Bilinear $\mathscr{H}_2$-optimal Model Order Reduction with applications to thermal parametric systems

Dissertation
zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr.rer.nat.)

von Dipl.-Math. Angelika Susanne Bruns
geb. am 06.01.1986 in Freudenstadt
genehmigt durch die Fakultät für Mathematik
der Otto-von-Guericke-Universität Magdeburg

Gutachter: Prof. Dr. Peter Benner
Prof. Dr. Tobias Damm

eingereicht am 20.03.2015
Verteidigung am 30.06.2015

Contents

Danksagung

Diese Arbeit wäre ohne die Beteiligung vieler Personen in dieser Art und Weise nicht möglich gewesen. Natürlich gilt mein Dank an erster Stelle meinem Betreuer Prof. Dr. Peter Benner, der sich bei meinen Besuchen in Magdeburg immer Zeit für mich nahm, mir hilfreiche Hinweise und Tipps gab und insbesondere das Wagnis Industrie- und "Numerik-fremde" Doktorandin einging. Herzlichen Dank!
Außerdem bedanke ich mich herzlich bei Prof. Dr. Tobias Damm für die Übernahme der Zweitkorrektur.
Die CSC Gruppe am MPI in Magdeburg hat mich bei meinen Besuchen immer freundlich aufgenommen. Danke für eure Hilfsbereitschaft und diverse schöne Abende in Magdeburg, Schloss Ringberg, im Harz und in Kroatien. Ein spezieller Dank geht an meine Mentorin Dr. Ulrike Baur.
Meine Promotionszeit bei Bosch wäre nicht das Gleiche gewesen ohne meine Kollegen aus der "Mathematikerecke". Ich danke hier insbesondere Dr. Katrin Schumacher und Dr. Rudy Eid für die hervorragende Betreuung - nicht nur fachlich, auch persönlich habe ich viel von euch gelernt. Ebenso bedanke ich mich bei allen anderen Kollegen aus der CR/ARH, die mir vor allem bei Fragen zu thermischen Simulationen weiter geholfen haben, zusätzlich danke ich Dr. Kilian Kriener und Thomas Heid von ED/ESY3.
Ich danke außerdem meinen Eltern und meiner Schwester für ihre Unterstützung während nun schon über 10 Jahren Mathematik. Danke an Sophie für über 6 Jahre Horizonterweiterung.
Johannes — Danke. Für alles. Ich liebe dich.

Zusammenfassung

Wird in der Industrie eine neue Komponente entwickelt, so spielen Computersimulationen mittlerweile eine wichtige Rolle. Immer schnellere und immer genauere Simulationsmodelle werden gewünscht, damit Zeit und Kosten gespart werden können. Mit Hilfe von Modellordnungsreduktion (MOR) kann man aus großen, mit der Finite Elemente Methode erstellten Modellen kleine und genaue Modelle erhalten, die dann in kurzer Zeit simuliert werden können. Immer häufiger wird auch gefordert, die Variation von Parametern im großen Finite Elemente Modell auf die kleinen reduzierten Modelle zu übertragen. Diese Parameter beschreiben beispielsweise verschiedene Randbedingungen, die im Modell abgebildet werden, genauso wie Änderungen in der Geometrie (z.B. Variation von Längen). Mit Hilfe von Methoden aus der parametrischen Modellordnungsreduktion (pMOR) können diese Parameterabhängigkeiten auch im reduzierten Modell erhalten und zur Simulation von unterschiedlichen Szenarien genutzt werden.
Anstatt die heute üblichen Verfahren zur pMOR zu benutzen, werden in dieser Arbeit die parametrischen Modelle, die eine spezielle Parameterabhängigkeit zeigen, in bilineare Modelle umgeschrieben. Nun können auch Verfahren zur bilinearen Modellordnungsreduktion angewandt werden, insbesondere Verfahren zur $\mathscr{H}_2$-optimalen Reduktion. Ziel dieser $\mathscr{H}_2$-optimalen Verfahren ist es, den Fehler zwischen dem Ausgangsmodell und dem reduzierten Modell in der $\mathscr{H}_2$-Norm zu minimieren. Wir verwenden zum einen den sogenannten **B**ilinear **I**nterpolatory **R**ational **K**rylov **A**lgorithm (BIRKA) von Benner und Breiten [12]. Außerdem entwickeln wir neue bilineare $\mathscr{H}_2$-optimale Algorithmen, die auf Optimierungsverfahren auf Grassmann-Mannigfaltigkeiten beruhen.
Die theoretischen Grundlagen der thermischen Modellierung werden erklärt und auf die erstellten Modelle von Elektromotoren angewandt. Parametrische

Modelle können aus den Finite Elemente Modellen durch eine Analyse der Gleichungen abgeleitet werden. Die Parameter sind einerseits Größen, die das thermische Verhalten während des Betriebs erklären und andererseits Größen, die Variationen in der Geometrie des Motors beschreiben. Diese Parameter sollen in den reduzierten Modellen erhalten bleiben.
Während die neu entwickelten Algorithmen noch nicht reif für die Reduktion von großen Modellen sind, wird in der Arbeit gezeigt, dass die Reduktion mit BIRKA zu guten reduzierten Modellen führt. Allerdings müssen dazu verschiedene Nachbesserungen an der Reduktionsmethodik vorgenommen werden, beispielsweise müssen Methoden zur Stabilitätserhaltung angewandt werden. In Modellen mit Variationen in der Geometrie, werden zusätzlich zum ursprünglichen BIRKA nach der Reduktion noch Interpolationsverfahren verwendet, um reduzierte Modelle mit der Parameterabhängigkeit des Originalmodells zu erhalten.

Summary

The design process of a new component in industry is nowadays almost always accompanied by computer simulations. In order to save time and money, fast and accurate models for the simulation of the component are required. Using Model Order Reduction (MOR) large models obtained by Finite Element simulations can be reduced to small models possessing the same behavior as the original. Often it is required to obtain reduced models, where the dependence in one or several parameters (for example the length or width of a part) of the original model is preserved. Using so called parametric Model Order Reduction (pMOR) the parameters in the reduced model can be varied and the models can be used for fast simulation of several scenarios.
Instead of using the commonly employed methods from pMOR, methods from bilinear Model Order Reduction will be used within this work, as parametric models with a certain form of parameter dependence can be rewritten as bilinear models. We focus on methods from bilinear $\mathscr{H}_2$-optimal Model Order Reduction, as their objective is to minimize the error between the original and the reduced model measured in the $\mathscr{H}_2$-norm. First, the **B**ilinear **I**nterpolatory **R**ational **K**rylov **A**lgorithm (BIRKA) developed by Benner and Breiten [12] is used. Second, we derive new bilinear $\mathscr{H}_2$-optimal algorithms based on optimization on Grassmann manifolds.
The foundations of thermal modeling and their application to thermal simulations of electrical motors using Finite Element software will be explained. Parametric models suitable for pMOR can be derived from a Finite Element software analyzing the underlying equations. Two classes of parameters will be considered: Constants influencing the thermal behavior of the model and changes in the geometry of the model.
Using the newly developed optimization algorithms for $\mathscr{H}_2$-optimal MOR,

we find that they are not yet ready for the reduction of large parametric models as encountered in our thermal simulations. In contrast, the BIRKA performs well for the reduction of these models. However, several modifications on the reduction methods need to be performed to assure, for example, the preservation of stability during the reduction. For the reduction of models with parameters resulting from changes in the geometry, interpolation procedures need to be applied after the reduction to transfer the parameter dependence of the original to the reduced model.

List of Figures

List of Tables

List of Algorithms

Notations

$\mathbb{R}$	field of real numbers
$\mathbb{C}$	field of complex numbers
$\mathrm{Re}(z)$	real part of $z \in \mathbb{C}$
$\mathrm{Im}(z)$	imaginary part of $z \in \mathbb{C}$
A^T	transpose of the matrix $A \in \mathbb{C}^{n \times m}$
A^*	conjugate transpose of the matrix $A \in \mathbb{C}^{n \times m}$
A^{-1}	inverse of the matrix $A \in \mathbb{C}^{n \times n}$
A^{-*}	inverse, conjugate transpose of the matrix $A \in \mathbb{C}^{n \times n}$
I_n	identity matrix of dimension $n \times n$
$\sigma_i(A)$	i-th singular value of a matrix $A \in \mathbb{C}^{n \times m}$, singular values ordered by magnitude
$\lambda_i(A)$, $\lambda_i(A, E)$	i-th eigenvalue of a matrix $A \in \mathbb{C}^{n \times n}$ in some ordering, generalized eigenvalue of the pencil $A - \lambda E$ in some ordering
$\|\|A\|\|_2 = \sigma_{\max}(A)$	spectral norm of a matrix
$\mathrm{vec}(A) = \left[a_{11}, \ldots, a_{1n}, a_{21}, \ldots, a_{mn}\right]^T$	the vector formed by stacking the columns of the matrix $A \in \mathbb{C}^{n \times m}$
$\det(A)$	determinant of a matrix A
$\mathrm{rk}(A)$	rank of a matrix A

$\mathrm{tr}(A) = \sum_{k=1}^{n} a_{kk}$	trace of a matrix A
$\mathrm{span}(A)$	space spanned by the columns of the matrix A
$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1m}B \\ \vdots & & \vdots \\ a_{n1}B & & a_{nm}B \end{bmatrix}$	Kronecker product of two matrices $A \in \mathbb{C}^{n \times m}$ and $B \in \mathbb{C}^{k \times l}$
$\mathrm{diag}(a_{11}, \dots, a_{nn}) = \begin{bmatrix} a_{11} & & 0 \\ & \ddots & \\ 0 & & a_{nn} \end{bmatrix}$	matrix with diagonal $(a_{11}, \dots, a_{nn})$
$\mathcal{O}_n \subset \mathbb{R}^{n \times n}$	orthogonal real matrices i.e. matrices with $A^T A = A A^T = I_n$
P	reachability Gramian
Q	observability Gramian
$\kappa_2(A) = \|\|A\|\|_2 \|\|A^{-1}\|\|_2 = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$	2-norm condition number

CHAPTER 1

Introduction

1.1. Motivation

In industry, simulations are an important tool in the design process of a new component. In order to save time and money, fast and accurate models for simulation are needed. Model Order Reduction (MOR) is a powerful method to obtain small and accurate models from large Finite Element models. More and more often, Finite Element models are used, which contain several parameters. Such parameters can be lengths and heights as well as physical behavior. These parametrized models will often be used to find optimal designs by using optimization w.r.t. the given parameters. As the Finite Element models are large, optimization runs can easily exceed the computation capacities. It is hence desirable to reduce models while preserving the parameter dependency. This is the objective of parametric Model Order Reduction (pMOR). Recently, Benner and Breiten [11] presented a method to rewrite linear parametric models into bilinear models. This allows bilinear Model Order Reduction methods to be used for parametric Model Order Reduction. The resulting reduced order model should be a good approximation of the original model. Within the framework of $\mathscr{H}_2$-optimal Model Order Reduction, the error can be measured and minimized in the $\mathscr{H}_2$-norm. In this work, we will examine bilinear $\mathscr{H}_2$-optimal methods for the reduction of linear parametric systems, which have been applied to and further developed on thermal models of electrical motors.

1.2. Dissertation overview

In Chapter 2, we review results from Linear Algebra, Differential Geometry and Systems Theory. The concepts will be stated for linear and bilinear systems.
Chapter 3 provides the reader with the foundations of heat transfer modeling. The underlying physical effects (heat conductance, convective heat transfer, radiation) will be reviewed and the mode of operation and the thermal modeling of an electrical motor will be described. Three different electrical motor models have been built and will be presented. Chapter 4 gives an overview over the equations that are solved during heat transfer modeling, and the procedure to obtain parametric models by careful analysis of these equations.
In Chapter 5, methods for Model Order Reduction (MOR) will be discussed. First, methods for linear MOR will be reviewed, followed by a discussion of methods for the reduction of parameter dependent models (parametric MOR). It is possible to rewrite parametric models with a certain parameter dependency as bilinear models, and hence methods from bilinear MOR will be considered. Of particular interest are methods from the class of $\mathscr{H}_2$-optimal bilinear MOR, as their objective is to minimize the error between original and reduced model. First, we review existing methods and state the **B**ilinear **I**nterpolatory **R**ational **K**rylov **A**lgorithm (BIRKA) [12]. Second, we develop algorithms for the reduction of bilinear systems via optimization on Grassmann manifolds. These methods are of interest, as they preserve stability during the reduction process.
The objective of Chapter 6 is the discussion of several issues that were encountered while applying BIRKA to thermal models. These issues are examined, and strategies for their mitigation will be developed. Especially preservation of stability during the calculation is crucial. Results for BIRKA and the new $\mathscr{H}_2$-optimal methods will be given in Chapters 7 and 8. Whereas the new methods are not yet applicable to large systems, BIRKA performs well on bilinear systems that have been obtained from linear parametric systems. First, only physical parameters are considered. Second, we present results for systems with a parameter dependency resulting from changes in geometry, which can only be rewritten partially as bilinear systems. For such systems, parametric reduced order models can then be obtained by an interpolation procedure.

1.3. Thesis contributions

The main contributions of this thesis are:

- One objective of this thesis is MOR of thermal electrical motor models. Hence, it is shown how matrices suitable for pMOR can be obtained from Comsol®, a Finite Element Software. To do so, the equations which are solved by the Software are used to theoretically reconstruct the dependence in parameters of the model (cf. Chapter 4).
- In contrast to other works about pMOR, in this thesis the reduction of the parametric models is done using BIRKA [12]. Several issues where encountered when the algorithm was applied: One class of parameters leads to a non-singular stiffness matrix, in several cases there is the need to scale other system matrices to fulfill a Kronecker product approximation and in addition, BIRKA does not preserve stability. All these issues have been resolved, and we show results for the reduction of a motor model from $n = 41,199$ degrees of freedom to a reduced order of $r = 300$. This has been done for 13 physical parameters.
- In addition, models with geometrical variations are considered. After the reduction with BIRKA, several interpolation strategies between the reduced order models obtained in several parameter points have been compared.
- Finally, we develop new $\mathscr{H}_2$-optimal bilinear methods for MOR using optimization on Grassmann manifolds. These methods can preserve stability for symmetric systems matrices, and their applicability to small models will be proved.

CHAPTER 2

Mathematical prerequisites

In this first theoretical chapter, some results from different areas of mathematics are reviewed. First, general results from Linear Algebra will be presented, followed by a closer look on some definitions from Differential Geometry. The last section provides the reader with an introduction to linear and bilinear systems theory.

2.1. Linear Algebra

Within this section we review the decomposition of matrices, the properties of the Kronecker product and provide the reader with basic knowledge on matrix pertubation theory.

2.1.1. Matrices and their decompositions. Most of the matrices in this work are symmetric, which is why we state the definition here.

Definition 2.1.1. A matrix $A \in \mathbb{R}^{n\times n}$ is called *symmetric* if $A = A^T$. A symmetric matrix is *positive (semi)definite*, denoted by $A > (\geq)0$, if $x^T Ax > (\geq)0$ for all vectors $0 \neq x \in \mathbb{R}^n$. It is *negative (semi)definite*, denoted by $A < (\leq)0$, if $x^T Ax < (\leq)0$ for all vectors $0 \neq x \in \mathbb{R}^n$.

We will often refer to the following two matrix decompositions, the eigenvalue and the singular value decomposition.

Definition 2.1.2 (Generalized eigenvalue decomposition [38, Section 7.7])**.** If $A, B \in \mathbb{C}^{n\times n}$, then the set of all matrices of the form $A - \lambda B$ with $\lambda \in \mathbb{C}$ is a *pencil*. The *generalized eigenvalues* of $A - \lambda B$ are elements of the set $\lambda(A, B)$ defined as

$$\lambda(A, B) = \{z \in \mathbb{C} : \det(A - zB) = 0\}.$$

If $\lambda \in \lambda(A, B)$ and $0 \neq x \in \mathbb{C}^n$ satisfies

$$Ax = \lambda Bx, \tag{2.1}$$

then x is an *eigenvector* of $A - \lambda B$. The problem of finding nontrivial solutions to (2.1) is the *generalized eigenvalue problem*. If B is nonsingular, $\lambda(A, B) = \lambda(B^{-1}A)$ holds.

Theorem 2.1.3 (The singular value decomposition (SVD) [38, Theorem 2.4.1])**.** *If $A \in \mathbb{R}^{m\times n}$, then there exist orthogonal matrices $U = [u_1, \ldots, u_m] \in \mathbb{R}^{m\times m}$ and $V = [v_1, \ldots, v_n] \in \mathbb{R}^{n\times n}$ such that*

$$U^T AV = \operatorname{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m\times n}, \tag{2.2}$$

with $p = \min(m, n)$ where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$.

The σ_j will be called *singular values*. If it shall be clarified that they result from a singular value decomposition of the matrix A, we denote them by $\sigma_j(A)$. Let r be such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_p = 0$. Then $\operatorname{rk}(A) = r$ and A can be decomposed in the following way:

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T.$$

Using matrices, we will write this decomposition as follows:

$$A = U_r \Sigma_r V_r^T, \tag{2.3}$$

with $U_r \in \mathbb{R}^{m\times r}$, $\Sigma_r \in \mathbb{R}^{r\times r}$ and $V_r \in \mathbb{R}^{n\times r}$ and refer to it as the *compact singular value decomposition*.

2.1.2. Properties of the Kronecker product. The following matrix product is referred to as the Kronecker product:

Definition 2.1.4. For two matrices $A \in \mathbb{C}^{n\times m}$ and $B \in \mathbb{C}^{k\times l}$, the *Kronecker product* is defined as:

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1m}B \\ \vdots & & \vdots \\ a_{n1}B & & a_{nm}B \end{bmatrix}.$$

The Kronecker product has the following properties (see for example [38], Section 12.3):

$$\begin{aligned}
(A \otimes B)^T &= A^T \otimes B^T, \text{ with } A \in \mathbb{C}^{n\times m}, B \in \mathbb{C}^{k\times l}, \\
(A \otimes B)^{-1} &= A^{-1} \otimes B^{-1}, \text{ with } A \in \mathbb{C}^{n\times m}, B \in \mathbb{C}^{k\times l}, \\
(A \otimes B) \otimes C &= A \otimes (B \otimes C), \text{ with } A \in \mathbb{C}^{n\times m}, B \in \mathbb{C}^{k\times l} \text{ and } C \in \mathbb{C}^{s\times q}, \\
(AC \otimes BD) &= (A \otimes B)(C \otimes D), \\
&\text{with } A \in \mathbb{C}^{n\times m}, B \in \mathbb{C}^{k\times l}, C \in \mathbb{C}^{m\times s} \text{ and } D \in \mathbb{C}^{l\times q},
\end{aligned}$$

but in general $A \otimes B \neq B \otimes A$! In addition one obtaines (with $A \in \mathbb{C}^{n\times m}$, $B \in \mathbb{C}^{k\times l}$):

$$\begin{aligned}
\mathrm{rk}(A \otimes B) &= \mathrm{rk}(A) \cdot \mathrm{rk}(B), \\
\det(A \otimes B) &= \det(A)^n \cdot \det(B)^m \text{ for } A \in \mathbb{R}^{m\times m} \text{ and } B \in \mathbb{R}^{n\times n}, \\
\mathrm{tr}(A \otimes B) &= \mathrm{tr}(A) \cdot \mathrm{tr}(B), \\
||A \otimes B||_2 &= ||A||_2 \cdot ||B||_2.
\end{aligned}$$

If $C = AXB$ for $C \in \mathbb{R}^{n\times m}, A \in \mathbb{R}^{n\times k}, X \in \mathbb{R}^{k\times l}$ and $B \in \mathbb{R}^{l\times m}$ then one obtains for the Kronecker product and the vec operator:

$$\mathrm{vec}(C) = (B^T \otimes A)\,\mathrm{vec}(X). \tag{2.4}$$

2.1.3. Matrix pertubation theory. The connection between the eigenvalues of two matrices will be needed within this work. The following results have been established in the context of matrix pertubation theory, the relation of the eigenvalues of a pertubed Matrix $M + S$ and the unpertubed matrix M will be examined.

Theorem 2.1.5 (Bauer-Fike,[38, Theorem 7.2.2]). *If μ is an eigenvalue of $M+S \in \mathbb{C}^{n\times n}$ and $X^{-1}MX = diag(\lambda_1, \dots, \lambda_n)$, then*

$$\min_{i=1,\dots,n} |\lambda_i - \mu| \leq \kappa_2(X)||S||_2. \tag{2.5}$$

Corollary 2.1.6. *Let $X^{-1}MX = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$, and $M+S \in \mathbb{C}^{n\times n}$. For every eigenvalue $\lambda(M+S)$ an eigenvalue $\lambda_i(M)$ exists such that $|\lambda_i(M) - \lambda(M+S)| \leq \kappa_2(X)||S||_2$.*

The next results show the connection between the eigenvalues of two real symmetric matrices A and B.

Proposition 2.1.7 (Weyl,[60, Theorem 4.8, Corollary 4.9]). *Let $A, B \in \mathbb{R}^{n\times n}$ be two symmetric matrices. Let $\lambda_i(A)$ and $\lambda_i(B)$ for $i = 1, \dots, n$ be the eigenvalues of A and B with $\lambda_1(A) \geq \cdots \geq \lambda_n(A)$ and $\lambda_1(B) \geq \cdots \geq \lambda_n(B)$. Then it holds:*

$$\lambda_i(A+B) \in [\lambda_i(A) + \lambda_n(B), \lambda_i(A) + \lambda_1(B)] \text{ for } i = 1, \dots, n. \tag{2.6}$$

Corollary 2.1.8 ([60, Corollary 4.10]). *Under the assumptions of Proposition 2.1.7 it holds*

$$|\lambda_i(A+B) - \lambda_i(A)| \leq ||B||_2 \text{ for } i = 1, \dots, n. \tag{2.7}$$

2.2. Differential geometry

In Section 5.5.4, several algorithms based on optimization on manifolds will be derived. For a more detailed presentation of this topic, we refer to [1] and [30]. Let $\mathcal{O}_r$ denote the set of the orthogonal matrices in $\mathbb{R}^{r\times r}$.

Definition 2.2.1 (Stiefel manifold [1, Section 3.3.2]). For $r \leq n$, the *Stiefel manifold* is defined as the set of all $n \times r$ orthonormal matrices:

$$\mathrm{St}(r,n) := \{X \in \mathbb{R}^{n\times r} | X^T X = I_r\}.$$

Clearly, $\mathrm{St}(r,n) \subset \mathbb{R}^{n\times r}$. It can be shown that $\mathrm{St}(r,n)$ is a compact submanifold of $\mathbb{R}^{n\times r}$ (cf. [1, Section 3.3.2]). The *tangent space* of a Stiefel manifold at $X \in \mathrm{St}(r,n)$ is defined as follows (cf. [1, Example 3.5.2]):

$$\mathcal{T}_X\mathrm{St}(r,n) = \{Z \in \mathbb{R}^{n\times r} | X^T Z + Z^T X = 0\}.$$

Heading for an algorithm for the gradient flow, the gradient of a function on the manifold has to be calculated. Therefore, we first have to provide a concept of direction and length of a tangent vector. This leads to the

definition of an inner product on the tangent space. For a Stiefel manifold, the *inner product* is defined as

$$\langle \xi, \eta \rangle = \operatorname{tr}(\xi^T \eta) \text{ with } \xi, \eta \in \mathcal{T}_X \mathrm{St}(r, n). \tag{2.8}$$

The *gradient in X of a function F on a Stiefel manifold* is defined to be the tangent vector ∇F such that

$$\operatorname{tr}(F_X^T Y) = \operatorname{tr}((\nabla F)^T (I - \frac{1}{2} X X^T) Y), \tag{2.9}$$

holds for all tangent vectors $Y \in \mathcal{T}_X \mathrm{St}(r, n)$. Here, F_X is the matrix of all partial derivatives of F with respect to X, i.e.:

$$(F_X)_{ij} = \frac{\partial F}{\partial X_{ij}}. \tag{2.10}$$

Solving equation (2.9) leads to the following expression for the gradient:

$$\nabla F = F_X - X F_X^T X. \tag{2.11}$$

The *Grassmann manifold* $\mathrm{Gr}(r, n)$, $r \leq n$, is defined as the set of all r-dimensional subspaces of $\mathbb{R}^n$. Following [30], it can be seen as a quotient manifold in the following way: Two matrices U_1 and U_2 in $\mathrm{St}(r, n)$ are equivalent, if they span the same r-dimensional subspace. This holds if and only if $U_1 = U_2 Q$ for an orthogonal matrix $Q \in \mathbb{R}^{r \times r}$. The equivalence class $[U]$ of a point $U \in \mathrm{St}(r, n)$ can be defined as:

$$[U] = \{UQ | Q \in \mathcal{O}_r\}.$$

The map

$$G : \mathrm{Gr}(r, n) \to \mathrm{St}(r, n)/\mathcal{O}_r$$

is a bijection. We will therefore consider the Grassmann manifold as this quotient manifold of $\mathrm{St}(r, n)$. A matrix $U \in \mathrm{St}(r, n)$ represents a whole equivalence class in $\mathrm{Gr}(r, n)$. The *tangent space of the Grassmann manifold* can be described as follows [30, Section 2.5]:

$$\mathcal{T}_X \mathrm{Gr}(r, n) = \{Z \in \mathbb{R}^{n \times r} | X^T Z = 0\}. \tag{2.12}$$

On a manifold, the shortest connection between two points is called a geodesic. Let $X(0) = X$ and $\dot{X}(0) = H$. Let $H = W \Sigma V^T$ be the compact singular value decomposition (cf. equation (2.3)) of H with $W \in \mathbb{R}^{n \times r}$, $\Sigma, V \in \mathbb{R}^{r \times r}$. The geodesic can be described as [30, Section 2.5.1]:

$$X(t) = \begin{bmatrix} XV & W \end{bmatrix} \begin{bmatrix} \cos \Sigma t \\ \sin \Sigma t \end{bmatrix} V^T. \tag{2.13}$$

For a Grassmann manifold, the inner product is defined as

$$\langle \xi, \eta \rangle = \operatorname{tr}(\xi^T \eta), \text{ with } \xi, \eta \in \mathcal{T}_X \mathrm{Gr}(r, n). \tag{2.14}$$

The *gradient in X of a function F on the Grassmann manifold* is defined to be the tangent vector ∇F such that

$$\operatorname{tr}(F_X^T Y) = \operatorname{tr}((\nabla F)^T Y), \tag{2.15}$$

holds for all tangent vectors $Y \in \mathcal{T}_X \mathrm{Gr}(r, n)$. Solving equation (2.15) leads to the following expression for the gradient [30, Section 2.5.3]:

$$\nabla F = F_X - X X^T F_X. \tag{2.16}$$

We will also need the following definition:

Definition 2.2.2 ([1, Definition 4.2.1])**.** Given a function F on $\mathrm{St}(r, n)$ or $\mathrm{Gr}(r, n)$, a sequence $\{\eta_k\}$, $\eta_k \in \mathcal{T}_{x_k} \mathrm{St}(r, n)$ or $\eta_k \in \mathcal{T}_{x_k} \mathrm{Gr}(r, n)$ is *gradient-related* if, for any subsequence $\{x_k\}_{k \in \mathcal{K}}$ of $\{x_k\}$ that converges to a non-critical point of F, the corresponding subsequence $\{\eta_k\}_{k \in \mathcal{K}}$ is bounded and satisfies

$$\lim_{k \to \infty} \sup_{k \in \mathcal{K}} \langle \nabla F(x_k), \eta_k \rangle < 0. \tag{2.17}$$

2.3. Systems theory

Many physical phenomena, chemical reactions, biological processes or models for the forecast of financial processes can be mathematically described by the same class of systems, so called dynamical systems. External influences that have a direct impact on the behavior of the system are called inputs. The behavior of the systems will be monitored within a certain time range and at certain points, the system's outputs. The connection between the inputs and the outputs will often be measured and referred to as the system's input-output-relationship. A dynamical system can be described by a differential equation. In this work, two kinds of dynamical systems will be considered: linear and bilinear systems.

2.3.1. Linear Systems. In the following section some basic knowledge on linear dynamical systems will be reviewed, such as stability, observability, controllability, balanced systems, norms of systems and the input-output relationship.

Definition 2.3.1. A *linear system* Σ_{lin} of order n is a system of ordinary differential equations of the following form:

$$\Sigma_{\text{lin}} : \begin{cases} E\dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), x(0) = x_0, \end{cases} \tag{2.18}$$

where $E, A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$. The input $u(t) \in \mathbb{R}^m$ can be time-dependent just as the states $x(t) \in \mathbb{R}^n$ and the output $y(t) \in \mathbb{R}^p$ are. The value of $x(0) = x_0$ is called initial value. The space $\mathcal{X}$ containing all states $x(t)$ is called state space.

2.3.1.1. *Stability.* Systems with bounded solution trajectories $x(t)$ are of special importance. This characteristic of a system is referred to as stability. For linear systems (c.f. system (2.18)) with nonsingular E, stability is defined as follows:

Definition 2.3.2 (c.f. [63] Chapter 2.7,[5] Chapter 5.8,[61] Chapter 3.2.1)**.** The system

$$E\dot{x}(t) = Ax(t), E \text{ nonsingular},$$

is *asymptotically stable* if

(i) For all $x^0 \in \mathbb{R}^n$ the initial value problem $E\dot{x}(t) = Ax(t), x(0) = x^0$, has a solution and for every $\varepsilon > 0$ there exists a $\delta > 0$ such that $||x(t)||_2 < \varepsilon$ for all $t \geq 0$ and for all $||x(0)||_2 < \delta$ (Lyapunov stability).

(ii) There exists $\delta > 0$ such that $x(t) \to 0$ as $t \to \infty$ if $||x(0)||_2 < \delta$.

Theorem 2.3.3 ([63] Corollary 2.11, [61] Theorem 3.7)**.** *The system*

$$E\dot{x}(t) = Ax(t), E \text{ nonsingular},$$

is asymptotically stable if and only if all the eigenvalues of $\lambda E - A$ lie in the open left half-plane.

We will therefore speak of a stable system, if all the eigenvalues of $\lambda E - A$, E nonsingular, lie in the open left half-plane. In this case, the eigenvalues of the pencil $\lambda E - A$ are those of the matrix $E^{-1}A$.

2.3.1.2. *Controllability, Observability and Balanced Systems.* During the analysis of a linear system (2.18) one might ask how the system is affected by the input $u(t)$. The following two characterisations will be considered.

Definition 2.3.4 ([5]). $x^* \in \mathbb{R}^n$ is *reachable* (from the origin $x(0) = 0$) if there exist an admissible input function and $t_e < \infty$ such that $x(t_e) = x^*$ holds (and hence $x(t_e) = x^*$ belongs to the state space of a linear system (2.18)).

Definition 2.3.5 ([5, 46]). A nonzero state $x(0) = x_0$ is *controllable* if there exists an admissible input function such that the system can be transformed from x_0 to any given end state $x(t_e)$ within a finite time $[0, t_e]$.

For linear continuous time systems the concepts of controllability and reachability coincide (cf. [5], Theorem 4.18). Hence, the following concepts will be developed for the controllability of a linear system. In the following chapters we will need the concept of the controllability Gramian.

Definition 2.3.6 ([61] Lemma 4.57). Consider a stable linear system (2.18) with E nonsingular. The *controllability Gramian* can be defined as follows:

$$P = \frac{1}{2\pi} \int_{-\infty}^{\infty} (i\omega E - A)^{-1} BB^* (i\omega E - A)^{-*} d\omega. \tag{2.19}$$

If one considers the eigenvalue decomposition of P, the eigenvalues measure the degree of controllability, whereas the eigenvectors corresponding to the largest eigenvalues can be understood as the directions in which the system is easy to control.

