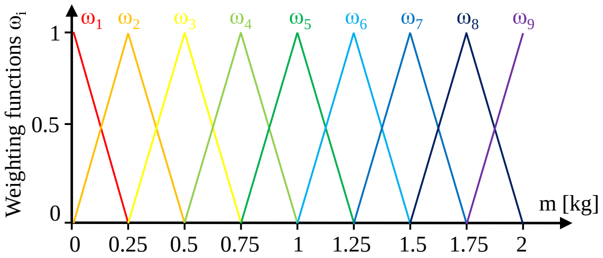


Figure 8.4: Weighting functions $w_i := w_i(m)$ corresponding to points $m_1, \dots, m_9$.

8.3.3 Control Task

A controller is to be designed using the interpolated system (8.3) which enables the change of the steady-state point $\mathbf{x}(t_\mathrm{S}) = \mathbf{x}(0) = (0, 0, \mathbf{0}, \mathbf{0}) \to \mathbf{x}(t_\mathrm{E}) = (0.5\mathrm{m}, 0, \mathbf{0}, \mathbf{0})$ with $t \in [0,\ 5]$s and $t_\mathrm{S} = 1$s for the hanging chain with mass m_int. It is required that

- the end time t_E is as small as possible.
- the constraint on the force $|F| \leq 300$N is taken into account.
- the chain is free of oscillations as soon as it reaches its target.

For this, we set the following reference trajectory $y_\mathrm{n,ref}$ for the control variable $\tilde{y}_\mathrm{n} \approx y_\mathrm{n}$

$$y_\mathrm{n,ref}(t) = 0.5\mathrm{m}\,\sigma(t - t_\mathrm{S}) \tag{8.5}$$

where $\sigma(t)$ is the Heaviside step function. In the following we consider two control concepts which are expected to fulfill the requirements of the control task: a two-degree-of-freedom control in Section 8.3.4 and MPC in Section 8.3.5.

8.3.4 Two-Degree-of-Freedom Control

Controller design

At first, a two-degree-of-freedom control which is shown in Figure 8.5 is designed. It consists of a feedforward trajectory control and a feedback loop including a controller in order to tackle deviations from the trajectory. Additionally, an observer is implemented in order to estimate the reduced state vector.

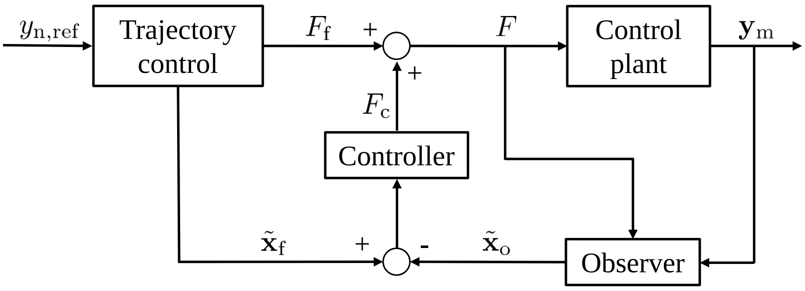


Figure 8.5: Structure of a two-degree-of-freedom control with a trajectory generation and a feedback loop including an observer and a controller.

Firstly, the feedforward trajectory control is designed. One prominent method for this is a flatness-based tracking control, see e.g [166]. As the interpolated LTI system (8.3) is found to be controllable, a flat output z_f is defined as

$$z_\mathrm{f} = \boldsymbol{\lambda}(m_\mathrm{int})^T \tilde{\mathbf{x}}_\mathrm{f} \tag{8.6}$$

where $\tilde{\mathbf{x}}_\mathrm{f}$ is the flatness-based state vector of the plant model and $\boldsymbol{\lambda}(m_\mathrm{int})^T$ is taken as the last row of the inverse controllability matrix

$$\boldsymbol{\lambda}(m_\mathrm{int})^T = [0, \dots, 0, 1][\tilde{\mathbf{b}}(m_\mathrm{int}), \tilde{\mathbf{A}}(m_\mathrm{int})\tilde{\mathbf{b}}(m_\mathrm{int}), \dots, \tilde{\mathbf{A}}(m_\mathrm{int})^{q-1}\tilde{\mathbf{b}}(m_\mathrm{int})]^{-1}. \tag{8.7}$$

Then, the vector containing the flatness coordinates is

$$\tilde{\mathbf{x}}_\mathrm{f}^* = \begin{pmatrix} z_\mathrm{f} \\ \vdots \\ \overset{(q-1)}{z_\mathrm{f}} \end{pmatrix} = \underbrace{\begin{pmatrix} \boldsymbol{\lambda}(m_\mathrm{int})^T \\ \vdots \\ \boldsymbol{\lambda}(m_\mathrm{int})^T \tilde{\mathbf{A}}(m_\mathrm{int})^{q-1} \end{pmatrix}}_{\mathbf{T}_\mathrm{f}(m_\mathrm{int})} \tilde{\mathbf{x}}_\mathrm{f}. \tag{8.8}$$

The flatness-based output $\tilde{y}_\mathrm{f}$ of the plant is with $\tilde{\mathbf{c}}_\mathrm{c}(m_\mathrm{int})^T = \tilde{\mathbf{C}}(m_\mathrm{int})(3,:)$ as the third row of the output matrix which corresponds to the control variable

$$\tilde{y}_\mathrm{f} = \underbrace{\tilde{\mathbf{c}}_\mathrm{c}(m_\mathrm{int})^T \mathbf{T}_\mathrm{f}(m_\mathrm{int})^{-1}}_{\tilde{\mathbf{c}}_\mathrm{c}^*(m_\mathrm{int})^T} \tilde{\mathbf{x}}_\mathrm{f}^*. \tag{8.9}$$

Next, the trajectory of the flat output z_f, which has to be differentiable $q = 12$ times, is calculated with a polynomial approach of degree $2q + 1 = 25$ in the time interval $t \in [t_\mathrm{S},\ t_\mathrm{E}]$, see [166]. This is obtained by allocating initial and final conditions with $t_\mathrm{E} = t_\mathrm{S} + T_\mathrm{f}$ to the flat output and its derivatives. As we use steady-state working points for the initial and final conditions, the derivatives of the flat output are found to be zero with $\overset{(1)}{z}(t_\mathrm{S}) = \overset{(1)}{z}(t_\mathrm{E}) = \ldots = \overset{(q)}{z}(t_\mathrm{S}) = \overset{(q)}{z}(t_\mathrm{E}) = 0$. Then, the remaining initial and final conditions for the flat output can by calculated using (8.9) with

$$z_\mathrm{f}(t_\mathrm{S}) = \frac{\tilde{y}_\mathrm{f}(t_\mathrm{S})}{\tilde{\mathbf{c}}_{\mathrm{c},1}^*(m_\mathrm{int})^T} = 0, \tag{8.10}$$

$$z_\mathrm{f}(t_\mathrm{E}) = \frac{\tilde{y}_\mathrm{f}(t_\mathrm{E})}{\tilde{\mathbf{c}}_{\mathrm{c},1}^*(m_\mathrm{int})^T} = \frac{0.5\mathrm{m}}{\tilde{\mathbf{c}}_{\mathrm{c},1}^*(m_\mathrm{int})^T} \tag{8.11}$$

where $\tilde{\mathbf{c}}_{\mathrm{c},1}^*(m_\mathrm{int})^T$ is the first entry of $\tilde{\mathbf{c}}_\mathrm{c}^*(m_\mathrm{int})^T$. Finally, the trajectory for the input is

$$F_\mathrm{f} = -\boldsymbol{\lambda}(m_\mathrm{int})^T \tilde{\mathbf{A}}(m_\mathrm{int})^q \mathbf{T}(m_\mathrm{int})^{-1} \mathbf{x}_\mathrm{f}^* + \overset{(q)}{z_\mathrm{f}}. \tag{8.12}$$

The left-over design variable T_f is iteratively increased until the flatness-based feedforward control meets the constraint $|F_\mathrm{f}| \leq 300\mathrm{N}$.

Secondly, the control law of the state feedback controller with a constant feedback gain $\mathbf{r}_\mathrm{c}(m_\mathrm{int})^T \in \mathbb{R}^{1\times 12}$, see e.g. [99], is

$$F_\mathrm{c} = \mathbf{r}_\mathrm{c}(m_\mathrm{int})^T(\tilde{\mathbf{x}}_\mathrm{f} - \tilde{\mathbf{x}}_\mathrm{o}), \tag{8.13}$$

where $\tilde{\mathbf{x}}_\mathrm{o}$ is the state estimate provided by an observer. We use a linear-quadratic regulator which is obtained by solving the Riccati equation using the interpolated plant (8.3). The equation of the observer is

$$\dot{\tilde{\mathbf{x}}}_\mathrm{o} = \Big(\tilde{\mathbf{A}}(m_\mathrm{int}) - \mathbf{L}_\mathrm{o}(m_\mathrm{int})\tilde{\mathbf{C}}_\mathrm{m}(m_\mathrm{int})\Big)\tilde{\mathbf{x}}_\mathrm{o} + \tilde{\mathbf{b}}(m_\mathrm{int})F + \mathbf{L}_\mathrm{o}(m_\mathrm{int})\mathbf{y}_\mathrm{m}, \tag{8.14}$$

where $\mathbf{L}_\mathrm{o}(m_\mathrm{int}) \in \mathbb{R}^{12\times 2}$ is the observer gain which we choose according to the Kalman filter design, see e.g. [99], and $\tilde{\mathbf{C}}_\mathrm{m}(m_\mathrm{int}) = \tilde{\mathbf{C}}(m_\mathrm{int})(1:2,:)$ are the first two rows of the output matrix which correspond to the measured signals.

Numerical and experimental results

In this section numerical and experimental results for maneuver (8.5) are presented using the two-degree-of-freedom control for the mass of the load $m_\mathrm{int} = 1.6\mathrm{kg}$, where the design variable is calculated to be $T_\mathrm{f} = 3.6\mathrm{s}$.

Firstly, simulation results of the chain using two different controllers which both drive the nonlinear system (8.1) are compared. The movement of the chain and of the load is shown in Figure 8.6, where the time is represented by the transition from transparent to nontransparent. The red line illustrates the chain driven by the controller that is designed based on the interpolated system, whereas for the chain represented by the black line the controller is constructed based on a directly reduced system for $m_\mathrm{int} = 1.6\mathrm{kg}$ and $q = 12$ using TBR. One can verify there is a relatively small error due to interpolation which is indicated by the discrepancy between the red and black line. One can also see that there is no oscillation at the end of the maneuver.

Secondly, results for maneuver (8.5) obtained by a simulation with the nonlinear model and measurement results obtained for the system on the test rig are compared for the case where the controller is designed using the interpolated reduced system. The simulated and measured force F is compared in Figure 8.7. It can be seen that the force does not violate the constraint of $|F| \leq 300\mathrm{N}$. In Figure 8.8 and 8.9 the measured position of the car y_w and the measured angle of the first chain link ϕ_1 are compared

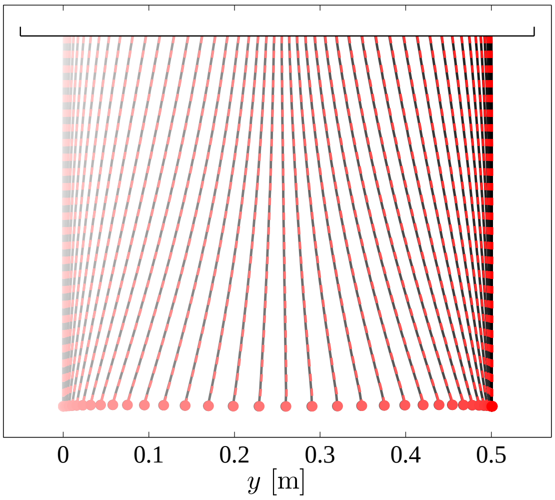


Figure 8.6: Simulation results of the chain for the maneuver from 0m to 0.5m comparing two controllers which both drive the nonlinear model: a controller designed by using the directly reduced system (black) and one by using the interpolated reduced system (red) at parameter value $m_{\mathrm{int}} = 1.6$kg.

to their simulated counterparts, respectively. One can verify that the simulated and measured signals match well. Additionally, the position of the load y_{n} obtained by the simulation is plotted in Figure 8.8. It reaches its target without oscillations which can also be observed in the experiment. To conclude, the controller based on the interpolated model fulfills the requirements of maneuver (8.5).