Proposition 2.3.7 ([61] Corollary 4.58). *Consider a stable linear system (2.18) with E nonsingular. The controllability Gramian P (2.19) exists and is the unique Hermitian solution to the following Lyapunov equation:*

$$AXE^T + EXA^T + BB^T = 0. \tag{2.20}$$

In addition, P is positive definite if and only if the system is controllable.

In practice, we will often be able to measure the output $y(t)$ of a linear system (2.18). If the input $u(t)$ and the output $y(t)$ are known, we want to reconstruct the states $x(t)$. This leads to the concept of observability.

Definition 2.3.8 ([46]). A linear system (2.18) is completely *observable*, if the initial state x_0 can be reconstructed from the behavior of the input $u(t)$ and the output $y(t)$ within a finite time interval $[0, t_e]$.

Again, we will need the concept of the systems observability Gramian.

Definition 2.3.9. Consider a stable linear system (2.18) with E nonsingular. The *observability Gramian* Q is defined as follows:

$$Q = E^T \tilde{Q} E,$$

with

$$\tilde{Q} = \frac{1}{2\pi} \int_{-\infty}^{\infty} (i\omega E - A)^{-*} C^* C (i\omega E - A)^{-1} d\omega. \tag{2.21}$$

The interpretation is similar to the controllability case: If one considers the eigenvalue decomposition of Q, the eigenvalues measure the degree of observability, whereas the largest eigenvectors can be understood as the directions in which the system is easy to observe.

Proposition 2.3.10 ([61] Corollary 4.58)**.** *Consider a stable linear system (2.18) with E nonsingular. The matrix $\tilde{Q}$ (see Definition 2.3.9) exists and is the unique Hermitian solution to the following Lyapunov equation:*

$$A^T X E + E^T X A + C^T C = 0. \tag{2.22}$$

In addition, $\tilde{Q}$ and therefore also the observability Gramian Q is positive definite if and only if the system is observable.

A balanced representation of a linear dynamical system is a representation of the system in which every state is "equally" reachable and observable. This section introduces the concepts which will be needed for the Balanced Truncation Model Order Reduction in Section 5.2.1. The reader should note that there exist several other balanced representations beside the one presented here. They can be found in the work by Gugercin and Antoulas [40] and the references therein.

Definition 2.3.11 ([61, Definition 7.5])**.** The Hankel singular values, denoted by ς_j, of a stable linear system (2.18) with E nonsingular are the square-roots of the eigenvalues of PQ.

Proposition 2.3.12 ([61, Corollary 7.7])**.** *A stable linear system (2.18) with E nonsingular is controllable and observable if and only if its Hankel singular values are non-zero.*

Definition 2.3.13 ([61, Definition 7.10])**.** A stable linear system (2.18) with E nonsingular is called *balanced*, if the controllability and the observability Gramians are equal and diagonal.

Every stable, controllable and observable linear system with E nonsingular can be transformed into a balanced representation. To do so, one computes the Cholesky factorization of the Gramians

$$P = RR^T \text{ and } \tilde{Q} = L^T L,$$

which exists due to the positive definiteness of P and $\tilde{Q}$ (cf. Propositions 2.3.7 and 2.3.10). Computing the QR decomposition of the Cholesky factors L and R leads to the following decomposition with orthogonal matrices Q_c and Q_o:

$$R^T = Q_c \tilde{R}^T \text{ and } L = Q_o \tilde{L}.$$

It is obvious that $P = RR^T = \tilde{R}\tilde{R}^T$ and $\tilde{Q} = L^T L = \tilde{L}^T \tilde{L}$. The Hankel singular values can now be computed via the singular values of $\tilde{L}E\tilde{R}$:

$$\varsigma_j^2 = \lambda_j(P \underbrace{E^T \tilde{Q} E}_{Q}) = \lambda_j(\tilde{R}\tilde{R}^T E^T \tilde{L}^T \tilde{L} E) = \lambda_j(\tilde{R}^T E^T \tilde{L}^T \tilde{L} E \tilde{R}) = \sigma_j^2(\tilde{L}E\tilde{R}),$$

with the singular value decomposition

$$\tilde{L}E\tilde{R} = U_b \Sigma V_b^T,$$

and orthogonal U_b, V_b and $\Sigma = \operatorname{diag}(\varsigma_1, \ldots, \varsigma_n)$. The matrices of the linear system can now be transformed to a balanced system representation:

$$W_b^T E T_b,\ W_b^T A T_b,\ W_b^T B,\ C T_b,$$

where

$$W_b = \tilde{L}^T U_b \Sigma^{-1/2}, \quad T_b = \tilde{R} V_b \Sigma^{-1/2}, \quad W_b^{-1} = T_b^T E^T, \quad T_b^{-1} = W_b^T E.$$

The Gramian (as the observability and the controllability Gramian coincide cf. Definition 2.3.13) of the balanced system is obtained from those of the original system in the following way:

$$T_b^{-1} P T_b^{-T} = \Sigma = W_b^{-1} \tilde{Q} W_b^{-T} = T_b^T Q T_b.$$

2.3.1.3. *Systems norms and spaces and input-output relationship.* As the objective is to approximate the given original models, one needs to be able to quantify the difference between the original and the reduced system, or generally speaking, between two dynamical systems. To do so, several different spaces and their norms, both in the time and in the frequency domain, need to be considered.

Definition 2.3.14 ([5, Section 5.1.2])**.** Let $f : \mathcal{I} \to \mathbb{R}^n$, with $\mathcal{I} \in \{\mathbb{R}, \mathbb{R}_-, \mathbb{R}_+, [a, b]\}$ be a vector valued function. The *Lebesgue space* $\mathcal{L}_2^n(\mathcal{I})$ is defined as:

$$\mathcal{L}_2^n(\mathcal{I}) = \left\{ f : \mathcal{I} \to \mathbb{R}^n : \left(\int_{t \in \mathcal{I}} ||f(t)||_2^2 \right)^{\frac{1}{2}} < \infty \right\}. \tag{2.23}$$

In our models, input and output will be considered as functions in these spaces: $u(t) \in \mathcal{L}_2^m(\mathcal{I})$ and $y(t) \in \mathcal{L}_2^p(\mathcal{I})$ with $t \in \mathcal{I}$ (cf. the definition of a linear system (2.18)). Usually, one is interested in a relationship between input and output. As such a relationship in the time domain is described by a convolution which is often difficult to calculate, the relation is often examined in the frequency domain. There, it can easily be determined by a product of matrices, as we will see in this section. For the transformation from time to frequency domain the Laplace transformation is used.

Definition 2.3.15 ([18, Section 15.2])**.** The *Laplace transform* of a function $f : \mathbb{R}^+ \to \mathbb{R}$ is defined as

$$F(s) = \mathcal{L}\{f(t)\}(s) = \int_0^\infty f(t) e^{-st} \mathrm{d}t, \tag{2.24}$$

with

$$\mathcal{L}\{f'(t)\}(s) = sF(s) - f(0). \tag{2.25}$$

For a vector, the Laplace transform has to be seen element wise. We transform the linear system (assuming $x(0) = x_0 = 0$):

$$\begin{aligned} & \mathcal{L}\{E\dot{x}(t)\}(s) = \mathcal{L}\{Ax(t) + Bu(t)\}(s) \\ \Rightarrow \quad & E\mathcal{L}\{\dot{x}(t)\}(s) = A\mathcal{L}\{x(t)\}(s) + B\mathcal{L}\{u(t)\}(s) \\ \Rightarrow \quad & sEX(s) = AX(s) + BU(s) \\ \Rightarrow \quad & X(s) = (sE - A)^{-1} BU(s), \end{aligned}$$

and $Y(s) = CX(s)$. This leads to the following connection between the input and the output:

$$Y(s) = C(sE - A)^{-1} BU(s).$$

Definition 2.3.16. The *transfer function* $H : \mathbb{C} \to \mathbb{C}^{p \times m}$ of the linear system (2.18) is defined as

$$H(s) := C(sE - A)^{-1} B. \tag{2.26}$$

Functions in frequency domain will often be interpreted as functions of a complex variable. A detailed description of frequency domain spaces for linear systems can be found in [5]. Here we use Hardy spaces $\mathscr{H}_2$ and $\mathscr{H}_\infty$. The following system norms can then be established using the transfer function $H(s)$ and the corresponding Hardy space norms:

Definition 2.3.17 ([5, Section 5.1.3])**.** The $\mathscr{H}_2$ *norm* of a stable system is defined as

$$||\Sigma_{\text{lin}}||_{\mathscr{H}_2} := \left(\int_{-\infty}^{\infty} \text{tr}(H^*(-iy)H(iy))dy \right)^{\frac{1}{2}}. \tag{2.27}$$

The $\mathscr{H}_\infty$ norm of a stable system is defined as

$$||\Sigma_{\text{lin}}||_{\mathscr{H}_\infty} := \sup_{y\in\mathbb{R}} \left(\sigma_{\max}(H(iy))\right), \tag{2.28}$$

with maximal singular value $\sigma_{\max}$.

Proposition 2.3.18 ([5])**.** *It holds:*

$$||\Sigma_{\text{lin}}||_{\mathscr{H}_2} = \sqrt{\text{tr}(B^*QB)} = \sqrt{\text{tr}(CPC^*)}, \tag{2.29}$$

for the systems Gramians as defined in (2.21) *and* (2.19).

2.3.2. Bilinear Systems. The second class of dynamical systems which will be considered in this thesis are bilinear systems. An overview and examples can be found in [49].

Definition 2.3.19. A *bilinear system* of order n is a system of differential equations of the following form:

$$\Sigma_{\text{bil}} : \begin{cases} E\dot{x}(t) = Ax(t) + \sum_{k=1}^{m} N_k u_k(t)x(t) + Bu(t), \\ y(t) = Cx(t), \quad x(0) = x_0, \end{cases} \tag{2.30}$$

where $E, A, N_k \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$. The input $u(t) \in \mathbb{R}^m$ can be time-dependent just as the states $x(t) \in \mathbb{R}^n$ and the output $y(t) \in \mathbb{R}^p$ are. The value of $x(0) = x_0$ is called initial value.

In this section, only systems with $E \neq I_n$, E nonsingular, will be considered.

2.3.2.1. *Volterra series representation.* A connection between the systems input and output can be established by using the following Volterra series representation for the states of bilinear systems established by Mohler [49]. We will consider systems with E nonsingular.

$$x(t) = \sum_{i=1}^{\infty} \int_0^{\infty} \cdots \int_0^{\infty} \sum_{k_1,k_2,\ldots,k_i=1}^{m} e^{E^{-1}A(\tau_1)} E^{-1} N_{k_1} \cdot \\ \cdot e^{E^{-1}A(\tau_2-\tau_1)} E^{-1} N_{k_2} e^{E^{-1}A(\tau_3-\tau_2)} \cdots E^{-1} N_{k_{i-1}} e^{E^{-1}A(\tau_i-\tau_{i-1})} E^{-1} b_{k_i} \cdot \\ \cdot u_{k_1}(t-\tau_1) \cdots u_{k_i}(t-\tau_i) d\tau_1 \ldots d\tau_i. \tag{2.31}$$

The input-output relationship of the system can then be defined as:

$$y(t) = \sum_{i=1}^{\infty} \int_0^{\infty} \cdots \int_0^{\infty} \sum_{k_1,k_2,\ldots,k_i=1}^{m} Ce^{E^{-1}A(\tau_1)} E^{-1} N_{k_1} \cdot \\ \cdot e^{E^{-1}A(\tau_2-\tau_1)} E^{-1} N_{k_2} e^{E^{-1}A(\tau_3-\tau_2)} \cdots E^{-1} N_{k_{i-1}} e^{E^{-1}A(\tau_i-\tau_{i-1})} E^{-1} b_{k_i} \\ \cdot u_{k_1}(t-\tau_1) \cdots u_{k_i}(t-\tau_i) d\tau_1 \ldots d\tau_i, \cdot \tag{2.32}$$

with columns b_{k_i} of B and Volterra kernels defined as:

$$h_i^{(k_1,\ldots,k_i)}(\tau_1,\ldots,\tau_i) = Ce^{E^{-1}A\tau_1} E^{-1} N_{k_1} e^{E^{-1}A(\tau_2-\tau_1)} \cdot \ldots \\ \ldots \cdot E^{-1} N_{k_{i-1}} e^{E^{-1}A(\tau_i-\tau_{i-1})} E^{-1} b_{k_i}, \tag{2.33}$$

where $i = 1, 2, \ldots,\ k_i = 1, \ldots, m$, and $\tau_{i+1} \geq \tau_i \geq 0$. The input-output relation can now be written as:

$$y(t) = \sum_{i=1}^{\infty} \int_0^{\infty} \cdots \int_0^{\infty} \sum_{k_1,k_2,\ldots,k_i=1}^{m} h_i^{(k_1,\ldots,k_i)}(\tau_1,\ldots,\tau_i) \\ \cdot \left(\prod_{j=1}^{i} u_{k_j}(t-\tau_j) \right) d\tau_1 \ldots d\tau_i. \tag{2.34}$$

In practice, the Volterra kernels $h_i^{(k_1,\ldots,k_i)}(\tau_1,\ldots,\tau_i)$ need to be examined in the frequency domain as well. Therefore we need a multivariate Laplace transform:

Definition 2.3.20 ([24]). Given a function $f(t_1, \ldots, t_n)$ defined on $\mathbb{R}^n$ define its *Laplace transform* $F(s_1, \ldots, s_n)$ by:

$$F(s_1, \ldots, s_n) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(t_1, \ldots, t_n) \exp\left(-\sum_{k=1}^{n} t_k s_k\right) dt_1 \ldots dt_n. \tag{2.35}$$

We can now transform the Volterra kernels.

Definition 2.3.21. The *i-th order transfer function of the Volterra kernel*

$$h_i^{(k_1,\ldots,k_i)}(\tau_1, \ldots, \tau_i) = C e^{E^{-1}A\tau_1} E^{-1} N_{k_1} e^{E^{-1}A(\tau_2-\tau_1)} \ldots E^{-1} N_{k_{i-1}} e^{E^{-1}A(\tau_i-\tau_{i-1})} E^{-1} b_{k_i},$$

is defined as

$$H_i^{(k_1,\ldots,k_i)}(s_1, \ldots, s_i) = C(s_i E - A)^{-1} N_{k_1} (s_{i-1} E - A)^{-1} \ldots N_{k_{i-1}} (s_1 E - A)^{-1} b_{k_i}. \tag{2.36}$$

By taking $\overline{N} = [N_1 \ \ldots \ N_m]$, this definition can be rewritten simultaneously for all N_k by using Kronecker products:

$$\begin{aligned} H_i(s_1, \ldots, s_i) =& C(s_i E - A)^{-1} \overline{N} [I_m \otimes (s_{i-1} E - A)^{-1}](I_m \otimes \overline{N}) \ldots \\ & \cdot [\underbrace{I_m \otimes \cdots \otimes I_m}_{i-2 \text{ times}} \otimes (s_2 E - A)^{-1})](\underbrace{I_m \otimes \cdots \otimes I_m}_{i-2 \text{ times}} \otimes \overline{N}) \\ & \cdot [\underbrace{I_m \otimes \cdots \otimes I_m}_{i-1 \text{ times}} \otimes (s_1 E - A)^{-1})](\underbrace{I_m \otimes \cdots \otimes I_m}_{i-1 \text{ times}} \otimes B). \end{aligned} \tag{2.37}$$

In addition, Bruni et al. [19] examined the convergence of the Volterra series and established the following result:

Proposition 2.3.22. *If the Volterra series in* (2.31) *converges, then it uniformly converges to the solution of the bilinear system* (2.30). *For bounded inputs the Volterra series* (2.31) *converges on any finite time interval* $[0, t_e]$.

The convergence of the Volterra series is connected to the stability of the system.

2.3.2.2. *Stability.* The notion of stability for bilinear systems differs from that for linear systems. For bounded inputs, the following definition of stability applies:

Definition 2.3.23 ([72, 59])**.** The bilinear system (2.30) is called *bounded-input-bounded-output (BIBO) stable*, if for any bounded input, the output is bounded on $[0, \infty)$. An input/output is called *bounded* if it satisfies the following condition: $||u||_\infty = max_j sup_{t\in[0,\infty)}|u_j(t)| < M$.

Siu and Schetzen [59] combined convergence of the Volterra series with BIBO stability. They showed the following sufficient condition for BIBO stability.

Theorem 2.3.24 ([59])**.** *Let a bilinear system (2.30) with nonsingular E be given, and let the pencil $A - \lambda E$ be stable, i.e. there exist real scalars $\beta, \alpha \in \mathbb{R}$ with $\beta > 0$ and $0 < \alpha \leq -\max_i(\mathrm{Re}(\lambda_i((A, E))))$ such that*

$$||e^{E^{-1}At}||_2 \leq \beta e^{-\alpha t},\ t \geq 0. \tag{2.38}$$

Assume $||u(t)|| = \sqrt{\sum_{k=1}^{m}|u_k(t)|^2} \leq M$ uniformely on $[0, \infty)$ with $M > 0$ and denote $\Gamma = \sum_{k=1}^{m}||E^{-1}N_k||_2$. Then the system is BIBO stable if $\Gamma < \frac{\alpha}{M\beta}$.

The bilinear system is hence stable if the matrices N_k are sufficiently bounded.

2.3.2.3. *Reachability, observability and balanced representation.* As for linear systems, the concepts of reachability, observability and balanced representation exist for bilinear systems. However, the concepts need to be generalized, which will be done in the following section.

Definition 2.3.25 ([25, 56])**.** A state $x(t_e)$ of a bilinear system (2.30) is *reachable* (from the origin $x(0) = 0$) if there exists an admissible input function that maps the origin of the state space into the state $x(t_e)$ in a finite interval of time $[0, t_e]$.

Definition 2.3.26 ([56])**.** A bilinear system (2.30) is called *(span) reachable* if the space of all reachable states $\mathcal{X}^{\text{reach}}$ spans $\mathbb{R}^n$.

For a bilinear system (2.30) with $E \neq I$ nonsingular, the following statements for reachability can be derived. Let

$$P_1(t_1) = e^{E^{-1}At_1}E^{-1}B,$$

$$P_i(t_1, \ldots, t_i) = e^{E^{-1}At_i}E^{-1}[N_1P_{i-1} \quad N_2P_{i-1} \ldots N_mP_{i-1}], \quad i = 2, 3, \ldots$$

Definition 2.3.27 ([72]). If it exists, the *reachability Gramian* is defined as

$$P = \sum_{i=1}^{\infty} \int_0^{\infty} \cdots \int_0^{\infty} P_i P_i^* \, dt_1 \ldots dt_i. \tag{2.39}$$

Zhang and Lam [72] established the following theorem for the existence of the reachability Gramian:

Theorem 2.3.28 ([72]). *The reachability Gramian* (2.39) *exists, if*

(i) *the pencil* $A - \lambda E$ *is stable, with*

$$||e^{E^{-1}At}||_2 \leq \beta e^{-\alpha t}, \, t \geq 0, \tag{2.40}$$

where $\beta > 0$ *and* $0 < \alpha \leq -\max_i(\mathrm{Re}(\lambda_i(A, E))$, $\beta, \alpha \in \mathbb{R}$.

(ii) $\Gamma_1 < \frac{\sqrt{2\alpha}}{\beta}$, *with* $\Gamma_1^2 = ||\sum_{k=1}^{m} E^{-1} N_k N_k^T E^{-T}||_2$.

The connection of P to the bilinear Lyapuonv equations and the reachability of the system can now be established:

Theorem 2.3.29 ([72]). *Suppose* $A - \lambda E$ *is stable, and the reachability Gramian* P *exists. Then*

(i) P *satisfies the following bilinear Lyapunov equation:*

$$AXE^T + EXA^T + \sum_{k=1}^{m} N_k X N_k^T + BB^T = 0. \tag{2.41}$$

(ii) *The bilinear system* (2.30) *is reachable if and only if* P *is positive definite.*

Proposition 2.3.30 ([72]). *If* (2.41) *has a unique solution, then the solution* P *is symmetric.*

For linear stable systems, it is known that if the Lyapuonv equation has a unique solution it is the reachability (controllability) Gramian. For bilinear systems, however, it is possible that a unique solution to the Lyapunov equation is not the reachability Gramian. Consider for example the following bilinear system (cf. [72]):

$$\dot{x} \quad = \quad -x + 2xu + u.$$

This leads to the solution of the Lyapunov equation $p = -\frac{1}{2}$. But the integrals $\tilde{p}_i = \int p_i p_i^T$ lead to $\tilde{p}_i = 2^{i-2}$, which gives $p = \sum_{i=1}^{\infty} 2^{i-2}$ which does not converge — hence the reachability Gramian does not exist.

This behavior is summarized in the following theorem:

Theorem 2.3.31 ([72]). *Suppose $A - \lambda E$ is stable.*

- (2.41) *has a positive (semi) definite solution X if and only if the reachability Gramian* (2.39) *exists and converges to a positive semidefinite matrix $\hat{X}$ satisfying* (2.41).
- *If* (2.41) *has a unique positive (semi) definite solution X, then* (2.39) *converges to X and therefore X is the reachability Gramian.*

For a bilinear system (2.30) with E nonsingular the following statements for observability can be derived. Let

$$Q_1(t_1) = Ce^{E^{-1}At_1},$$

$$Q_i(t_1, \ldots, t_i) = [Q_{i-1}E^{-1}N_1 \quad Q_{i-1}E^{-1}N_2 \ldots Q_{i-1}E^{-1}N_m]^T e^{E^{-1}At_i}, i = 2, 3, \ldots$$

Definition 2.3.32 ([72]). If it exists, the *observability Gramian* is defined as

$$Q = \sum_{i=1}^{\infty} \int_0^{\infty} \cdots \int_0^{\infty} Q_i^* Q_i dt_1 \ldots dt_i. \tag{2.42}$$

Zhang and Lam [72] established the following theorem for the existence of the observability matrix:

Theorem 2.3.33 ([72]). *The observability matrix* (2.42) *exists, if*

(i) *the pencil $A - \lambda E$ is stable, with*

$$||e^{E^{-1}At}||_2 \leq \beta e^{-\alpha t}, t \geq 0, \tag{2.43}$$

where $\beta > 0$ and $0 < \alpha \leq -\max_i(\mathrm{Re}(\lambda_i(A, E))$, $\beta, \alpha \in \mathbb{R}$.

(ii) $\Gamma_1 < \frac{\sqrt{2\alpha}}{\beta}$, *with* $\Gamma_1^2 = ||\sum_{k=1}^m E^{-1}N_k N_k^T E^{-T}||_2$.

Theorem 2.3.34. *Suppose $A - \lambda E$ is stable, and the observability Gramian exists. Then*

(i) *$E^{-T}QE^{-1}$ satisfies the following bilinear Lyapunov equation:*

$$A^T Y E + E^T Y A + \sum_{k=1}^{m} N_k^T Y N_k + C^T C = 0. \tag{2.44}$$

(ii) *The bilinear system* (2.30) *is observable if and only if Q is positive definite.*

Theorem 2.3.35 ([72]). *Suppose $A - \lambda E$ is stable.*

- (2.44) *has a positive (semi) definite solution Y if and only if the observability Gramian* (2.42) *exists and converges to a positive semidefinite matrix $\hat{Q}$ satisfying* (2.44) *for $E^{-T}\hat{Q}E^{-1}$.*
- *If* (2.44) *has a unique positive (semi) definite solution Y, then* (2.42) *converges to $Q = E^T Y E$ and Q is the reachability Gramian.*

A balanced representation of a bilinear system can be obtained in the same way as in the linear case. Assume the bilinear system is BIBO stable, and the Gramians P and Q exist and are positive definite. They can be decomposed as

$$P = RR^T \text{ and } Q = L^T L.$$

By using the singular value decompositon of

$$LER = U_b \Sigma V_b^T,$$

one obtains

$$W_b^T E T_b,\ W_b^T A T_b,\ W_b^T N_k T_b,\ W_b^T B,\ C T_b,$$

where

$$W_b = L^T U_b \Sigma^{-1/2}, \quad T_b = R V_b \Sigma^{-1/2}, \quad W_b^{-1} = T_b^T E^T, \quad T_b^{-1} = W_b^T E.$$

Details can be found in [42, 2] and the references therein.

2.3.2.4. *$\mathscr{H}_2$-norm of a bilinear system.*

Definition 2.3.36. The *$\mathscr{H}_2$-norm* of a bilinear system is defined as

$$||\Sigma_{\text{bil}}||^2_{\mathscr{H}_2} = \text{tr}\left(\sum_{i=1}^{\infty}\int_0^\infty \cdots \int_0^\infty \sum_{k_1,k_2,\ldots,k_i=1}^{m} h_i^{(k_1,\ldots,k_i)}(s_1,\ldots,s_i)\cdot \right.$$
$$\left. \cdot (h_i^{(k_1,\ldots,k_i)}(s_1,\ldots,s_i))^T ds_1 \ldots ds_i\right), \tag{2.45}$$

with Volterra kernels $h_i^{(k_1,\ldots,k_i)}(s_1,\ldots,s_i)$ defined in (2.33).

Zhang and Lam [72] showed, that the bilinear $\mathscr{H}_2$-norm satisfies the same property as the linear norm:

Theorem 2.3.37. *For a bilinear system (2.30) if $A - \lambda E$ is stable and the reachability Gramian P (or the observability Gramian Q) exists, then its $\mathscr{H}_2$-norm can be computed from*

$$||\Sigma_{\text{bil}}||_{\mathscr{H}_2} = \sqrt{\text{tr}(CPC^T)} \text{ (or } = \sqrt{\text{tr}(B^TQB)}), \tag{2.46}$$

where P (or $E^{-T}QE^{-1}$) satisfies (2.41) (or (2.44)).

Benner and Breiten [12] showed that the bilinear $\mathscr{H}_2$-norm can equivalently be written as:

Theorem 2.3.38 ([12])**.** *Let Σ_{bil} be a stable bilinear system. Then it holds that*

$$||\Sigma_{\text{bil}}||^2_{\mathscr{H}_2} = \text{vec}(I_p)^T(C \otimes C)\cdot \cdot \left(-A \otimes E - E \otimes A - \sum_{k=1}^{m} N_k \otimes N_k\right)^{-1} (B \otimes B)\text{vec}(I_m). \tag{2.47}$$

CHAPTER 3

Modeling of heat transfer problems

The design of a new product is a complex process with many experts involved. From the idea to the final concept, a close cooperation between design engineers, simulation experts, test engineers and manufacturing specialists is required. After setting up a first design, this design is examined by a team of simulation experts. Depending on the requirements, different analyses need to be conducted. Several physical aspects need to be taken into account, like mechanical deformations, fluid flows, electromagnetic effects and thermal analyses. Depending on the evaluation of the simulation results, the design will be improved. A prototype of the optimized product is then fabricated and thoroughly tested in a series of experiments. Until arriving at the final product, all new designs will be simulated — hence simulation plays a major role. In the final stage of the product development, simulation and experiment should coincide. The main part is now designing the manufacturing process, which also might involve changes in the design, which again need to be examined by simulation and experiment. Finally, the

new component is carefully designed, can be manufactured and the production can start!
As explained above, simulation is an important part of the product design process. Having the ability of simulating different designs instead of building them can save a lot of time and money. It is desirable to obtain models of the product that lead to accurate results. The more complex the models get the longer the simulations take. This — in turn — shows the need for small and accurate models, which can, for example, be obtained by Model Order Reduction (cf. Chapter 5).
This work focuses on the thermal modeling of electrical motors. The underlying physical effects, the mode of operation of an electrical motor and the model parametrization and creation will be the key aspects of this chapter.

3.1. Thermal Modeling

For a thermal analysis, several physical effects have to be considered and can be modeled based on the three main types of heat transfer: heat conductance, convection and radiation. For a broad overview of heat and mass transfer see for example the book of Baehr and Stephan [7].

3.1.1. Heat Conductance. Temperature gradients lead to energy transfer by heat conductance. The heat flux $\dot{q}(x,t)$ (in $\frac{W}{m^2}$ at time t and location x) describes the energy transfer in a conductive material. The heat flux quantifies the amount of heat which flows through a certain area. Fourier's law states the proportionality between heat flux and the temperature gradient:

$$\dot{q} = -k \cdot \mathrm{grad}(T). \tag{3.1}$$

The constant k is called thermal conductivity. Strictly speaking, it depends on temperature, but in many applications it is well approximated by a constant. Thermal conductivities are known for many materials: Metals usually have high thermal conductivities ($10\frac{W}{mK} - 10^3\frac{W}{mK}$), while the thermal conductivities of fibres and foams are small ($10^{-2}\frac{W}{mK} - 1\frac{W}{mK}$). They can therefore be used as insulators.
For two solids in contact, the heat leaving one body has to be absorbed by the other. For the heat flux, this leads to the following equation on the interface:

$$\left(k_1 \frac{\partial T_1}{\partial \mathbf{n}}\right)_I = \left(k_2 \frac{\partial T_2}{\partial \mathbf{n}}\right)_I,$$

where T_1 and T_2 are the temperature on the first and second solid, and $\frac{\partial}{\partial \mathbf{n}}$ is the derivative in normal direction. If the two materials are closely attached to each other, the temperature on the interface is the same:

$$(T_1)_I = (T_2)_I.$$

In some situations, the two surfaces are not directly connected, but separated by a small gap. This gap is filled with air or an insulation material and leads to a low thermal conductance. This thermal resistance can be modeled on the interface by a thermal contact conductance coefficient (or contact heat transfer coefficient) h_c leading to the following equation for the flux:

$$\left(k_1 \frac{\partial T_1}{\partial \mathbf{n}}\right)_I = h_c[(T_1)_I - (T_2)_I].$$

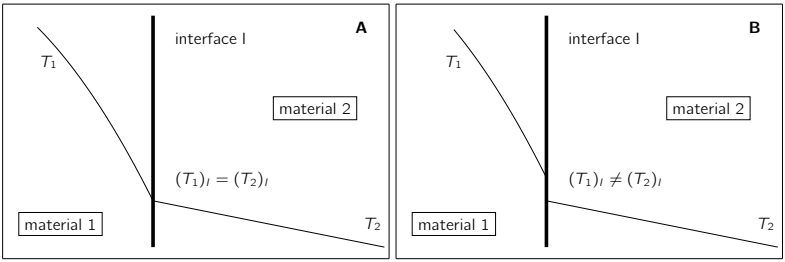


Figure 3.1. Temperature on the interface between two solids in contact with each other. **A**: no contact resistance, **B**: contact resistance

3.1.2. Convective Heat Transfer. In a fluid, heat is not only transferred by conduction, but also by the movement of the molecules within the fluid. These two effects are summarized as convective heat transfer, which is often referred to as convection. A special case is the heat transfer between a fluid and a solid. The characteristics of the fluid layer close to the solid have the greatest effect on the heat transfer between the two materials. Hence, the velocity and the temperature within this layer have to be modeled and analyzed, which is not a trivial task. The modeling of heat transfer in combined solid and fluid systems is often called conjugate

heat transfer modeling. For large and complex models, a conjugate heat transfer analysis can be too time consuming, because a fine discretisation of the boundary layer is required. Hence, a heat transfer coefficient h is introduced, which describes the heat transfer between fluid and solid. It allows an analysis of the heat transfer without explicit treatment of the fluid. The heat flux on the boundary between fluid and solid is then modeled by the following equation:

$$\dot{q}_I = h(T_{\text{solid}} - T_{\text{fluid}}).$$

The values of the heat transfer coefficients h can be determined by measurements or simulations of the fluid flow. Different fluids (air, water) and different types of convection result in different values for the heat transfer coefficients. Forced convection occurs whenever the fluid is forced to flow in a certain direction in contrast to free (or natural) convection. For free convection between air and a solid, the values of the heat transfer coefficients range from $5\frac{W}{m^2K} - 25\frac{W}{m^2K}$, while for forced convection in hot air they range from $20\frac{W}{m^2K}$ to $300\frac{W}{m^2K}$. The highest heat transfer coefficients can be measured in boiling water or condensating vapor, with values up to $10^5\frac{W}{m^2K} - 10^6\frac{W}{m^2K}$.