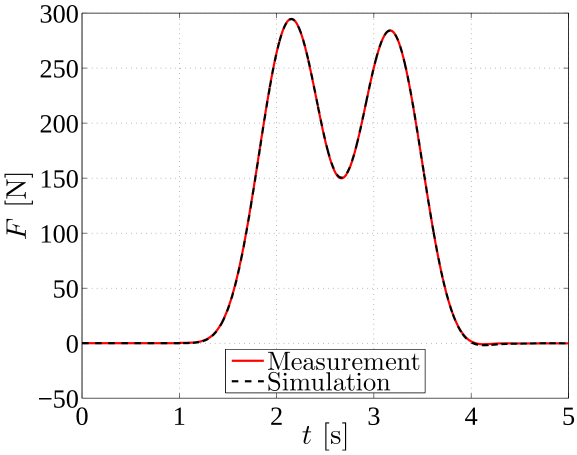


Figure 8.7: Comparison of the force obtained from simulation with the nonlinear model and from measurement on the test rig in each case using the two-degree-of-freedom control designed with the interpolated model at parameter value $m_{\mathrm{int}} = 1.6$kg.

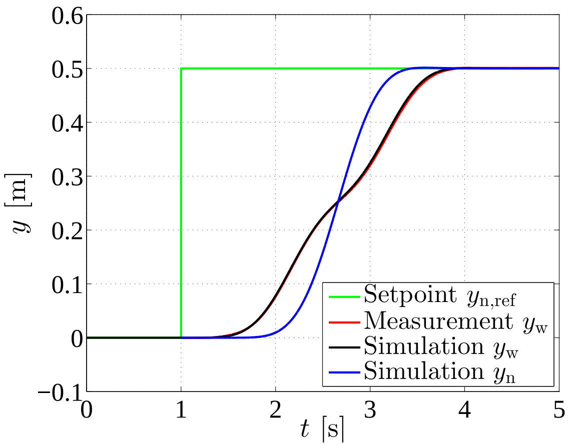


Figure 8.8: Comparison of the position of the car from simulation and measurement and simulated position of the load.

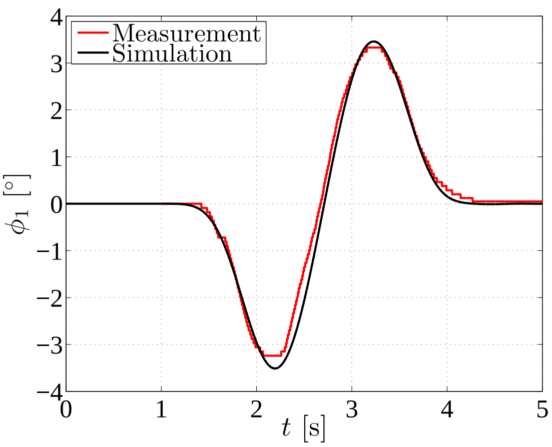


Figure 8.9: Comparison of the angle of the first chain link from simulation and measurement.

8.3.5 Model Predictive Control

Controller Design

An alternative to the two-degree-of-freedom control is a feedback control whose structure is depicted in Figure 8.10. It consists of a model predictive controller and an observer in order to estimate the reduced state vector. For the latter a Kalman filter is implemented according to Section 8.3.4. The advantages of MPC are, firstly, that constraints can be directly incorporated into the controller design process and, secondly, that it reacts to future values of the reference trajectory.

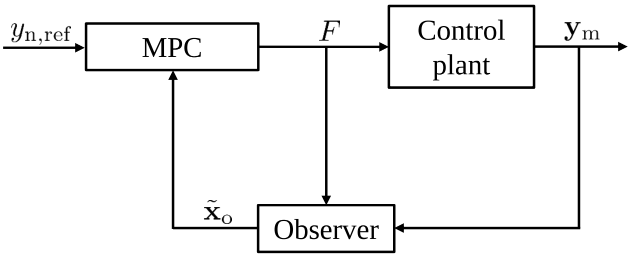


Figure 8.10: Structure of a feedback loop including a model predictive controller and an observer.

For the design of a model predictive controller please refer to e.g. [114, 40]. We use the following design process: Firstly, the interpolated model (8.3) is converted to a discrete-time LTI state-space system with discrete time variable $k \in \mathbb{Z}_0^+$

$$\begin{aligned} \tilde{\mathbf{x}}(k+1) &= \tilde{\mathbf{A}}_\mathrm{d}(m_\mathrm{int})\tilde{\mathbf{x}}(k) + \tilde{\mathbf{b}}_\mathrm{d}(m_\mathrm{int})F(k) \\ \tilde{y}_\mathrm{n}(k) &= \tilde{\mathbf{c}}_\mathrm{d}(m_\mathrm{int})^T\tilde{\mathbf{x}}(k) \end{aligned} \tag{8.15}$$

with $\tilde{\mathbf{c}}_\mathrm{d}(m_\mathrm{int})^T = \tilde{\mathbf{C}}(m_\mathrm{int})(3,:)$ as the third row of the output matrix corresponding to the control variable. The current state is provided by the observer. Secondly, a cost function is minimized subject to some constraints for current and future time steps up to the prediction horizon $N_\mathrm{p} = 850$. Here, a quadratic cost function with weighting matrices $\mathbf{Q}_\mathrm{MPC} = 16\mathbf{I}_{N_\mathrm{p}}$ and $\mathbf{R}_\mathrm{MPC} = \mathbf{I}_{N_\mathrm{p}}$ is considered:

$$J_k = (\tilde{\mathbf{y}}_{\mathrm{n},k} - \mathbf{y}_{\mathrm{n,ref},k})^T\mathbf{Q}_\mathrm{MPC}(\tilde{\mathbf{y}}_{\mathrm{n},k} - \mathbf{y}_{\mathrm{n,ref},k}) + \Delta\mathbf{F}_k^T\mathbf{R}_\mathrm{MPC}\Delta\mathbf{F}_k. \tag{8.16}$$

The predicted output $\tilde{\mathbf{y}}_{\mathrm{n},k}$ and input signal $\mathbf{F}_k$ are

$$\tilde{\mathbf{y}}_{\mathrm{n},k} := \Big(\tilde{y}_\mathrm{n}(k+1), \ldots, \tilde{y}_\mathrm{n}(k+N_p)\Big)^T \tag{8.17}$$

$$\mathbf{F}_k := \Big(F(k), \ldots, F(k+N_p-1)\Big)^T. \tag{8.18}$$

The vector $\mathbf{y}_{\mathrm{n,ref},k}$ is the reference trajectory and $\Delta\mathbf{F}_k = \mathbf{F}_k - \mathbf{F}_{k-1}$ is the control deviation for the prediction horizon. The following constraints are applied to the input and control deviation

$$\begin{aligned} (-300\mathrm{N}, \ldots, -300\mathrm{N}) &\leq \mathbf{F}_k \leq (300\mathrm{N}, \ldots, 300\mathrm{N}) \\ (-0.7\mathrm{N}, \ldots, -0.7\mathrm{N}) &\leq \Delta\mathbf{F}_k \leq (0.7\mathrm{N}, \ldots, 0.7\mathrm{N}). \end{aligned} \tag{8.19}$$

In every time step, the optimization problem (8.16) and (8.19) delivers the solution $\Delta\mathbf{F}_k^*$. In order to reduce the computational effort, only the first $N_\mathrm{c} = 10$ deviations are assumed to have a non-zero value, where N_c is the control horizon. Then, the input signal $\mathbf{F}_k^*$ is calculated and the first entry $F^*(k)$ is applied to the system.

Numerical and Experimental Results

Numerical and experimental results for maneuver (8.5) are presented using the feedback control for the mass of the load $m_\mathrm{int} = 0.6\mathrm{kg}$.

Firstly, simulation results of the chain using again two different controllers which both drive the nonlinear system (8.1) are compared. The movement of the chain and load varying in time is shown in Figure 8.11. The red line illustrates again the chain driven by the controller that is designed based on the interpolated system, whereas for the chain represented by the black line the controller is constructed based on a directly reduced system for $m_{\text{int}} = 0.6\text{kg}$ and $q = 12$ using TBR. One can verify that there is a relatively small error due to interpolation which is indicated by the discrepancy between the red and black line and which grows in time. One can see there is a small oscillation at the end of the maneuver.

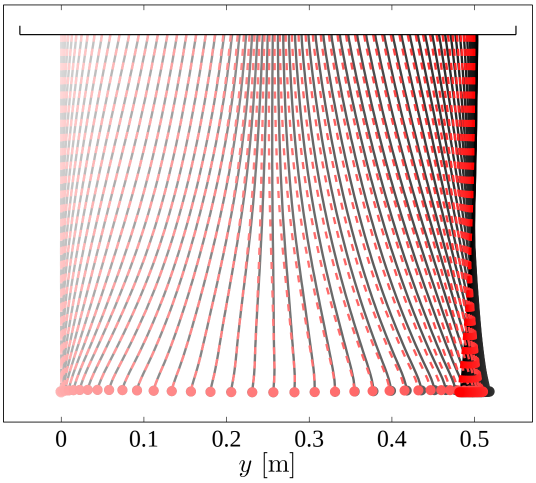


Figure 8.11: Simulation results of the chain for the maneuver from 0m to 0.5m comparing two controllers which both drive the nonlinear model: a controller designed by using the directly reduced system (black) and one by using the interpolated reduced system (red) at parameter value $m_{\text{int}} = 0.6\text{kg}$.

Secondly, results for maneuver (8.5) obtained by a simulation with the nonlinear model and measurement results obtained for the system on the test rig are compared for the case where the feedback control is designed using the interpolated reduced system. The simulated and measured force F is compared in Figure 8.12. It can be seen that the force does not violate the constraint of $|F| \leq 300\text{N}$. In Figure 8.13 and 8.14 the measured position of the car y_{w} and the measured angle of the first chain link ϕ_1 are compared to their simulated counterparts, respectively. One can verify that the simulated and measured signals match well. The car starts moving 0.85s, which is the prediction horizon, before the change of the setpoint from $y_{\text{n,ref}} = 0\text{m}$ to $y_{\text{n,ref}} = 0.5\text{m}$ at $t_{\text{S}} = 1\text{s}$. Hence, the target value is reached earlier than for the case of the flatness-based approach, but it is computationally more expensive as in every time step an optimization problem needs to be solved. Additionally, the position of the load y_{n} obtained by the simulation is plotted in Figure 8.13. One can see for the simulation there is one oscillation at the tip until it reaches its target which can also be seen in Figure 8.11 and which is confirmed in the experiment. To conclude, the controller based on the interpolated model fulfills the requirements of maneuver (8.5).

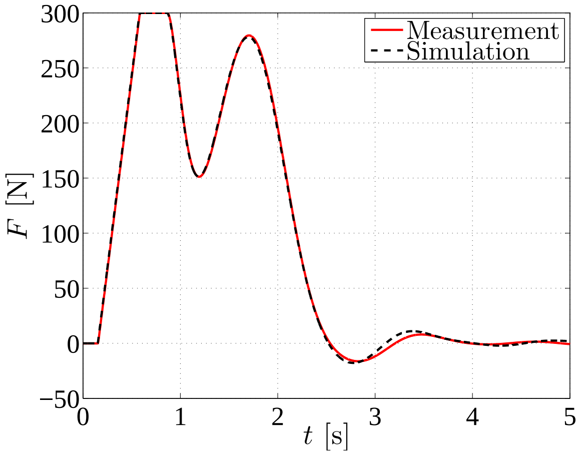


Figure 8.12: Comparison of the force obtained from simulation with the nonlinear model and from measurement on the test rig in each case using the model predictive controller designed with the interpolated model at parameter value $m_{\text{int}} = 0.6$kg.

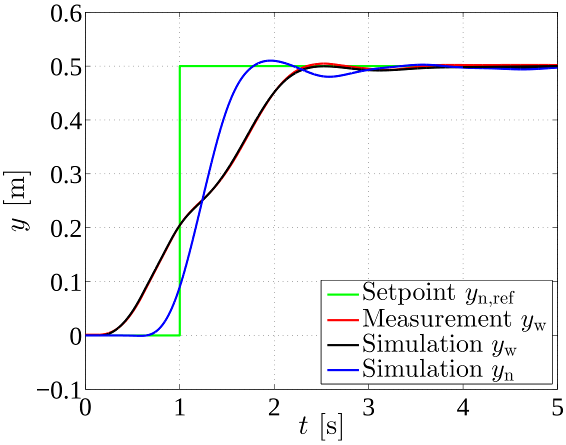


Figure 8.13: Comparison of the position of the car from simulation and measurement and simulated position of the load.

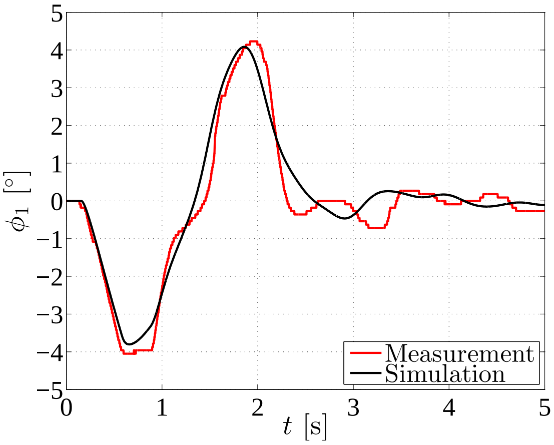


Figure 8.14: Comparison of the angle of the first chain link from simulation and measurement.

9 Conclusions

9.1 Summary

This thesis deals with model order reduction of parameter-dependent LTI systems. The considered approach is based on the interpolation of reduced system matrices obtained for a set of grid points. A general framework was proposed which can be applied by the user as a construction kit for pMOR by matrix interpolation. It illustrates the steps and presents the different options for each step. In most cases, the user does not have enough insight into the physics of the model in order to decide on the options leading to the most accurate reduced model. As a remedy, a Black-Box method was presented which automatically determines the best interpolation method and the optimal grid points in the parameter space and, additionally, monitors the accuracy of the reduced model. Hence, this automatic approach enables users without deep knowledge of pMOR to obtain a parameter-dependent reduced system of a desired accuracy using as few grid points as possible. The benefit of this method was demonstrated to decrease the simulation time for examples arising in microsystems technology, mechatronics and structural mechanics. It was pointed out that the interpolation of system matrices does not necessarily lead to asymptotically stable systems even if the locally reduced systems are asymptotically stable. Hence, a stability-preserving method based on semidefinite programming was presented which minimally modifies the ROBs compared to the ones which are adjusted for accuracy reasons. Since common grid-based approaches suffer from the curse of dimensionality, leading to costs that grow exponentially with the number of parameters, sparse-grid-based interpolation was introduced for pMOR by matrix interpolation. This enables us to apply interpolation procedures for high-dimensional parameter spaces. In addition, the general framework was extended for the case that the locally reduced system can have different sizes. For this, the system matrices are resized using pseudoinverses for a meaningful interpolation. Finally, it was demonstrated using an example at a test rig that pMOR by matrix interpolation can easily be used for controller design of parametric high-order systems.

It was pointed out in this thesis that the transformation matrices $\mathbf{T}_i$ and $\mathbf{M}_i$ play a crucial role in the proposed framework. Hence, the possible choices are summarized in the following. The matrices $\mathbf{T}_i$ are calculated so that the right ROBs of the local systems are adjusted for accuracy reasons using the DS/MAC/PS approach (3.8)/(3.16)/(3.13). With the calculation of $\mathbf{M}_i$ we can enforce one of the following objectives:

- O1) It can be used to adjust the left ROBs of the local systems for accuracy reasons with the DS/MAC/PS approach (3.23)/(3.24)/(3.25).

- O2) One can use STABLE which delivers $\mathbf{M}_i = \mathbf{P}_i\hat{\mathbf{E}}_i\mathbf{T}_i$. It minimally modifies the choice from O1) so that stability preservation is guaranteed.

- O3) The matrices $\mathbf{M}_i = (\hat{\mathbf{E}}_i\mathbf{T}_i)^{-T}$ from equation (3.26) reduce the number of interpolants and hence, the computational effort as it leads to $\tilde{\mathbf{E}}_i = \mathbf{I}_q$.

Due to duality, we can calculate matrices $\mathbf{M}_i$ so that the left ROBs are adjusted for accuracy reasons with the DS/MAC/PS approach. Then, matrices $\mathbf{T}_i$ can be computed so that one of the objectives above is satisfied.

9.2 Outlook

It was pointed out in Section 2.3.2 that interpolating locally reduced system matrices is a very promising approach for the reduction of parameter-dependent systems. However, a lot of open questions were raised such as stability preservation or the interpolation of differently-sized models. In this thesis, solutions were developed for all these issues. In addition, an efficient method selection and refinement algorithm was presented. These methods were proposed for the interpolation of system matrices. However, they can directly be applied to other local bases approaches such as the interpolation of transfer functions or the interpolation of the ROBs because the presented algorithms are independent of the interpolated values.

As a next step, model order reduction of nonlinear systems based on the interpolation of locally reduced system matrices can be examined. First approaches are proposed in [112, 104]. However, the authors found that there are many design parameters which confront the user with difficulties to apply this method. The results presented in this thesis for the reduction of parameter-dependent systems might contribute to the development of a self-contained reduction method for nonlinear systems. Intermediate steps for developing a Black-Box method with stability preservation could be:

- A special case of nonlinear systems are switched systems. Instead of interpolating the locally reduced systems, we switch between them with regard to a time-dependent switching signal. First results for the reduction of switched systems with stability preservation are proposed in [74]. This method is influenced by the results presented in Section 5.

- In the next step, the reduction of systems whose matrices comprise time-dependent parameters can be considered. First results are presented in [156, 63, 47] which trace the interpolation of locally reduced system matrices at different time points back to a spatial sampling of the parameter domain. Hence, it seems promising to apply the Black-Box method presented in this thesis for an automatic reduction of these systems.

Appendix A

Interpolation

A.1 Interpolation Problem

Consider a d-dimensional function $f : \mathcal{D} \to \mathbb{R}$ with domain $\mathcal{D} \subseteq \mathbb{R}^d$ that is to be interpolated [50, 43]. Assume one has sampled f at the grid points $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\} \subset \mathcal{D}$ to obtain the values $y_1, \ldots, y_N \in \mathbb{R}$ where $y_i = f(\mathbf{p}_i)$ for each $i \in \{1, \ldots, N\}$. From the data $\{(\mathbf{p}_1, y_1), \ldots, (\mathbf{p}_N, y_N)\}$ an approximation $\hat{f} : \mathcal{D} \to \mathbb{R}$ with $\hat{f}(\mathbf{p}) = \sum_{i=1}^{N} \alpha_i \omega_i(\mathbf{p})$ is constructed with basis functions $\omega_1, \ldots, \omega_N : \mathcal{D} \to \mathbb{R}$ and coefficients $\alpha_1, \ldots, \alpha_N \in \mathbb{R}$ that satisfies the interpolation condition:

$$\hat{f}(\mathbf{p}_i) = y_i \;\; \forall \, i \in \{1, \ldots, N\}. \tag{A.1}$$

A special class of basis functions is given in the following definition.

Definition A.1 ([59])**.** The basis functions $\omega_1(\mathbf{p}), \ldots, \omega_N(\mathbf{p})$ are called *cardinal basis functions* if they fulfill

$$\omega_i(\mathbf{p}_j) = \delta_{ij} = \begin{cases} 1, & \text{if } j = i \\ 0, & \text{if } j \neq i. \end{cases} \tag{A.2}$$

A.2 Gridded Data Interpolation

Introduction

Gridded data interpolation relies on an ordered relationship among the grid points which are indexed along the coordinate directions [151]. Assume that the u-th direction includes N_u points which are arranged as follows:

$$p_1^u < p_2^u < \ldots < p_{N_u}^u \;\; \forall \, u \in \{1, \ldots, d\}. \tag{A.3}$$

Then, the grid is defined by arranging all dimensions

$$\mathcal{P} = \{p_1^1, \dots, p_{N_1}^1\} \times \dots \times \{p_1^d, \dots, p_{N_d}^d\}. \tag{A.4}$$

The resulting grid is called a *regular grid* for which an example is given in Figure A.1. A special case is the *full grid* for which the neighboring points in every direction have equal spacing. An example for a full grid is given in Figure A.2.

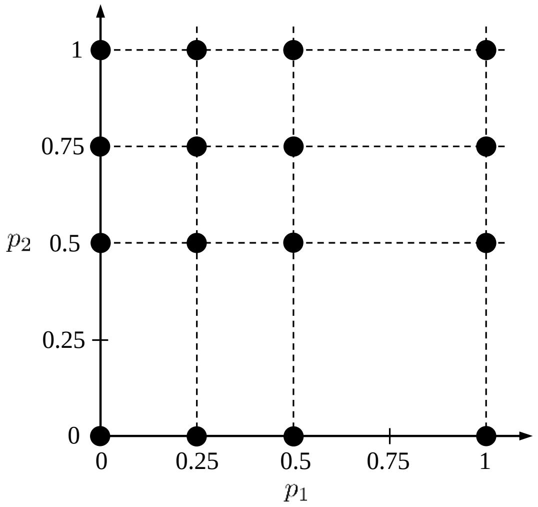


Figure A.1: Regular grid with two parameters p_1, p_2 and refinement in the top left corner.

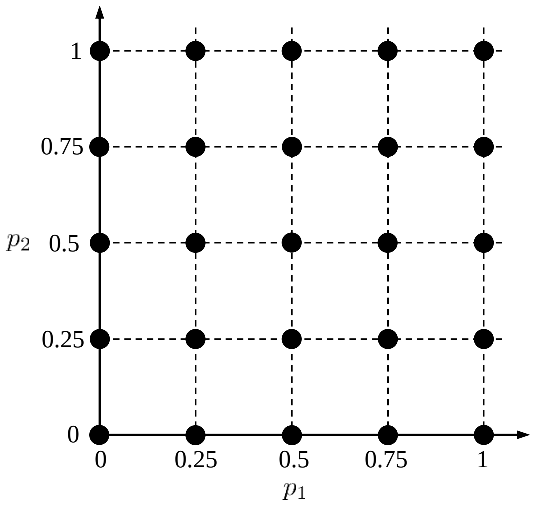


Figure A.2: Full grid with two parameters p_1, p_2 and grid mesh width $\Delta p_1 = \Delta p_2 = 0.25$.

Application

We can use a regular grid if the domain allows the rectangular structure of the grid. This is the case if the domain is rectangular itself and if there are no constraints on the domain which prohibit the regularity of the grid.

Discussion

A regular grid has the beneficial property that the gridded structure allows to locate the neighbor points of a grid point easily because every point has a fixed number of predefined neighbors. Hence, the distance between two points can easily be obtained. In addition, it allows to define operations such as the derivative. A disadvantage of such a grid is the difficulty to keep its structure during a refinement procedure. This can be observed in Figure A.1, where we wish local refinement in the top left corner. However, we must insert additional points to preserve the rectangular structure of the grid. In the following, two approaches for gridded data interpolation are reviewed by means of a 1-dimensional parameter space.

A.2.1 Polynomial Interpolation

A set of N distinct data points $\{(p_1, y_1), \dots, (p_N, y_N)\}$ is given. We are looking for a polynomial of degree at most $N-1$

$$\hat{f}(p) = \sum_{i=1}^{N} \alpha_i p^{i-1} \tag{A.5}$$

fulfilling the interpolation condition (A.1). The resulting system of linear equations has a unique solution and delivers the coefficients $\alpha_1, \dots, \alpha_N$ [152, 3]. The interpolation polynomial in the Lagrange form is given by

$$\hat{f}(p) = \sum_{i=1}^{N} y_i \omega_i(p), \tag{A.6}$$

where the coefficients are the function values at the grid points $y_1, \dots, y_N$ and the basis functions are given by

$$\omega_i(p) = \prod_{j=1, j\neq i}^{N} \frac{p - p_j}{p_i - p_j} \quad \forall\, i \in \{1, \dots, N\}. \tag{A.7}$$

The Lagrange basis functions fulfill the property of cardinal basis functions (A.2).