3.1.3. Radiation. Every material emits energy to its environment by electromagnetic waves. This type of energy transfer is called thermal radiation or heat radiation. The internal energy of a body is converted into electromagnetic waves and transmitted to its surroundings. Similarly, a body simultaneously absorbs energy in the form of radiation and transforms it to internal energy. If a heat flux by radiation is modeled, it is done by the following equation:

$$\dot{q} = \epsilon\sigma(T^4 - T_s^4),$$

where σ is the Stefan-Boltzmann constant ($5.67 \cdot 10^{-8}\frac{W}{m^2K^4}$), and ϵ is the emissivity — the ability of a body to emit radiation. Strictly speaking, this material property is dependent on the temperature and the condition of the body's surface. Typical values are 0.90 for wood at $293K$ or 0.049 for aluminum at $443K$. As the temperatures T of the material and T_s of the surroundings are raised to the power of four, the effect of the radiation is large at high temperatures.

3.2. The heat equation

The law of energy conservation for thermal systems can be stated in terms of the first law of thermodynamics [28, 62]: The change in internal energy of a closed system is the sum of the heat supplied and the work added to the system.
In this section, $\dot{G}$ will denote the amount of the quantity G supplied to the system during a time $\mathrm{d}t$. First, the expression for the heat supplied to the system is derived. The governing equation for the heat flux into a surface element $\mathrm{d}A$, caused by the heat $\dot{Q}(x,t)$ (at time t and location x) is the following [7]:

$$\frac{\mathrm{d}\dot{Q}(x,t)}{\mathrm{d}A} = -\dot{\mathrm{q}}(\mathrm{x},\mathrm{t}) \cdot \mathbf{n}. \tag{3.2}$$

Integration over the surface and using the Gauss theorem leads to the following equation for the heat:

$$\dot{Q}(x,t) = -\int_{(A)} \dot{\mathrm{q}}(\mathrm{x},\mathrm{t}) \cdot \mathbf{n}\mathrm{d}A = -\int_{(V)} \mathrm{div}(\dot{\mathrm{q}}(\mathrm{x},\mathrm{t}))\mathrm{d}V. \tag{3.3}$$

The work added to the system can be described by a time dependent power density $S(x,t)$ per volume area (measured in $\frac{W}{m^3}$). Integration leads to the following expression for the work [7]:

$$\dot{W} = \int_{(V)} S(x,t)\mathrm{d}V. \tag{3.4}$$

The change in internal energy $U(x,t)$ can be stated using the specific heat capacity C. It specifies the heat that must be supplied to increase the temperature by $\mathrm{d}T$. The change in internal energy for this temperature change can then be calculated from the heat capacity and the mass of the body [7]:

$$\mathrm{d}U(x,t) = mC\mathrm{d}T(x,t) = \int_{(V)} \rho\mathrm{d}V \cdot C\mathrm{d}T(x,t). \tag{3.5}$$

As heat conduction in a solid body is considered, the changes in volume and density due to temperature and pressure changes are small and can be neglected, leading to:

$$\frac{\mathrm{d}U(x,t)}{\mathrm{d}t} = \int_{(V)} \rho C \frac{\partial T(x,t)}{\partial t}\mathrm{d}V. \tag{3.6}$$

Using the law of energy conservation for thermal systems, the equations (3.3),(3.6) and (3.4) result in:

$$\int_{(V)} \left(\rho C \frac{\partial T(x,t)}{\partial t} + \mathrm{div}(\dot{q}(x,t)) - S(x,t) \right) dV = 0. \tag{3.7}$$

This integral is equal to zero for any chosen region only when the integrand is zero. Therefore the following equation can be derived:

$$\rho C \frac{\partial T(x,t)}{\partial t} = -\mathrm{div}(\dot{q}(x,t)) + S(x,t). \tag{3.8}$$

Using Fourier's law (3.1) the so called heat equation is obtained:

$$\rho C \frac{\partial T(x,t)}{\partial t} = k \Delta T(x,t) + S(x,t). \tag{3.9}$$

3.3. Boundary and Interface conditions

To determine the thermal behavior of a component, the temperature field $T(x,t)$ (dependent on location x and time t) has to be examined. The temperature field $T(x,t)$ within a domain $\Omega \subset \mathbb{R}^3$ for times $t \in [0, t_{end}]$ can be calculated using the heat equation (3.9) with constant material properties ρ, C, k and a heat source S. The derivation of the heat equation can be found in Section 3.2.
On interfaces and outer surfaces, now called boundaries and denoted as $\Gamma \subset \mathbb{R}^2$, different conditions have to be specified, depending on the situation of interest. They are mathematically formulated as follows:

- Dirichlet boundary conditions:

  $$T(x,t) = T_D(t) \text{ on the boundary } \Gamma_D.$$

  These conditions correspond to fixed temperatures on surfaces.
- Neumann boundary conditions:

  $$-k \frac{\partial T(x,t)}{\partial \mathbf{n}} = \dot{q}_N \text{ in } \Gamma_N,$$

  where $\dot{q}_N$ is a given heat flux on the boundary.
- Robin boundary conditions:

  $$-k \frac{\partial T(x,t)}{\partial \mathbf{n}} = h(T - T_\infty) \text{ in } \Gamma_R,$$

  where h denotes the heat transfer coefficient defined in Section 3.1.2. T_∞ is the temperature of the surrounding domain.

- Interface conditions: A thermal resistance between two surfaces can be modeled on the interface by a thermal contact conductance coefficient, as shown in Section 3.1.1. The interface I will be considered as two surfaces: I_1 with temperature T_1 and I_2 with temperature T_2. The following equation applies:

$$k_2 \frac{\partial T|_{I_1}(x,t)}{\partial \mathbf{n}} = -k_1 \frac{\partial T|_{I_2}(x,t)}{\partial \mathbf{n}} = h_c \left(T(x,t)\,|_{I_1} - T(x,t)|_{I_2} \right).$$

3.4. Mode of operation of an electrical motor

An electrical motor converts electrical energy into mechanical work, which is produced by the interaction of an electrical current and a magnetic field. One part of the motor — the so called stator — consists of several coils wound around an iron core. When a voltage is applied, a current is induced in the coil. Inside the counterpart — the so called rotor — a magnetic field is generated either by a permanent magnet or by an electromagnet. The interaction of this magnetic field with the current in the stator results in a rotation of the rotor.
Actuating the motor with electrical currents leads to an increase in temperature in its different components due to thermal losses. It is important to analyze the influence of this temperature change on the materials of the motor, as it affects its life-span. This is done by carrying out a thermal analysis.

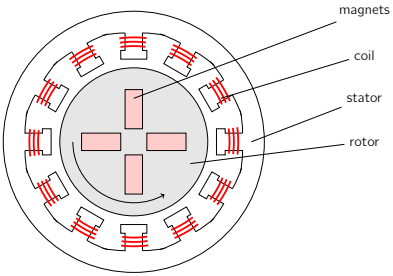
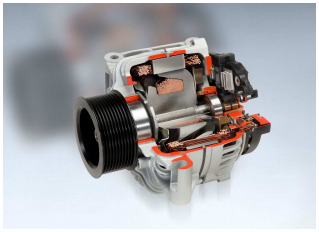


Figure 3.2. Drawing of a slice through an electrical motor.

(a) Drive unit and generator in one: the Bosch integrated motor generator.



(b) Generator for commercial vehicles. The configuration of the coils is the same as for an electrical motor.

Figure 3.3. Two components manufactured by Robert Bosch GmbH illustrating the structure of an electrical motor. Photos by courtesy of Robert Bosch GmbH.

3.5. Thermal modeling of an electrical motor

The main heat source in the electrical motor are thermal losses, resulting from the current in the coil of the stator and/or rotor. The motor has to fulfill various operational requirements and therefore different current profiles have to be considered. The temperature on certain parts of the motor (for example the flange) should not exceed a specified upper limit because these parts are in contact with other temperature sensitive components. This upper limit is built into the model as a fixed temperature (Dirichlet boundary condition, cf. Section 3.3).
The motor is surrounded by air, therefore convection has to be considered. The motor needs to work in a large temperature range (arctic winter, tropical summer), therefore different ambient temperatures are examined in the model. Varying the heat transfer coefficients represent different cooling strategies or different interaction scenarios of the motor with its environment (Robin boundary condition, cf. Section 3.3).
Various parts of the motor are not directly attached to each other and the resulting thermal resistance has to be modeled by a contact heat transfer coefficient. Varying this parameter, the small gap between the two materials can be considered as filled with air or an insulation material (Interface

condition, cf. Section 3.3). The motor is built from various materials such as steel, copper and plastics. These materials have different properties, among others the density ρ, the specific heat C and the thermal conductivity k. Here, these material parameters will not be varied. As the motor temperature remains relatively small, the effect of radiation is not of great importance, and will therefore be neglected.
The thermal analyses within this work have been conducted using Comsol Multiphysics®, version 3.5a. This software provides the user with an environment for the modeling of dynamical systems. In our case, the heat equation (3.9) on the electrical motor model is solved, using the boundary conditions and interface conditions as explained above.
Different motor models have been examined. First, only one coil and parts of the stator are considered. The resulting geometry, which is provided with the different boundary and interface conditions as well as heat sources and material properties, can be seen in Figure 3.4.

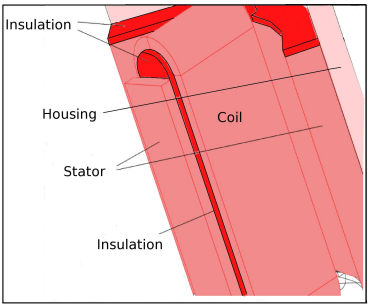


Figure 3.4. The Comsol® model simulates the heat transfer in a stator slice, without the rotor.

Second, a complete motor is modeled. Details for this model are given in the next chapter, as on top of the underlying physics, changes in geometry are incorporated.

CHAPTER 4

Model parametrization

The heat dissipation in a component can be determined by a Comsol® simulation as explained in Chapter 3. Comsol® is based on the Finite Element Method. Having knowledge of the mathematical background allows one to construct parametrized models, which can then be treated by Model Order Reduction. In this Chapter, we focus on the parametrization of thermal models. Two types of parameters will be considered: physical parameters and parameters resulting from variations in geometry. The latter require a detailed analysis of the underlying equations, which will be the main subject of this chapter.

4.1. Discretization of the heat equation

As given in Section 3.3, the temperature field of a component can be determined by solving the heat equation (3.9). This is done by a spatial discretization using the Finite Element Method (cf. for example [16]). To do so, the domain on which the equation is solved is divided into smaller domains, so called elements. On these elements, special basis functions $\psi_j(x)$ will be considered. By using them, incorporating the boundary and interface conditions and the weak formulation of the heat equation, one is

able to discretize the equation.

$$\begin{aligned}
&\int_\Omega \psi_j(x)\rho C\frac{\partial T(x,t)}{\partial t}dx + \int_\Omega \nabla\psi_j(x)\cdot k\nabla T(x,t)dx \\
&+\int_{\Gamma_R} \psi_j(x)hT(x,t)ds \\
&+\int_{I_1} \psi_j(x)h_cT(x,t)ds - \int_{I_2} \psi_j(x)h_cT(x,t)ds \\
&=\int_\Omega \psi_j(x)S(x,t)dx + \int_{\Gamma_N} \psi_j(x)(-\dot{q}_N)ds + \int_{\Gamma_R} \psi_j(x)hT_\infty ds.
\end{aligned} \tag{4.1}$$

The material parameters ρ, C and k are taken as constant. With finite element basis functions $\psi_k(x)$ the temperature is approximated as follows,

$$T(x,t) \approx \sum_{k=1}^{N} \mathrm{T}_k(t)\psi_k(x).$$

By plugging this into equation (4.1), the following discretized equation is obtained:

$$E\dot{T}(t) = (A + hN_1 + h_cN_2)T(t) + B\cdot\begin{bmatrix} S(t) \\ hT_\infty \\ \dot{q}_N \\ T_D \end{bmatrix}, \tag{4.2}$$

where the entries of the matrices are given as:

$$\begin{aligned}
E_{kj} &= \rho C\int_\Omega \psi_k(x)\psi_j(x)dx, \\
A_{kj} &= k\int_\Omega \nabla\psi_k(x)\cdot\nabla\psi_j(x)dx, \\
(N_1)_{kj} &= \int_{\Gamma_R} \psi_k(x)\psi_j(x)dx, \\
(N_2)_{kj} &= \int_{I_1} \psi_k(x)\psi_j(x)dx - \int_{I_2} \psi_k(x)\psi_j(x)dx,
\end{aligned}$$

$$B_{j1} = \int_{\Omega} \psi_j(x)dx,$$
$$B_{j2} = -\int_{\Gamma_N} \psi_j(x)ds,$$
$$B_{j3} = \int_{\Gamma_R} \psi_j(x)ds.$$

The entries of the fourth column B_{j4} are obtained from an elimination of the corresponding Dirichlet boundary nodes after the discretization. As $\int \psi_k(x)\psi_j(x)dx = \int \psi_j(x)\psi_k(x)dx$ and $\int \nabla\psi_k(x)\cdot\nabla\psi_j(x)dx = \int \nabla\psi_j(x)\cdot\nabla\psi_k(x)dx$, the matrices E, A and N_k for the considered class of systems are symmetric and E is in addition positive definite.

4.2. Physical parametrization

In the discretized form of the heat equation (4.2), two types of physical parameters appear: Heat transfer coefficients h resulting from convection (cf. Section 3.1.2) and given as Robin boundary conditions (cf. Section 3.3), and the contact heat transfer coefficients h_c, resulting from heat conduction (cf. Section 3.1.1) on the interface of two model parts (cf. Section 3.3).

4.3. Geometric variations

For a change in geometry, Comsol® 3.5a uses the so called "moving mesh" [51]. The mesh can be deformed, moved and scaled using transformations given by the user, or — in the case where the physical processes transform the model — are calculated by Comsol®. The underlying equations are those of an arbitrary Lagrangian-Eulerian (ALE) framework. It basically transforms the mesh from a reference frame to a material or spatial frame. A more detailed description of this framework can be found in [29] and the references therein. In our special case, we will incorporate scaling functions in order to scale the model, and just scale the mesh, not deform or move it (often called "mesh morphing").

4.3.1. Modeling of scalings in the motor model. The model of an electrical motor requires essentially two different scaling functions. The first one is a simple linear scaling, which is used to scale the model in z-direction. The second one is the nonlinear scaling of an annulus. The inner

radius is kept constant and the outer radius is scaled. It will be used for the scaling of housing and stator. The two different scalings are illustrated in Figures 4.1 and 4.2a, whereas Figure 4.2b gives an idea how a complete scaling of the housing would look like. The scalings can be described via the following functions:

Definition 4.3.1. Let $\Omega = [0, a] \times [0, b] \in \mathbb{R}^2$ and $\mu \geq 0$. A *linear scaling function* to increase the size of the rectangle Ω in x-direction is defined as follows:

$$G_\mu : \Omega \to \Omega_s \subset \mathbb{R}^2, \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} (1+\mu)x \\ y \end{pmatrix}. \tag{4.3}$$

Definition 4.3.2. Let Ω be an annulus with inner radius R. Let $\gamma \geq 1$. The *annulus scaling function* will be defined as follows:

$$F_\gamma : \Omega \to \Omega_s \subset \mathbb{R}^2, \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \left[\gamma + (1-\gamma)\frac{R}{\sqrt{x^2+y^2}}\right] \begin{pmatrix} x \\ y \end{pmatrix}. \tag{4.4}$$

These scaling functions need to be inserted in the Comsol® model to scale the modeled motor parts.

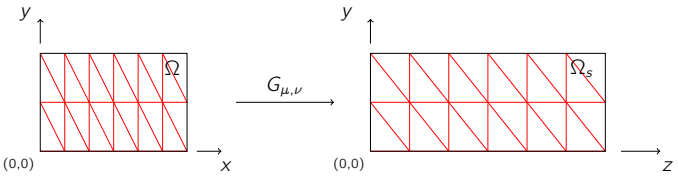
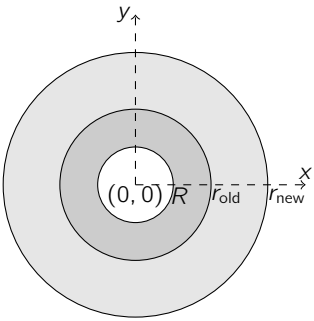
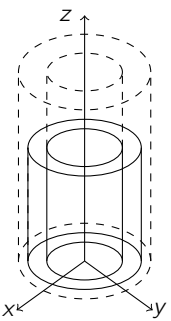


Figure 4.1. Simple linear scaling of a rectangle.

The variation of the height of the stator, rotor and housing will be modeled by a linear scaling defined by a linear function G_θ. The flange will also change the height, it is modeled by a function G_μ. The stator and housing will be scaled using nonlinear functions F_γ and F_η, respectively. This is shown in Figure 4.3. For the modeling of geometric variations, the stator will in addition be simplified as a hollow cylinder. In contrast to our first model (cf. Figure 3.4) the coils will be modeled as cuboids within the stator. This can be seen in Figure 4.4, a top view of the Comsol model.



(a) Scaling of an annulus R - r_{old} to the annulus R - r_{new}



(b) Scaling of the housing both in z and x, y-direction

Figure 4.2. Two scalings needed for the geometry variation of an electrical motor

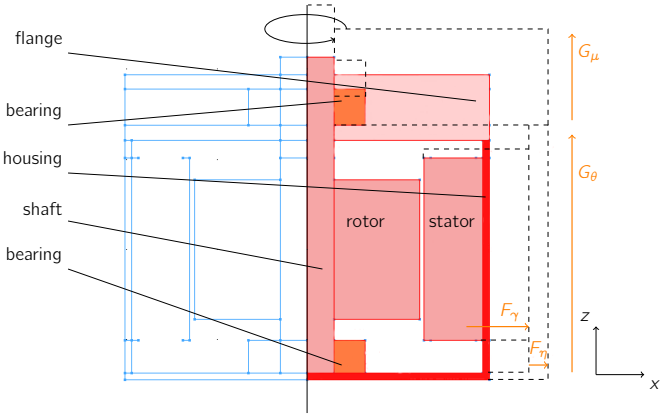


Figure 4.3. Rotationally symmetric slice through the complex Comsol motor model showing the different scaling functions

4.3.2. Parametrized system formulation. Using the weak formulation of the heat equation as given in equation (4.1), it is possible to obtain a parametrized model depending on the different scalings. Moosmann [50] showed in his thesis, that scalings can be incorporated in the model by transforming the basis functions from an unscaled to a scaled element and additionally use substitution in the integrals. We will basically use this approach for the scaling of our models. First, we state that for the defined linear scalings (4.3) it holds:

$$\psi = \psi^s \circ G_\mu. \tag{4.5}$$

However, for the nonlinear scaling F_γ given in equation (4.4) this is not true anymore. To overcome this difficulty, we will need to consider only the scaling of the finite element mesh. In our Comsol® model, we use triangular mesh elements in the scaled annuli. Hence we need to scale triangles as illustrated in Figure 4.5.

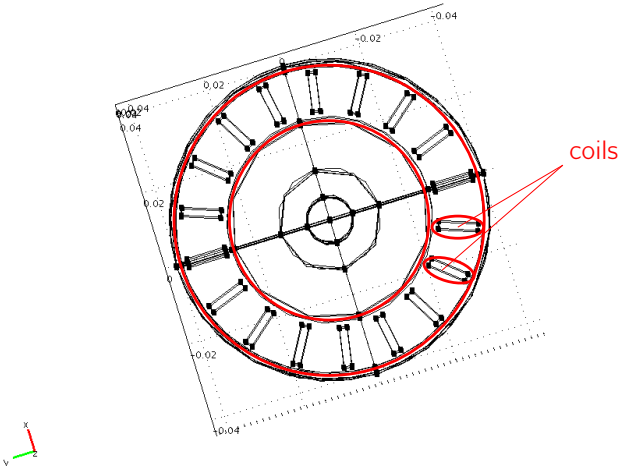


Figure 4.4. Model parametrized in geometry, top view. Simplified modeling of the stator with coils.

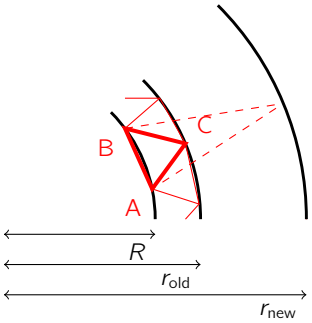


Figure 4.5. Scaling of a triangular mesh element in the annulus.

Knowing how the vertices of the triangles will be scaled using the non-linear function F_γ (cf. equation (4.4)), it is possible to calculate linear functions $G_{\gamma,j}$ for the scaling of the mesh in the annuli in the following way: The vertices of the triangle $(x_{A,j}, y_{A,j})$, $(x_{B,j}, y_{B,j})$ and $(x_{C,j}, y_{C,j})$ lie on circles with radii $r_{A,j}$,$r_{B,j}$ and $r_{C,j}$. Using the scaling function F_γ leads to the following scaling of the vertex A, which can be calculated for B and C in the same way:

$$\begin{pmatrix} x_{A,j} \\ y_{A,j} \end{pmatrix} \mapsto \underbrace{\left(\gamma + \frac{(1-\gamma)R}{r_{A,j}}\right)}_{=:D_{A,j}(\gamma)} \begin{pmatrix} x_{A,j} \\ y_{A,j} \end{pmatrix}.$$

We are now able to calculate a linear function $G_{\gamma,j}$ that maps the vertices of a triangle $\mathcal{T}_j = ((x_{A,j}, y_{A,j}), (x_{B,j}, y_{B,j}), (x_{C,j}, y_{C,j}))$ to the vertices of the scaled triangle $\mathcal{T}_j^s = (D_{A,j}(\gamma)(x_{A,j}, y_{A,j}), D_{B,j}(\gamma)(x_{B,j}, x_{B,j}), D_{C,j}(\gamma)(x_{C,j}, x_{C,j}))$:

$$\begin{aligned} &G_{\gamma,j} : \mathcal{T}_j \to \mathcal{T}_j^s \subset \mathbb{R}^2 \\ &\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \gamma K_{1,j} + K_{2,j} \\ \gamma K_{3,j} + K_{4,j} \end{pmatrix} + \begin{pmatrix} \gamma K_{5,j} + K_{6,j} & \gamma K_{7,j} + K_{8,j} \\ \gamma K_{9,j} + K_{10,j} & \gamma K_{11,j} + K_{12,j} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \end{aligned}$$

with constants $K_{1,j}$ to $K_{12,j}$ depending on the vertex coordiantes $(x_{A,j}, y_{A,j})$, $(x_{B,j}, y_{B,j})$, $(x_{C,j}, y_{C,j})$ and the radii $r_{A,j}$, $r_{B,j}$, $r_{C,j}$, R. The reader should note, that for every triangular mesh element a different scaling function $G_{\gamma,j}$ is needed, as it depends on the vertices. For later calculations, we state here

the Jacobian matrix of the inverse functions and the Jacobian determinant of the functions $G_{\gamma,j}$:

$$\left(J_{G_{\gamma,j}}\right)^{-1} = \frac{1}{\det J_{G_{\gamma,j}}} \begin{pmatrix} \gamma K_{11,j} + K_{12,j} & -\gamma K_{9,j} - K_{10,j} \\ -\gamma K_{7,j} - K_{8,j} & \gamma K_{5,j} + K_{6,j} \end{pmatrix}, \tag{4.6a}$$

$$\begin{aligned} \det J_{G_{\gamma,j}} &= \gamma^2(K_{5,j}K_{11,j} - K_{9,j}K_{7,j}) + K_{6,j}K_{12,j} - K_{10,j}K_{8,j} \\ &\quad + \gamma(K_{5,j}K_{12,j} + K_{6,j}K_{11,j} - K_{9,j}K_{8,j} - K_{10,j}K_{7,j}) \\ &= \gamma^2 d_{2,j} + \gamma d_{1,j} + d_{0,j} \\ &=: d_j(\gamma). \end{aligned} \tag{4.6b}$$

For the linear scaling (cf. equation (4.3)) the corresponding inverse Jacobian and determinant are given by:

$$\left(J_{G_\mu}\right)^{-1} = \begin{pmatrix} \frac{1}{1+\mu} & 0 \\ 0 & 1 \end{pmatrix}, \tag{4.7a}$$

$$\det J_{G_\mu} = 1 + \mu. \tag{4.7b}$$

In most of the motor parts, both scalings need to be incorporated. For example, the stator is scaled linearly in z and nonlinearly in x, y-direction. Hence a function in $\mathbb{R}^3$ will be used:

$$\begin{aligned} &G_{\gamma,\theta,j}((x,y,z)) \\ &= \begin{pmatrix} \gamma K_{1,j} + K_{2,j} \\ \gamma K_{3,j} + K_{4,j} \\ 0 \end{pmatrix} + \begin{pmatrix} \gamma K_{5,j} + K_{6,j} & \gamma K_{7,j} + K_{8,j} & 0 \\ \gamma K_{9,j} + K_{10,j} & \gamma K_{11,j} + K_{12,j} & 0 \\ 0 & 0 & 1+\theta \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \end{aligned}$$

The corresponding inverse Jacobian and Jacobian determinant are:

$$\begin{aligned} \left(J_{G_{\gamma,\theta,j}}\right)^{-1} &= \begin{pmatrix} \frac{\gamma K_{11,j} + K_{12,j}}{d_j(\gamma)} & \frac{-\gamma K_{9,j} - K_{10,j}}{d_j(\gamma)} & 0 \\ \frac{-\gamma K_{7,j} - K_{8,j}}{d_j(\gamma)} & \frac{\gamma K_{5,j} + K_{6,j}}{d_j(\gamma)} & 0 \\ 0 & 0 & \frac{1}{1+\theta} \end{pmatrix} \\ \det J_{G_{\gamma,j}} &= (1+\theta) d_j(\gamma), \end{aligned}$$

with $d_j(\gamma)$ as given in (4.6b). Let $\mathfrak{x} = (x, y, z)$ and $\mathfrak{x}^s = (x^s, y^s, z^s)$ be scaled coordinates. Using the weak formulation of the heat equation (4.1),

the equation (4.5) and substitution, one obtains for one entry of the scaled matrix E^s with a linear function G:

$$\begin{aligned} E^s_{kl} &= \rho C_p \int_{\Omega^s} \psi^s_k(\mathfrak{x}^s)\psi^s_l(\mathfrak{x}^s)\mathrm{d}\mathfrak{x}^s \qquad (4.8)\\ &= \rho C_p \int_{\Omega^s} (\psi_k \circ G^{-1})(\mathfrak{x}^s)(\psi_l \circ G^{-1})(\mathfrak{x}^s)\mathrm{d}\mathfrak{x}^s \\ &= \rho C_p \int_{G^{-1}(\Omega^s)} \psi_k(G^{-1}(G(\mathfrak{x})))\psi_l(G^{-1}(G(\mathfrak{x})))|\det J_G(\mathfrak{x})|\mathrm{d}\mathfrak{x} \\ &= \rho C_p \int_{\Omega} \psi_l(\mathfrak{x})\psi_k(\mathfrak{x})|\det J_G(\mathfrak{x})|\mathrm{d}\mathfrak{x}. \end{aligned}$$

Considering the function $G_{\gamma,\theta,j}$ one obtains $|\det J_{G_{\gamma,\theta,j}}(\mathfrak{x})| = (1+\theta)d_j(\gamma)$, depending on the mesh element $\mathcal{T}_j$. However, as for all j every $d_j(\gamma)$ is a polynomial of degree two in γ, the matrix E^s can be written as

$$E^s = (1+\theta)(\gamma^2 E_2 + \gamma E_1 + E_0).$$

For the matrix A the calculation of a dependency in the parameter for one entry of the scaled matrix A^s is more complicated:

$$\begin{aligned} A^s_{kl} &= \lambda \int_{\Omega_s} \nabla\psi^s_k(\mathfrak{x}^s)\nabla\psi^s_l(\mathfrak{x}^s)d\mathfrak{x}^s \qquad (4.9)\\ &= \lambda \int_{\Omega_s} \nabla\left((\psi_k \circ G^{-1})(\mathfrak{x}^s)\right)\nabla\left((\psi_l \circ G^{-1})(\mathfrak{x}^s)\right)d\mathfrak{x}^s \\ &= \lambda \int_{\Omega_s} \nabla\psi_k(G^{-1}(\mathfrak{x}^s))J_{G^{-1}}(\mathfrak{x}^s)\nabla\psi_l(G^{-1}(\mathfrak{x}^s))J_{G^{-1}}(\mathfrak{x}^s)d\mathfrak{x}^s \\ &= \lambda \int_{\Omega=G^{-1}(\Omega_s)} \nabla\psi_k(\mathfrak{x})J_{G^{-1}}(G(\mathfrak{x}))\nabla\psi_l(\mathfrak{x})J_{G^{-1}}(G(\mathfrak{x}))|\det J_G(\mathfrak{x})|d\mathfrak{x}. \end{aligned}$$

For the functions $G_{\gamma,\theta,j}$, the calculation of the integral (4.9) needs to be done in the mesh elements on which the basis functions ψ^s_k and ψ^s_l are supported, i.e.

$$\begin{aligned} A^s_{kl} &= \lambda \int_{\mathcal{T}^s_1 \cup \cdots \cup \mathcal{T}^s_{\text{end}}} \nabla\psi^s_k(\mathfrak{x}^s)\nabla\psi^s_l(\mathfrak{x}^s)d\mathfrak{x}^s \\ &= \lambda \sum_j \int_{\mathcal{T}^s_j} \nabla\psi^s_k(\mathfrak{x}^s)\nabla\psi^s_l(\mathfrak{x}^s)d\mathfrak{x}^s. \end{aligned}$$

The integral on one mesh element leads to (using equation (4.9)):

$$
\begin{aligned}
&A^s_{klj}\\
=&\int_{\mathcal{T}^s_j} \nabla\psi^s_k(\mathfrak{x}^s)\nabla\psi^s_l(\mathfrak{x}^s)d\mathfrak{x}^s\\
=&\lambda\int_{\mathcal{T}_j=G^{-1}_{\gamma,\theta,j}(\mathcal{T}^s_j)} \nabla\psi_k(\mathfrak{x})J_{G^{-1}_{\gamma,\theta,j}}(G_{\gamma,\theta,j}(\mathfrak{x}))\nabla\psi_l(\mathfrak{x})J_{G^{-1}_{\gamma,\theta,j}}(G_{\gamma,\theta,j}(\mathfrak{x}))|\det J_{G_{\gamma,\theta,j}}(\mathfrak{x})|d\mathfrak{x}\\
=&\lambda\int_{\mathcal{T}_j}\begin{bmatrix}\partial_1\psi_l(\mathfrak{x}) & \partial_2\psi_l(\mathfrak{x}) & \partial_3\psi_l(\mathfrak{x})\end{bmatrix}\begin{pmatrix}\frac{\gamma K_{11,j}+K_{12,j}}{d_j(\gamma)} & \frac{-\gamma K_{9,j}-K_{10,j}}{d_j(\gamma)} & 0\\ \frac{-\gamma K_{7,j}-K_{8,j}}{d_j(\gamma)} & \frac{\gamma K_{5,j}+K_{6,j}}{d_j(\gamma)} & 0\\ 0 & 0 & \frac{1}{1+\theta}\end{pmatrix}\\
&\cdot\left(\begin{bmatrix}\partial_1\psi_k(\mathfrak{x}) & \partial_2\psi_k(\mathfrak{x}) & \partial_3\psi_k(\mathfrak{x})\end{bmatrix}\begin{pmatrix}\frac{\gamma K_{11,j}+K_{12,j}}{d_j(\gamma)} & \frac{-\gamma K_{9,j}-K_{10,j}}{d_j(\gamma)} & 0\\ \frac{-\gamma K_{7,j}-K_{8,j}}{d_j(\gamma)} & \frac{\gamma K_{5,j}+K_{6,j}}{d_j(\gamma)} & 0\\ 0 & 0 & \frac{1}{1+\theta}\end{pmatrix}\right)^T\\
&\cdot|(1+\theta)d_j(\gamma)|\mathrm{d}\mathfrak{x}\\
=&\frac{1+\theta}{d_j(\gamma)}\cdot\gamma^2\int_{\mathcal{T}_j}\varphi^0_{kl}(\mathfrak{x})\mathrm{d}\mathfrak{x}+\frac{1+\theta}{d_j(\gamma)}\cdot\gamma\int_{\mathcal{T}_j}\varphi^1_{kl}(\mathfrak{x})\mathrm{d}\mathfrak{x}\\
&+\frac{1+\theta}{d_j(\gamma)}\int_{\mathcal{T}_j}\varphi^2_{kl}(\mathfrak{x})\mathrm{d}\mathfrak{x}+\frac{d_j(\gamma)}{1+\theta}\int_{\mathcal{T}_j}\varphi^3_{kl}(\mathfrak{x})\mathrm{d}\mathfrak{x},
\end{aligned}
$$

with functions φ^i_{kl} depending on the derivatives $\partial_1\psi_k$, $\partial_1\psi_l$, $\partial_2\psi_k$, $\partial_2\psi_l$, $\partial_3\psi_k$, $\partial_3\psi_l$ and the constants $K_{i,j}$. One matrix entry A^s_{kl}, considered as a function of γ depends on different $d_j(\gamma)$ and $K_{i,j}$.