A.2.2 Piecewise Polynomial Interpolation

One can also look for a piecewise polynomial in each interval $p \in [p_i,\ p_{i+1}]$ [152, 3]. A common choice is the linear spline which represents a set of line segments between two neighboring data points. The linear spline interpolant can be represented by

$$\hat{f}(p) = \sum_{i=1}^{N} y_i \omega_i(p), \tag{A.8}$$

where the coefficients are the function values at the grid points $y_1, \dots, y_N$ and the basis functions for $i \in \{1, \dots, N\}$ are given by

$$\omega_i(p) = \begin{cases} \frac{p - p_{i-1}}{p_i - p_{i-1}}, & \text{if } p \in [p_{i-1},\ p_i] \\ \frac{p_{i+1} - p}{p_{i+1} - p_i}, & \text{if } p \in [p_i,\ p_{i+1}]. \end{cases} \tag{A.9}$$

These basis functions are often referred to as *hat functions*. They fulfill the property of cardinal basis functions (A.2). Another common choice is the cubic spline which constructs in each interval $k \in \{1, \ldots, N-1\}$ a cubic polynomial:

$$\hat{f}^k(p) = \sum_{j=1}^{4} \alpha_j^k (p - p_i)^{j-1}. \tag{A.10}$$

The polynomials need to fulfill the interpolation condition (A.1) and they need to be two times differentiable at the interior nodes. Together with two boundary conditions, this leads to a strictly diagonally dominant system of linear equations. Its solution delivers the coefficients $\alpha_1^k, \ldots, \alpha_4^k$ for each k.

A.3 Scattered Data Interpolation

Introduction

Scattered data interpolation assumes a set of points which have no structure or order between their relative locations [163]. A corresponding grid is called an *irregular grid* for which an example is given in Figure A.3. A grid whose grid points have approximately the same distance from each other and capture all regions of the domain $\mathcal{D}$ is referred to as *space-filling design* for which an example is shown in Figure A.4. There are various methods for space-filling designs such as Latin hypercube sampling [116], Halton or Sobol quasi-random sequence [92, 149].

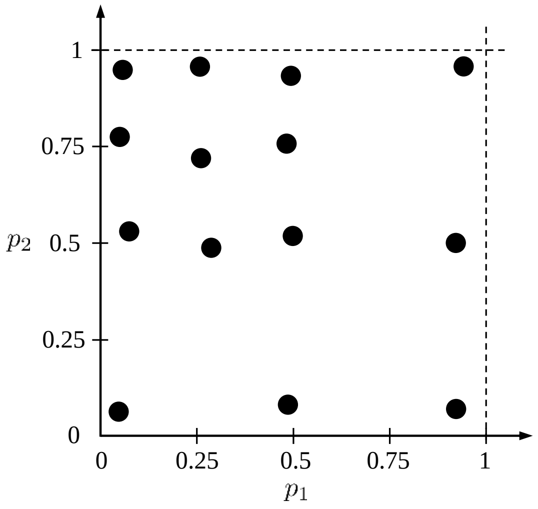


Figure A.3: Irregular grid with two parameters p_1, p_2 and refinement in the top left corner.

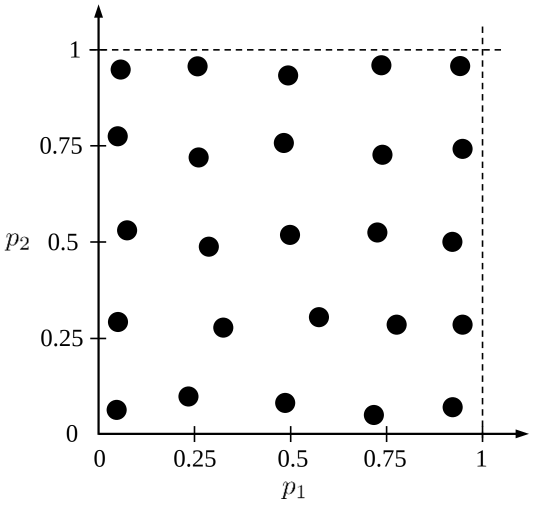


Figure A.4: Irregular grid with two parameters p_1, p_2 and a space-filling design.

Application

We can use irregular grids for every set of grid points. Hence, there are no demands on the domain. The domain does not have to be rectangular and can have any constraints. For that reason, we can use irregular grids in situations where it is not possible to have a regular grid. This is for example the case for models arising in meteorology because weather measurements are only available at irregularly located observation stations.

Discussion

Irregular grids have the advantage that they can easily be applied for multidimensional parameter spaces. In addition, they can easily take constraints on domain $\mathcal{D}$ into account. A disadvantage is that the grid points do not have predefined neighbor points and hence, operations such as calculating the distance are expensive or even not possible such as calculating a derivative. In the following, two approaches for scattered data interpolation are presented.

A.3.1 Radial Basis Functions

Consider again a d-dimensional function $f : \mathcal{D} \to \mathbb{R}$ that we want to interpolate using the scattered data $\{(\mathbf{p}_1, y_1), \ldots, (\mathbf{p}_N, y_N)\}$. The interpolant of the radial basis function (RBF) scheme is

$$\hat{f}(\mathbf{p}) = \sum_{i=1}^{N} \alpha_i \phi_i(||\mathbf{p} - \mathbf{p}_i||_2, c) \tag{A.11}$$

with basis functions $\phi_1, \ldots, \phi_N : \mathbb{R}_0^+ \to \mathbb{R}$ and coefficients $\alpha_1, \ldots, \alpha_N \in \mathbb{R}$ [163, 35]. A radial basis function $\phi_i(||\mathbf{p} - \mathbf{p}_i||_2, c)$ is a function which is radially symmetric around a point $\mathbf{p}_i$ defined in terms of the Euclidean distance. It can depend on the shape parameter c which influences the smoothness of the function. Some common radial basis functions are shown in Table A.1.

Table A.1: Some common radial basis functions.

Name of RBF	Definition
Linear	$\phi_i(\lVert\mathbf{p} - \mathbf{p}_i\rVert_2) = \lVert\mathbf{p} - \mathbf{p}_i\rVert_2$
Thin plate spline	$\phi_i(\lVert\mathbf{p} - \mathbf{p}_i\rVert_2) = \lVert\mathbf{p} - \mathbf{p}_i\rVert_2^2 \ln(\lVert\mathbf{p} - \mathbf{p}_i\rVert_2)$
Cubic	$\phi_i(\lVert\mathbf{p} - \mathbf{p}_i\rVert_2) = \lVert\mathbf{p} - \mathbf{p}_i\rVert_2^3$
Gaussian	$\phi_i(\lVert\mathbf{p} - \mathbf{p}_i\rVert_2, c) = \mathrm{e}^{-(c\lVert\mathbf{p} - \mathbf{p}_i\rVert_2)^2}$, $c > 0$
Multiquadric	$\phi_i(\lVert\mathbf{p} - \mathbf{p}_i\rVert_2, c) = \sqrt{\lVert\mathbf{p} - \mathbf{p}_i\rVert_2^2 + c^2}$, $c > 0$
Inverse multiquadric	$\phi_i(\lVert\mathbf{p} - \mathbf{p}_i\rVert_2, c) = \frac{1}{\sqrt{\lVert\mathbf{p} - \mathbf{p}_i\rVert_2^2 + c^2}}$, $c > 0$

Once the basis functions are chosen, the coefficients $\alpha_1, \dots, \alpha_N$ are computed by solving a system of linear equations which arises when the data is inserted into the interpolation condition (A.1):

$$\underbrace{\begin{bmatrix} \phi_1(||\mathbf{p}_1 - \mathbf{p}_1||_2, c) & \phi_2(||\mathbf{p}_1 - \mathbf{p}_2||_2, c) & \dots & \phi_N(||\mathbf{p}_1 - \mathbf{p}_N||_2, c) \\ \phi_1(||\mathbf{p}_2 - \mathbf{p}_1||_2, c) & \phi_2(||\mathbf{p}_2 - \mathbf{p}_2||_2, c) & \dots & \phi_N(||\mathbf{p}_2 - \mathbf{p}_N||_2, c) \\ \vdots & \vdots & & \vdots \\ \phi_1(||\mathbf{p}_N - \mathbf{p}_1||_2, c) & \phi_2(||\mathbf{p}_N - \mathbf{p}_2||_2, c) & \dots & \phi_N(||\mathbf{p}_N - \mathbf{p}_N||_2, c) \end{bmatrix}}_{\boldsymbol{\phi}} \underbrace{\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix}}_{\boldsymbol{\alpha}} = \underbrace{\begin{bmatrix} f(\mathbf{p}_1) \\ f(\mathbf{p}_2) \\ \vdots \\ f(\mathbf{p}_N) \end{bmatrix}}_{\mathbf{f}}. \tag{A.12}$$

The $N \times N$ interpolation matrix $\boldsymbol{\phi}$ is symmetric because of $||\mathbf{p}_i - \mathbf{p}_j||_2 = ||\mathbf{p}_j - \mathbf{p}_i||_2$ for $i, j \in \{1, \dots, N\}$. A unique solution exists for the system of equations (A.12) if and only if $\boldsymbol{\phi}$ is nonsingular. This is guaranteed for the linear, Gaussian, multiquadric and inverse multiquadric basis functions from Table A.1 if $N \geq 2$ and if the data points are all different [117]. The thin plate spline and the cubic basis function may or may not lead to nonsingular $\boldsymbol{\phi}$. For such cases it is useful to add low order polynomials $p_1(\mathbf{p}), \dots, p_M(\mathbf{p})$ and the corresponding coefficients $\gamma_1, \dots, \gamma_M$ to the interpolant (A.11)

$$\hat{f}(\mathbf{p}) = \sum_{i=1}^{N} \alpha_i \phi_i(||\mathbf{p} - \mathbf{p}_i||_2, c) + \sum_{j=1}^{M} \gamma_j p_j(\mathbf{p}). \tag{A.13}$$

As now the system of equations (A.13) is underdetermined, we use the side conditions

$$\sum_{i=1}^{N} \alpha_i p_j(\mathbf{p}_i) = 0 \qquad \forall \;\; j \in \{1, \dots, M\} \tag{A.14}$$

and solve the linear system of $N + M$ equations

$$\begin{bmatrix} \boldsymbol{\phi} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\gamma} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} \tag{A.15}$$

with $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_M]^T$ and

$$\mathbf{P} = \begin{bmatrix} p_1(\mathbf{p}_1) & \dots & p_M(\mathbf{p}_1) \\ \vdots & & \vdots \\ p_1(\mathbf{p}_N) & \dots & p_M(\mathbf{p}_N) \end{bmatrix}. \tag{A.16}$$

A.3.2 Kriging

Consider again a d-dimensional function $f : \mathcal{D} \to \mathbb{R}$ that we want to interpolate using the scattered data $\{(\mathbf{p}_1, y_1), \dots, (\mathbf{p}_N, y_N)\}$. Kriging or Gaussian process regression models this function by a Gaussian process $F(\mathbf{p}) \sim \mathcal{N}\Big(\hat{f}(\mathbf{p}), \hat{\sigma}(\mathbf{p})^2\Big)$ with mean $\hat{f}(\mathbf{p})$ and variance $\hat{\sigma}(\mathbf{p})^2$ [106, 46, 140]. The Gaussian correlation function is

$$\phi_i(||\mathbf{p} - \mathbf{p}_i||_2) = \mathrm{e}^{-\theta||\mathbf{p}-\mathbf{p}_i||_2^2} \tag{A.17}$$

with correlation length θ. The vector of correlations is

$$\mathbf{r}(\mathbf{p}) = [\phi_1(||\mathbf{p} - \mathbf{p}_1||_2), \phi_2(||\mathbf{p} - \mathbf{p}_2||_2), \dots, \phi_N(||\mathbf{p} - \mathbf{p}_N||_2)]^T \tag{A.18}$$

and the correlation matrix is

$$\mathbf{R} = [\mathbf{r}(\mathbf{p}_1), \mathbf{r}(\mathbf{p}_2), \dots, \mathbf{r}(\mathbf{p}_N)]. \tag{A.19}$$

This delivers the Kriging predictor for the mean

$$\hat{f}(\mathbf{p}) = \mathbf{r}(\mathbf{p})^T \mathbf{R}^{-1}(\bar{\mathbf{f}} - \bar{\boldsymbol{\mu}}) + \mu(\mathbf{p}) \tag{A.20}$$

with a low-order polynomial obtained from regression $\mu(\mathbf{p})$ and

$$\bar{\mathbf{f}} = [f(\mathbf{p}_1), \dots, f(\mathbf{p}_N)]^T, \tag{A.21}$$

$$\bar{\boldsymbol{\mu}} = [\mu(\mathbf{p}_1), \dots, \mu(\mathbf{p}_N)]^T. \tag{A.22}$$

The Kriging predictor also gives a variance estimate $\hat{\sigma}(\mathbf{p})^2$ as a measure of precision using an estimate of the unadjusted variance σ^2. The hyperparameters θ and σ^2 can be determined using maximum likelihood estimation. Comparing the RBF interpolant (A.13) and the Kriging predictor (A.20), it can be observed that RBF interpolation and Kriging can be written in the same form. However, the first one is constructed using various basis functions whereas the latter employs a correlation matrix and vector. In addition, Kriging provides a variance estimate [148].

A.4 Sparse-Grid-Based Interpolation

Consider again the interpolation problem defined in Section A.1 where function $f(\mathbf{p})$ is approximated by $\hat{f}(\mathbf{p}) = \sum_{i=1}^{N} \alpha_i \phi_i(\mathbf{p})$ with coefficients $\alpha_1, \ldots, \alpha_N$ and basis functions which we denote with $\phi_1(\mathbf{p}), \ldots, \phi_N(\mathbf{p})$ in the context of sparse grids. They span the approximation space $V = \operatorname{span}\{\phi_1(\mathbf{p}), \ldots, \phi_N(\mathbf{p})\}$. The sparse-grid-based interpolation method, see e.g. [37], firstly specifies what basis functions $\phi_1(\mathbf{p}), \ldots, \phi_N(\mathbf{p})$ to use, secondly, on which points $\mathbf{p}_1, \ldots, \mathbf{p}_N$ to sample the function $f(\mathbf{p})$, and, thirdly, how to construct the coefficients $\alpha_1, \ldots, \alpha_N$. Furthermore, this is done optimally with respect to the interpolation error in the $\mathcal{L}_2$-norm if the function $f(\mathbf{p})$ has bounded mixed derivatives up to order two.

Firstly, take the standard hat function $\phi : [-1, 1] \to \mathbb{R}$ with $\phi(p) = \max(1 - |p|, 0)$. Then, the one-dimensional hierarchical basis function $\phi_{l,i} : [0, 1] \to \mathbb{R}$ centered at the sparse grid point $p_{l,i} = i \cdot 2^{-l}$ is defined as

$$\phi_{l,i}(p) = \phi(2^l \cdot p - i)\,. \tag{A.23}$$

The $l \in \mathbb{N}$ is called the level and the $i \in \mathbb{N}$ the index of the basis function $\phi_{l,i}$ and grid point $p_{l,i}$, respectively. In Figure A.5(left), the hierarchical basis is plotted up to level three. We extend (A.23) to the d-dimensional case with

$$\phi_{\boldsymbol{l},\boldsymbol{i}}(\mathbf{p}) = \prod_{j=1}^{d} \phi_{l_j,i_j}(p_j), \tag{A.24}$$

where level $\boldsymbol{l} = (l_1, \ldots, l_d)$ and index $\boldsymbol{i} = (i_1, \ldots, i_d)$ are vectors in $\mathbb{N}^d$ and $\mathbf{p} = (p_1, \ldots, p_d)$ is a vector in $[0, 1]^d$. We associate these hierarchical basis functions in so-called hierarchical increments: Let $\Phi_{\boldsymbol{l}}$ be a set that contains all basis functions (A.24) with level $\boldsymbol{l}$ and index in $\{\boldsymbol{i} \in \mathbb{N}^d \,|\, 1 \leq i_j < 2^{l_j},\, i_j \text{ odd}, 1 \leq j \leq d\}$. Such a set $\Phi_{\boldsymbol{l}}$ spans the hierarchical space $W_{\boldsymbol{l}} = \operatorname{span}(\Phi_{\boldsymbol{l}})$. These hierarchical increments give us a versatile way to define approximation spaces. For example,

$$V_\ell^{\mathrm{FG}} = \bigoplus_{||\boldsymbol{l}||_\infty \leq \ell} W_{\boldsymbol{l}} \tag{A.25}$$

is the space of piecewise d-linear functions of level $\ell \in \mathbb{N}$. This space corresponds to a full grid shown in Figure A.5(a) with mesh width $h = 2^{-\ell}$. The level ℓ controls how many grid points we want to have. Figure A.5(left) shows the grid points and the basis functions spanning V_3^{FG}. Of course, this full grid approach becomes computationally

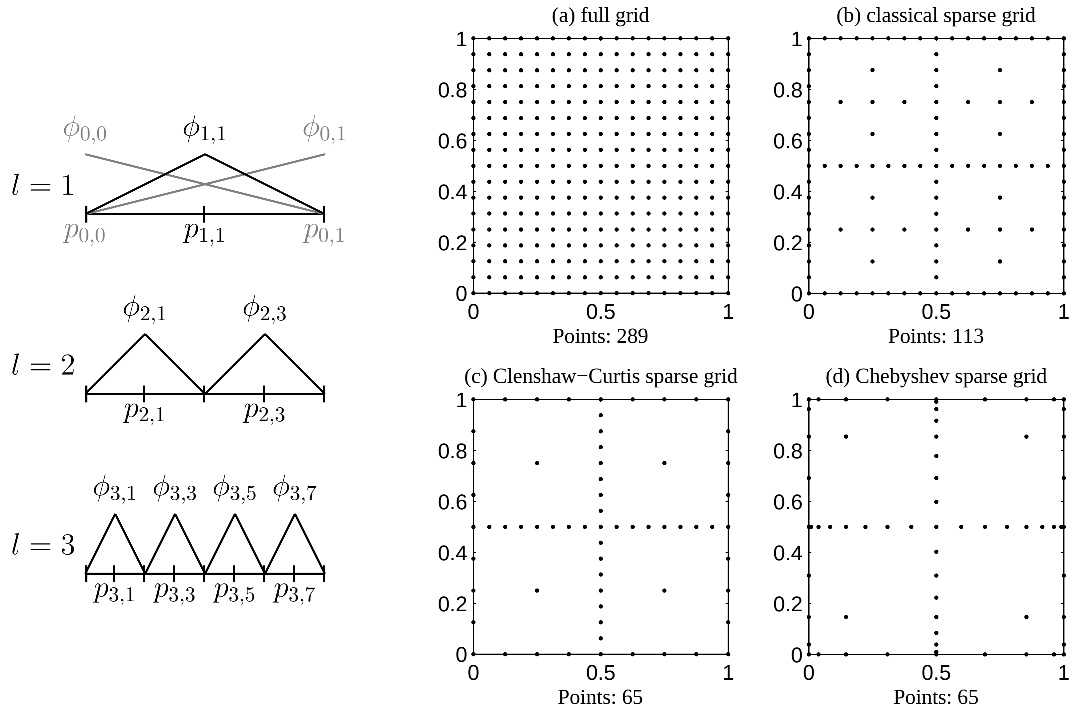


Figure A.5: On the left, hierarchical basis functions from level 0 (boundary points) to level 3. On the right, a comparison of full and sparse grids. In (a) a full grid with mesh width $h = 2^{-4}$ and in (b) the classical sparse grid of level 4 corresponding to (A.26). Two variants, the Clenshaw-Curtis and the Chebyshev sparse grid, in (c) and (d).

infeasible for large dimensions d due to the curse of dimensionality. The number of grid points and basis functions is in $\mathcal{O}(2^{\ell d})$ and thus grows exponentially with the dimension d. It is well known that the interpolation accuracy is in $\mathcal{O}(2^{-2\ell})$ for this type of grids and spaces. The sparse grid space is then defined as

$$V_\ell^{\mathrm{SG}} = \bigoplus_{||\boldsymbol{l}||_1 \leq \ell+d-1} W_{\boldsymbol{l}} \tag{A.26}$$

which determines the basis functions of a sparse grid space and thus the grid points of a sparse grid. The sparse grid space V_ℓ^{SG} is spanned only by $\mathcal{O}(2^\ell \ell^{d-1})$ hierarchical basis functions, and the accuracy of the sparse grid interpolant $\hat{f}^{\mathrm{SG}} \in V_\ell^{\mathrm{SG}}$ is in $\mathcal{O}(2^{-2\ell}\ell^{d-1})$ in $\mathcal{L}_2$-norm if the function $f(\mathbf{p})$ has bounded mixed derivatives up to order two. Thus, we save a lot of grid points while the accuracy of the interpolant deteriorates only slightly compared to the interpolation on the full grid.

The discussion above with definition (A.26) and basis functions (A.24) leads to the classical sparse grid structure shown in Figure A.5(b). There exist other sparse grid structures—see Figure A.5(c) and (d)—which all have similar asymptotic properties but which might lead to different results for some applications, see [103, 102] and the references therein.

So far, we have introduced the basis functions (A.24) as well as the grid points of a sparse grid space (A.26). Finally, we have to discuss how to compute the sparse grid interpolant $\hat{f}^{\mathrm{SG}}(\mathbf{p}) = \sum_{l=1}^{\ell} \sum_{i=1}^{N_l} \alpha_{l,i}\phi_{l,i}(\mathbf{p})$ with $\sum_{i=1}^{\ell} N_l = N$. We denote in the remainder of this thesis for simplicity's sake with $\phi_{l,1}(\mathbf{p}), \dots, \phi_{l,N_l}(\mathbf{p})$ all basis functions of current level l with number of points N_l. For this, we need to determine the coefficients $\alpha_{1,1}, \dots, \alpha_{\ell,N_\ell} \in \mathbb{R}$ which are also called *hierarchical surpluses*. This is done by the so-called hierarchisation procedure. This is a highly recursive algorithm that allows us to compute these coefficients with linear complexity in the number of grid points. Thus, this is cheaper than to solve a system of linear equations as required in, e.g., Kriging or RBF approaches. We do not go into details of the hierarchisation procedure here but refer to [37] for details.

Overall, the three introduced building blocks—hierarchical basis, sparse grid structure, hierarchisation procedure—form an efficient interpolation method that is well-suited for (moderately) high-dimensional problems.

Appendix B

Proof of Proposition 3.1

Proof. Recall the two conditions:

- C1: Orthonormal bases $\mathbf{V}_i, \mathbf{V}_0$ with $\mathbf{V}_i^T\mathbf{V}_i = \mathbf{I}_q$ and $\mathbf{V}_0^T\mathbf{V}_0 = \mathbf{I}_q$.
- C2: The orthonormal bases span the same subspace with $\mathrm{span}(\mathbf{V}_i) = \mathrm{span}(\mathbf{V}_0)$ and $\mathrm{rank}([\mathbf{V}_i, \mathbf{V}_0]) = \mathrm{rank}(\mathbf{V}_i) = \mathrm{rank}(\mathbf{V}_0) = q$.

Firstly, we proof that $\mathbf{V}_i^T\mathbf{V}_0 \in \mathbb{O}_q$ is orthogonal. For this, we start with the overdetermined system of linear equations

$$\mathbf{V}_i\mathbf{T}_i = \mathbf{V}_0 \tag{B.1}$$

which has a single unique solution $\mathbf{T}_i \in \mathbb{R}^{q\times q}$ because of condition C2 [68]. Hence, the Penrose condition (D.5) is fulfilled:

$$\mathbf{V}_i(\mathbf{V}_i^T\mathbf{V}_i)^{-1}\mathbf{V}_i^T\mathbf{V}_0 = \mathbf{V}_0 \tag{B.2}$$

or with condition C1

$$\mathbf{V}_0^T\mathbf{V}_i\mathbf{V}_i^T\mathbf{V}_0 = \mathbf{I}_q \tag{B.3}$$

and finally

$$(\mathbf{V}_i^T\mathbf{V}_0)^T(\mathbf{V}_i^T\mathbf{V}_0) = \mathbf{I}_q. \tag{B.4}$$

Hence, we know that $\mathbf{V}_i^T\mathbf{V}_0 \in \mathbb{O}_q$ is orthogonal. Secondly, we compare the three different approaches:

- For the MAC approach (3.16) we obtain

$$\mathbf{T}_i = (\mathbf{V}_0^T\mathbf{V}_i)^{-1} = (\mathbf{V}_i^T\mathbf{V}_0)^{-T} = \mathbf{V}_i^T\mathbf{V}_0. \tag{B.5}$$

- For the DS approach (3.8) we obtain using condition C1

$$\mathbf{T}_i = (\mathbf{V}_i^T\mathbf{V}_i)^{-1}\mathbf{V}_i^T\mathbf{V}_0 = \mathbf{I}_q\mathbf{V}_i^T\mathbf{V}_0 = \mathbf{V}_i^T\mathbf{V}_0. \tag{B.6}$$

- For the PS approach (3.13) recall the optimization problem

$$\mathbf{T}_i = \arg\max_{\mathbf{T}_i \in \mathbb{O}(q)} \operatorname{tr}\left(\mathbf{T}_i^T \mathbf{V}_i^T \mathbf{V}_0\right). \tag{B.7}$$

  It reaches its maximum value $\operatorname{tr}(\mathbf{I}_q)$ for $\mathbf{T}_i = \mathbf{V}_i^T \mathbf{V}_0 \in \mathbb{O}_q$.