For a different matrix entry A^s_{gh} and a different mesh element $\mathcal{T}_{\hat{j}}$ one obtains:

$$
\begin{aligned}
A^s_{gh}=&\frac{1+\theta}{d_{\hat{j}}(\gamma)}\cdot\gamma^2\int_{\mathcal{T}_{\hat{j}}}\varphi^0_{gh}(\mathfrak{x})\mathrm{d}\mathfrak{x}+\frac{1+\theta}{d_{\hat{j}}(\gamma)}\cdot\gamma\int_{\mathcal{T}_{\hat{j}}}\varphi^1_{gh}(\mathfrak{x})\mathrm{d}\mathfrak{x}\\
&+\frac{1+\theta}{d_{\hat{j}}(\gamma)}\int_{\mathcal{T}_{\hat{j}}}\varphi^2_{gh}(\mathfrak{x})\mathrm{d}\mathfrak{x}+\frac{d_{\hat{j}}(\gamma)}{1+\theta}\int_{\mathcal{T}_{\hat{j}}}\varphi^3_{gh}(\mathfrak{x})\mathrm{d}\mathfrak{x},
\end{aligned}
$$

with different denominators $d_j(\gamma)$ and $d_{\hat{j}}(\gamma)$. Hence it is not possible to find an easy affine dependency in the parameter γ like for the matrix E. We

state now the discretized heat equation with a parametrization in geometry. For the ease of presentation, we only consider changes in two parameters γ (resulting originally from a nonlinear scaling (4.4)) and θ (resulting from a linear scaling (4.3)):

$$E(\gamma,\theta) = (1+\theta)(\gamma^2 E_2 + \gamma E_1 + E_0), \tag{4.10}$$

$$A(\gamma,\theta) = (1+\theta)A_1(\gamma) + \frac{1}{1+\theta}A_2(\gamma). \tag{4.11}$$

The matrices N_1 and N_2 (cf. equation (4.2)) have the same dependency in the parameters as E. The calculation of the parameter dependency for the matrices B resulting from the right hand side of equation (4.1) can be executed by using substitution and equation (4.5) for the integral $\int_\Gamma \psi_k^s(\mathfrak{x}^s)d\mathfrak{x}^s$. For the different columns of B however, it is important to note that only those boundaries or parts of the model that will be affected by the scaling will actually change. If for example only the height of the stator changes, the Dirichlet boundary condition on top of the flange will not be affected. Assuming that θ changes the height of the stator and γ scales it in x,y-direction, the corresponding scalings will be as follows:

$$B_h(\gamma,\theta) = (1+\theta)(\gamma^2 B_{h,2} + \gamma B_{h,1} + B_{h,0}), \tag{4.12}$$

$$B_{T_0}(\gamma) = (\gamma^2 B_{T_0,2} + \gamma B_{T_0,1} + B_{T_0,0}), \tag{4.13}$$

$$B_S(\gamma,\theta) = (1+\theta)(\gamma^2 B_{S,2} + \gamma B_{S,1} + B_{S,0}), \tag{4.14}$$

where B_h refers to the outer surface of the housing with a Robin boundary condition, B_{T_0} refers to a Dirichlet boundary condition on the flange, whereas B_S models the heat source in the coils and $B = [B_h \quad B_{T_0} \quad B_S]$. For the two additional scalings of flange (original linear scaling with parameter μ) and housing (original nonlinear scaling with parameter ν), the generalization is straightforward.

Figure 4.6 shows results of a simulation of the parametrized model for $t = 600s$ without any scalings. After the discretization one obtains a system with $n = 71,978$ degrees of freedom. It is obvious, that the coils in the stator are the main heat sources. The temperature on the flange remains fixed at $T_0 = 348.15K$, whereas interaction between the model and the environment is given by convection.

To simplify the analysis of the geometry variations, it is convenient to have a model with the same physical behavior and the same scaling functions but with fewer degrees of freedom. Therefore a simplified model was built.

It consists of rotor, stator, housing and flange. The geometry can be seen in Figure 4.7. A result of a simulation of the heat flux with scaling of the stator in z- and x, y-direction is shown in Figure 4.8.

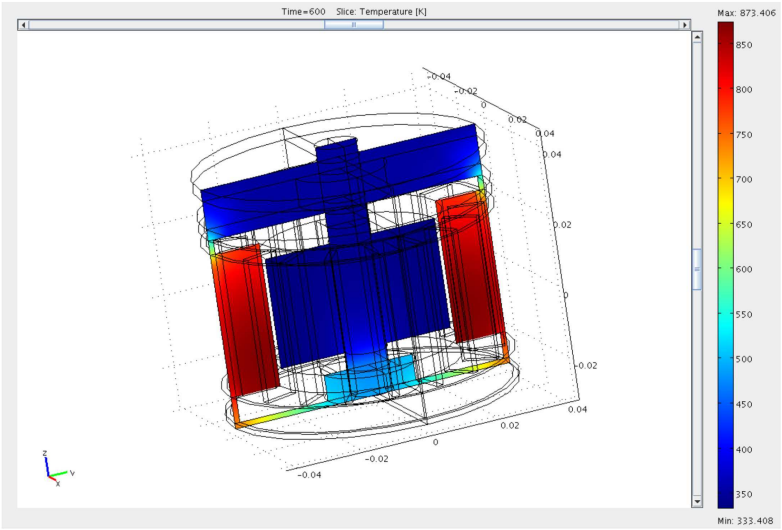


Figure 4.6. Simulation of large model — no scaling function was applied.

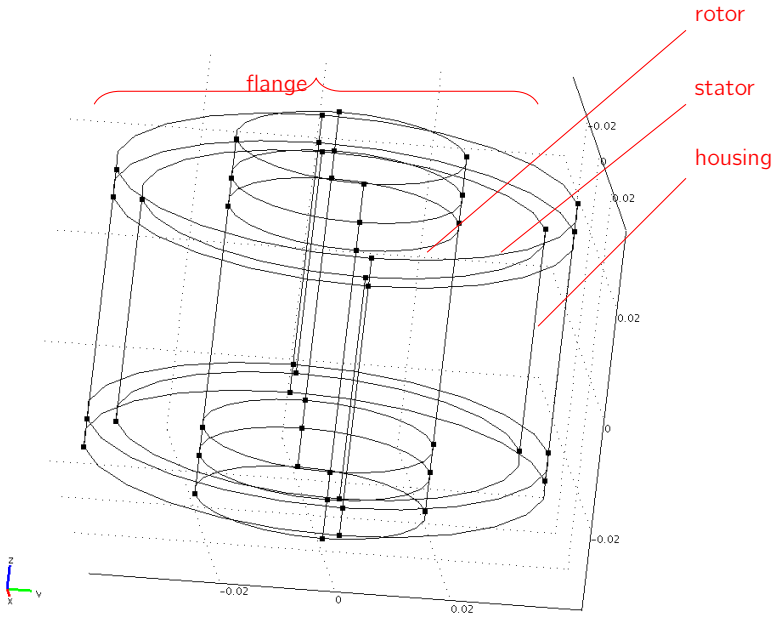


Figure 4.7. Simplified motor model.

In this chapter we have shown that it is possible to obtain parametrized models by an analysis of the underlying equations. By inserting scaling functions into Comsol®, the scaling of an electrical motor model can be analyzed, and these scalings can be represented by parameters. First, linear scalings have been considered (cf. equation (4.3)). Inserting them in the finite element discretization of the heat equation shows that these scalings can be considered as affine parameters (cf. the parameter θ in equations (4.10) to (4.14)). Second, nonlinear scalings have been examined (cf. equation (4.4)). They can be considered as linear scalings by using the scaling of the underlying mesh and hence a parameter dependency can be obtained by inserting these linear scalings into the finite element discretization of the heat equation.

However these originally non-linear scalings lead to a non-affine parameter dependency for the matrix A (cf. parameter γ in equation (4.11)). Having derived the parameter dependency of our models, methods from parametric Model Order Reduction (cf. Section 5.3) can be applied to obtain small reduced order models. In addition, several Comsol® models for the thermal analysis of electrical motors have been built and presented in this chapter.

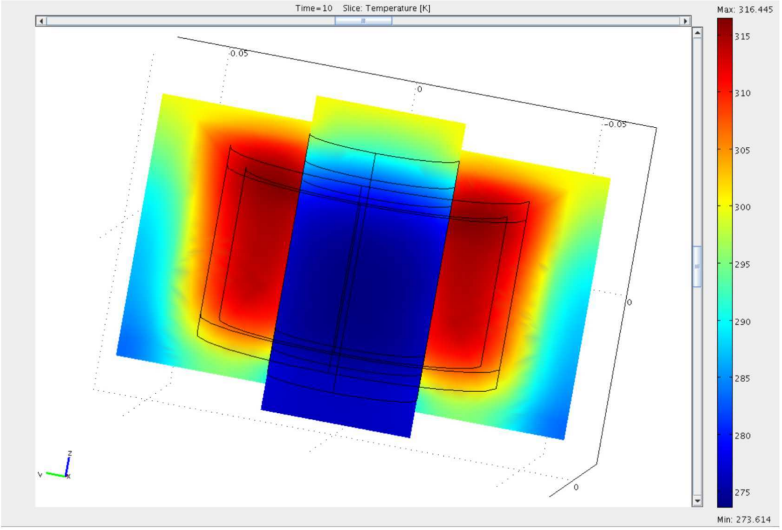


Figure 4.8. Simplified motor model after the scaling and a short thermal simulation

CHAPTER 5

Model Order Reduction

Model Order Reduction (MOR) is a powerful method to reduce the dimension of large dynamical systems and therefore the simulation time significantly while guaranteeing a very good approximation of the original output. The simulation of a linear system

$$\Sigma_{\text{lin}} : \begin{cases} E\dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), \quad x(0) = x_0, \end{cases} \tag{2.18}$$

where $E, A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $u(t) \in \mathbb{R}^m$, $x(t) \in \mathbb{R}^n$ and $y(t) \in \mathbb{R}^p$ requires a large amount of time if the number of degrees of freedom n is large. The main idea of projection based MOR is to find matrices that project the system onto a low-dimensional subspace and by that obtain a reduced model:

$$\hat{\Sigma}_{\text{lin}} : \begin{cases} \hat{E}\dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \hat{B}u(t), \\ \hat{y}(t) = \hat{C}\hat{x}(t), \quad \hat{x}(0) = \hat{x}_0 \end{cases} \tag{5.1}$$

with $\hat{E}, \hat{A} \in \mathbb{R}^{r\times r}$, $\hat{B} \in \mathbb{R}^{r\times m}$, $\hat{C} \in \mathbb{R}^{p\times r}$, $u(t) \in \mathbb{R}^m$, $\hat{x}(t) \in \mathbb{R}^r$ and $\hat{y}(t) \in \mathbb{R}^p$, where $r \ll n$. A reduced order model is extremely useful, when not only one, but a large number of simulations needs to be done with different input scenarios (e.g. in optimization, parameter studies or feedback control) as it reduces the simulation time significantly.
In this chapter, the theory of MOR is reviewed. First, methods for MOR of linear systems are stated (cf. Section 5.2), followed by a short introduction to parametric Model Order Reduction (pMOR), in Section 5.3. A certain class of linear parametric systems can be reformulated as bilinear systems (cf. Section 5.3.2) and hence reduced using bilinear Model Order Reduction. Methods for bilinear MOR will be reviewed (cf. Sections 5.4 and 5.5), and a new bilinear $\mathscr{H}_2$-optimal reduction method, based on optimization on Grassmann manifolds is derived in Section 5.5.4.

5.1. Projection-based MOR and the error system

The following two definitions state the main properties of a projector.

Definition 5.1.1. A *projector* is a matrix $\mathcal{P} \in \mathbb{R}^{n\times n}$ with $\mathcal{P}^2 = \mathcal{P}$. $\mathcal{P}$ is the projection onto a subspace $\mathcal{V} \subset \mathbb{R}^n$ if $\text{range}(\mathcal{P}) = \mathcal{V}$. $\mathcal{P}$ is an *orthogonal projector (or Galerkin projection)* if $\mathcal{P} = \mathcal{P}^T$, otherwise an *oblique projector (or Petrov-Galerkin projection)*.

Definition 5.1.2. If $V = [v_1, \dots, v_k]$ is a basis of $\mathcal{V}$, then $\mathcal{P}_\mathcal{V} = V(V^TV)^{-1}V^T$ is a projector onto $\mathcal{V}$. Let $\mathcal{W}$ be another k-dimensional subspace of $\mathbb{R}^n$. The projector $\mathcal{P}_{\mathcal{V}\mathcal{W}} = V(W^TV)^{-1}W^T$, projects onto $\mathcal{V}$ along $\mathcal{W}$.

Assume that the original state $x(t) \in \mathbb{R}^n$ approximately lies in a low-dimensional subspace $\mathcal{V}$ with $\dim(\mathcal{V}) = r \ll n$, hence $x(t)$ can be approximated by a linear combination of basis vectors of $\mathcal{V}$: $x(t) \approx V\hat{x}(t)$, with $\hat{x}(t) \in \mathbb{R}^r$. By inserting this into the original linear system, one obtains:

$$\begin{aligned} EV\dot{\hat{x}}(t) &= AV\hat{x}(t) + Bu(t) + \varepsilon(t), \\ y(t) &\approx CV\hat{x}(t). \end{aligned} \tag{5.2}$$

As $EV\dot{\hat{x}}(t) \in \text{span}(EV)$, one can project $AV\hat{x}(t) + Bu(t)$ onto EV along a subspace W which is orthogonal to the residual (i.e. $W^T\varepsilon(t) = 0$ holds)

and the reduced-order model can then be obtained:

$$\hat{\Sigma}_{\text{lin}}: \begin{cases} \overbrace{W^T E V}^{\hat{E}} \dot{\hat{x}}(t) = \overbrace{W^T A V}^{\hat{A}} \hat{x}(t) + \overbrace{W^T B}^{\hat{B}} u(t), \\ \hat{y}(t) = \underbrace{CV}_{\hat{C}} \hat{x}(t), \end{cases} \tag{5.3}$$

where $\hat{E}, \hat{A} \in \mathbb{R}^{r \times r}$, $\hat{B} \in \mathbb{R}^{r \times m}$, $\hat{C} \in \mathbb{R}^{p \times r}$ and $\hat{y}(t) \in \mathbb{R}^p$. Determining suitable matrices V and W is the main aim of projective Model Order Reduction.

It remains to determine if the reduced order model is a good approximation to the original. The outputs of the reduced model and the original model will therefore be compared:

$$y^{\text{err}}(t) = y(t) - \hat{y}(t).$$

Accordingly, one can derive the following error system:

$$\Sigma_{\text{lin}}^{\text{err}}: \begin{cases} \begin{bmatrix} E & 0 \\ 0 & \hat{E} \end{bmatrix} \begin{bmatrix} \dot{x}(t) \\ \dot{\hat{x}}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & \hat{A} \end{bmatrix} \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix} + \begin{bmatrix} B \\ \hat{B} \end{bmatrix} u(t), \\ y^{\text{err}}(t) = Cx(t) - \hat{C}\hat{x}(t) = \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix}. \end{cases}$$

Let $x^{\text{err}}(t) = \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix}$ be the states of the error system. The transfer function of the error system can be stated as:

$$\begin{aligned} H^{\text{err}}(s) &= \overbrace{\begin{bmatrix} C & -\hat{C} \end{bmatrix}}^{C^{\text{err}}} \overbrace{\left(s \begin{bmatrix} E & 0 \\ 0 & \hat{E} \end{bmatrix} - \begin{bmatrix} A & 0 \\ 0 & \hat{A} \end{bmatrix} \right)^{-1}}^{(sE^{\text{err}} - A^{\text{err}})^{-1}} \overbrace{\begin{bmatrix} B \\ \hat{B} \end{bmatrix}}^{B^{\text{err}}} \\ &= C(sE - A)^{-1}B - \hat{C}(s\hat{E} - \hat{A})^{-1}\hat{B} \\ &= H(s) - \hat{H}(s). \end{aligned}$$

In the frequency domain it holds

$$Y^{\text{err}}(s) = H^{\text{err}}(s)U(s). \tag{5.4}$$

It is now the objective to minimize $Y^{\text{err}}(s)$, the error between the original output function $Y(s)$ and the reduced output $\hat{Y}(s)$. By using Parseval's theorem (cf. [5]), one obtains that the $\mathcal{L}_2$-norms (cf. Section 2.3.1.3) of $Y^{\text{err}}(s)$ and $U(s)$ in the frequency domain and $y^{\text{err}}(t)$ and $u(t)$ in the

time domain coincide. To quantify the Model Order Reduction error, it is desirable to find an error bound of the following structure:

$$||y^{\text{err}}||_{\mathcal{L}_2^p} \leq \epsilon ||u^{\text{err}}||_{\mathcal{L}_2^m}.$$

By using the input-output relationship (5.4) in the frequency domain, the ϵ can be given as the difference between the transfer functions $H(s)$ and $\hat{H}(s)$, which can be measured in the $\mathscr{H}_\infty$-norm of the error system:

$$||\Sigma_{\text{lin}}^{\text{err}}||_{\mathscr{H}_\infty} = ||\Sigma_{\text{lin}} - \hat{\Sigma}_{\text{lin}}||_{\mathscr{H}_\infty} = \sup_{y \in \mathbb{R}} \left(\sigma_{\max}(H(iy) - \hat{H}(iy))\right). \tag{5.5}$$

For the error of the impulse response, the $\mathscr{H}_2$-norm can be used, and computated in the following way:

$$||\Sigma_{\text{lin}}^{\text{err}}||_{\mathscr{H}_2} = ||\Sigma_{\text{lin}} - \hat{\Sigma}_{\text{lin}}||_{\mathscr{H}_2} = \sqrt{\text{tr}((B^{\text{err}})^* Q^{\text{err}} B^{\text{err}})} = \sqrt{\text{tr}(C^{\text{err}} P^{\text{err}} (C^{\text{err}})^*)}, \tag{5.6}$$

where P^{err} and Q^{err} are the Gramians of the error system.

5.2. MOR of linear systems

In the following Section, several methods for the reduction of linear systems will be shortly reviewed. Numerous researchers have been working on the reduction of this class of systems in the last three decades. For a detailed introduction, we refer to the book of Antoulas [5] and the references therein. We assume that all linear systems we consider throughout this section are reachable, observable and stable. In addition, the matrix E is always invertible.

5.2.1. Balanced Truncation. Consider a stable, observable and controllable linear system (cf. (2.18), Chapter 2.3.1.2). The basic idea of the balanced truncation method is to eliminate the states in which the system is difficult to observe and difficult to reach. The following derivation of Balanced Truncation follows basically the dissertation of Stykel [61], whereas we examine only systems with nonsingular E matrix. As the system is stable, observable and controllable, the controllability (2.19) and the observability Gramian (2.21) exist and are positive definite, they can be factorised by using a Cholesky factorization: $P = \tilde{R}^T \tilde{R}$ and $Q = \tilde{L}\tilde{L}^T$ with $\tilde{R} \in \mathbb{R}^{n \times n}$ and $\tilde{L} \in \mathbb{R}^{n \times n}$ of full rank. The second step consists of calculating the singular

value decomposition of the product:

$$\tilde{L}E\tilde{R} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix},$$

with $U_1, V_1 \in \mathbb{R}^{n\times r}$, $U_2, V_2 \in \mathbb{R}^{n\times(n-r)}$ having orthogonal columns and $\Sigma_1 = \operatorname{diag}(\varsigma_1, \ldots \varsigma_r)$ and $\Sigma_2 = \operatorname{diag}(\varsigma_{r+1}, \ldots \varsigma_n)$ are matrices containing the Hankel singular values ordered in descending order. A balanced reduced realization can now be computed using

$$W = \tilde{L}^T U_1 \Sigma_1^{-1/2} \in \mathbb{R}^{n\times r} \text{ and } V = \tilde{R} V_1 \Sigma_1^{-1/2} \in \mathbb{R}^{n\times r}.$$

The reduction of the linear system is then performed using W and V as projections in the following way:

$$\hat{E} = W^T E V, \quad \hat{A} = W^T A V, \quad \hat{B} = W^T B, \quad \hat{C} = CV.$$

The quality of the approximation can be measured in the $\mathcal{H}_\infty$-norm according to the following error bound:

Theorem 5.2.1 ([5]). *The $\mathcal{H}_\infty$-norm of the error system is bounded by the sum of neglected Hankel singular values:*

$$||\Sigma_{\text{lin}} - \hat{\Sigma}_{\text{lin}}||_{\mathcal{H}_\infty} \leq 2(\varsigma_{r+1} + \cdots + \varsigma_n).$$

5.2.2. Krylov subspace methods. The main idea behind Krylov subspace methods, which are widely used for the reduction of linear systems, consists of comparing the summands of series expansions of the original and reduced systems transfer functions. Various authors contributed to the development of this technique, for a deeper insight we refer to the book of Antoulas [5] and the references therein.

Definition 5.2.2. The *ℓ-th (block) Krylov subspace* for $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ is defined as follows:

$$\mathcal{K}_\ell(A, B) = \operatorname{span}\{B, AB, \ldots, A^{\ell-1}B\}. \tag{5.7}$$

Definition 5.2.3 ([5, 31, 36]). The *moments* of a transfer function $H(s)$ evaluated at $s = s_0 \in \mathbb{C}$ are

$$m_k(s_0) = (-1)^k \frac{\mathrm{d}^k}{\mathrm{d}s^k} H(s)|_{s=s_0}.$$

It holds

$$m_k(s_0) = C((s_0E - A)^{-1}E)^k (s_0E - A)^{-1}B, \quad k = 0, 1, \ldots.$$

The moments m_k are the coefficients of a Laurent series expansion of the transfer function $H(s)$ around s_0. Expanding at infinity leads to the definition of the so called Markov parameters:

Definition 5.2.4. The *Markov parameters* (also called the moments at infinity) of a system are defined as:

$$M_j = C(E^{-1}A)^j E^{-1}B, \quad j = 0, 1, \dots.$$

The moments and Markov parameters of the original and the reduced system can now be compared. The objective of the so called "moment matching" methods is to find projection matrices such that a certain number of these moments are equal for the reduced and the original system without the need of explicitely calculating them. The following theorem shows how to choose the projection matrices in order to achieve moment matching. They are formulated for the case in which a matching around $s_0 = 0$ is desired.

Theorem 5.2.5 ([58]). *If the columns of the matrices V and W used in (5.3) form bases for the Krylov subspaces $\mathcal{K}_{\ell_1}(A^{-1}E, (E^{-1}A)^{r_1}A^{-1}B)$ and $\mathcal{K}_{\ell_2}(A^{-T}E^T, (E^{-T}A^T)^{r_2}A^{-T}C)$, respectively, both with rank q, where q is a multiple of m and p, then the first $\frac{q-r_1}{m} + \frac{q-r_2}{p}$ moments and $\frac{r_1}{m} + \frac{r_2}{p}$ Markov parameters of the original and the reduced order system match.*

A reduced model calculated using the Krylov subspaces above leads to a good approximation of the original model, as long as a sufficient number of moments and Markov parameters is matched. This guarantees that the reduced transfer function is an approximation of the original one.

The calculation of the matrices V and W can be done by using the Arnoldi or the Lanczos algorithm. They can be found in [5].
In the case where a moment matching around several different points s_k is considered, the following theorem has been shown by Gallivan et al. [36]:

Theorem 5.2.6 ([36]). *If*

$$\bigcup_{k=1}^{K} \mathcal{K}_{J_k}((s_kE - A)^{-1}E, (s_kE - A)^{-1}B) \subset \mathrm{Im}(V),$$

and

$$\bigcup_{k=1}^{K} \mathcal{K}_{L_k}((s_kE - A)^{-T}E^T, (s_kE - A)^{-T}C^T) \subset \mathrm{Im}(W).$$

where the points $s_k \in \mathbb{C}$ are chosen such that the pencils $s_kE - A$ are invertible for all $k \in \{1, \dots, K\}$, then the $J_k + L_k$ moments at s_k of the original linear system Σ_{lin} (2.18) and those of the reduced linear system $\hat{\Sigma}_{\text{lin}}$ match, provided the matrices $s_k\hat{E} - \hat{A}$ are invertible.

Moment-matching around points s_k is nothing else than assuring that the reduced transfer function interpolates the original transfer function at points s_k.

5.2.3. Rational Interpolation. First, we only consider so called single input single output systems (SISO), i.e. systems with $C^T, B \in \mathbb{R}^n$. The projection matrix V is now obtained by

$$V = [(s_1E - A)^{-1}B, \dots, (s_rE - A)^{-1}B], \tag{5.8}$$

with distinct parameters $s_1, \dots, s_r$. Let W be such that $W^TV = I_r$. The following interpolation conditions can be established:

Proposition 5.2.7 ([5], Proposition 11.6)**.** *The transfer function of the reduced system $\hat{\Sigma}_{\text{lin}}$ as in (5.3) obtained by using V as given in (5.8) and W with $W^TV = I_r$, interpolates the transfer function of the original system Σ_{lin} at the points s_k, that is*

$$H(s_k) = C(s_kE - A)^{-1}B = \hat{C}(s_k\hat{E} - \hat{A})^{-1}\hat{B} = \hat{H}(s_k), \quad k = 1, \dots, r.$$

Using the matrix V defined as in (5.8) would hence lead to a matching of one moment around each interpolation point s_k (cf. Theorem 5.2.6). The interpolation conditions have been examined for two sided projections as well. It is possible to establish interpolation conditions for the derivatives:

Proposition 5.2.8 ([5])**.** *Let $\hat{\Sigma}_{\text{lin}}$ as in (5.3) with*

$$V = [(s_0E - A)^{-1}B, (s_0E - A)^{-2}B, \dots, (s_0E - A)^{-r}B],$$

and $W^TV = I_r$. It interpolates the transfer function of Σ at s_0 together with $r - 1$ derivatives at the same point:

$$\begin{aligned}\frac{(-1)^k}{k!}\frac{\mathrm{d}^k}{\mathrm{d}s^k}H(s)|_{s=s_0} &= C(s_0E - A)^{-(k+1)}B \\ &= \hat{C}(s_0\hat{E} - \hat{A})^{-(k+1)}\hat{B} = \frac{(-1)^k}{k!}\frac{\mathrm{d}^k}{\mathrm{d}s^k}\hat{H}(s)|_{s=s_0},\end{aligned}$$

where $k = 1, 2, \dots, r - 1$.

Consider the following matrices

$$V = \begin{bmatrix}(s_1E - A)^{-1}B & \dots & (s_rE - A)^{-1}B\end{bmatrix}, \tag{5.9}$$

$$W = \begin{bmatrix}(s_{r+1}E - A)^{-T}C^T & \dots & (s_{2r}E - A)^{-T}C^T\end{bmatrix}. \tag{5.10}$$

Then the following proposition derives the interpolation conditions for a system with two projection matrices:

Proposition 5.2.9 ([5]). *Assuming full rank $V, W \in \mathbb{R}^{n\times r}$ given as in (5.9) and (5.10), the transfer function of the projected system $\hat{\Sigma}_{\text{lin}}$ defined by (5.3) interpolates the transfer function of Σ_{lin} at the points s_i, $i = 1, \dots 2r$.*

Using Theorem 5.2.6, one obtains that V and W as defined in equations (5.9) and (5.10) lead to a matching of 2 moments around each point s_k.

For systems with multiple inputs and multiple outputs (MIMO) corresponding interpolation conditions — the so called tangential interpolation conditions — have been examined by several researchers [35, 65, 43, 41]. The following theorem can be obtained:

Theorem 5.2.10 ([35, 65, 43, 41]). *Let $V, W \in \mathbb{R}^{n\times r}$ be of full rank. Let $s_k \in \mathbb{C}$, $r_k \in \mathbb{R}^{m\times 1}$ and $l_k \in \mathbb{R}^{1\times p}$ be interpolation points and left and right tangential directions. Let the points s_k be chosen such that $s_kE - A$ is non-singular. If for all $k = 1, \dots, r$ it holds*

$$(s_kE - A)^{-1}Br_k \in \text{span}(V),$$
$$(s_kE - A)^{-T}C^Tl_k^T \in \text{span}(W),$$

the reduced system (W^TEV, W^TAV, W^TB, CV) satisfies:

$$l_k\hat{H}(s_k) = l_kH(s_k), \tag{5.11}$$

$$\hat{H}(s_k)r_k = H(s_k)r_k, \tag{5.12}$$

$$l_k\hat{H}'(s_k)r_k = l_kH'(s_k)r_k, \text{ for all } k = 1, \dots, r. \tag{5.13}$$

It remains the task of choosing interpolation points s_k and interpolation directions r_k, l_k such that the obtained reduced model is a good approximation to the original one. This problem has been examined in the context of $\mathscr{H}_2$-optimal Model Order Reduction, which we will review in the next section.

5.2.4. $\mathscr{H}_2$-optimal Model Order Reduction. The objective of the $\mathscr{H}_2$-optimal MOR is to find a reduced system $(\hat{E}, \hat{A}, \hat{B}, \hat{C})$ such that the error of the system examined in the $\mathscr{H}_2$-norm $||\Sigma_{\text{lin}} - \hat{\Sigma}_{\text{lin}}||_{\mathscr{H}_2}$ is minimized.

5.2.4.1. *Interpolation-based $\mathscr{H}_2$-optimality conditions.* With the aim of minimizing the $\mathscr{H}_2$-norm of the error system $||\Sigma_{\text{lin}} - \hat{\Sigma}_{\text{lin}}||_{\mathscr{H}_2}$, the derivation of this norm using the system Gramians representation (2.29) is considered, following the derivation given by van Dooren et al. [64].

Let $P^{\text{err}} = \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix}$ and $Q^{\text{err}} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix}$ be the solutions of the Lyapunov equations of the error system:

$$A^{\text{err}} P^{\text{err}} (E^{\text{err}})^T + E^{\text{err}} P^{\text{err}} (A^{\text{err}})^T + B^{\text{err}} (B^{\text{err}})^T = 0,$$
$$(A^{\text{err}})^T Q^{\text{err}} E^{\text{err}} + (E^{\text{err}})^T Q^{\text{err}} A^{\text{err}} + (C^{\text{err}})^T C^{\text{err}} = 0,$$

where

$$A^{\text{err}} = \begin{bmatrix} A & 0 \\ 0 & \hat{A} \end{bmatrix}, \quad B^{\text{err}} = \begin{bmatrix} B \\ \hat{B} \end{bmatrix}, \quad E^{\text{err}} = \begin{bmatrix} E & 0 \\ 0 & \hat{E} \end{bmatrix}, \quad C^{\text{err}} = \begin{bmatrix} C \\ -\hat{C} \end{bmatrix}.$$

We aim at minimizing

$$\mathcal{J} := ||\Sigma_{\text{lin}} - \hat{\Sigma}_{\text{lin}}||_{\mathscr{H}_2}^2 = \text{tr}(C^{\text{err}} P^{\text{err}} (C^{\text{err}})^T) = \text{tr}((B^{\text{err}})^T Q^{\text{err}} B^{\text{err}}). \quad (5.14)$$

We can rewrite $\mathcal{J}$ as:

$$\begin{aligned} \mathcal{J} &= \text{tr}(B^T Q_{11} B + 2B^T Q_{12} \hat{B} + \hat{B}^T Q_{22} \hat{B}) \\ &= \text{tr}(C P_{11} C^T - 2C P_{12} \hat{C}^T + \hat{C} P_{22} \hat{C}^T). \end{aligned} \quad (5.15)$$

The gradient of a matrix valued function can be defined as follows:

Definition 5.2.11 ([64])**.** The *gradient* of a real scalar function $f(X)$ of a matrix variable $X \in \mathbb{R}^{n \times m}$ is the matrix $\nabla_X f(X) \in \mathbb{R}^{n \times m}$ defined by

$$[\nabla_X f(X)]_{ij} = \frac{\text{d}}{\text{d}X_{ij}} f(X), \quad i = 1, \ldots, n, \quad j = 1, \ldots, m.$$

The calculation of the gradient with respect to each of the system matrices leads to (cf. [64]):

$$\begin{aligned} \nabla_{\hat{E}} \mathcal{J} &= 2(Q_{22} \hat{A} P_{22} + Q_{12}^T A P_{12}), \\ \nabla_{\hat{A}} \mathcal{J} &= 2(Q_{22} \hat{E} P_{22} + Q_{12}^T E P_{12}), \\ \nabla_{\hat{B}} \mathcal{J} &= 2(Q_{22} \hat{B} + Q_{12}^T B), \\ \nabla_{\hat{C}} \mathcal{J} &= 2(\hat{C} P_{22} - C P_{12}). \end{aligned}$$

For an optimal interpolation point, the gradient of the function $\mathcal{J}$ must be zero. This leads to the following conditions:

Theorem 5.2.12 (Wilson conditions for systems with $E \neq I_n$, E nonsingular)**.** *If the reduced transfer function $\hat{H}(s)$ minimizes $\mathcal{J}$, then the following holds:*

$$\begin{aligned} Q_{22}\hat{A}P_{22} + Q_{12}^T A P_{12} &= 0, \\ Q_{22}\hat{E}P_{22} + Q_{12}^T E P_{12} &= 0, \\ Q_{22}\hat{B} + Q_{12}^T B &= 0, \\ \hat{C}P_{22} - C P_{12} &= 0. \end{aligned} \tag{5.16}$$

One directly concludes that the following proposition holds:

Proposition 5.2.13 (cf. [64])**.** *For every stationary point of $\mathcal{J}$ where P_{22} and Q_{22} are invertible, we have the following identities:*

$$\hat{E} = W^T E V, \quad \hat{A} = W^T A V, \quad \hat{B} = W^T B, \quad \hat{C} = C V,$$

with $W := -Q_{12}Q_{22}^{-1}$ and $V := P_{12}P_{22}^{-1}$, P_{12}, P_{22}, Q_{12} and Q_{22} satisfy the following Sylvester and Lyapunov equations:

$$A P_{12}\hat{E}^T + E P_{12}\hat{A}^T + B\hat{B}^T = 0, \tag{5.17}$$

$$A^T Q_{12}\hat{E} + E^T Q_{12}\hat{A} - C^T\hat{C} = 0, \tag{5.18}$$

$$\hat{A}P_{22}\hat{E}^T + \hat{E}P_{22}\hat{A}^T + \hat{B}\hat{B}^T = 0, \tag{5.19}$$

$$\hat{A}^T Q_{22}\hat{E} + \hat{E}^T Q_{22}\hat{A} + \hat{C}^T\hat{C} = 0. \tag{5.20}$$

Remark 5.2.14. An $\mathscr{H}_2$-optimal reduced order model fulfills the Wilson conditions given in Theorem 5.2.12. A model fulfilling the Wilson conditions is not necessarily to be $\mathscr{H}_2$-optimal!

If one wants to calculate an $\mathscr{H}_2$-optimal reduced order model, one might think of iteratively solving the Sylvester equations (5.17) and (5.18) starting from a (randomly) chosen reduced order model and updating V and W (and hence the reduced model) in every step.

It is possible to establish the equivalence between these Wilson conditions for $\mathscr{H}_2$-optimality and recently obtained interpolation conditions (cf. for example [43]). They have been first derived for the SISO case by Gugercin and coworkers [41] and then independently generalized to the

MIMO case not only by Gugercin but also by Van Dooren and coworkers [64], as well as Bunse-Gerstner and coworkers [21]. As a first derivation of interpolation conditions was done by Meier and Luenberger in 1976 [48], we will refer to these conditions as the Meier-Luenberger conditions.

Theorem 5.2.15 (Meier-Luenberger conditions)**.** *Given a linear stable system with transfer function $H(s)$, if $\hat{H}(s)$ is the best stable approximation of H with respect to the $\mathscr{H}_2$-norm, then the following conditions hold (for $k = 1, \ldots, r$):*

$$\tilde{C}_k^T \hat{H}(-\hat{\lambda}_k) = \tilde{C}_k^T H(-\hat{\lambda}_k), \tag{5.21}$$

$$\hat{H}(-\hat{\lambda}_k)\tilde{B}_k = H(-\hat{\lambda}_k)\tilde{B}_k, \tag{5.22}$$

$$\tilde{C}_k^T \hat{H}'(-\hat{\lambda}_k)\tilde{B}_k = \tilde{C}_k^T H'(-\hat{\lambda}_k)\tilde{B}_k, \tag{5.23}$$

with $\tilde{C} = \hat{C}X$ and $\tilde{B} = \hat{B}^T Y$, where Y, X are the left and right eigenvectors of $\hat{A} - \lambda\hat{E}$ and have been calculated such that $Y^*\hat{A}X = \operatorname{diag}(\hat{\lambda}_1, \ldots, \hat{\lambda}_r)$ and $Y^*\hat{E}X = I_r$.

The connection between Theorems 5.2.10 and 5.2.15 can now be seen: If

$$(-\hat{\lambda}_k E - A)^{-1} B\tilde{B}_k \in \operatorname{span}(V),$$

and

$$(-\hat{\lambda}_k E - A)^{-T} C^T \tilde{C}_k^T \in \operatorname{span}(W),$$

hold for the projections V and W, the conditions (5.21) – (5.23) are satisfied. This leads to Algorithm 1, widely known as IRKA (**I**nterpolatory **R**ational **K**rylov **A**lgorithm) [41, 6]. It has also been derived in a slightly different version as MIRIam (**M**IMO **I**terative **R**ational **I**nterpolation **A**lgorithm) by Bunse-Gerstner and coworkers [43, 21].

5.2.4.2. *$\mathscr{H}_2$-optimal models via optimization on manifolds.* Another approach has been developed by Yan and Lam in 1999 [69]. They assume that the reduced order model (5.3) has been generated by a one sided projection $U = V = W$ and, hence, $\mathcal{J}$ can be perceived as a function of U [69]:

$$\mathcal{J}(U) = \operatorname{tr}(BB^T(Q_{11} + UQ_{22}U^T + 2Q_{12}U^T)) \tag{5.24}$$

$$= \operatorname{tr}(C^T C(P_{11} + UP_{22}U^T - 2P_{12}U^T)), \tag{5.25}$$

where $\mathcal{J}(U)$ coincides with $\mathcal{J}$ as given in equation (5.15) by inserting $\hat{B} = U^T B$ and $\hat{C} = CU$. Yan and Lam [69] have shown that minimizing $\mathcal{J}(U)$

Algorithm 1 IRKA as given in [6].

Input: Initial selection of interpolation points $\sigma_1, \dots, \sigma_r$ and initial left and right tangential directions $l_1, \dots, l_r \in \mathbb{R}^{1\times p}$ and $q_1, \dots, q_r \in \mathbb{R}^{m\times 1}$.
Output: Reduced order model $\hat{E} = W^T E V$, $\hat{A} = W^T A V$, $\hat{B} = W^T B$, $\hat{C} = CV$.

1: $V = [(\sigma_1 E - A)^{-1} B q_1, \dots, (\sigma_r E - A)^{-1} B q_r]$
2: $W = [(\sigma_1 E - A)^{-T} C^T l_1, \dots, (\sigma_r E - A)^{-T} C^T l_r]$
3: **while** not converged **do**
4: $\quad \hat{E} = W^T E V$, $\hat{A} = W^T A V$, $\hat{B} = W^T B$, $\hat{C} = CV$
5: $\quad$ Compute $Y^* \hat{A} X = \mathrm{diag}(\lambda_1, \dots, \lambda_r)$ and $Y^* \hat{E} X = I_r$, where Y^* and X are the left and right eigenvectors of $\lambda \hat{E} - \hat{A}$.
6: $\quad$ Set $\sigma_k \leftarrow -\lambda_k$ and $l_k^* \leftarrow e_k^T Y^* \hat{B}$ $q_k \leftarrow \hat{C} X e_k$
7: $\quad V = [(\sigma_1 E - A)^{-1} B q_1, \dots, (\sigma_r E - A)^{-1} B q_r]$
8: $\quad W = [(\sigma_1 E - A)^{-T} C^T l_1, \dots, (\sigma_r E - A)^{-T} C^T l_r]$
9: **end while**

can be seen as the following minimization problem on the Stiefel manifold $\mathrm{St}(r, n) := \{X \in \mathbb{R}^{n\times r}, r \le n | X^T X = I_r\}$:

$$\begin{aligned} &\text{Minimize } \mathcal{J}(U) \text{ over } U \in \mathrm{St}(r, n) \\ &\text{subject to the stability of the reduced system.} \end{aligned} \tag{5.26}$$

Using tools from differential geometry, they derived an iterative gradient flow algorithm calculating a new projection matrix U in every step until a minimum of $\mathcal{J}(U)$ is reached. This method has recently been further developed by Xu and Zeng [68]. For a deeper insight in the used theory, the reader is referred to [69, 68] or Section 5.5.4, where the corresponding theory will be derived for bilinear systems.

5.3. Parametric Model Order Reduction (pMOR)

In applications, parameters are often incorporated in the linear models, for example geometric variations or physical effects (cf. Section 4). Hence, it is desirable to find methods to reduce these models, keeping their parameter dependency. An overview of methods for parametric model order reduction can be found in [13]. A parametric model is defined as follows:

Definition 5.3.1. A *linear parametric system* of order n is a matrix differential equation of the following form:

$$\Sigma_{\text{lin}}(\mathfrak{p}): \quad \begin{cases} E(\mathfrak{p})\dot{x}(t,\mathfrak{p}) = A(\mathfrak{p})x(t,\mathfrak{p}) + B(\mathfrak{p})u(t), \\ y(t,\mathfrak{p}) = C(\mathfrak{p})x(t,\mathfrak{p}), \end{cases} \tag{5.27}$$

where $E(\mathfrak{p}), A(\mathfrak{p}) \in \mathbb{R}^{n\times n}$, $B(\mathfrak{p}) \in \mathbb{R}^{n\times m}$, $C(\mathfrak{p}) \in \mathbb{R}^{p\times n}$. The system depends on $\mathfrak{p} = (\mathfrak{p}^1, \ldots, \mathfrak{p}^d) \in \Omega \subset \mathbb{R}^d$ — a set of parameters in a (usually bounded) domain Ω. It holds $u(t) \in \mathbb{R}^m$, $x(t,\mathfrak{p}) \in \mathbb{R}^n$ and $y(t,\mathfrak{p}) \in \mathbb{R}^p$.

The aim of parametric Model Order Reduction (pMOR) is to reduce the system (5.27) while preserving the dependency on the parameters:

$$\hat{\Sigma}_{\text{lin}}(\mathfrak{p}): \quad \begin{cases} \hat{E}(\mathfrak{p})\dot{\hat{x}}(t,\mathfrak{p}) = \hat{A}(\mathfrak{p})\hat{x}(t,\mathfrak{p}) + \hat{B}(\mathfrak{p})u(t), \\ \hat{y}(t,\mathfrak{p}) = \hat{C}(\mathfrak{p})\hat{x}(t,\mathfrak{p}), \end{cases} \tag{5.28}$$

with $\hat{E}(\mathfrak{p}), \hat{A}(\mathfrak{p}) \in \mathbb{R}^{r\times r}$, $\hat{B}(\mathfrak{p}) \in \mathbb{R}^{r\times m}$, $\hat{C}(\mathfrak{p}) \in \mathbb{R}^{p\times r}$, $u(t) \in \mathbb{R}^m$, $\hat{x}(t,\mathfrak{p}) \in \mathbb{R}^r$ and $\hat{y}(t,\mathfrak{p}) \in \mathbb{R}^p$.

For the one/two parameter case, early methods were developed by Weile et al. [67] using moment matching. These methods were transferred to the multiparameter case by Daniel et. al [27], Farle et al. [32] and Feng et al. [33]. After a multivariate Taylor series expansion around the parameter points and frequencies, projection matrices are then calculated using moment matching. However, as the number of parameters increases, the order of the model increases as well which leads to large reduced orders. In addition to this approach, several other interpolation methods for pMOR have been proposed. Baur et al. [9] extend the statements in Sections 5.2.3 and 5.2.4 to parametric systems. Baur and Benner propose to interpolate the systems transfer function [10]. Other methods interpolate the reduced system's matrices. These methods have been developed independently by Panzer et al. [53] and Amsallem et al. [3]. Recent research by Geuss et al. [37] showed that both methods can be seen within the same interpolation framework. These two interpolation methods will be reviewed within this section.

Prior to stating the theory of the interpolation methods, we want to draw attention to a special class of linear parametric systems, having the following special parameter dependency (which we present for $E(\mathfrak{p})$, it is

valid for all other matrices as well):

$$E(\mathfrak{p}) = E_0 + \sum_{j=1}^{M} f_j(\mathfrak{p}) E_j. \tag{5.29}$$

This is called an affine parameter dependency and is convenient in practice, as parameters and matrices are independent. The system matrices can be reduced as follows:

$$\hat{E}(\mathfrak{p}) = W^T E(\mathfrak{p}) V = W^T E_0 V + \sum_{j=1}^{M} f_j(\mathfrak{p}) W^T E_j V. \tag{5.30}$$

The benefit of an affine parameter dependency is that the matrices E_j can be reduced a priori. For a new parameter $\mathfrak{p}_{\text{new}} = (\mathfrak{p}^1_{\text{new}}, \ldots, \mathfrak{p}^d_{\text{new}})$, only the functions f_j need to be evaluated and the reduced matrix $E(\mathfrak{p}_{\text{new}})$ can be easily calculated.
Instead of using interpolation to obtain reduced order models, it is also possible to establish projections V and W that are valid in the whole parameter domain Ω. Often, this is done by concatenating the projections obtained for the reduction in several parameter points:

$$V = [V(\mathfrak{p}_1), \ldots, V(\mathfrak{p}_K)], \quad W = [W(\mathfrak{p}_1), \ldots, W(\mathfrak{p}_K)].$$

Certainly, there might be linearly dependent columns in different $V(\mathfrak{p}_i), V(\mathfrak{p}_j)$ or $W(\mathfrak{p}_i), W(\mathfrak{p}_j)$, which can be eliminated, while finding an orthogonal basis of the overall subspace by means of an SVD. After the SVD-step one obtains $V \in \mathbb{R}^{n \times r^V_{\text{all}}}$ with $r \leq r^V_{\text{all}} \leq rK$ and $W \in \mathbb{R}^{n \times r^W_{\text{all}}}$ with $r \leq r^W_{\text{all}} \leq rK$ depending on the significance of the sampling points $\mathfrak{p}_1, \ldots, \mathfrak{p}_K$. Hence the order of the reduced model for a parameter $\mathfrak{p}_{\text{new}}$ might increase. If r^W_{all} and r^V_{all} are different, for example $r^W_{\text{all}} \geq r^V_{\text{all}}$, one can choose $r_{\text{all}} = r^V_{\text{all}}$, taking only the first r_{all} columns of W. If r^W_{all} is much larger than r^V_{all}, a one-sided projection setting $V = W$ can be tried, as using only the first r_{all} columns of W can lead to a loss of information. In addition, the original model needs to be assembled in the new point $\mathfrak{p}_{\text{new}}$ prior to the reduction (cf. (5.29)) which is then performed in the following way:

$$\hat{E}(\mathfrak{p}_{\text{new}}) = W^T E(\mathfrak{p}_{\text{new}}) V, \qquad \hat{A}(\mathfrak{p}_{\text{new}}) = W^T A(\mathfrak{p}_{\text{new}}) V,$$
$$\hat{B}(\mathfrak{p}_{\text{new}}) = W^T B(\mathfrak{p}_{\text{new}}), \qquad \hat{C}(\mathfrak{p}_{\text{new}}) = C(\mathfrak{p}_{\text{new}}) V.$$

In the case where the parameter dependency is affine (as given in equation (5.29)), it is not necessary to assemble the matrices in the new point $\mathfrak{p}_{\text{new}}$, only the functions f_j need to be evaluated (cf. (5.30)). Hence this method will often be used when an affine parameter dependency is given.

5.3.1. Parametric MOR via interpolation of the systems matrices. In this work, we will focus on the works, where the parametric reduced order models will be interpolated. As recently noted by Geuss et al. [37], the present known methods [53, 3] for the interpolation of reduced order models can be seen within a general framework. We are going to follow Geuss' presentation. It basically consists of four steps:

(1) Sample the parameter space and obtain models in points $\mathfrak{p}_1, \ldots, \mathfrak{p}_K$:

$$\Sigma_{\text{lin}}(\mathfrak{p}_j) \text{ with } E(\mathfrak{p}_j),\ A(\mathfrak{p}_j),\ B(\mathfrak{p}_j),\ C(\mathfrak{p}_j) \text{ for } j = 1, \ldots, K.$$

(2) Calculate reduced order models using techniques from linear MOR (cf. Section 5.2) in points $\mathfrak{p}_1, \ldots, \mathfrak{p}_K$:

$$\hat{\Sigma}_{\text{lin}}(\mathfrak{p}_j) \text{ with } \hat{E}(\mathfrak{p}_j),\ \hat{A}(\mathfrak{p}_j),\ \hat{B}(\mathfrak{p}_j),\ \hat{C}(\mathfrak{p}_j) \text{ for } j = 1, \ldots, K,$$

using projection matrices $V(\mathfrak{p}_j)$ and $W(\mathfrak{p}_j)$.

(3) Adjust the reduced order bases.

(4) Choose the interpolation manifold and the interpolation method to obtain a reduced system $\hat{\Sigma}_{\text{lin}}(\mathfrak{p}_{\text{new}})$.

5.3.1.1. *Adjusting the reduced order bases.* The subspaces $\mathcal{V}_j$ and $\mathcal{W}_j$ spanned by the columns of matrices $V(\mathfrak{p}_j) \in \mathbb{R}^{n\times r}$ and $W(\mathfrak{p}_j) \in \mathbb{R}^{n\times r}$ need to be adjusted, as the different reduced models $\hat{\Sigma}_{\text{lin}}(\mathfrak{p}_j)$ do not lie in the same state space. Hence, one needs to transform the models into the same coordinate system by using matrices $M_j \in \mathbb{R}^{r\times r}$ and $T_j \in \mathbb{R}^{r\times r}$ prior to the interpolation:

$$\begin{aligned}
\overline{E}_j &= M_j^T \hat{E}(\mathfrak{p}_j) T_j, \\
\overline{A}_j &= M_j^T \hat{A}(\mathfrak{p}_j) T_j, \\
\overline{B}_j &= M_j^T \hat{B}(\mathfrak{p}_j), \\
\overline{C}_j &= \hat{C}(\mathfrak{p}_j) T_j, \text{ for } j = 1, \ldots, K.
\end{aligned}$$

First, we will consider the subspaces $\mathcal{V}_j$. After choosing a reference subspace $R_V \in \mathbb{R}^{n\times r}$, state transformations T_j can be calculated such that the reduced states can be transferred to the reference subspace, i.e. $\hat{x}(t,\mathfrak{p}_j) = T_j\overline{x}(t,\mathfrak{p}_j)$ holds. There are three main approaches for the choice of the reference subspace:

- Single reference subspace:
  This first method has been developed by Amsallem et al. [3]. One of the bases $V(\mathfrak{p}_{j_0})$ is chosen as reference:
  $$R_V = V(\mathfrak{p}_{j_0}).$$
  It is not clear for which $j_0 \in \{1,\dots,K\}$ the best interpolated reduced order models will be obtained. A good guess might be the j_0 closest to the interpolation point.
- Non-weighted SVD:
  Following Panzer et al. [53], first an SVD of all given reduced order bases $V(\mathfrak{p}_1)$ to $V(\mathfrak{p}_K)$ needs to be calculated:
  $$U\Sigma Z^T = [V(\mathfrak{p}_1),\dots,V(\mathfrak{p}_K)].$$
  The reference subspace will then be chosen as: $R_V = U(:,1:r)$, the first r columns of U.
- Weighted SVD [53]: The reference subspace will now be calculated as:
  $$U\Sigma Z^T = [\omega_1(\mathfrak{p})V(\mathfrak{p}_1),\dots,\omega_K(\mathfrak{p})V(\mathfrak{p}_K)],$$
  with $R_V = U(:,1:r)$, where $\omega_j(\mathfrak{p})$ are parameter dependent weights. Accordingly, a new reference subspace needs to be calculated for every new parameter. Using this approach, subspaces where the corresponding $\mathfrak{p}_j$ lie near the interpolation point will be "automatically" favoured.

Amsallem et al. [3] and Geuss et al. [37] noted that the matrix T_j can be calculated under the assumption, that the vectors of $\overline{V}(\mathfrak{p}_j) = V(\mathfrak{p}_j)T_j$ and R_V are in good correlation. They make use of the so called Modal Assurance Criterion (MAC):

$$\mathrm{MAC}(u,w) = \frac{|u^T w|^2}{(u^T u)(w^T w)},$$

with vectors $w,u \in \mathbb{R}^n$. Details can be found in [37, 3] and the references therein. In our case, we want the vectors v_j^i, the i-th column of $\overline{V}(\mathfrak{p}_j)$, and

R_V^l, the l-th column of R, to be in good correlation. They are normalized and hence the MAC reduces to:

$$\mathrm{MAC}(v_j^i, R_V^l) = |v_j^i, R_V^l|^2.$$

According to Geuss [37], there are two possibilities for the fulfillment of the MACs.

- Strong fulfillment:
  Assuming good correlation for the corresponding vectors, i.e.

  $$\mathrm{MAC}(v_j^k, R_V^k) = |v_j^k, R_V^k|^2 = 1, k = 1, \ldots, r,$$

  and no correlation between the non corresponding vectors, i.e.

  $$\mathrm{MAC}(v_j^i, R_V^l) = |v_j^i, R_V^l|^2 = 0,\ i \neq l,\ i, l = 1, \ldots, r,$$

  one obtains:

  $$T_j^T V(\mathfrak{p}_j)^T R_V = I_r.$$

  Hence one can choose T_j as:

  $$T_j = (R_V^T V(\mathfrak{p}_j))^{-1}.$$

  Obtained by a different derivation, Panzer et al. [53] use the same matrices T_j for the transformation.
- Weak fulfillment:
  This approach has been developed by Amsallem et al. [3]. Instead of finding a correlation for the whole matrix, only the diagonal elements will be considered. They shall be maximized, given by the following equation:

  $$T_j = \arg\max_{T_j} \mathrm{tr}\left(T_j^T V(\mathfrak{p}_j)^T R_V\right).$$

  A solution to this problem can be obtained by using the SVD of $V(\mathfrak{p}_j)^T R_V = U_j \Sigma_j Z_j^T$ for orthogonal matrices T_j:

  $$\begin{aligned} T_j &= \arg\max_{T_j \in \mathcal{O}_r} \mathrm{tr}\left(T_j^T U_j \Sigma_j Z_j^T\right) \\ &= \arg\max_{T_j \in \mathcal{O}_r} \mathrm{tr}\left(Z_j^T T_j^T U_j \Sigma_j\right), \end{aligned}$$

  where $T_j = U_j Z_j^T$ solves the problem.

We have now given the explanations for the adjustment of the right reduced order bases. For the adjustment of the left reduced order bases Geuss et al. [37] propose to use the dual systems of the reduced order systems and proceed as for the right reduced order bases. Considering the approaches given by Panzer and Amsallem and coworkers [53, 3], they can be incorporated in this framework as well. The following transformation matrices M_j have been proposed:

- A strong fulfillment of the MACs leads to the choice
$$M_j = (R_W^T W(\mathfrak{p}_j))^{-1},$$
with R_W obtained by using one of the three given possibilities given for R_V and using W instead of V.
- A weak fulfillment of the MACs leads to
$$\begin{aligned} M_j &= \arg \max_{M_j \in \mathcal{O}_r} \operatorname{tr}\left(M_j^T W(\mathfrak{p}_j)^T R_W\right) \\ &= U_j Z_j^T, \end{aligned}$$
by using the SVD of $W(\mathfrak{p}_j)^T R_W$.
- Panzer et al. [53] propose to use $R_W = R_V$ and hence obtain $M_j = (R_V^T W(\mathfrak{p}_j))^{-1}$.
- In the approach by Amsallem et al. [3] an adjustment of the left subspaces is not given. However, the obtained reduced order models can be multiplied by $\hat{E}(\mathfrak{p}_j)^{-1}$ which will lead to the choice
$$M_j = \hat{E}(\mathfrak{p}_j)^{-T} = \left(V(\mathfrak{p}_j)^T (E(\mathfrak{p}_j))^T W(\mathfrak{p}_j)\right)^{-1},$$
where the reference subspace is given by $R_W = E(\mathfrak{p}_j)V(\mathfrak{p}_j)$.

Manifold	$\mathbb{R}^{q_1 \times q_2}$	Nonsingular matrices
$\operatorname{Exp}_X(\Gamma)$	$X + \Gamma$	$\exp(\Gamma)X$
$\operatorname{Log}_X(Y)$	$Y - X$	$\log(YX^{-1})$

Table 5.1. Exponential and logarithm mappings for different manifolds.

5.3.1.2. *Choosing the interpolation manifold.* After the adjustment of the bases, it remains to interpolate the transformed matrices $\overline{E}_j, \overline{A}_j, \overline{B}_j$, and $\overline{C}_j$. Amsallem et al. [3] propose to interpolate on tangential spaces of a certain matrix manifold $\mathcal{M}$. For a reference point $X \in \mathcal{M}$, the exponential mapping

$$\mathrm{Exp}_X : \mathcal{T}_X\mathcal{M} \to \mathcal{M} \tag{5.31}$$

and the logarithm mapping

$$\mathrm{Log}_X : \mathcal{M} \supset \mathcal{U}_X \to \mathcal{T}_X\mathcal{M} \tag{5.32}$$

define the connection between a manifold and a tangential space. In our case, two different manifolds will be considered. The first is the manifold of the real matrices with k rows and l columns: $\mathbb{R}^{k\times l}$. The second is the one of nonsingular matrices in $\mathbb{R}^{k\times k}$. The definitions for the exponential and the logarithm mapping can be found in Table 5.1. The maps *exp* and *log* are the matrix exponential and logarithm, respectively. After choosing one reference model from all the transformed reduced models, the remaining models will be interpolated in the tangential space with respect to the reference model. Hence, for a fixed reference matrix $\overline{A}(\mathfrak{p}_{\ell_0})$, the other matrices need to be mapped to the tangential space $\mathcal{T}_{\overline{A}(\mathfrak{p}_{\ell_0})}\mathcal{M}$ by the logarithm mapping: $\Gamma_j = \mathrm{Log}_{\overline{A}(\mathfrak{p}_{\ell_0})}(\overline{A}(\mathfrak{p}_j))$. The obtained Γ_j will now be interpolated using a suitable interpolation method which leads to the matrix $\Gamma_{\text{new}} \subset \mathcal{T}_{\overline{A}(\mathfrak{p}_{\ell_0})}\mathcal{M}$ for a parameter sample $\mathfrak{p}_{\text{new}}$. This matrix is transformed to the manifold $\mathcal{M}$ using the exponential mapping and gives $\overline{A}(\mathfrak{p}_{\text{new}})$.
In contrast to Amsallem et al. [3], Panzer et al. [53] however simply interpolate the matrices without mapping the matrices on tangential manifolds. In Chapter 8, we are going to compare different approaches using this framework and apply them to our bilinear systems (cf. Section 8.2):

- We follow Amsallem et al. [3]: Use a fixed reference subspace and obtain T_j by a weak fulfillment of the MACs and M_j by inversion of $\hat{E}(\mathfrak{p}_j)$.
- As given by Panzer et al. [53], we use a reference subspace given by a (weighted) SVD of all underlying matrices $V(\mathfrak{p}_j)$, and obtain $T_j = (R_V^T V(\mathfrak{p}_j))^{-1}$ and $M_j = (R_V^T W(\mathfrak{p}_j))^{-1}$.

5.3.2. Parametric systems as bilinear systems. For parametric models with a special affine parameter dependency, it is possible to transform them

into bilinear models. This transformation was originally given by Breiten and Benner in [11].
Consider the following affine parametric system:

$$\Sigma_{\text{lin}}(\mathfrak{p}): \quad \begin{cases} E\dot{x}(t,\mathfrak{p}) = \left(A + \sum_{k=1}^{\overline{m}} f(\mathfrak{p}_k)A_k\right)x(t,\mathfrak{p}) + \widetilde{B}\widetilde{u}(t), \\ y(t,\mathfrak{p}) = Cx(t,\mathfrak{p}), \end{cases} \tag{5.33}$$

with $E, A, A_i \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{p\times n}$, $\widetilde{B} \in \mathbb{R}^{n\times\widetilde{m}}$. Define $N_k = A_k$ for $k = 1,\dots,\overline{m}$ and $N_k = 0$ for $k = \overline{m}+1,\dots,\overline{m}+\widetilde{m}$. In addition let $m := \overline{m}+\widetilde{m}$, and let the first $\overline{m}$ columns of the new B be zero. For the columns $\overline{m}+1$ to m use the matrix $\widetilde{B}$. Finally, set $u(t) = \begin{bmatrix} f(\mathfrak{p}_1) & \dots & f(\mathfrak{p}_{\overline{m}}) & \widetilde{u}(t) \end{bmatrix}^T$. The steps above result in a bilinear system:

$$\Sigma_{\text{bil}}: \quad \begin{cases} E\dot{x}(t) = Ax(t) + \sum_{k=1}^{m} N_k u_k(t)x(t) + Bu(t), \\ y(t) = Cx(t). \end{cases} \tag{5.34}$$

The transformation of such parametric models results in bilinear models, where all parameters can be seen as inputs. Bilinear Model Order Reduction needs to be applied for the reduction, which is now “parameter free", as in contrast to the methods for parametric model order reduction which have been discussed in the previous sections, there is no interpolation procedure needed to obtain parametric reduced order models, as it is not necessary to consider the newly obtained inputs in the reduction process. The linear parametric models given by a physical parametrization (cf. equation (4.2)) of the electrical motor model have exactly the structure of (5.33) and hence bilinear model order reduction methods can be applied to obtain a parametric reduced order model.
However, constant inputs u_k (as resulting from parametric systems) are not $\mathcal{L}_2^m$ functions (as the integrals $\int_{-\infty}^{\infty} u_k^2 d\omega$ do not exist) and hence strictly speaking not admissible input functions. During the reduction, the system is reduced without “knowing" anything about the inputs. A good reduced order model can hence be calculated using bilinear reduction methods. In addition, the condition for BIBO-Stability (cf. Theorem 2.3.24) can be fulfilled for constant inputs as well.