Hence, all three approaches lead to the solution $\mathbf{T}_i = \mathbf{V}_i^T \mathbf{V}_0 \in \mathbb{O}_q$. ■

Appendix C

Convex Cones

Consider the following definitions:

Definition C.1 ([82])**.** A set $\mathcal{C}$ is called a *cone* if for every $x \in \mathcal{C}$ and $\theta \geq 0$ we have $\theta x \in \mathcal{C}$.

Definition C.2 ([82])**.** A set $\mathcal{C}$ is called a *convex cone* if for every $x_1, x_2 \in \mathcal{C}$ and $\theta_1, \theta_2 \geq 0$ we have $\theta_1 x_1 + \theta_2 x_2 \in \mathcal{C}$.

Appendix D

Pseudoinverse

As we will employ the pseudoinverse as introduced by [120, 26, 132] in this thesis, the nomenclature is introduced in this appendix. Consider a matrix $\mathbf{Q} \in \mathbb{R}^{a \times b}$. If its columns are linearly independent for $a \geq b$, the left inverse $\mathbf{Q}^{(l)+} \in \mathbb{R}^{b \times a}$ with $\mathbf{Q}^{(l)+}\mathbf{Q} = \mathbf{I}_b$ is denoted with

$$\mathbf{Q}^{(l)+} = (\mathbf{Q}^T\mathbf{Q})^{-1}\mathbf{Q}^T. \tag{D.1}$$

If matrix $\mathbf{Q}$ has linearly independent rows for $a \leq b$, the right inverse $\mathbf{Q}^{(r)+} \in \mathbb{R}^{b \times a}$ with $\mathbf{Q}\mathbf{Q}^{(r)+} = \mathbf{I}_a$ is

$$\mathbf{Q}^{(r)+} = \mathbf{Q}^T(\mathbf{Q}\mathbf{Q}^T)^{-1}. \tag{D.2}$$

For the calculation of the pseudoinverse there exists a numerically more efficient approach using Singular Value Decomposition (SVD), see [147]. It calculates the SVD of $\mathbf{Q} = \mathbf{U}\mathbf{\Sigma}\mathbf{Z}^T$, where $\mathbf{\Sigma}$ is a diagonal matrix containing the singular values. Then, $\mathbf{\Sigma}^{(r/l)+}$ is a diagonal matrix containing the reciprocal singular values of $\mathbf{\Sigma}$. This leads to the (right/left) pseudoinverse $\mathbf{Q}^{(r/l)+}$:

$$\mathbf{Q}^{(r/l)+} = \mathbf{Z}\mathbf{\Sigma}^{(r/l)+}\mathbf{U}^T. \tag{D.3}$$

We use formula (D.3) for the calculation of the pseudoinverse. One application of the pseudoinverse is the solution of the matrix equality

$$\mathbf{Q}\mathbf{X} = \mathbf{Y} \tag{D.4}$$

with $\mathbf{X} \in \mathbb{R}^{b \times o}, \mathbf{Y} \in \mathbb{R}^{a \times o}$. A solution exists for matrix equation (D.4) if and only if the Penrose condition

$$\mathbf{Q}\mathbf{Q}^{(r/l)+}\mathbf{Y} = \mathbf{Y} \tag{D.5}$$

is fulfilled [132]. Then, the solution with minimum $\|\mathbf{X}\|_F$ is $\mathbf{X} = \mathbf{Q}^{(r/l)+}\mathbf{Y}$, where $\|\cdot\|_F$ denotes the Frobenius norm. If (D.5) holds, then the solution $\mathbf{X}$ is unique if and only if $\mathbf{Q}$ has full column rank. If the Penrose condition is not fulfilled, $\mathbf{X} = \mathbf{Q}^{(r/l)+}\mathbf{Y}$ is the unique best approximate solution of equation (D.4) in the sense of the method of least squares minimizing the optimization problem [133]

$$\mathbf{X} = \arg\min_{\mathbf{X}\in\mathbb{R}^{b\times o}} \|\mathbf{QX} - \mathbf{Y}\|_F. \tag{D.6}$$

Bibliography

[1] *MOR Wiki*. Available from http://www.modelreduction.org.

[2] *Oberwolfach Model Reduction Benchmark Collection*. Available from http://www.simulation.uni-freiburg.de/downloads/benchmark/.

[3] A. S. Ackleh, E. J. Allen, R. B. Kearfott, and P. Seshaiyer. *Classical and Modern Numerical Analysis*. Chapman & Hall/CRC, Boca Raton, Florida, 2009.

[4] D. M. Allen. The Relationship between Variable Selection and Data Augmentation and a Method for Prediction. *Technometrics*, 16(1):125–127, 1974.

[5] D. Amsallem. *Interpolation on Manifolds of CFD-Based Fluid and Finite Element-Based Structural Reduced-Order Models for Online Aeroelastic Predictions*. PhD thesis, Stanford University, 2010.

[6] D. Amsallem, J. Cortial, K. Carlberg, and C. Farhat. A Method for Interpolating on Manifolds Structural Dynamics Reduced-Order Models. *Int. J. Numer. Methods Engrg.*, 80(9):1241–1258, 2009.

[7] D. Amsallem, J. Cortial, and C. Farhat. Towards Real-Time CFD-Based Aeroelastic Computations Using a Database of Reduced-Order Information. *AIAA Journal*, 48(9):2029–2037, 2010.

[8] D. Amsallem, J. Cortial, and C. Farhat. On-Demand CFD-Based Aeroelastic Predictions Using a Database of Reduced-Order Bases and Models. *Proc. of the 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, Orlando, Florida, 2009.

[9] D. Amsallem, S. Deolalikar, F. Gurrola, and C. Farhat. Model Predictive Control under Coupled Fluid-Structure Constraints Using a Database of Reduced-Order Models on a Tablet. *Proc. of the 21st AIAA Computational Fluid Dynamics Conference*, San Diego, CA, 2013.

[10] D. Amsallem and C. Farhat. Interpolation Method for the Adaptation of Reduced-Order Models to Parameter Changes and Its Application to Aeroelasticity. *AIAA Journal*, 46(7):1803–1813, 2008.

[11] D. Amsallem and C. Farhat. An Online Method for Interpolating Linear Parametric Reduced-Order Models. *SIAM J. Sci. Comput.*, 33(5):2169–2198, 2011.

[12] D. Amsallem and C. Farhat. Stabilization of Projection-Based Reduced-Order Models. *Int. J. Numer. Methods Engrg.*, 91(4):358–377, 2012.

[13] D. Amsallem and C. Farhat. Aeroelastic Analysis of F-16 and F-18/A Configurations Using Adapted CFD-Based Reduced-Order Models. *Proc. of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Honolulu, Hawaii, 2007.

[14] B. D. O. Anderson and Y. Liu. Controller Reduction: Concept and Approaches. *IEEE Trans. Autom. Control*, 34(8):802–812, 1989.

[15] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems.* SIAM, Philadelphia, PA, 2005.

[16] E. Arian, M. Fahl, and E. W. Sachs. Trust-Region Proper Orthogonal Decomposition for Flow Control. Technical report, ICASE, NASA Langley Research Center, 2000.

[17] S. Arlot and A. Celisse. A Survey of Cross-Validation Procedures for Model Selection. *Statistics Surveys*, 4:40–79, 2010.

[18] V. Aute, K. Saleh, O. Abdelaziz, S. Azarm, and R. Radermacher. Cross-Validation Based Single Response Adaptive Design of Experiments for Kriging Metamodeling of Deterministic Computer Simulations. *Struct. Multidisc. Optim.*, 48(3):581–605, 2013.

[19] U. Baur, C. A. Beattie, P. Benner, and S. Gugercin. Interpolatory Projection Methods for Parameterized Model Reduction. *SIAM J. Sci. Comput.*, 33(5):2489–2518, 2011.

[20] U. Baur and P. Benner. Model Reduction for Parametric Systems Using Balanced Truncation and Interpolation. *at-Automatisierungstechnik*, 57(8):411–420, 2009.

[21] U. Baur, P. Benner, A. Greiner, J. G. Korvink, J. Lienemann, and C. Moosmann. Parameter Preserving Model Reduction for MEMS Applications. *Math. Comp. Model. Dyn. Sys.*, 17(4):297–317, 2011.

[22] T. Bechtold, D. Hohfeld, E. B. Rudnyi, and M. Guenther. Efficient Extraction of Thin-Film Thermal Parameters from Numerical Models via Parametric Model Order Reduction. *J. Micromech. Microeng.*, 20(4):045030, 2010.

[23] P. Benner, S. Gugercin, and K. Willcox. A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Review*, 57(4):483–531, 2015.

[24] P. Benner, M. Köhler, and J. Saak. Sparse-Dense Sylvester Equations in $\mathcal{H}_2$-Model Order Reduction. Preprint MPIMD/11-11, Max Planck Institute Magdeburg, Dec. 2011. Available from `http://www.mpi-magdeburg.mpg.de/preprints/`.

[25] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, Berlin, Germany, 2006.

[26] A. Bjerhammar. Application of Calculus of Matrices to the Method of Least Squares with Special Reference to Geodetic Calculations. *Trans. of the Royal Inst. of Technology*, 49, 1951.

[27] B. N. Bond and L. Daniel. Stable Reduced Models for Nonlinear Descriptor Systems Through Piecewise-Linear Approximation and Projection. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 28(10):1467–1480, 2009.

[28] B. N. Bond and L. Daniel. Stabilizing Schemes for Piecewise-Linear Reduced Order Models via Projection and Weighting Functions. *Proc. of the IEEE/ACM Int. Conf. Comput.-Aided Des.*, pages 860–867, San Jose, CA, 2007.

[29] B. N. Bond and L. Daniel. Guaranteed Stable Projection-Based Model Reduction for Indefinite and Unstable Linear Systems. *Proc. of the IEEE/ACM Int. Conf. Comput.-Aided Des.*, pages 728–735, San Jose, CA, 2008.

[30] J. Borggaard, K. R. Pond, and L. Zietsman. Parametric Reduced Order Models Using Adaptive Sampling and Interpolation. *Proc. of the 19th IFAC World Congress*, pages 7773–7778, Cape Town, South Africa, 2014.

[31] S. Boyd and V. Balakrishnan. A Regularity Result for the Singular Values of a Transfer Matrix and a Quadratically Convergent Algorithm for Computing its $\mathcal{L}_\infty$-norm. *Systems and Control Letters*, 15(1):1–7, 1990.

[32] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.

[33] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[34] N. A. Bruinsma and M. Steinbuch. A Fast Algorithm to Compute the $\mathcal{H}_\infty$-Norm of a Transfer Function Matrix. *Systems and Control Letters*, 14(4):287–293, 1990.

[35] M. D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, Cambridge, UK, 2008.

[36] T. Bui-Thanh, K. Willcox, and O. Ghattas. Model Reduction for Large-Scale Systems with High-Dimensional Parametric Input Space. *SIAM J. Sci. Comput.*, 30(6):3270–3288, 2008.

[37] H.-J. Bungartz and M. Griebel. Sparse Grids. *Acta Numerica*, 13:147–269, 2004.

[38] S. Burgard, O. Farle, and R. Dyczij-Edlinger. An h Adaptive Sub-Domain Framework for Parametric Order Reduction. *IEEE Trans. Magn.*, 51(3):7208504, 2015.