5.4. Bilinear Model Order Reduction

The reduction of bilinear systems as given by equation (2.30) (or (5.34)) obtained attention within the last 20 years. The methods developed for linear systems can often be transferred to bilinear systems.

Throughout this Section, we assume the bilinear systems (2.30) to be reachable, observable and BIBO stable. In addition, we assume the existence of the Gramians of the system, and only systems with E nonsingular will be considered.

5.4.1. The error system. As in the linear case, we need to quantify the quality of the approximation. Hence, the error between the original and the reduced order model needs to be measured. The error system is defined as follows:

$$\Sigma_{\text{bil}}^{\text{err}}: \quad \begin{cases} \begin{bmatrix} E & 0 \\ 0 & \hat{E} \end{bmatrix} \begin{bmatrix} \dot{x}(t) \\ \dot{\hat{x}}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & \hat{A} \end{bmatrix} \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix} + \sum_{k=1}^{m} \begin{bmatrix} N_k & 0 \\ 0 & \hat{N}_k \end{bmatrix} \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix} u_k + \begin{bmatrix} B \\ \hat{B} \end{bmatrix} u(t), \\ y(t) - \hat{y}(t) = \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix}. \end{cases} \tag{5.35}$$

The reachability Gramian of the error system $P^{\text{err}} = \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix}$ satisfies the following generalized Lyapunov equation:

$$\begin{bmatrix} A & \hat{A} \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \begin{bmatrix} E^T & \hat{E}^T \end{bmatrix} + \begin{bmatrix} E & \hat{E} \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \begin{bmatrix} A^T & \hat{A}^T \end{bmatrix} + \sum_{k=1}^{m} \begin{bmatrix} N_k & \hat{N}_k \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \begin{bmatrix} N_k^T & \hat{N}_k^T \end{bmatrix} + \begin{bmatrix} B \\ \hat{B} \end{bmatrix} \begin{bmatrix} B^T & \hat{B}^T \end{bmatrix} = 0. \tag{5.36}$$

Using the observability Gramian $Q^{\text{err}} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix}$ one obtains that

$$Y^{\text{err}} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} = \begin{bmatrix} E^{-T} & \hat{E}^{-T} \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix} \begin{bmatrix} E^{-1} & \hat{E}^{-1} \end{bmatrix}, \tag{5.37}$$

satisfies the following Lyapunov equation:

$$\begin{bmatrix} A^T & \\ & \hat{A}^T \end{bmatrix} \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} \begin{bmatrix} E & \\ & \hat{E} \end{bmatrix} + \begin{bmatrix} E^T & \\ & \hat{E}^T \end{bmatrix} \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} \begin{bmatrix} A & \\ & \hat{A} \end{bmatrix} + \sum_{k=1}^{m} \begin{bmatrix} N_k^T & \\ & \hat{N}_k^T \end{bmatrix} \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} \begin{bmatrix} N_k & \\ & \hat{N}_k \end{bmatrix} + \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} C^T \\ -\hat{C}^T \end{bmatrix} = 0. \tag{5.38}$$

The $\mathscr{H}_2$-norm of the error system will now be used to measure the error between the original and the reduced order model. Using the error system Gramians this can be done in the following way:

$$\begin{aligned} ||\Sigma_{\text{bil}}^{\text{err}}||_{\mathscr{H}_2} = ||\Sigma_{\text{bil}} - \hat{\Sigma}_{\text{bil}}||_{\mathscr{H}_2} &= \sqrt{\text{tr}\left(\begin{bmatrix} C & -\hat{C} \end{bmatrix} P^{\text{err}} \begin{bmatrix} C^T \\ -\hat{C}^T \end{bmatrix}\right)} \\ &= \sqrt{\text{tr}\left(\begin{bmatrix} B^T & \hat{B}^T \end{bmatrix} Q^{\text{err}} \begin{bmatrix} B \\ \hat{B} \end{bmatrix}\right)}. \end{aligned} \tag{5.39}$$

In addition, using the definition of the $\mathscr{H}_2$-norm given by Benner and Breiten as in (2.47), the norm of the corresponding error system can hence be given as:

$$\begin{aligned} \mathcal{J} &= ||\Sigma_{\text{bil}}^{\text{err}}||_{\mathscr{H}_2}^2 \\ &= \text{vec}(I_{2p})^T (\begin{bmatrix} C & -\hat{C} \end{bmatrix} \otimes \begin{bmatrix} C & -\hat{C} \end{bmatrix}) \\ &\times \left(- \begin{bmatrix} A & \\ & \hat{A} \end{bmatrix} \otimes \begin{bmatrix} E & \\ & \hat{E} \end{bmatrix} - \begin{bmatrix} E & \\ & \hat{E} \end{bmatrix} \otimes \begin{bmatrix} A & \\ & \hat{A} \end{bmatrix} - \sum_{k=1}^{m} \begin{bmatrix} N_k & \\ & \hat{N}_k \end{bmatrix} \otimes \begin{bmatrix} N_k & \\ & \hat{N}_k \end{bmatrix} \right)^{-1} \\ &\times \left(\begin{bmatrix} B \\ \hat{B} \end{bmatrix} \otimes \begin{bmatrix} B \\ \hat{B} \end{bmatrix} \right) \text{vec}(I_{2m}). \end{aligned} \tag{5.40}$$

5.4.2. Bilinear Balanced Truncation. Already in 1993, Al-Baiyat and Bettayeb [2] applied balancing methods to special (so called k-power) bilinear systems. Recent results have been obtained by Hartmann et al. [42]. As given in Section 2.3.2.3, the bilinear Gramians can be decomposed as

$$P = RR^T \text{ and } Q = L^T L.$$

By using the singular value decompositon of

$$LER = U_b \Sigma V_b^T,$$

one obtains

$$W_b^T E T_b,\ W_b^T A T_b,\ W_b^T N_k T_b,\ W_b^T B,\ C T_b,$$

where

$$W_b = L^T U_b \Sigma^{-1/2}, \quad T_b = R V_b \Sigma^{-1/2}, \quad W_b^{-1} = T_b^T E^T, \quad T_b^{-1} = W_b^T E.$$

If the Hankel singular values given by $\Sigma = \mathrm{diag}(\varsigma_1, \ldots, \varsigma_n)$ show a decay and $\varsigma_{d+1} \ll \varsigma_d$ holds, one can approximate the original model by using

$$W = L^T U_1 \Sigma_1^{-1/2}, \quad T = R V_1 \Sigma_1^{-1/2},$$

with

$$LER = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix},$$

$U_1, V_1 \in \mathbb{R}^{n \times r}$, $U_2, V_2 \in \mathbb{R}^{n \times (n-r)}$ having orthogonal columns and $\Sigma_1 = \mathrm{diag}(\varsigma_1, \ldots \varsigma_d)$, $\Sigma_2 = \mathrm{diag}(\varsigma_{d+1}, \ldots \varsigma_n)$.

5.4.3. Bilinear Krylov Subspace Methods. Model Order Reduction for bilinear systems via Krylov subspaces has been examined by several researchers such as Philipps [54], Condon and Ivanov [23], Breiten and Damm [17], Bai and Skoogh [8], and Lin and coworkers [45]. Moment matching can be achieved by series expansions of the multivariate transfer functions as given in (2.37). For ease of presentation, we assume $E = I_n$ throughout the following section. A multimoment can be defined as:

Definition 5.4.1 ([45],[34])**.** Let Σ_{bil} be a bilinear system as given in (2.30). For nonnegative integers $m_1, \ldots, m_i$, a *multimoment* $\mathcal{H}_i^{(m_1,\ldots,m_i)}(s_1, \ldots, s_i)$ of the transfer function $H_i(s_1, \ldots, s_i)$ as given in (2.37) is defined as

$$\begin{aligned} \mathcal{H}_i^{(m_1,\ldots,m_i)}(s_1, \ldots, s_i) =& (-1)^i C(s_i I_n - A)^{-m_i} \mathbf{N} [I_m \otimes (s_{i-1} I_n - A)^{-m_{i-1}} \mathbf{N}] \ldots \\ & \cdot [\underbrace{I_m \otimes \cdots \otimes I_m}_{i-2 \text{ times}} \otimes (s_2 I_n - A)^{-m_2} \mathbf{N}] \\ & \cdot [\underbrace{I_m \otimes \cdots \otimes I_m}_{i-1 \text{ times}} \otimes (s_1 I_n - A)^{-m_1} B], \end{aligned} \tag{5.41}$$

where $\mathbf{N} = [N_1 \quad \ldots \quad N_m]$.

To ensure moment matching, Krylov subspaces (cf. 5.2.2) need to be built. Often (see f.e. [8, 45, 17]), the following Krylov subspaces are used for moment matching around $s = 0$:

$$\text{span}(V^{(1)}) = \mathcal{K}_q(A^{-1}, A^{-1}B),$$
$$\text{span}(V^{(i)}) = \bigcup_{k=1}^{m} \mathcal{K}_q(A^{-1}, A^{-1}N_k V^{(i-1)}),$$
$$\text{span}(V) = \text{span}\left(\bigcup_{i=1}^{r} \text{span}(V^{(i)})\right).$$

Moment matching in points other than the origin can be guaranteed by the following result given by Flagg [34]:

Theorem 5.4.2 ([34], Subsystem Interpolation)**.** *Let* $\{\xi_j\}_{j=1}^k$, $\{\zeta_j\}_{j=1}^k \subset \mathbb{C}$ *and vectors* $c^T \in \mathbb{C}^p$ *and* $b \in \mathbb{C}^m$ *be given. Define* $\mathbf{b}_j = \mathbf{1}_j \otimes b$ *and* $\mathbf{N}^{\oplus T} = \left[N_1^T, \ldots, N_m^T\right]$ *where* $\mathbf{1}_j$ *is a column of* m^{j-1} *ones. To construct a reduced order system that matches all the multimoments* $\mathcal{H}_j^{(l_1,\ldots,l_j)}(\xi_1, \ldots, \xi_j)\mathbf{b}_j$ *and* $c\mathcal{H}_j^{(l_1,\ldots,l_j)}(\zeta_j, \ldots, \zeta_1)$ *for* $j = 1, \ldots, k$ *and* $l_1, \ldots, l_j = 1, \ldots, q$*, construct the matrices* V *and* W *as follows:*

$$\text{span}(V^{(1)}) = \mathcal{K}_q\{(\xi_1 I - A)^{-1}, (\xi_1 I - A)^{-1}Bb\},$$
$$\text{span}(W^{(1)}) = \mathcal{K}_q\{(\zeta_1 I - A)^{-*}, (\zeta_1 I - A)^{-*}C^* c^*\},$$
$$\text{span}(V^{(j)}) = \mathcal{K}_q\{(\xi_j I - A)^{-1}, (\xi_j I - A)^{-1}\mathbf{N}(I_m \otimes V^{(j-1)})\} \text{ for } j = 2, \ldots, k,$$
$$\text{span}(W^{(j)}) = \mathcal{K}_q\{(\zeta_j I - A)^{-*}, (\zeta_j I - A)^{-*}\mathbf{N}^{\oplus T}(I_m \otimes W^{(j-1)})\} \text{ for } j = 2, \ldots, k,$$
$$\text{span}(V) = \text{span}\{\bigcup_{j=1}^{k} \text{span}(V^{(j)})\},$$
$$\text{span}(W) = \text{span}\{\bigcup_{j=1}^{k} \text{span}(W^{(j)})\}.$$

Provided $\tilde{W}^T = (W^T V)^{-1} W^T$ *is defined, the reduced system* $\hat{A} = \tilde{W}^T A V$*,* $\hat{N}_k = \tilde{W}^T N_k V$*,* $\hat{C} = CV$ *and* $\hat{B} = \tilde{W}^T B$ *satisfies:*

$$\mathcal{H}_j^{(l_1,\ldots,l_j)}(\xi_1, \ldots, \xi_j)\mathbf{b}_j = \hat{\mathcal{H}}^{(l_1,\ldots,l_j)}(\xi_1, \ldots, \xi_j)\mathbf{b}_j$$

and

$$c\mathcal{H}_j^{(l_1,\ldots,l_j)}(\zeta_1, \ldots, \zeta_j) = c\hat{\mathcal{H}}^{(l_1,\ldots,l_j)}(\zeta_1, \ldots, \zeta_j)$$

for $j = 1, \ldots, k$ *and* $l_1, \ldots, l_k = 1, \ldots, q$*.*

Using this moment matching of multimoments would involve a strategy for finding points $\{\xi_j\}_{j=1}^k$, $\{\zeta_j\}_{j=1}^k \subset \mathbb{C}$ and vectors $c^T \in \mathbb{C}^p$ and $b \in \mathbb{C}^m$ such that the reduced model delivers a good approximation to the original model. The advantage of this approach is that it does not depend on the convergence of the underlying Volterra series, which might not be known a priori (cf. the definition of BIBO stability and the convergence of the Volterra series given in Section 2.3.2). In addition to the moment matching approach, one might think of the interpolation of the multivariate transfer functions $H_i(s_1, \ldots, s_i)$, or — in other words — the interpolation of the Volterra series. This approach has been examined by Flagg [34] in his dissertation and resulted in a derivation of interpolation conditions for the Volterra series representation of a bilinear system. Flagg was able to establish a connection between Volterra series interpolation and the results concerning the $\mathscr{H}_2$-optimal conditions for bilinear systems recently derived by Zhang and Lam [72] and Benner and Breiten [12].

5.5. $\mathscr{H}_2$ - optimal bilinear Model Order Reduction

As in the linear case, one is interested in $\mathscr{H}_2$-optimal bilinear MOR. Within this section, necessary $\mathscr{H}_2$-optimality conditions for bilinear systems are obtained by deriving the $\mathscr{H}_2$-norm (5.39) of the error system (5.35). First, the bilinear Wilson conditions originally obtained by Zhang and Lam [72] will be derived. Using a different approach, Benner and Breiten [12] obtained the **B**ilinear **I**nterpolatory **R**ational **K**rylov **A**lgorithm (BIRKA), a generalization to bilinear systems of the linear IRKA (Algorithm 1). In addition, we will derive a new $\mathscr{H}_2$-optimal algorithm relying on optimization on Grassmann manifolds, which is a generalization of the methods given in the linear case by Yan and Lam [69] and Xu and Zeng [68].
As the Finite Element Discretisation of industrial models leads to systems with $E \neq I_n$, we need to incorporate E in our derivation. We can not simply invert the matrix E as due to their large dimension, the inversion would be numerically expensive or even impossible. Hence, we will derive optimality conditions for systems with $E \neq I_n$, E nonsingular, which have not been stated elsewhere. All systems will be assumed to be reachable, observable, BIBO stable and the Gramians shall exist.

5.5.1. Wilson conditions for bilinear systems. Defining $\mathcal{C} = \left[\begin{smallmatrix} C^T \\ -\hat{C}^T \end{smallmatrix}\right] \left[\begin{smallmatrix} C & -\hat{C} \end{smallmatrix}\right]$, the norm of the error system can be given as:

$$\mathcal{J} = ||\Sigma_{\text{bil}}^{\text{err}}||_{\mathscr{H}_2}^2 = \operatorname{tr}\left(\left[\begin{smallmatrix} C & -\hat{C} \end{smallmatrix}\right] P^{\text{err}} \left[\begin{smallmatrix} C^T \\ -\hat{C}^T \end{smallmatrix}\right]\right) = \operatorname{tr}\left(P^{\text{err}}\mathcal{C}\right). \tag{5.42}$$

By differentiating the norm (5.42) and using the Lyapunov equations (5.36) and (5.38) we obtain the following conditions (for a detailed derivation see Appendix A.1):

$$\begin{aligned}
\hat{E} &= -Y_{22}^{-1}Y_{12}^T E P_{12} P_{22}^{-1}, && (5.43)\\
\hat{A} &= -Y_{22}^{-1}Y_{12}^T A P_{12} P_{22}^{-1}, && (5.44)\\
\hat{N}_k &= -Y_{22}^{-1}Y_{12}^T N_k P_{12} P_{22}^{-1}, \text{ for } k = 1, \ldots, m, && (5.45)\\
\hat{B} &= -Y_{22}^{-1}Y_{12}^T B, && (5.46)\\
\hat{C} &= C P_{12} P_{22}^{-1}, && (5.47)
\end{aligned}$$

with Y_{ij} as given in (5.37) and P_{ij} as in (5.36). This leads to the following theorem:

Theorem 5.5.1 ([72]). *If the reduced system $\hat{\Sigma}_{\text{bil}}$, which is reachable and observable, is an $\mathscr{H}_2$-optimal reduced order model for the system Σ_{bil} and the reachability and observability Gramians P^{err} and Q^{err} exist, then there exist matrices $W, V \in \mathbb{R}^{n\times r}$ such that*

$$\hat{E} = W^T E V,\ \hat{A} = W^T A V,\ \hat{N}_k = W^T N_k V,\ \hat{B} = W^T B,\ \hat{C} = CV. \tag{5.48}$$

They can be obtained by equations (5.43) to (5.44) as $W := -Y_{12}Y_{22}^{-1}$ and $V := P_{12}P_{22}^{-1}$.

Remark 5.5.2. Inserting the observability Gramian Q^{err} in the equations leads to the projections for the system multiplied by E^{-1}:

$$\begin{aligned}
\hat{E} &= -Y_{22}^{-1}Y_{12}^T E P_{12} P_{22}^{-1}\\
&= -\hat{E}Q_{22}^{-1}\hat{E}^T\hat{E}^{-T}Q_{12}^T E^{-1} E P_{12} P_{22}^{-1},\\
&\Rightarrow\ I_r = -Q_{22}^{-1}Q_{12}^T P_{12} P_{22}^{-1},\\
\hat{A} &= -Y_{22}^{-1}Y_{12}^T A P_{12} P_{22}^{-1}\\
&= -\hat{E}Q_{22}^{-1}\hat{E}^T\hat{E}^{-T}Q_{12}^T E^{-1} A P_{12} P_{22}^{-1},\\
&\Rightarrow\ \hat{E}^{-1}\hat{A} = -Q_{22}^{-1}Q_{12}^T E^{-1} A P_{12} P_{22}^{-1},
\end{aligned}$$

with analogue calculations for N_k, B and C.

5.5.2. The optimality conditions derived by Benner and Breiten. As in the case of the Wilson conditions, Benner and Breiten deduce the optimality conditions by differentiating the $\mathscr{H}_2$-norm of the error system (5.40). In contrast to their derivation, we need to consider $E \neq I_n$, E nonsingular. The obtained reduced system can be written as $(\hat{A}, \hat{N}_k, \hat{B}, \hat{C})$ after multiplying with $\hat{E}^{-1}$ from the left, and hence we will assume $\hat{E} = I_r$. In addition, we assume that $\hat{A}$ is diagonalizable.
It is possible to rewrite the representation of the $\mathscr{H}_2$-norm as given in (5.40) by using:

$$\hat{A} = S\Lambda S^{-1},\ \tilde{B}^T = S^{-1}\hat{B},\ \tilde{C} = \hat{C}S,\ \tilde{N}_k^T = S^{-1}(\hat{N})_k S,$$

which leads to:

$$\begin{aligned}
\mathcal{J} &= ||\Sigma_{\text{bil}}^{\text{err}}||_{\mathscr{H}_2}^2 \\
&= \text{vec}(I_{2p})^T([\,C\ -\tilde{C}\,] \otimes [\,C\ -\tilde{C}\,]) \\
&\times \left(-\begin{bmatrix} A & \\ & \Lambda \end{bmatrix} \otimes \begin{bmatrix} E & \\ & I_r \end{bmatrix} - \begin{bmatrix} E & \\ & I_r \end{bmatrix} \otimes \begin{bmatrix} A & \\ & \Lambda \end{bmatrix} - \sum_{k=1}^{m} \begin{bmatrix} N_k & \\ & \tilde{N}_k^T \end{bmatrix} \otimes \begin{bmatrix} N_k & \\ & \tilde{N}_k^T \end{bmatrix} \right)^{-1} \\
&\times \left(\begin{bmatrix} B \\ \tilde{B}^T \end{bmatrix} \otimes \begin{bmatrix} B \\ \tilde{B}^T \end{bmatrix} \right) \text{vec}(I_{2m}).
\end{aligned} \tag{5.49}$$

Derivations with respect to the eigenvalues of the reduced system $\Lambda = \text{diag}(\hat{\lambda}_1, \ldots, \hat{\lambda}_r)$ and the matrices $\tilde{N}_k, \tilde{B}$, and $\tilde{C}$ lead to the following optimality conditions (their derivation can be found in Appendix A.2):

$$\begin{aligned}
&\text{vec}(I_p)^T(\tilde{C} \otimes C) \left(-I_r \otimes A - \Lambda \otimes E - \sum_{k=1}^{m} \tilde{N}_k^T \otimes N_k \right)^{-1} \\
&(e_i e_i^T \otimes E) \left(-I_r \otimes A - \Lambda \otimes E - \sum_{k=1}^{m} \tilde{N}_k^T \otimes N_k \right)^{-1} (\tilde{B}^T \otimes B)\text{vec}(I_m) \\
&= \text{vec}(I_p)^T(\tilde{C} \otimes \hat{C}) \left(-I_r \otimes \hat{A} - \Lambda \otimes I_r - \sum_{k=1}^{m} \tilde{N}_k^T \otimes \hat{N}_k \right)^{-1} \\
&(e_i e_i^T \otimes I_r) \left(-I_r \otimes \hat{A} - \Lambda \otimes I_r - \sum_{k=1}^{m} \tilde{N}_k^T \otimes \hat{N}_k \right)^{-1} (\tilde{B}^T \otimes \hat{B})\text{vec}(I_m),
\end{aligned} \tag{5.50}$$

$$\begin{aligned}
&\mathrm{vec}(I_p)^T(\tilde{C}\otimes C)\left(-I_r\otimes A-\Lambda\otimes E-\sum_{k=1}^{m}\tilde{N}_k{}^T\otimes N_k\right)^{-1}\\
&(e_ie_j^T\otimes N)\left(-I_r\otimes A-\Lambda\otimes E-\sum_{k=1}^{m}\tilde{N}_k{}^T\otimes N_k\right)^{-1}(\tilde{B}^T\otimes B)\mathrm{vec}(I_m)\\
&=\mathrm{vec}(I_p)^T(\tilde{C}\otimes \hat{C})\left(-I_r\otimes \hat{A}-\Lambda\otimes I_r-\sum_{k=1}^{m}\tilde{N}_k{}^T\otimes \hat{N}_k\right)^{-1}\\
&(e_ie_j^T\otimes \hat{N})\left(-I_r\otimes \hat{A}-\Lambda\otimes I_r-\sum_{k=1}^{m}\tilde{N}_k{}^T\otimes \hat{N}_k\right)^{-1}(\tilde{B}^T\otimes \hat{B})\mathrm{vec}(I_m),
\end{aligned}\tag{5.51}$$

$$\begin{aligned}
&\mathrm{vec}(I_p)^T(\tilde{C}\otimes C)\left(-I_r\otimes A-\Lambda\otimes E-\sum_{k=1}^{m}\tilde{N}_k{}^T\otimes N_k\right)^{-1}\\
&\qquad\cdot(e_je_i^T\otimes B)\mathrm{vec}(I_m)\\
&=\mathrm{vec}(I_p)^T(\tilde{C}\otimes \hat{C})\left(-I_r\otimes \hat{A}-\Lambda\otimes I_r-\sum_{k=1}^{m}\tilde{N}_k{}^T\otimes \hat{N}_k\right)^{-1}\\
&\qquad\cdot(e_je_i^T\otimes \hat{B})\mathrm{vec}(I_m),
\end{aligned}\tag{5.52}$$

$$\begin{aligned}
&\mathrm{vec}(I_p)^T(e_ie_j^T\otimes C)\left(-I_r\otimes A-\Lambda\otimes E-\sum_{k=1}^{m}\tilde{N}_k{}^T\otimes N_k\right)^{-1}\\
&\qquad\cdot(\tilde{B}^T\otimes B)\mathrm{vec}(I_m)\\
&=\mathrm{vec}(I_p)^T(e_ie_j^T\otimes \hat{C})\left(-I_r\otimes \hat{A}-\Lambda\otimes I_r-\sum_{k=1}^{m}\tilde{N}_k{}^T\otimes \hat{N}_k\right)^{-1}\\
&\qquad\cdot(\tilde{B}^T\otimes \hat{B})\mathrm{vec}(I_m).
\end{aligned}\tag{5.53}$$

The following theorem shows the connection between an optimal reduced order model and the conditions (5.50) — (5.53).

Theorem 5.5.3 ([12]). *Let Σ_{bil} denote a BIBO stable bilinear system. Assume that $\hat{\Sigma}_{\text{bil}}$ is a reduced bilinear system of order r that minimizes the $\mathscr{H}_2$-norm of the error system among all other bilinear systems of dimension r. Then, $\hat{\Sigma}_{\text{bil}}$ fulfills the conditions (5.50) — (5.53).*

5.5.3. Algorithms resulting from the $\mathscr{H}_2$-optimality conditions. Now it is possible to obtain two different algorithms for the calculation of bilinear optimal reduced order models. First, as seen in the context of the Wilson conditions, optimal models can be obtained by using $W = -Y_{12}Y_{22}^{-1}$ and $V = P_{12}P_{22}^{-1}$ (cf. Theorem 5.5.1). Hence it holds $\text{span}(Y_{12}) \subset W$ and $\text{span}(P_{12}) \subset V$. It is sufficient to determine Y_{12} and P_{12} which can be done by solving Sylvester equations obtained by splitting the equations (5.36) and (5.38). This leads to the following algorithm (for a more detailed insight we refer to the derivation of Benner and Breiten [12]):

Algorithm 2 Generalized Sylvester iteration (cf. [12]).

Input: $E, A, N_k, B, C, \hat{E}, \hat{A}, \hat{N}_k, \hat{B}, \hat{C}$
Output: $\hat{E}^{\text{opt}}, \hat{A}^{\text{opt}}, \hat{N}_k^{\text{opt}}, \hat{B}^{\text{opt}}\hat{C}^{\text{opt}}$

1: **while** not converged **do**
2: Solve

$$A X \hat{E}^T + E X \hat{A}^T + \sum_{k=1}^{m} N_k X \hat{N}_k + B\hat{B}^T = 0 \tag{5.54}$$

3: Solve

$$A^T Y \hat{E} + E^T Y \hat{A} + \sum_{k=1}^{m} N_k Y \hat{N}_k - C^T \hat{C} = 0 \tag{5.55}$$

4: $V = \text{orth}(X)$, $W = \text{orth}(Y)$ % orth computes an orthonormal basis
5: $\hat{E} = W^T E V$, $\hat{A} = W^T A V$, $\hat{N}_k = W^T N_k V$, $\hat{B} = W^T B$,
6: **end while**
7: $\hat{E}^{\text{opt}} = \hat{E}$, $\hat{A}^{\text{opt}} = \hat{A}$, $\hat{N}_k^{\text{opt}} = \hat{N}_k$, $\hat{B}^{\text{opt}} = \hat{B}$, $\hat{C}^{\text{opt}} = \hat{C}$

Theorem 5.5.4 ([12]). *If Algorithm 2 converges, then* $\hat{E}^{\text{opt}}$, $\hat{A}^{\text{opt}}$, $\hat{N}_k^{\text{opt}}$, $\hat{B}^{\text{opt}}$ *and* $\hat{C}^{\text{opt}}$ *fulfill the Wilson optimality conditions (5.43)-(5.47).*

Proof. The proof of this Theorem can be found in the Appendix A.3. □

As we derived the optimality conditions according to Breiten and Benner [12] by using reduced systems assuming $\hat{E} = I_r$, we obtain for the solution of the bilinear Sylvester equations (5.54) and (5.55):

$$\begin{aligned}
\text{vec}(X) &= \left(-I_r \otimes A - \hat{A} \otimes E - \sum_{k=1}^{m} \hat{N}_k \otimes N_k\right)^{-1} \text{vec}(B\hat{B}^T) \\
&= \left(-SS^{-1} \otimes A - S\Lambda S^{-1} \otimes E - \sum_{k=1}^{m} S\tilde{N}_k^T S^{-1} \otimes N_k\right)^{-1} (\hat{B} \otimes B)\text{vec}(I_m) \\
&= \left((S \otimes I_n)\left(-I_r \otimes A - \Lambda \otimes E - \sum_{k=1}^{m} \tilde{N}_k^T \otimes N_k\right)\left(S^{-1} \otimes I_n\right)\right)^{-1} (\hat{B} \otimes B)\text{vec}(I_m) \\
&= (S \otimes I_n) \underbrace{\left(-I_r \otimes A - \Lambda \otimes E - \sum_{k=1}^{m} \tilde{N}_k^T \otimes N_k\right)^{-1} (\tilde{B}^T \otimes B)\text{vec}(I_m)}_{\text{vec}(V)},
\end{aligned}$$

and

$$\begin{aligned}
\text{vec}(Y) &= \left(I_r^T \otimes A^T + \hat{A}^T \otimes E^T + \sum_{k=1}^{m} \hat{N}_k^T \otimes N_k^T\right)^{-1} (\hat{C}^T \otimes C^T)\text{vec}(I_p) \\
&= \left(S^{-T}S^T \otimes A^T + S^{-T}\Lambda S^T \otimes E^T + \sum_{k=1}^{m} S^{-T}\tilde{N}_k S^T \otimes N_k^T\right)^{-1} (\hat{C}^T \otimes C^T)\text{vec}(I_p) \\
&= \left(-S^{-T} \otimes I_n\right)\left(-I_r \otimes A^T - \Lambda \otimes E^T - \sum_{k=1}^{m} \tilde{N}_k \otimes N_k^T\right)^{-1} \left(S^T \otimes I_n\right)(\hat{C}^T \otimes C^T)\text{vec}(I_p) \\
&= \left(-S^{-T} \otimes I_n\right) \underbrace{\left(-I_r \otimes A^T - \Lambda \otimes E^T - \sum_{k=1}^{m} \tilde{N}_k \otimes N_k^T\right)^{-1} (\tilde{C}^T \otimes C^T)\text{vec}(I_p)}_{\text{vec}(W)},
\end{aligned}$$

This leads to the fact that $\text{span}(X) \subset V$ and $\text{span}(Y) \subset W$. Instead of solving the Sylvester equations as given in (5.54) and (5.55), we can use the vectorized form of the Sylvester equations to calculate an optimal reduced model, which leads to Algorithm 3.

Algorithm 3 Bilinear IRKA for systems with $E \neq I$, E nonsingular (cf. [12]).

Input: $E, A, N_k, B, C, \hat{A}, \hat{N}_k, \hat{B}, \hat{C}$
Output: $\hat{A}^{\text{opt}}, \hat{N}_k^{\text{opt}}, \hat{B}^{\text{opt}}, \hat{C}^{\text{opt}}$

1: **while** not converged **do**
2: $\hat{A} = S\Lambda S^{-1}$, $\tilde{B}^T = S^{-1}\hat{B}$, $\tilde{C} = \hat{C}S$ $\tilde{N}_k^T = S^{-1}\hat{N}_k S$
3: $\text{vec}(V) = \left(-I_r \otimes A - \Lambda \otimes E - \sum_{k=1}^m \tilde{N}_k{}^T \otimes N_k\right)^{-1} (\tilde{B}^T \otimes B)\text{vec}(I_m)$
4: $\text{vec}(W) = \left(-I_r \otimes A^T - \Lambda \otimes E^T - \sum_{k=1}^m \tilde{N}_k \otimes N_k^T\right)^{-1} (\tilde{C}^T \otimes C^T)\text{vec}(I_p)$
5: $V = \text{orth}(V)$, $W = \text{orth}(W)$ % orth computes an orthonormal basis
6: $\hat{A} = (W^T EV)^{-1} W^T AV$, $\hat{N}_k = (W^T EV)^{-1} W^T N_k V$, $\hat{B} = (W^T EV)^{-1} W^T B$, $\hat{C} = CV$
7: **end while**
8: $\hat{A}^{\text{opt}} = \hat{A}$, $\hat{N}_k^{\text{opt}} = \hat{N}_k$, $\hat{B}^{\text{opt}} = \hat{B}$, $\hat{C}^{\text{opt}} = \hat{C}$

The convergence of Algorithm 3 will be measured in terms of the change in the eigenvalues of the reduced system. In every iteration the change in the eigenvalues between the last two iterations is checked. If it is sufficiently small, the algorithm stops and returns the final reduced order model.

5.5.4. $\mathscr{H}_2$-optimal MOR by using methods from differential geometry. We will establish a new result for the derivation of $\mathscr{H}_2$-optimal bilinear reduced order models. For ease of presentation we will assume $E = I_n$. As a system with E invertible is equivalent to the system multiplied by E^{-1}, this is possible. In addition, a generalization to systems with $E \neq I_n$ should be possible.

5.5.4.1. *The minimization problem.* As in the preceding sections we are going to minimize the $\mathscr{H}_2$-norm of the error system. However, we use a different approach, which was originally given for linear systems by Yan and Lam in 1999 [69]. It is based on minimizing the norm on the Stiefel manifold. This approach was recently transferred to Grassmann manifolds by Xu and Zeng [68]. We will now develop the methods for the bilinear case. In contrast to the methods in the previous sections, these methods directly preserve the BIBO stability of the model. Hence there is no need for stabilization methods that can be used for example to stabilize reduced order models obtained by BIRKA see Section 6.2.

First, the objective function for the minimization has to be found. We define the following function:

$$\begin{aligned}\mathcal{J}(W,V) &= \mathcal{J}(W^TAV, W^TN_1V, \ldots, W^TN_mV, W^TB, CV)\\ &:= ||\Sigma_{\text{bil}}^{\text{err}}||^2_{\mathscr{H}_2}\\ &= \text{tr}(\begin{bmatrix}C & -\hat{C}\end{bmatrix} P^{\text{err}} \begin{bmatrix}C^T\\ -\hat{C}^T\end{bmatrix})\\ &= \text{tr}(CP_{11}C^T - 2CP_{12}\hat{C}^T + \hat{C}P_{22}\hat{C}^T)\\ &= \text{tr}(C^TCP_{11} - 2V^TC^TCP_{12} + V^TC^TCVP_{22})\\ &= \text{tr}(C^TC(P_{11} - 2P_{12}V^T + VP_{22}V^T)),\end{aligned}$$

with P^{err} as given in equation (5.36), where $\begin{bmatrix}E & \hat{E}\end{bmatrix} = I_{nr}$. The reader should note that P_{12} and P_{22} depend on the reduced model and hence are functions of V and W. The problem of finding an $\mathscr{H}_2$-optimal reduced order model can be stated as a minimization problem of the form:

Minimize $\mathcal{J}(W^TAV, W^TN_kV, W^TB, CV)$ with respect to $(W,V) \in \mathbb{R}^{n\times r} \times \mathbb{R}^{n\times r}$ subject to $W^TV = I_r$ and $\hat{\Sigma}_{\text{bil}}$ is BIBO stable. (5.56)

If we use $W^T = V^\dagger = (V^TV)^{-1}V^T$ or $V^T = W^\dagger = (W^TW)^{-1}W^T$, the matrices W and V satisfy $W^TV = I_r$ if they have full rank. The following modified problem can therefore be considered:

Minimize $\mathcal{J}(V) := \mathcal{J}(V^\dagger AV, V^\dagger N_kV, V^\dagger B, CV)$ over $V \in \mathbb{R}^{n\times r}$ subject to the BIBO stability of $\hat{\Sigma}_{\text{bil}}$ reduced with $V^\dagger$ and V. (5.57)

This modified problem is an approximation to the original problem (5.56). It finds reduced models in a subset of the reduced models that would be considered while solving (5.56). It holds:

$$\begin{aligned}\mathcal{J}(V) &= \text{tr}(C^TCP_{11} - 2V^TC^TCP_{12} + V^TC^TCVP_{22})\\ &= \text{tr}(C^TC(P_{11} - 2P_{12}V^T + VP_{22}V^T)).\end{aligned} \tag{5.58}$$

Define $U = V(V^TV)^{-1/2}$. Let the reachability Gramian of the error system obtained by reducing the original system with U be

$$\tilde{P}^{\text{err}} = \begin{bmatrix}P_{11} & \tilde{P}_{12}\\ \tilde{P}_{12}^T & \tilde{P}_{22}\end{bmatrix}.$$

Let the reachability Gramian of the system reduced with V and $V^\dagger$ be P^{err}. If $\tilde{P}^{\text{err}}$ and P^{err} are the unique solutions to the Lyapunov equations of the respective error systems, then one concludes that

$$P_{22} = (V^TV)^{-1/2}\tilde{P}_{22}(V^TV)^{-1/2} \qquad \text{and} \qquad P_{12} = \tilde{P}_{12}(V^TV)^{-1/2}.$$

This can be seen in the following Lemma.

Lemma 5.5.5. *Using V and $V^\dagger$ for the reduction or using U (one-sided) respectively, and assuming that the corresponding Lyapunov equations of the error systems have unique solutions, leads to the following connection between the systems Gramians: $P_{22} = (V^TV)^{-1/2}\tilde{P}_{22}(V^TV)^{-1/2}$ and $P_{12} = \tilde{P}_{12}(V^TV)^{-1/2}$, where the matrices with $\tilde{\ }$ correspond to the system with U.*

Proof. If the original model has been reduced with U, one obtains

$$\tilde{P}^{\text{err}} = \begin{bmatrix} P_{11} & \tilde{P}_{12} \\ \tilde{P}_{12}^T & \tilde{P}_{22} \end{bmatrix},$$

the solution of the following Lyapunov equation:

$$\begin{aligned}
&\begin{bmatrix} A & \\ & (V^TV)^{-\frac{1}{2}}V^TAV(V^TV)^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} P_{11} & \tilde{P}_{12} \\ \tilde{P}_{12}^T & \tilde{P}_{22} \end{bmatrix} \begin{bmatrix} E^T & \\ & (V^TV)^{-\frac{1}{2}}V^TE^TV(V^TV)^{-\frac{1}{2}} \end{bmatrix} \\
&+ \begin{bmatrix} E & \\ & (V^TV)^{-\frac{1}{2}}V^TEV(V^TV)^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} P_{11} & \tilde{P}_{12} \\ \tilde{P}_{12}^T & \tilde{P}_{22} \end{bmatrix} \begin{bmatrix} A^T & \\ & (V^TV)^{-\frac{1}{2}}V^TA^TV(V^TV)^{-\frac{1}{2}} \end{bmatrix} \\
&+ \sum_{k=1}^{m} \begin{bmatrix} N_k & \\ & (V^TV)^{-\frac{1}{2}}V^TN_kV(V^TV)^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} P_{11} & \tilde{P}_{12} \\ \tilde{P}_{12}^T & \tilde{P}_{22} \end{bmatrix} \begin{bmatrix} N_k^T & \\ & (V^TV)^{-\frac{1}{2}}V^TN_k^TV(V^TV)^{-\frac{1}{2}} \end{bmatrix} \\
&+ \begin{bmatrix} B \\ (V^TV)^{-\frac{1}{2}}V^TB \end{bmatrix} \begin{bmatrix} B^T & B^TV(V^TV)^{-\frac{1}{2}} \end{bmatrix} = 0. \qquad (5.59)
\end{aligned}$$

If the reduction has been performed with V and $V^\dagger = (V^TV)^{-1}V^T$, one obtains:

$$\begin{aligned}
&\begin{bmatrix} A & \\ & (V^TV)^{-1}V^TAV \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \begin{bmatrix} E^T & \\ & V^TE^TV(V^TV)^{-1} \end{bmatrix} \\
&+ \begin{bmatrix} E & \\ & (V^TV)^{-1}V^TEV \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \begin{bmatrix} A^T & \\ & V^TA^TV(V^TV)^{-1} \end{bmatrix} \\
&+ \sum_{k=1}^{m} \begin{bmatrix} N_k & \\ & (V^TV)^{-1}V^TN_kV \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \begin{bmatrix} N_k^T & \\ & V^TN_k^TV(V^TV)^{-1} \end{bmatrix} \\
&+ \begin{bmatrix} B \\ (V^TV)^{-1}V^TB \end{bmatrix} \begin{bmatrix} B^T & B^TV(V^TV)^{-1} \end{bmatrix} = 0. \qquad (5.60)
\end{aligned}$$

Multiplying equation (5.60) with $\begin{bmatrix} I_n & \\ & (V^TV)^{\frac{1}{2}} \end{bmatrix}$ from the left and the right yields:

$$\begin{aligned}
&\begin{bmatrix} A & \\ & (V^TV)^{-\frac{1}{2}}V^TAV(V^TV)^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} I_n & \\ & (V^TV)^{\frac{1}{2}} \end{bmatrix} \\
&\qquad \cdot \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \begin{bmatrix} I_n & \\ & (V^TV)^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} E^T & \\ & (V^TV)^{-\frac{1}{2}}V^TE^TV(V^TV)^{-\frac{1}{2}} \end{bmatrix} \\
&+ \begin{bmatrix} E & \\ & (V^TV)^{-\frac{1}{2}}V^TEV(V^TV)^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} I_n & \\ & (V^TV)^{\frac{1}{2}} \end{bmatrix} \\
&\qquad \cdot \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \begin{bmatrix} I_n & \\ & (V^TV)^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} A^T & \\ & (V^TV)^{-\frac{1}{2}}V^TA^TV(V^TV)^{-\frac{1}{2}} \end{bmatrix} \\
&+ \sum_{k=1}^{m} \begin{bmatrix} N_k & \\ & (V^TV)^{-\frac{1}{2}}V^TN_kV(V^TV)^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} I_n & \\ & (V^TV)^{\frac{1}{2}} \end{bmatrix} \\
&\qquad \cdot \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \begin{bmatrix} I_n & \\ & (V^TV)^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} N_k^T & \\ & (V^TV)^{-\frac{1}{2}}V^TN_k^TV(V^TV)^{-\frac{1}{2}} \end{bmatrix} \\
&+ \begin{bmatrix} B \\ (V^TV)^{-\frac{1}{2}}V^TB \end{bmatrix} \begin{bmatrix} B^T & B^TV(V^TV)^{-\frac{1}{2}} \end{bmatrix} = 0. \qquad (5.61)
\end{aligned}$$

Under the assumption that (5.61) and (5.59) hold, one obtains (as equation (5.59) has a unique solution):

$$\begin{bmatrix} P_{11} & \tilde{P}_{12} \\ \tilde{P}_{12}^T & \tilde{P}_{22} \end{bmatrix} = \begin{bmatrix} I_n & \\ & (V^TV)^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \begin{bmatrix} I_n & \\ & (V^TV)^{\frac{1}{2}} \end{bmatrix}, \qquad (5.62)$$

which leads to

$$\tilde{P}_{12}^T = (V^TV)^{\frac{1}{2}}P_{12}^T,$$

and

$$\tilde{P}_{22} = (V^TV)^{\frac{1}{2}}P_{22}(V^TV)^{\frac{1}{2}}.$$

□

One can then show that the functions $\mathcal{J}(V)$ and $\mathcal{J}(U)$ are equal:

$$\begin{aligned}\mathcal{J}(V) =& \operatorname{tr}(C^TCP_{11} - 2V^TC^TCP_{12} + V^TC^TCVP_{22})\\ =& \operatorname{tr}(C^TCP_{11} - 2(V^TV)^{-1/2}V^TC^TC\tilde{P}_{12} + (V^TV)^{-1/2}V^TC^TCV(V^TV)^{-1/2}\tilde{P}_{22})\\ =& \mathcal{J}(U)\\ =& \mathcal{J}(U^TAU, U^TN_1U, \ldots, U^TN_mU, U^TB, CU).\end{aligned}$$

Hence the following minimization problem is equivalent to (5.57):

> Minimize $\mathcal{J}(U) := \mathcal{J}(U^TAU, U^TN_kU, U^TB, CU)$ over $U \in \mathbb{R}^{n\times r}$ with $U^TU = I_r$ subject to the BIBO stability of $\hat{\Sigma}_{\text{bil}}$ the reduced bilinear system calculated with U.

As U is an element of the Stiefel manifold $\operatorname{St}(r,n)$ (cf. Section 2.2) the minimization problem can be stated on this manifold:

> Minimize $\mathcal{J}(U) := \mathcal{J}(U^TAU, U^TN_kU, U^TB, CU)$ over $U \in \operatorname{St}(r,n)$ subject to the BIBO stability of $\hat{\Sigma}_{\text{bil}}$, the reduced bilinear system calculated with U. (5.63)

Before we can state the minimization problem on the Grassmann manifold (cf. Section 2.2), we need the following Lemma:

Lemma 5.5.6. *For an orthogonal matrix $Q \in \mathbb{R}^{r\times r}$ it holds $\mathcal{J}(U) = \mathcal{J}(UQ)$.*

Proof. It holds (cf. (2.45)):

$$\mathcal{J}(U) = ||\Sigma_{\text{bil}}^{\text{err}}||_{\mathcal{H}_2}^2 = \operatorname{tr}\left(\sum_{i=1}^{\infty}\int_0^{\infty}\cdots\int_0^{\infty}\sum_{k_1,k_2,\ldots,k_i=1}^{m}(h^{\text{err}})_i^{(k_1,\ldots,k_i)}(s_1,\ldots,s_i) \cdot ((h^{\text{err}})_i^{(k_1,\ldots,k_i)}(s_1,\ldots,s_i))^T ds_1\ldots ds_i\right).$$

The Volterra kernels of the error system are:

$$
\begin{aligned}
&(h^{\mathrm{err}})_i^{(k_1,\dots,k_i)}(s_1,\dots,s_i)\\
&= \begin{bmatrix} C & -\hat{C} \end{bmatrix} e^{\left(\begin{bmatrix} A & \\ & \hat{A} \end{bmatrix}^{s_i}\right)} \begin{bmatrix} N_{k_1} & \\ & \hat{N}_{k_1} \end{bmatrix} e^{\left(\begin{bmatrix} A & \\ & \hat{A} \end{bmatrix}^{s_{i-1}}\right)} \begin{bmatrix} N_{k_2} & \\ & \hat{N}_{k_2} \end{bmatrix} \cdots\\
&\qquad \cdots \begin{bmatrix} N_{k_{i-1}} & \\ & \hat{N}_{k_{i-1}} \end{bmatrix} e^{\left(\begin{bmatrix} A & \\ & \hat{A} \end{bmatrix}^{s_1}\right)} \begin{bmatrix} b_{k_i} \\ \hat{b}_{k_i} \end{bmatrix}\\
&= \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} e^{As_i} N_{k_1} e^{As_{i-1}} N_{k_2} \dots N_{k_{i-1}} e^{As_1} & \\ & e^{\hat{A}s_i} \hat{N}_{k_1} e^{\hat{A}s_{i-1}} \hat{N}_{k_2} \dots \hat{N}_{k_{i-1}} e^{\hat{A}s_1} \end{bmatrix} \begin{bmatrix} b_{k_i} \\ \hat{b}_{k_i} \end{bmatrix}.
\end{aligned}
$$

The Volterra kernels coincide for UQ and U with $QQ^T = Q^TQ = I_r$ because

$$
\begin{aligned}
&\begin{bmatrix} C & -CUQ \end{bmatrix}\\
&\cdot \begin{bmatrix} e^{As_i} N_{k_1} \dots N_{k_{i-1}} e^{As_1} & \\ & e^{Q^TU^TAUQs_i} Q^TU^TN_{k_1}UQ \dots Q^TU^TN_{k_{i-1}}UQe^{Q^TU^TAUQs_1} \end{bmatrix}\\
&\qquad \cdot \begin{bmatrix} b_{k_i} \\ Q^TU^Tb_{k_i} \end{bmatrix}\\
&= \begin{bmatrix} C & -CU \end{bmatrix}\\
&\cdot \begin{bmatrix} e^{As_i} N_{k_1} \dots N_{k_{i-1}} e^{As_1} & \\ & QQ^Te^{U^TAUs_i} QQ^TU^TN_{k_1}U \dots QQ^TU^TN_{k_{i-1}}UQQ^Te^{U^TAUs_1} \end{bmatrix}\\
&\qquad \cdot \begin{bmatrix} b_{k_i} \\ QQ^TU^Tb_{k_i} \end{bmatrix}\\
&= \begin{bmatrix} C & -CU \end{bmatrix} \begin{bmatrix} e^{As_i} N_{k_1} \dots N_{k_{i-1}} e^{As_1} & \\ & e^{U^TAUs_i} U^TN_{k_1}U \dots U^TN_{k_{i-1}}Ue^{U^TAUs_1} \end{bmatrix} \begin{bmatrix} b_{k_i} \\ U^Tb_{k_i} \end{bmatrix},
\end{aligned}
$$

and we conclude $\mathcal{J}(U) = \mathcal{J}(UQ)$. □

We can now state the minimization problem on the Grassmann manifold:

> Minimize $\mathcal{J}(U)$ over $[U] \in \mathrm{Gr}(r,n)$ subject to the BIBO stability of $\hat{\Sigma}_{\mathrm{bil}}$ reduced with U. (5.64)

5.5.4.2. *The bilinear fast gradient flow algorithm.* We will now calculate the gradients $\nabla_S\mathcal{J}$ and $\nabla_G\mathcal{J}$ of the objective function $\mathcal{J}(U)$ on the Stiefel and the Grassmann manifolds. A minimum of the objective function $\mathcal{J}(U)$ needs to satisfy $\nabla_S\mathcal{J} = 0$ or $\nabla_G\mathcal{J} = 0$, respectively. As shown before (cf. Section 2.2), the gradients need to satisfy the following equations:

$$\nabla_S\mathcal{J} = \mathcal{J}_U - U\mathcal{J}_U^TU, \tag{2.11}$$

$$\nabla_G \mathcal{J} = \mathcal{J}_U - UU^T \mathcal{J}_U. \tag{2.16}$$

Hence, one needs the matrix of all partial derivatives of $\mathcal{J}$ with respect to U, i.e.:

$$(\mathcal{J}_U)_{ij} = \frac{\partial \mathcal{J}}{\partial U_{ij}}. \tag{5.65}$$

Let E_{ij} be the single-entry matrix having a one in entry (i,j) and zeros elsewhere. We derive:

$$\begin{aligned}(\mathcal{J}_U)_{ij} &= \frac{\partial}{\partial U_{ij}} \mathrm{tr}\left(C^T C(P_{11} - 2P_{12}U^T + UP_{22}U^T)\right) \qquad (5.66)\\ &= \mathrm{tr}\left(C^T C\left(E_{ij}P_{22}U^T + U\frac{\partial P_{22}}{\partial U_{ij}}U^T + UP_{22}E_{ij}^T - 2\frac{\partial P_{12}}{\partial U_{ij}}U^T - 2P_{12}E_{ij}^T\right)\right)\\ &= \mathrm{tr}\left(\underbrace{\frac{\partial P_{22}}{\partial U_{ij}}U^T C^T CU}_{(*)} \underbrace{-2\,U^T C^T C\frac{\partial P_{12}}{\partial U_{ij}}}_{(+)} + 2\left(C^T CUP_{22} - C^T CP_{12}\right)E_{ij}^T\right).\end{aligned}$$

By splitting the Lyapunov equations of the error system ((5.36) and (5.38)), the following Lyapunov and Sylvester equations can be obtained:

$$AP_{11} + P_{11}A^T + \sum_{k=1}^{m} N_k P_{11} N_k^T + BB^T = 0, \tag{5.67}$$

$$A^T Q_{11} + Q_{11}A + \sum_{k=1}^{m} N_k^T Q_{11} N_k + C^T C = 0, \tag{5.68}$$

$$U^T AUP_{22} + P_{22}U^T A^T U + \sum_{k=1}^{m} U^T N_k UP_{22}U^T N_k^T U + U^T BB^T U = 0, \tag{5.69}$$

$$U^T A^T UQ_{22} + Q_{22}U^T AU + \sum_{k=1}^{m} U^T N_k^T UQ_{22}U^T N_k U + U^T C^T CU = 0, \tag{5.70}$$

$$AP_{12} + P_{12}U^T A^T U + \sum_{k=1}^{m} N_k P_{12} U^T N_k^T U + BB^T U = 0, \tag{5.71}$$

$$A^T Q_{12} + Q_{12} U^T A U + \sum_{k=1}^{m} N_k^T Q_{12} U^T N_k U - C^T C U = 0. \tag{5.72}$$

Differentiating the equations (5.69) and (5.71) with respect to U leads to:

$$\begin{aligned} &E_{ij}^T A U P_{22} + U^T A E_{ij} P_{22} + U^T A U \frac{\partial P_{22}}{\partial U_{ij}} + \frac{\partial P_{22}}{\partial U_{ij}} U^T A^T U + P_{22} E_{ij}^T A^T U + P_{22} U^T A^T E_{ij} \\ &+ \sum_{k=1}^{m} E_{ij}^T N_k U P_{22} U^T N_k^T U + \sum_{k=1}^{m} U^T N_k E_{ij} P_{22} U^T N_k^T U + \sum_{k=1}^{m} U^T N_k U \frac{\partial P_{22}}{\partial U_{ij}} U^T N_k^T U \\ &+ \sum_{k=1}^{m} U^T N_k U P_{22} E_{ij}^T N_k^T U + \sum_{k=1}^{m} U^T N_k U P_{22} U^T N_k^T E_{ij} + E_{ij}^T B B^T U + U^T B B^T E_{ij} = 0, \end{aligned} \tag{5.73}$$

and

$$\begin{aligned} &A \frac{\partial P_{12}}{\partial U_{ij}} + \frac{\partial P_{12}}{\partial U_{ij}} U^T A^T U + P_{12} E_{ij}^T A^T U + P_{12} U^T A^T E_{ij} + \sum_{k=1}^{m} N_k \frac{\partial P_{12}}{\partial U_{ij}} U^T N_k^T U \\ &+ \sum_{k=1}^{m} N_k P_{12} E_{ij}^T N_k^T U + \sum_{k=1}^{m} N_k P_{12} U^T N_k^T E_{ij} + B B^T E_{ij} = 0. \end{aligned} \tag{5.74}$$

We define

$$\begin{aligned} Z \quad = \quad & P_{22} E_{ij}^T A^T U + P_{22} U^T A^T E_{ij} + \sum_{k=1}^{m} E_{ij}^T N_k U P_{22} U^T N_k^T U \\ & + \sum_{k=1}^{m} U^T N_k E_{ij} P_{22} U^T N_k^T U + U^T B B^T E_{ij}. \end{aligned}$$

For the next step, we use the following Lemma:

Lemma 5.5.7. *Let $P, X \in \mathbb{R}^{n\times m}$ and $Q, Y \in \mathbb{R}^{n\times m}$. Let $A, C_j \in \mathbb{R}^{n\times n}, B, D_j \in \mathbb{R}^{m\times m}$, $j = 1, \ldots, h$. If P and Q satisfy*

$$AP + PB + \sum_{j=1}^{h} C_j P D_j + X = 0 \text{ and } A^T Q + Q B^T + \sum_{j=1}^{h} C_j^T P D_j^T + Y = 0,$$

then it holds

$$tr\left(Y^T P\right) = tr\left(X^T Q\right).$$

Proof.

$$\begin{aligned}
\mathrm{tr}(Y^T P) &= \mathrm{tr}((-A^T Q - QB^T - \sum_{j=1}^{h} C_j^T Q D_j^T)^T P) \\
&= -\mathrm{tr}(PA^T Q) - \mathrm{tr}(PQB^T) - \sum_{j=1}^{h} \mathrm{tr}(PC_j^T Q D_j^T) \\
&= -\mathrm{tr}(PA^T Q) - \mathrm{tr}(B^T PQ) - \sum_{j=1}^{h} \mathrm{tr}(D_j^T PC_j^T Q) \\
&= \mathrm{tr}((-AP - PB - \sum_{j=1}^{h} C_j P D_j)^T Q) \\
&= \mathrm{tr}(X^T Q).
\end{aligned}$$

□

This Lemma together with equations (5.70) and (5.73) gives part (*) of equation (5.66):

$$\mathrm{tr}(U^T C^T CU \frac{\partial P_{22}}{\partial U_{ij}}) = \mathrm{tr}((Z + Z^T)^T Q_{22}) = 2\mathrm{tr}(ZQ_{22}),$$

and together with equations (5.72) and (5.74), the lemma leads to part (+) of equation (5.66):

$$\begin{aligned}
\mathrm{tr}(-U^T C^T C \frac{\partial P_{12}}{\partial U_{ij}}) =& \mathrm{tr}((P_{12} E_{ij}^T A^T U + P_{12} U^T A^T E_{ij} + \sum_{k=1}^{m} N_k P_{12} E_{ij}^T N_k^T U \\
&+ \sum_{k=1}^{m} N_k P_{12} U^T N_k^T E_{ij} + BB^T E_{ij})^T Q_{12}).
\end{aligned}$$

Now the derivative $\mathcal{J}_U$ can be calculated:

$$\begin{aligned}
(\mathcal{J}_U)_{ij} =& 2tr\Big(ZQ_{22} + (C^T CUP_{22} - C^T CP_{12}) E_{ij}^T \\
&+ U^T AE_{ij} P_{12}^T Q_{12} + E_{ij}^T AUP_{12}^T Q_{12} + E_{ij}^T BB^T Q_{12} \\
&+ \sum_{k=1}^{m} U^T N_k E_{ij} P_{12}^T N_k^T Q_{12} + \sum_{k=1}^{m} E_{ij}^T N_k UP_{12}^T N_k^T Q_{12} \Big)
\end{aligned}$$

$$
\begin{aligned}
=&2tr\Bigg(P_{22}E_{ij}^T A^T U Q_{22} + P_{22}U^T A^T E_{ij} Q_{22} \\
&+ \sum_{k=1}^{m} E_{ij}^T N_k U P_{22} U^T N_k^T U Q_{22} + \sum_{k=1}^{m} U^T N_k E_{ij} P_{22} U^T N_k^T U Q_{22} \\
&+ U^T B B^T E_{ij} Q_{22} + A^T U Q_{12}^T P_{12} E_{ij}^T + A U P_{12}^T Q_{12} E_{ij}^T + B B^T Q_{12} E_{ij}^T \\
&+ \sum_{k=1}^{m} N_k^T U Q_{12}^T N_k P_{12} E_{ij}^T + \sum_{k=1}^{m} N_k U P_{12}^T N_k^T Q_{12} E_{ij}^T \\
&+ (C^T C P_{22} - C^T C P_{12}) E_{ij}^T \Bigg) \\
=&2tr\Bigg(\Bigg(A^T U Q_{12}^T P_{12} + A U P_{12}^T Q_{12} + B B^T Q_{12} + B B^T U Q_{22} \\
&+ \sum_{k=1}^{m} N_k^T U Q_{12}^T N_k P_{12} + \sum_{k=1}^{m} N_k U P_{12}^T N_k^T Q_{12} \\
&+ C^T C U P_{22} - C^T C P_{12} + A^T U Q_{22} P_{22} + A U P_{22} Q_{22} \\
&+ \sum_{k=1}^{m} N_k U P_{22} U^T N_k^T U Q_{22} + \sum_{k=1}^{m} N_k^T U Q_{22} U^T N_k U P_{22} \Bigg) E_{ij}^T \Bigg).
\end{aligned}
$$

By defining

$$
\begin{aligned}
R =& A^T U (Q_{12}^T P_{12} + Q_{22} P_{22}) + A U (P_{12}^T Q_{12} + P_{22} Q_{22}) + B B^T (Q_{12} + U Q_{22}) \\
&+ C^T C (U P_{22} - P_{12}) + \sum_{k=1}^{m} N_k^T U (Q_{12}^T N_k P_{12} + Q_{22} U^T N_k U P_{22}) \\
&+ \sum_{k=1}^{m} N_k U (P_{12}^T N_k^T Q_{12} + P_{22} U^T N_k^T U Q_{22}),
\end{aligned} \tag{5.75}
$$

we obtain

$$
\mathcal{J}_U = 2R. \tag{5.76}
$$

The gradient on the Stiefel manifold can now be determined:

$$
\begin{aligned}
\nabla_S \mathcal{J} &= \mathcal{J}_U - U \mathcal{J}_U^T U \\
&= 2(R - U R^T U).
\end{aligned} \tag{5.77}
$$

The gradient on the Grassmann manifold is:

$$\begin{aligned}\nabla_G \mathcal{J} &= \mathcal{J}_U - UU^T \mathcal{J}_U \\ &= 2(R - UU^T R). \end{aligned} \tag{5.78}$$

A minimum point of the function $\mathcal{J}(U)$ must satisfy the following conditions:

- On $\mathrm{St}(r,n)$: $(R - UR^T U) = 0$ and $U^T U = I_r$.
- On $\mathrm{Gr}(r,n)$: $(R - UU^T R) = 0$ and $U^T U = I_r$.

Using the definition of R, one obtains the following lemma.

Lemma 5.5.8. *It holds $U^T R = R^T U$ (i.e. $U^T R$ is symmetric), with R as given in (5.75).*

Proof. Using equations (5.69) to (5.72) one obtains:

$$\begin{aligned}
U^T R = {} & \overbrace{U^T \left(-C^T C + A^T U Q_{12}^T + \sum_{k=1}^{m} N_k^T U Q_{12}^T N_k \right)}^{-(A^T Q_{12})^T} P_{12} \\
& + \overbrace{U^T \left(C^T C U + A^T U Q_{22}^T + \sum_{k=1}^{m} N_k^T U Q_{22} U^T N_k U \right)}^{-Q_{22} U^T A U} P_{22} \\
& + \overbrace{U^T \left(B B^T + A U P_{12}^T + \sum_{k=1}^{m} N_k U P_{12}^T N_k^T \right)}^{(-A P_{12})^T} Q_{12} \\
& + \overbrace{U^T \left(B B^T U + A U P_{22} + \sum_{k=1}^{m} N_k U P_{22} U^T N_k^T U \right)}^{-P_{22} U^T A^T U} Q_{22} \\
& = -Q_{12}^T A P_{12} - Q_{22} U^T A U P_{22} - P_{12}^T A^T Q_{12} - P_{22} U^T A^T U Q_{22} \\
& = Q_{12}^T \left(P_{12} U^T A^T U + \sum_{k=1}^{m} N_k P_{12} U^T N_k^T U + B B^T U \right) \\
& + Q_{22} \left(P_{22} U^T A^T U + \sum_{k=1}^{m} U^T N_k U P_{22} U^T N_k^T U + U B B^T U \right)
\end{aligned}$$

$$
\begin{aligned}
&+ P_{12}^T \left(Q_{12} U^T A U + \sum_{k=1}^{m} N_k^T Q_{12} U^T N_k U - C^T C U \right) \\
&+ P_{22} \left(Q_{22} U^T A U + \sum_{k=1}^{m} U^T N_k^T U Q_{22} U^T N_k U + U^T C^T C U \right) \\
&= R^T U.
\end{aligned}
$$

□

Using the previous shown lemma, the following theorem results:

Theorem 5.5.9. *A minimum point $U \in \mathbb{R}^{n \times r}$ of the function $\mathcal{J}(U)$ must satisfy the conditions*

$$(R - UU^T R) = 0 \text{ and } U^T U = I_r, \tag{5.79}$$

regardless of whether the minimization is performed on the Stiefel or the Grassmann manifold.