[39] D. Butnaru, B. Peherstorfer, D. Pflüger, and H.-J. Bungartz. Fast Insight into High-Dimensional Parametrized Simulation Data. *Proc. of the 11th International Conference on Machine Learning and Applications*, pages 265–270, Boca Raton, Florida, 2012.

[40] E. F. Camacho and C. Bordons. *Model Predictive Control*. Springer, London, UK, 2nd edition, 2007.

[41] R. Castañé-Selga, R. Eid, and B. Lohmann. Stability Preservation in Projection-Based Model Order Reduction of Large Scale Systems. *Eur. J. Control*, 18(2):122–132, 2012.

[42] S. Chaturantabut and D. Sorensen. Nonlinear Model Reduction via Discrete Empirical Interpolation. *SIAM J. Sci. Comput.*, 32(5):2737–2764, 2010.

[43] E. W. Cheney. *Introduction to Approximation Theory.* AMS Chelsea Publishing, New York, NY, 1982.

[44] K.-W. E. Chu. The Solution of the Matrix Equation $AXB - CXD = Y$ and $(YA - DZ, YC - BZ) = (E, F)$. *Linear Algebra Appl.*, 93:93–105, 1987.

[45] B. Codrons. *Process Modelling for Control.* Springer, London, UK, 2005.

[46] N. Cressie. *Statistics for Spatial Data.* Wiley, New York, NY, 1993.

[47] M. Cruz Varona, M. Geuss, and B. Lohmann. Zeitvariante parametrische Modellordnungsreduktion am Beispiel von Systemen mit wandernder Last. In B. Lohmann and G. Roppenecker, editors, *Methoden und Anwendungen der Regelungstechnik. Erlangen-Münchener Workshops 2013 und 2014*, pages 57–70. Shaker-Verlag, Aachen, Germany, 2015.

[48] W. J. Culver. On the Existence and Uniqueness of the Real Logarithm of a Matrix. *Proc. of the American Mathematical Society*, 17(5):1146–1151, 1966.

[49] L. Daniel, O. Siong, K. Lee, and J. White. A Multiparameter Moment-Matching Model-Reduction Approach for Generating Geometrically Parameterized Interconnect Performance Models. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 23(5):678–693, 2004.

[50] P. J. Davis. *Interpolation and Approximation.* Dover Publications, New York, NY, 1975.

[51] J. Degroote, J. Vierendeels, and K. Willcox. Interpolation Among Reduced-Order Matrices to Obtain Parameterized Models for Design, Optimization and Probabilistic Analysis. *Int. J. Numer. Methods Fluids*, 63(2):207–230, 2010.

[52] L. C. W. Dixon and G. P. Szegö. The Global Optimization: An Introduction. In L. C. W. Dixon and G. P. Szegö, editors, *Towards Global Optimisation 2*, pages 1–15. North-Holland Publishing Company, Amsterdam, North-Holland, 1978.

[53] M. Drohmann and K. Carlberg. The ROMES Method for Statistical Modeling of Reduced-Order-Model Error. *SIAM/ASA J. Uncertainty Quantification*, 3(1):116–145, 2015.

[54] V. Druskin and V. Simoncini. Adaptive Rational Krylov Subspaces for Large-Scale Dynamical Systems. *Systems and Control Letters*, 60(8):546–560, 2011.

[55] R. Eid, R. Castañé-Selga, H. Panzer, T. Wolf, and B. Lohmann. Stability-Preserving Parametric Model Reduction by Matrix Interpolation. *Math. Comp. Model. Dyn. Sys.*, 17(4):319–335, 2011.

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

[57] D. J. Ewins. *Modal Testing, Theory, Practice and Applications.* Wiley, Taunton, UK, 2nd edition, 2000.

[58] O. Farle, S. Burgard, and R. Dyczij-Edlinger. Passivity Preserving Parametric Model-Order Reduction for Non-Affine Parameters. *Math. Comp. Model. Dyn. Sys.*, 17(3):279–294, 2011.

[59] G. E. Fasshauer. *Meshfree Approximation Methods with MATLAB.* Interdisciplinary Mathematical Sciences 6. World Scientific Publishing, Singapore, 2007.

[60] L. H. Feng, E. B. Rudnyi, and J. G. Korvink. Preserving the Film Coefficient as a Parameter in the Compact Thermal Model for Fast Electro-Thermal Simulation. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 24(12):1838–1847, 2005.

[61] F. Ferranti, G. Antonini, T. Dhaene, and L. Knockaert. Guaranteed Passive Parameterized Model Order Reduction of the Partial Element Equivalent Circuit (PEEC) Method. *IEEE Trans. Electromagn. Compat.*, 52(4):974–984, 2010.

[62] F. Ferranti, G. Antonini, T. Dhaene, L. Knockaert, and A. E. Ruehli. Physics-Based Passivity-Preserving Parameterized Model Order Reduction for PEEC Circuit Analysis. *IEEE Trans. Compon. Packag. Manuf. Technol.*, 1(3):399–409, 2011.

[63] J. Fiszer, T. Tamarozzi, B. Blockmans, and W. Desmet. A Time-Dependent Parametric Model Order Reduction Technique for Modelling Indirect Bearing Force Measurements. *Mechanism and Machine Theory*, 83:152–174, 2015.

[64] A. I. J. Forrester and A. J. Keane. Recent Advances in Surrogate-Based Optimization. *Progress in Aerospace Sciences*, 45(1-3):50–79, 2009.

[65] A. I. J. Forrester, A. Sobester, and A. J. Keane. Multi-fidelity Optimization via Surrogate Modelling. *Proc. of the Royal Society A*, 463(2088):3251–3269, 2007.

[66] A. I. J. Forrester, A. Sobester, and A. J. Keane. *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley, Chichester, UK, 2008.

[67] R. W. Freund. Model Reduction Methods Based on Krylov Subspaces. *Acta Numerica*, 12:267–319, 2003.

[68] G. Frobenius. Zur Theorie der linearen Gleichungen. *Journal für die reine und angewandte Mathematik*, 129:175–180, 1905.

[69] J. D. Gardiner, A. J. Laub, J. J. Amato, and C. B. Moler. Solution of the Sylvester Matrix Equation $AXB^T + CXD^T = E$. *ACM Transactions on Mathematical Software*, 18(2):223–231, 1992.

[70] W. K. Gawronski. *Dynamics and Control of Structures*. Springer, New York, NY, 1998.

[71] S. Geisser. The Predictive Sample Reuse Method with Applications. *Journal of the American Statistical Association*, 70(350):320–328, 1975.

[72] M. Geuss, D. Butnaru, B. Peherstorfer, H.-J. Bungartz, and B. Lohmann. Parametric Model Order Reduction by Sparse-Grid-Based Interpolation on Matrix Manifolds for Multidimensional Parameter Spaces. *Proc. of the European Control Conference*, pages 2727–2732, Strasbourg, France, 2014.

[73] M. Geuss, C. Choi, and B. Lohmann. Control of Parameter-Dependent High-Order Systems Using Parametric Model Reduction. *at-Automatisierungstechnik*, 62(7):523–533, 2014.

[74] M. Geuss and K. J. Diepold. An Approach for Stability-Preserving Model Order Reduction for Switched Linear Systems Based on Individual Subspaces. In B. Lohmann and G. Roppenecker, editors, *Methoden und Anwendungen der Regelungstechnik. Erlangen-Münchener Workshops 2011 und 2012*, pages 1–16. Shaker-Verlag, Aachen, Germany, 2013.

[75] M. Geuss, H. Panzer, I. Clifford, and B. Lohmann. Parametric Model Order Reduction Using Pseudoinverses for the Matrix Interpolation of Differently Sized Reduced Models. *Proc. of the 19th IFAC World Congress*, pages 9468–9473, Cape Town, South Africa, 2014.

[76] M. Geuss, H. Panzer, and B. Lohmann. On Parametric Model Order Reduction by Matrix Interpolation. *Proc. of the European Control Conference*, pages 3433–3438, Zurich, Switzerland, 2013.

[77] M. Geuss, H. Panzer, T. Wolf, and B. Lohmann. Stability Preservation for Parametric Model Order Reduction by Matrix Interpolation. *Proc. of the European Control Conference*, pages 1098–1103, Strasbourg, France, 2014.

[78] M. Giftthaler, T. Wolf, H. Panzer, and B. Lohmann. Parametric Model Order Reduction of Port-Hamiltonian Systems by Matrix Interpolation. *at-Automatisierungstechnik*, 62(9):619–628, 2014.

[79] K. Glover and D. J. N. Limebeer. Robust Multivariable Control System Design Using Optimal Reduced Order Plant Models. *Proc. of the American Control Conference*, pages 644–649, San Francisco, CA, 1983.

[80] P. J. Goddard and K. Glover. Controller Approximation: Approaches for Preserving $\mathcal{H}_\infty$-Performance. *IEEE Trans. Autom. Control*, 43(7):858–871, 1998.

[81] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, Maryland, 3rd edition, 1996.

[82] M. Grant and S. Boyd. Graph Implementations for Nonsmooth Convex Programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer, London, UK, 2008.

[83] M. Grant and S. Boyd. CVX: Matlab Software for Disciplined Convex Programming, Version 2.0, Aug. 2012. Available from `http://cvxr.com/cvx`.

[84] M. A. Grepl and A. T. Patera. A Posteriori Error Bounds for Reduced-Basis Approximations of Parametrized Parabolic Partial Differential Equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(1):157–181, 2005.

[85] E. J. Grimme. *Krylov Projection Methods for Model Reduction.* PhD thesis, University of Illinois at Urbana-Champaign, 1997.

[86] S. Gugercin, A. C. Antoulas, and C. Beattie. $\mathcal{H}_2$ Model Reduction for Large-Scale Linear Dynamical Systems. *SIAM J. Matrix Anal. and Appl.*, 30(2):609–638, 2008.

[87] S. Gugercin, A. C. Antoulas, C. A. Beattie, and E. Gildin. Krylov-Based Controller Reduction for Large-Scale Systems. *Proc. of the 43rd IEEE Conference on Decision and Control*, 3:3074–3077, Atlantis, Bahamas, 2004.

[88] P. K. Gunupudi, R. Khazaka, M. S. Nakhla, T. Smy, and D. Celo. Passive Parameterized Time-Domain Macromodels for High-Speed Transmission-Line Networks. *IEEE Trans. Microw. Theory Techn.*, 51(12):2347–2354, 2003.

[89] H. Gutmann. A Radial Basis Function Method for Global Optimization. *Journal of Global Optimization*, 19(3):201–227, 2001.

[90] B. Haasdonk and M. Ohlberger. Efficient Reduced Models and a Posteriori Error Estimation for Parametrized Dynamical Systems by Offline/Online Decomposition. *Math. Comp. Model. Dyn. Sys.*, 17(2):145–161, 2001.

[91] B. Haasdonk and M. Ohlberger. Reduced Basis Method for Finite Volume Approximations of Parametrized Linear Evolution Equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(2):277–302, 2008.

[92] J. H. Halton. Algorithm 247: Radical-Inverse Quasi-Random Point Sequence. *Communications of the ACM*, 7(12):701–702, 1964.

[93] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, New York, NY, 2nd edition, 2009.

[94] S. Hovland, J. T. Gravdahl, and K. Willcox. Explicit Model Predictive Control for Large-Scale Systems via Model Reduction. *AIAA Journal for Guidance, Control, and Dynamics*, 31(4):918–926, 2008.

[95] C. H. Jarvis. *Parameter Dependent Model Reduction for Complex Fluid Flows.* PhD thesis, Virginia Polytechnic Institute and State University, 2014.

[96] M. Johnson, L. Moore, and D. Ylvisaker. Minimax and Maximin Distance Designs. *Journal of Statistical Planning and Inference*, 26(2):131–148, 1990.

[97] D. R. Jones. A Taxonomy of Global Optimization Methods Based on Response Surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

[98] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[99] T. Kailath. *Linear Systems.* Prentice Hall, Englewood Cliffs, NJ, 1980.

[100] H. K. Khalil. *Nonlinear Systems.* Prentice Hall, Upper Saddle River, NJ, 3rd edition, 2002.

[101] A. Klimke. *Uncertainty Modeling Using Fuzzy Arithmetic and Sparse Grids.* PhD thesis, University of Stuttgart, 2006.

[102] A. Klimke. Sparse Grid Interpolation Toolbox – User's Guide. Technical Report IANS Report 2007/017, University of Stuttgart, 2007.

[103] A. Klimke and B. Wohlmuth. Algorithm 847: spinterp: Piecewise Multilinear Hierarchical Sparse Grid Interpolation in MATLAB. *ACM Transactions on Mathematical Software*, 31(4):561–579, 2005.

[104] D. Klis, S. Burgard, O. Farle, and R. Dyczij-Edlinger. Model Order Reduction for Nonlinear Eddy Current Problems. *Proc. of the MATHMOD*, pages 740–745, Vienna, Austria, 2012.

[105] R. Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *Proc. of the 14th International Joint Conference on Artificial Intelligence*, 2:1137–1143, 1995.

[106] D. G. Krige. A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand. *J. of the Chem., Metal. and Mining Soc. of South Africa*, 52(6):119–139, 1951.

[107] A. T. Leung and R. Khazaka. Parametric Model Order Reduction Technique for Design Optimization. *Proc. Intl. Symp. Circuits Syst.*, 2:1290–1293, Kobe, Japan, 2005.

[108] J. Löfberg. YALMIP: A Toolbox for Modeling and Optimization in MATLAB. *Proc. of the CACSD Conference*, pages 284–289, Taipei, Taiwan, 2004.

[109] J.-R. Li and J. White. Low Rank Solution of Lyapunov Equations. *SIAM J. Matrix Anal. and Appl.*, 24(1):260–280, 2002.

[110] D. Liberzon. *Switching in Systems and Control.* Birkhäuser, Boston, MA, 2003.

[111] D. Liberzon and A. S. Morse. Basic Problems in Stability and Design of Switched Systems. *IEEE Control Systems Magazine*, 19(5):59–70, 1999.

[112] B. Lohmann and R. Eid. Efficient Order Reduction of Parametric and Nonlinear Models by Superposition of Locally Reduced Models. In B. Lohmann and G. Roppenecker, editors, *Methoden und Anwendungen der Regelungstechnik. Erlangen-Münchener Workshops 2007 und 2008*, pages 1–20. Shaker-Verlag, Aachen, Germany, 2009.

[113] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard. DACE - A MATLAB Kriging Toolbox, Version 2.0, 2002. Available from `http://www2.imm.dtu.dk/projects/dace/`.

[114] J. M. Maciejowski. *Predictive Control with Constraints.* Pearson Education, Harlow, UK, 2002.

[115] O. Mason and R. Shorten. The Geometry of Convex Cones Associated with the Lyapunov Inequality and the Common Lyapunov Function Problem. *Electronic Journal of Linear Algebra*, 12:42–63, 2005.

[116] M. D. McKay, R. J. Beckman, and W. J. Conover. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 21(2):239–245, 1979.

[117] C. A. Micchelli. Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions. *Constructive Approximation*, 2(1):11–22, 1986.

[118] J. Mockus, V. Tiesis, and A. Zilinskas. The Application of Bayesian Methods for Seeking the Extremum. In L. C. W. Dixon and G. P. Szego, editors, *Towards Global Optimization 2*, pages 117–129. North-Holland Publishing Company, Amsterdam, North-Holland, 1978.

[119] B. C. Moore. Principal Component Analysis in Linear Systems: Controllability, Observability and Model Reduction. *IEEE Trans. Autom. Control*, 26(1):17–32, 1981.

[120] E. H. Moore. On the Reciprocal of the General Algebraic Matrix. *Bulletin of the American Mathematical Society*, 26:394–395, 1920.

[121] C. Moosmann, E. B. Rudnyi, A. Greiner, J. G. Korvink, and M. Hornung. Parameter Preserving Model Order Reduction of a Flow Meter. *Proc. of the NSTI Nanotech*, 3:684–687, Anaheim, CA, 2005.

[122] C. Mullis and R. Roberts. Synthesis of Minimum Roundoff Noise Fixed Point Digital Filters. *IEEE Trans. Circuits Syst.*, 23(9):551–562, 1976.

[123] D. Mustafa and K. Glover. Controller Reduction by $\mathcal{H}_\infty$-Balanced Truncation. *IEEE Trans. Autom. Control*, 36(6):668–682, 1991.

[124] G. Obinata and B. Anderson. *Model Reduction for Control System Design.* Springer, London, UK, 2001.

[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 report, Inst. of Automatic Control, TU München, 2009.

[126] H. 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. *Proc. of the American Control Conference*, pages 5512–5517, Washington, DC, 2013.

[127] H. Panzer, B. Kleinherne, and B. Lohmann. Analysis, Interpretation and Generalization of a Strictly Dissipative State Space Formulation of Second Order Systems. In B. Lohmann and G. Roppenecker, editors, *Methoden und Anwendungen der Regelungstechnik. Erlangen-Münchener Workshops 2011 und 2012*, pages 59–75. Shaker-Verlag, Aachen, Germany, 2013.

[128] H. Panzer, J. Mohring, R. Eid, and B. Lohmann. Parametric Model Order Reduction by Matrix Interpolation. *at-Automatisierungstechnik*, 58(8):475–484, 2010.

[129] A. Paul-Dubois-Taine and D. Amsallem. An Adaptive and Efficient Greedy Procedure for the Optimal Training of Parametric Reduced-Order Models. *Int. J. Numer. Methods Engrg.*, 102(5):1262–1292, 2015.

[130] B. Peherstorfer, S. Zimmer, and H.-J. Bungartz. Model Reduction with the Reduced Basis Method and Sparse Grids. In *Sparse Grids and Applications*, pages 223–242. Springer, Heidelberg, Germany, 2013.

[131] L. Peng, F. Liu, L. T. Pileggi, and S. R. Nassif. Modeling Interconnect Variability Using Efficient Parametric Model Order Reduction. *Proc. of the Design, Automation and Test in Europe Conference and Exhibition*, pages 958–963, Munich, Germany, 2005.

[132] R. Penrose. A Generalized Inverse for Matrices. *Proc. Camb. Phil. Soc.*, 51(3):406–413, 1955.

[133] R. Penrose. On Best Approximate Solutions of Linear Matrix Equations. *Proc. Camb. Phil. Soc.*, 52(1):17–19, 1956.

[134] T. Penzl. Numerical Solution of Generalized Lyapunov Equations. *Advances in Computational Mathematics*, 8(1):33–48, 1998.

[135] T. Penzl. A Cyclic Low-Rank Smith Method for Large Sparse Lyapunov Equations. *SIAM J. Sci. Comput.*, 21(4):1401–1418, 2000.

[136] N. Petit and P. Rouchon. Flatness of Heavy Chain Systems. *SIAM J. Control Optim.*, 40(2):475–495, 2001.

[137] C. Poussot-Vassal and C. Roos. Generation of a Reduced-Order LPV/LFT Model from a Set of Large-Scale MIMO LTI Flexible Aircraft Models. *Control Engineering Practice*, 20(9):919–930, 2012.

[138] M. J. D. Powell. *Approximation Theory and Methods.* Cambridge University Press, Cambridge, UK, 1981.

[139] A. Preumont. *Vibration Control of Active Structures.* Kluwer Acedemic Publishers, Dordrecht, The Netherlands, 2nd edition, 2002.

[140] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, Cambridge, MA, 2006.

[141] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and Analysis of Computer Experiments. *Statistical Science*, 4(4):409–423, 1989.

[142] E. R. Samuel, K. Chemmangat, D. Deschrijver, L. Knockaert, and T. Dhaene. Model Order Reduction of Parameterized State-Space Systems with Sequential Sampling. *Proc. of the 2013 International Symposium on Electromagnetic Compatibility*, pages 342–347, Brugge, Belgium, 2013.

[143] E. R. Samuel, F. Ferranti, L. Knockaert, and T. Dhaene. Passivity-Preserving Parameterized Model Order Reduction Using Singular Values and Matrix Interpolation. *IEEE Trans. Compon. Packag. Manuf. Technol.*, 3(6):1028–1037, 2013.

[144] G. Schelfhout and B. De Moor. A Note on Closed-Loop Balanced Truncation. *IEEE Trans. Autom. Control*, 41(10):1498–1500, 1996.

[145] M. Schonlau. *Computer Experiments and Global Optimization.* PhD thesis, University of Waterloo, 1997.

[146] M. Schonlau, W. J. Welch, and D. R. Jones. Global versus Local Search in Constrained Optimization of Computer Models. In N. Flournoy, W. F. Rosenberger, and W. K. Wong, editors, *New Developments and Applications in Experimental Design*, pages 11–25. Institute of Mathematical Statistics, Hayward, CA, 1998.

[147] N. Shinozaki, M. Sibuya, and K. Tanabe. Numerical Algorithms for the Moore-Penrose Inverse of a Matrix: Direct Methods. *Annals of the Institute of Statistical Mathematics*, 24(1):193–203, 1972.

[148] R. C. Smith. *Uncertainty Quantification, Theory, Implementation, and Applications.* SIAM, Philadelphia, PA, 2014.

[149] I. M. Sobol. On the Distribution of Points in a Cube and the Approximate Evaluation of Integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967.

[150] N. T. Son. A Real Time Procedure for Affinely Dependent Parametric Model Order Reduction Using Interpolation on Grassmann Manifolds. *Int. J. Numer. Methods Engrg.*, 93(8):818–833, 2012.

[151] H. Späth. *Two Dimensional Spline Interpolation Algorithms.* A K Peters, Wellesley, MA, 1995.

[152] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* Springer, New York, NY, 3rd edition, 2002.

[153] M. Stone. Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society. Series B*, 36(2):111–147, 1974.

[154] J. F. Sturm. Using SeDuMi 1.02, a MATLAB Toolbox for Optimization over Symmetric Cones. *Optimization Methods and Software*, 11(1-4):625–653, 1999.

[155] T. Stykel. *Analysis and Numerical Solution of Generalized Lyapunov Equations.* PhD thesis, TU Berlin, 2002.

[156] T. Tamarozzi, G. H. K. Heirman, and W. Desmet. An On-line Time Dependent Parametric Model Order Reduction Scheme with Focus on Dynamic Stress Recovery. *Comput. Methods Appl. Mech. Engrg.*, 268:336–358, 2014.

[157] D. Thull, D. Wild, and A. Kugi. Infinite-Dimensional Control of a Gantry Crane with Heavy Chains. *at-Automatisierungstechnik*, 53(8):400–410, 2005.

[158] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving Semidefinite-Quadratic-Linear Programs Using SDPT3. *Mathematical Programming Series B*, 95(2):189–217, 2003.

[159] L. Vandenberghe and S. Boyd. Semidefinite Programming. *SIAM Review*, 38(1):49–95, 1996.

[160] A. Varga and B. D. O. Anderson. Accuracy-Enhancing Methods for Balancing-Related Frequency-Weighted Model and Controller Reduction. *Automatica*, 39(5):919–927, 2003.

[161] D. S. Weile, E. Michielssen, E. Grimme, and K. Gallivan. A Method for Generating Rational Interpolant Reduced Order Models of Two-Parameter Linear Systems. *Appl. Math. Letters*, 12(5):93–102, 1999.

[162] S. Weisberg. *Applied Linear Regression.* Wiley, Hoboken, NJ, 3rd edition, 2005.

[163] H. Wendland. *Scattered Data Approximation.* Cambridge University Press, Cambridge, UK, 2005.

[164] P. M. R. Wortelboer, M. Steinbuch, and O. H. Bosgra. Iterative Model and Controller Reduction Using Closed-Loop Balancing, with Application to a Compact Disc Mechanism. *Int. J. Robust Nonlinear Control*, 9(3):123–142, 1999.

[165] A. Yousefi. *Preserving Stability in Model and Controller Reduction with Application to Embedded Systems.* PhD thesis, TU München, 2006.

[166] M. Zeitz. Differential Flatness: A Useful Method also for Linear SISO Systems. *at-Automatisierungstechnik*, 58(1):5–13, 2010.

[167] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control.* Prentice Hall, Englewood Cliffs, NJ, 1996.

[168] R. Zimmermann. A Locally Parametrized Reduced-Order Model for the Linear Frequency Domain Approach to Time-Accurate Computational Fluid Dynamics. *SIAM J. Sci. Comput.*, 36(3):508–537, 2014.