It is now the objective to find a zero of the gradient, i.e. a zero of $(R - UU^T R) = 0$. Following [69] and [68] this is done by using a gradient flow on the manifolds:

$$\dot{U} = \frac{\partial U}{\partial t} = U(t) U(t)^T R(t) - R(t). \tag{5.80}$$

Yan and Lam [69] propose to rewrite the equation (5.80) using the symmetry of $U^T R$ with $\Gamma = UR^T - RU^T$ skew-symmetric:

$$\dot{U} = \Gamma U. \tag{5.81}$$

They then suggest the following iteration for updating U:

$$U_{j+1} = \exp(t_j \Gamma_j) U_j. \tag{5.82}$$

Xu and Zeng [68] find the new projection matrix U_{j+1} by using the geodesic (i.e. the shortest connection of two points) on the Grassmann manifold:

$$U_{j+1} = U_j V_j \cos(t_j \Sigma_j) V_j^T + W_j \sin(t_j \Sigma_j) V_j^T, \tag{5.83}$$

with $-\nabla_G \mathcal{J}(U_j) = W_j \Sigma_j V_j^T$ (the SVD of $-\nabla_G \mathcal{J}(U_j)$). In addition they show that

$$\exp(t\Gamma) U = UV \cos(t\Sigma) V^T + W \sin(t\Sigma) V^T, \tag{5.84}$$

which is also true in the bilinear case. Hence, the calculation of U_{j+1} is the same in both approaches.

Remark 5.5.10. As the calculations of U_{j+1} using (5.82) or (5.83) lead to the same updated matrix, the optimization Yan and Lam performed in [69] was already based on a geodesic on a Grassmann manifold, which they were probably not aware of. Hence from a present point of view, they were in fact performing a reduction on a Grassmann manifold.

It now remains to choose the time step t_j such that a step in descent direction is performed, i.e. the condition

$$\mathcal{J}(U_j) \geq \mathcal{J}(U_{j+1}), \tag{5.85}$$

needs to be complied. In the linear case, Yan and Lam [69] propose two different time steps. One is based on the original matrices and chosen a priori, the other one is chosen in every step based on the original matrices and the corresponding matrix U_j. For linear systems and these time steps, the condition (5.85) is always satisfied. It is now possible to state the general optimization algorithm 4 for bilinear systems, inspired by the linear algorithm given by Yan and Lam [69].

Algorithm 4 GFA for bilinear systems (bilGFA).

Input: (A, N_k, B, C), maxIt : maximal number of iterations.
Output: Reduced model $(\hat{A}, \hat{N}_k, \hat{B}, \hat{C})$.

1: Choose a matrix $U_0 \in \mathbb{R}^{n\times r}$ such that $U_0^T U_0 = I_r$. Set $j = 0$.
2: **for** $j = 0 \to \text{maxIt} - 1$ **do**
3: Compute $P_{22}^j, Q_{22}^j, P_{12}^j, Q_{12}^j$ by solving the equations (5.69) - (5.72) for U_j.
4: Compute R_j by using equation (5.75).
5: Compute the gradient $\nabla\mathcal{J}(U_j) = R_j - U_j(U_j^T R_j)$.
6: Compute $\Gamma_j = U_j R_j^T - R_j U_j^T$.
7: Choose t_j.
8: Set $U_{j+1} = \exp(t_j\Gamma_j)U_j$.
9: **end for**
10: Calculate the reduced model: $\hat{A} = U_{\text{maxIt}}^T A U_{\text{maxIt}}$, $\hat{N}_k = U_{\text{maxIt}}^T N_k U_{\text{maxIt}}$, $\hat{B} = U_{\text{maxIt}}^T B$, $\hat{C} = C U_{\text{maxIt}}$.

For bilinear systems, the calculation of adaptive time steps t_j is not a straight forward generalization and requires further investigation. However, choosing an appropriate time step can be done by using the Armijo step size as proposed by Xu and Zeng [68]. With

$$\mathcal{U}_j(t) = U_j V_j \cos(t\Sigma_j) V_j^T + W_j \sin(t\Sigma_j) V_j^T,$$

the Armijo stepsize is $t_A = \delta^i \gamma$ where i is the smallest nonnegative integer such that

$$\mathcal{J}(U_j) - \mathcal{J}(\mathcal{U}_j(t_A)) \geq -\epsilon\delta^i\gamma \left\langle \nabla\mathcal{J}(U_j), -\nabla\mathcal{J}(U_j) \right\rangle, \tag{5.86}$$

holds for $\delta, \epsilon \in (0,1), \gamma > 0$. As $-\epsilon\delta^i\gamma \left\langle \nabla\mathcal{J}(U_j), -\nabla\mathcal{J}(U_j) \right\rangle$ is positive, it is obvious that

$$\mathcal{J}(U_j) \geq \mathcal{J}(\mathcal{U}_j(t_A)) = \mathcal{J}(U_{j+1}). \tag{5.87}$$

We are now at the point where all steps have been taken to define the optimization algorithm for a bilinear model. It is a further development of the linear fast gradient flow algorithm (FGFA) established by Xu and Zeng [68]. We will therefore call it the bilinear fast gradient flow algorithm (bilFGFA). Its main steps can be found under Algorithm 5.

The algorithm ends when the maximal number of iterations maxIt is reached. However this does not mean that the obtained reduced system $(\hat{A}, \hat{N}_k, \hat{B}, \hat{C})$ is an optimal model. Therefore, it is reasonable to check if the gradient $\nabla\mathcal{J}(U)$ converges to zero. If it is sufficiently small, the algorithm should stop.

5.5.4.3. *Analysis of the convergence behavior of the bilFGFA.* Starting from a BIBO stable original system, and reducing with bilFGFA, the resulting reduced system is not known to be BIBO stable. For symmetric matrices A and N_k, we can prove the following result, which ensures the BIBO stability of the reduced system:

Proposition 5.5.11. *Let* $||u(t)||_2 = \sqrt{\sum_{k=1}^{m} |u(t)|^2} \leq M$. *Let* (A, B, N_k, C) *be a bilinear system with*

$$\sum_{k=1}^{m} ||N_k||_2 < \frac{\alpha}{\beta M}, \text{ where } ||e^{At}||_2 \leq \beta e^{-\alpha t}, \max_{i=1,\dots,n} (\mathrm{Re}(\lambda_i(A))) < -\alpha \tag{5.89}$$

($\Longrightarrow$ system is BIBO stable, cf. Theorem 2.3.24) and symmetric A, N_k. Let $U \in \mathbb{R}^{n\times r}$ be orthogonal. Then the reduced system $\hat{\Sigma}_{\text{bil}}^{\text{err}}$ with $\hat{A} = U^T A U$, $\hat{B} = U^T B$, $\hat{N}_k = U^T N_k U$, $\hat{C} = CU$ is BIBO stable.

Algorithm 5 FGFA for bilinear systems (bilFGFA).

Input: (A, N_k, B, C), maxIt : maximal number of iterations.
Output: Reduced model $(\hat{A}, \hat{N}_k, \hat{B}, \hat{C})$.

1: Choose a matrix $U_0 \in \mathbb{R}^{n\times r}$ such that $U_0^T U_0 = I_r$. Set $j = 0$.
2: **for** $j = 0 \to \text{maxIt} - 1$ **do**
3: Compute $P_{22}^j, Q_{22}^j, P_{12}^j, Q_{12}^j$ by solving the equations (5.69) - (5.72) for U_j.
4: Compute R_j by using equation (5.75).
5: Compute the gradient $\nabla\mathcal{J}(U_j) = R_j - U_j(U_j^T R_j)$.
6: Compute the new search direction $F_j = -\nabla\mathcal{J}(U_j)$ and its SVD $F_j = W_j\Sigma_j V_j^T$.
7: Minimize $\mathcal{J}(\mathcal{U}_j(t))$ over $t \geq 0$, where

$$\mathcal{U}_j(t) = U_j V_j \cos(t\Sigma_j)V_j^T + W_j \sin(t\Sigma_j)V_j^T. \tag{5.88}$$

8: Set $t_j = t_{min}$ and $U_{j+1} = \mathcal{U}_j(t_j)$.
9: **end for**
10: Calculate the reduced model: $\hat{A} = U_{\text{maxIt}}^T A U_{\text{maxIt}}$, $\hat{N}_k = U_{\text{maxIt}}^T N_k U_{\text{maxIt}}$, $\hat{B} = U_{\text{maxIt}}^T B$, $\hat{C} = C U_{\text{maxIt}}$.

Proof. As the reduced matrix $\hat{A}$ and the original matrix A are symmetric, their eigenvalues are real and the following condition for the eigenvalues hold [60]:

$$\lambda_i(A) \geq \lambda_i(\hat{A}) \geq \lambda_{i+n-r}(A), \qquad i = 1, \ldots, r.$$

As A is stable, this leads to the condition

$$-\alpha > \lambda_i(A) \geq \lambda_i(\hat{A}), \qquad i = 1, \ldots, r. \tag{5.90}$$

Therefore, one can choose $\hat{\alpha} = \alpha$. As A and $\hat{A}$ are symmetric, they can be diagonalized by orthogonal matrices, and it holds:

$$||e^{At}||_2 \leq ||e^{Q^T\Lambda Q t}||_2 = ||Q^T e^{\Lambda t} Q||_2 \leq \underbrace{\overbrace{||Q^T||_2}^{=1}\overbrace{||Q||_2}^{=1}}_{=\beta}||e^{\Lambda t}||_2 \leq e^{-\alpha t} \text{ with } \beta = 1.$$

The same calculation leads to $||e^{\hat{A}t}||_2 \leq e^{-\hat{\alpha}t}$. Hence $\beta = \hat{\beta} = 1$ and $\hat{\alpha} = \alpha$.

For N_k and $\hat{N}_k$ symmetric, one knows that

$$||N_k||_2 = \max_{l=1,\dots,n} |\lambda_l(N_k)| \text{ and } ||\hat{N}_k||_2 = \max_{l=1,\dots,r} |\lambda_l(\hat{N}_k)|.$$

It also holds:

$$\lambda_i(N_k) \geq \lambda_i(\hat{N}_k) \geq \lambda_{i+n-r}(N_k), \qquad i = 1, \dots, r.$$

Therefore we conclude with $|\lambda_i(\hat{N}_k)| \leq \max\{|\lambda_1(N_k)|, |\lambda_n(N_k)|\}$:

$$||\hat{N}_k||_2 = \max_{l=1,\dots,r} \{|\lambda_l(\hat{N})|\} \leq \max\{|\lambda_1(N_k)|, |\lambda_n(N_k)|\} = ||N_k||_2.$$

We finish by calculating

$$\sum_{k=1}^{m} ||\hat{N}_k||_2 \leq \sum_{k=1}^{m} ||N_k||_2 < \frac{\alpha}{M\beta} = \frac{\hat{\alpha}}{M\hat{\beta}},$$

from which it follows that the reduced system is BIBO stable. □

Corollary 5.5.12. *If A and N_k are symmetric and the condition (5.89) holds, the error system is BIBO stable and it holds $\alpha^{\text{err}} = \alpha = \hat{\alpha}$, $\beta^{\text{err}} = \beta = \hat{\beta} = 1$ if the reduction is performed with an orthogonal $U \in \mathbb{R}^{n\times r}$.*

Proof. If the reduction is performed by an orthogonal U, then A^{err} and N_k^{err} are symmetric, as $\hat{A}$ and $\hat{N}_k$ stay symmetric. For a system fulfilling condition (5.89), it holds $\alpha = \hat{\alpha}$ and $\beta = \hat{\beta} = 1$ as shown in Proposition 5.5.11. As

$$\lambda_{\max}(A^{\text{err}}) = \lambda_{\max}\left(\begin{bmatrix} A & \\ & \hat{A} \end{bmatrix}\right) = \max\{\lambda_{\max}(A), \lambda_{\max}(\hat{A})\} \leq \max\{-\alpha, -\hat{\alpha}\}, = -\alpha,$$

one can choose $\alpha^{\text{err}} = \alpha = \hat{\alpha}$. The symmetric matrix A^{err} can be diagonalized by an orthogonal matrix, and it holds

$$||e^{A^{\text{err}}t}||_2 \leq ||e^{Q^T\Lambda^{\text{err}}Qt}||_2 = ||Q^T e^{\Lambda^{\text{err}}t} Q||_2 \leq \underbrace{\overbrace{||Q^T||_2}^{=1}\overbrace{||Q||_2}^{=1}}_{=\beta^{\text{err}}} ||e^{\Lambda^{\text{err}}t}||_2 \leq e^{-\alpha^{\text{err}}t} \leq e^{-\alpha t},$$

with $\beta^{\text{err}} = \beta = \hat{\beta} = 1$. Using $||\hat{N}_k||_2 \leq ||N_k||_2$ (cf. Proposition 5.5.11) one can conclude that $||N_k^{\text{err}}||_2 = \max\{||N_k||_2, ||\hat{N}_k||_2\} = ||N_k||_2$. As the original system satisfies the condition (5.89), one concludes

$$\sum_{k=1}^{m} ||N_k^{\text{err}}||_2 = \sum_{k=1}^{m} ||N_k||_2 < \frac{\alpha}{M\beta} = \frac{\alpha^{\text{err}}}{M\beta^{\text{err}}},$$

and hence the error system is BIBO stable. □

The following theorem states, that the gradient of the function converges to zero while using Algorithm 5.

Theorem 5.5.13. *Let A and N_k be symmetric and let*

$$||u(t)||_2 = \sqrt{\sum_{k=1}^{m} |u(t)|^2} \leq M.$$

For a bilinear system (A, B, N_k, C) with

$$\sum_{k=1}^{m} ||N_k||_2 < \frac{\alpha}{\beta M}, \text{ where } ||e^{At}||_2 \leq \beta e^{-\alpha t}, \max_{i=1,\dots,n} (\mathrm{Re}(\lambda_i(A))) < -\alpha \quad (5.89)$$

($\Longrightarrow$ system is BIBO stable), the Algorithm 5 provides BIBO stable reduced models and is globally convergent in the sense that for any initial projection matrix U_0 it holds

$$\lim_{j\to\infty} ||\nabla \mathcal{J}(U_j)|| = 0. \quad (5.91)$$

Proof. The reduced systems are BIBO stable (cf. Proposition 5.5.11). Hence $\hat{A}$ in particular is stable and therefore (as we assume all Gramians to exist), the $\mathscr{H}_2$-norm of the error system can be calculated using equation (5.42). It holds $\mathcal{J}(U) = ||\Sigma^{\mathrm{err}}||^2_{\mathscr{H}_2}$ and the function $\mathcal{J}(U) = \mathrm{tr}(C^T C(P_{11} - 2P_{12}U^T + UP_{22}U^T))$, seen as a function from $\mathbb{R}^{n\times r} \to \mathbb{R}$, is smooth. As $\mathrm{St}(r,n) \subset \mathbb{R}^{n\times r}$ is an embedded submanifold of $\mathbb{R}^{n\times r}$ and $\mathrm{G}(r,n) \cong \mathrm{St}(r,n)/\mathcal{O}_r$, $\mathcal{J}$ can be seen as a smooth function on the Grassmann manifold.

Using the condition for the Armijo stepsize,

$$\mathcal{J}(U_j) - \mathcal{J}(\mathcal{U}_j(t_A)) \geq -\epsilon\delta^i\gamma \langle \nabla\mathcal{J}(U_j), -\nabla\mathcal{J}(U_j)\rangle, \quad (5.92)$$

one can conclude, that

$$\mathcal{J}(U_j) \geq \mathcal{J}(\mathcal{U}_j(t_A)) = \mathcal{J}(U_{j+1}). \quad (5.93)$$

Using the convergence analysis provided by Absil et al. ([1] 4.3.1,4.3.2), the remaining steps of the proof can be executed:

First it will be shown, that for an infinite sequence $\{U_j\}$ generated by Algorithm 5, every accumulation point of $\{U_j\}$ is a critical point of $\mathcal{J}$.

We proceed by contradiction. Let there be a subsequence $\{U_j\}_{j\in\mathcal{K}}$ which converges to an U_* with $\nabla\mathcal{J}(U_*) \neq 0$. As it holds

$$\mathcal{J}(U_j) \geq \mathcal{J}(U_{j+1}), \tag{5.94}$$

it follows that the sequence $\mathcal{J}(U_j)$ converges to $\mathcal{J}(U_*)$. Using Algorithm 5 we know that the condition

$$\mathcal{J}(U_j) - \mathcal{J}(U_{j+1}) \geq -\epsilon t_A^j \left\langle \nabla\mathcal{J}(U_j), -\nabla\mathcal{J}(U_j)\right\rangle,$$

holds. The sequence $-\nabla\mathcal{J}(U_j)$ is gradient related (cf. Definition 2.2.2) and we know that $\mathcal{J}(U_j) - \mathcal{J}(U_{j+1})$ must converge to zero, hence $\{t_A^j\}_{j\in\mathcal{K}} \to 0$. As $t_A^j = \delta^{m_j}\gamma$ is the Armijo stepsize, there exists a $\overline{j}$ such that every $\mathcal{K} \ni j \geq \overline{j}$ satisfies the Armijo condition. Hence, for $\frac{t_A^j}{\delta}$ the Armijo condition is not fullfilled and it holds:

$$\mathcal{J}(U_j) - \mathcal{J}\left(\mathcal{U}_j\left(\frac{t_A^j}{\delta}, -\nabla\mathcal{J}(U_j)\right)\right) < -\epsilon\frac{t_A^j}{\delta}\left\langle \nabla\mathcal{J}(U_j), -\nabla\mathcal{J}(U_j)\right\rangle \quad \forall j \in \mathcal{K}, j \geq \overline{j}.$$

We define $\eta_j = \frac{-\nabla\mathcal{J}(U_j)}{||-\nabla\mathcal{J}(U_j)||}$ and $\alpha_j = \frac{t_A^j||-\nabla\mathcal{J}(U_j)||}{\delta}$. It can be shown by a simple calculation that $\mathcal{U}_j\left(\frac{t_A^j}{\delta}, -\nabla\mathcal{J}(U_j)\right) = \mathcal{U}_j\left(\alpha_j, \eta_j\right)$. We define the function

$$\hat{\mathcal{J}}_{U_j} = \mathcal{J}\circ\mathcal{U}_j : \mathcal{T}_{U_j} \to \mathrm{Gr}(r, n) \tag{5.95}$$

which allows us to rewrite the inequality above as:

$$\frac{\hat{\mathcal{J}}_{U_j}(0) - \hat{\mathcal{J}}_{U_j}(\alpha_j, \eta_j)}{\alpha_j} < -\epsilon\left\langle \nabla\mathcal{J}(U_j), \eta_j\right\rangle \quad \forall j \in \mathcal{K}, j \geq \overline{j}.$$

We can now use the mean value theorem to obtain for $t \in [0, \alpha_j]$:

$$-\mathrm{D}\hat{\mathcal{J}}_{U_j}(t, \eta_j)[\eta_j] < -\epsilon\left\langle \nabla\mathcal{J}(U_j), \eta_j\right\rangle \quad \forall j \in \mathcal{K}, j \geq \overline{j}. \tag{5.96}$$

A detailed explanation of the differential can be found in the book of Absil [1]. We already stated that $\{t_A^j\}_{j\in\mathcal{K}} \to 0$. As $-\nabla\mathcal{J}(U_j)$ is gradient related and hence bounded, it holds $\{\alpha_j\}_{j\in\mathcal{K}} \to 0$ as well. Every η_j has unit norm, and therefore they belong to a compact set. Hence there exists $\tilde{\mathcal{K}} \subset \mathcal{K}$ such that $\{\eta_j\}_{j\in\tilde{\mathcal{K}}} \to \eta_*$ for η_* with $||\eta_*|| = 1$. Since the metric on the tangential space is continuous, it holds $\mathrm{D}\hat{\mathcal{J}}_{U_j}(0, \eta_j)[\eta_j] = \left\langle \nabla\mathcal{J}(U_j), \eta_j\right\rangle$ (cf. Absil [1],

Chapters 3.6 and 4.4) and $\mathcal{J}$ is smooth, we take the limit over $\tilde{\mathcal{K}}$ in (5.96) which leads to:

$$-\langle \nabla\mathcal{J}(U_*), \eta_* \rangle \leq -\epsilon \langle \nabla\mathcal{J}(U_*), \eta_* \rangle . \tag{5.97}$$

Since $\epsilon < 1$, it follows that $\langle \nabla\mathcal{J}(U_*), \eta_* \rangle \geq 0$. But as $-\nabla\mathcal{J}(U_j)$ is gradient related, one has $\langle \nabla\mathcal{J}(U_*), \eta_* \rangle < 0$ which leads to a contradiction. Hence every accumulation point of $\{U_j\}$ is a critical point of $\mathcal{J}$.

It is left to show that $\lim_{j\to\infty} ||\nabla\mathcal{J}(U_j)|| = 0$ holds.

As $\mathrm{Gr}(r,n)$ is a compact manifold, the following set is compact (cf. [1]):

$$\mathcal{L} = \{U \in \mathrm{Gr}(r,n) : \mathcal{J}(U) \leq \mathcal{J}(U_0)\}.$$

We proceed by contradiction and assume that there is a subsequence $\{U_j\}_{j\in\mathcal{K}}$ and $\sigma > 0$ such that $||\nabla\mathcal{J}(U_j)|| > \sigma$ for all $j \in \mathcal{K}$. We see that $\{U_j\}_{j\in\mathcal{K}} \subset \mathcal{L}$ and since $\mathcal{L}$ is compact the sequence has an accumulation point U_* in $\mathcal{L}$. As the gradient is continuous it follows $||\nabla\mathcal{J}(U_*)|| \geq \sigma$ and U_* is not a critical point, which contradicts the statement shown before.

□

5.5.4.4. *The sequentially quadratic approximation.* In addition to their FGFA algorithm, Xu and Zeng [68] proposed a second algorithm, which they call sequentially quadratic approximation (SQA). The idea is to find a search direction by minimizing the function

$$\tilde{\mathcal{J}}(U) = \mathrm{tr}(C^T C(P_{11} + UP_{22}^j U^T - 2P_{12}^j U^T)), \tag{5.98}$$

in every iteration j and then to project the difference of U_j and the obtained minimal matrix $\tilde{U}$ on the tangential space $\mathcal{T}_{[U_j]}\mathrm{Gr}(r,n)$ and use this projection as the new search direction. Considering the bilinear Wilson conditions as given in Theorem 5.5.1, a minimum of $\tilde{\mathcal{J}}(U)$ could be obtained by using $\tilde{U} = P_{12}^j (P_{22}^j)^{-1}$. The difference between U_j and $\tilde{U}$ shall now be used as search direction. One has to note that $[\tilde{U}] \notin \mathrm{Gr}(r,n)$ in most cases, and hence $\tilde{U} - U_j$ is a difference defined in $\mathbb{R}^{n\times r}$. Nevertheless, after projecting onto $\mathcal{T}_{[U_j]}\mathrm{Gr}(r,n)$ with $\Pi = (I_n - U_j U_j^T)$ one obtains:

$$\Delta_j = \Pi(\tilde{U} - U_j) = \tilde{U} - U_j(U_j^T \tilde{U}). \tag{5.99}$$

Using this Δ_j and the negative gradient $-\nabla\mathcal{J}(U_j)$, one can define a gradient related sequence (cf. Definition 2.2.2).

Proposition 5.5.14. *If the sequence* $(\Delta_j)_j$ *is bounded and it holds* $c_1 < ||\Delta_j||$ *and* $\frac{\langle \nabla \mathcal{J}(U_j), \Delta_j \rangle}{||\nabla \mathcal{J}(U_j)|| \cdot ||\Delta_j||} < c_2$ *with* $c_1 > 0$ *and* $c_2 \in (-1, 0)$*, then the sequence* $(\Delta_j)_j$ *is gradient related.*

Proof. Let $(U_j)_{j \in \mathcal{K}}$ be a subsequence that converges to a non critical point of $\mathcal{J}$. One needs to show that the subsequence $(\Delta_j)_{j \in \mathcal{K}}$ is bounded and it holds

$$\lim_{j \to \infty} \sup_{j \in \mathcal{K}} \langle \nabla \mathcal{J}(U_j), \Delta_j \rangle < 0. \tag{5.100}$$

As the sequence $(-\nabla \mathcal{J}(U_j))_j$ is gradient related, it holds

$$\begin{aligned} & \lim_{j \to \infty} \sup_{j \in \mathcal{K}} \langle \nabla \mathcal{J}(U_j), -\nabla \mathcal{J}(U_j) \rangle < 0 \\ \Leftrightarrow \quad & \lim_{j \to \infty} \sup_{j \in \mathcal{K}} ||\nabla \mathcal{J}(U_j)|| > 0. \end{aligned}$$

It is assumed that

$$\frac{\langle \nabla \mathcal{J}(U_j), \Delta_j \rangle}{||\nabla \mathcal{J}(U_j)|| \cdot ||\Delta_j||} < c_2,$$

with $c_2 \in (-1, 0)$. Hence we obtain

$$\begin{aligned} & \frac{\langle \nabla \mathcal{J}(U_j), \Delta_j \rangle}{||\nabla \mathcal{J}(U_j)|| \cdot ||\Delta_j||} < c_2 \\ \Leftrightarrow \quad & \langle \nabla \mathcal{J}(U_j), \Delta_j \rangle < c_2 ||\nabla \mathcal{J}(U_j)|| \cdot ||\Delta_j|| \\ \Leftrightarrow \quad & \lim_{j \to \infty} \sup_{j \in \mathcal{K}} \langle \nabla \mathcal{J}(U_j), \Delta_j \rangle < c_2 \underbrace{c_1 \lim_{j \to \infty} \sup_{j \in \mathcal{K}} ||\nabla \mathcal{J}(U_j)||}_{>0} \\ \Leftrightarrow \quad & \lim_{j \to \infty} \sup_{j \in \mathcal{K}} \langle \nabla \mathcal{J}(U_j), \Delta_j \rangle < 0. \end{aligned}$$

□

As long as $||\Delta_j|| > c_1$ and $\frac{\langle \nabla J(U_j), \Delta_j \rangle}{||\nabla J(U_j)|| \cdot ||\Delta_j||} < c_2$ are fulfilled, the generated sequence $\{U_j\}$ is gradient related. If the inequalities are not fulfilled anymore, one can keep the sequence of the U_j gradient related by taking $-\nabla J(U_j)$ as new search direction. The following Algorithm 6 can be established.

Algorithm 6 SQA for bilinear systems (bilSQA).

Input: (A, N_k, B, C), parameters $c_1 > 0$, and $c_2 \in (-1, 0)$, maxIt : maximal number of iterations.

Output: Reduced model $(\hat{A}, \hat{N}_k, \hat{B}, \hat{C})$

1: Choose a matrix $U_0 \in \mathbb{R}^{n\times r}$ such that $U_0^T U_0 = I_r$. Set $j = 0$.
2: **for** $j = 0 \to \text{maxIt} - 1$ **do**
3: Compute $P_{22}^j, Q_{22}^j, P_{12}^j, Q_{12}^j$ by solving the equations (5.69) - (5.72) for U_j.
4: Compute $\tilde{U} = P_{12}^j (P_{22}^j)^{-1}$ and calculate Δ_j.
5: Compute R_j by using equation (5.75).
6: Compute the gradient $\nabla \mathcal{J}(U_j) = R_j - U_j(U_j^T R_j)$.
7: **if** Δ_j satisfies $||\Delta_j|| > c_1$ and $\frac{\langle \nabla \mathcal{J}(U_j), \Delta_j \rangle}{||\nabla \mathcal{J}(U_j)|| \cdot ||\Delta_j||} < c_2$ **then**.
8: Compute the search direction $F_j = \Delta_j$.
9: **else**
10: Use $F_j = -\nabla \mathcal{J}(U_j)$.
11: **end if**
12: Compute $F_j = W_j \Sigma_j V_j^T$.
13: Minimize $\mathcal{J}(\mathcal{U}_j(t))$ over $t \geq 0$ where

$$\mathcal{U}_j(t) = U_j V_j \cos(t\Sigma_j) V_j^T + W_j \sin(t\Sigma_j) V_j^T. \tag{5.101}$$

14: Set $t_j = t_{\min}$ and $U_{j+1} = \mathcal{U}_j(t_j)$.
15: **end for**
16: Calculate the reduced model: $\hat{A} = U_{\text{maxIt}}^T A U_{\text{maxIt}}$, $\hat{N}_k = U_{\text{maxIt}}^T N_k U_{\text{maxIt}}$, $\hat{B} = U_{\text{maxIt}}^T B$, $\hat{C} = C U_{\text{maxIt}}$.

In this Chapter, we have reviewed and stated methods from linear MOR (Balanced Truncation, Krylov Subspace Methods and $\mathscr{H}_2$-optimal MOR), parametric MOR and bilinear MOR, with a special focus on bilinear $\mathscr{H}_2$-optimal MOR. Two main approaches for bilinear $\mathscr{H}_2$-optimal MOR have been presented. First, the interpolatory approach leading to the Bilinear Interpolatory Krylov Algorithm (BIRKA, cf. Algorithm 3, [12]) has been stated. It has been extended to systems with $E \neq I_n$, E nonsingular. Second, new algorithms for the $\mathscr{H}_2$-optimal MOR have been derived. They rely on methods from optimization on Grassmann manifolds and their main advantage is the preservation of stability. For bilinear systems with A and N_k

symmetric, both convergence and stability preservation of the algorithms have been proven. However, for non-symmetric systems this remains an open problem and can be the objective of future research.

CHAPTER 6

Challenges when applying BIRKA to thermal industrial models

In this chapter we will focus on the applicability of BIRKA to the presented thermal models. Several strategies need to be developed to overcome the challenges that accompany the adoption of a new algorithm within an industrial context. They can be found in the next sections.

6.1. Kronecker product appproximation

The original BIRKA (cf. Algorithm 3) calculates the projection matrices for model order reduction via the following Kronecker products:

$$\operatorname{vec}(V) = \left(-I_{\hat{n}} \otimes A - \Lambda \otimes E - \sum_{k=1}^{m} \widetilde{N_k}^T \otimes N_k \right)^{-1} (\tilde{B}^T \otimes B) \operatorname{vec}(I_m), \tag{6.1a}$$

$$\operatorname{vec}(W) = \left(-I_{\hat{n}} \otimes A^T - \Lambda \otimes E^T - \sum_{k=1}^{m} \widetilde{N}_k \otimes N_k^T \right)^{-1} (\tilde{C}^T \otimes C^T) \operatorname{vec}(I_p). \tag{6.1b}$$

However — for large systems — this calculation of the projection matrices V and W is not feasible due to the Kronecker product, which rapidly increases the number of the equations to be handled. Benner and Breiten [12] propose an iterative method to overcome this difficulty. For the calculation of the projection matrices, a Neumann Series is employed in the following way:

$$\begin{aligned}
\operatorname{vec}(V) &= \left(-I_r \otimes A - \Lambda \otimes E - \sum_{k=1}^{m} \tilde{N}_k{}^T \otimes N_k\right)^{-1} (\tilde{B}^T \otimes B)\operatorname{vec}(I_m) \\
&\underbrace{=}_{(\times)} \sum_{i=0}^{\infty} \left[(-I_r \otimes A - \Lambda \otimes E)^{-1}(\sum_{k=1}^{m} \tilde{N}_k{}^T \otimes N_k)\right]^i \\
&\qquad \cdot (-I_r \otimes A - \Lambda \otimes E)^{-1}(\tilde{B}^T \otimes B)\operatorname{vec}(I_m) \\
&= \underbrace{(-I_r \otimes A - \Lambda \otimes E)^{-1}(\tilde{B}^T \otimes B)vec(I_m)}_{\operatorname{vec}(V^1)} \\
&\qquad \underbrace{+ (-I_r \otimes A - \Lambda \otimes E)^{-1}(\sum_{k=1}^{m} \tilde{N}_k{}^T \otimes N_k)\operatorname{vec}(V^1)}_{\operatorname{vec}(V^2)} \\
&\qquad \cdots \underbrace{+ (-I_r \otimes A - \Lambda \otimes E)^{-1}(\sum_{k=1}^{m} \tilde{N}_k{}^T \otimes N_k)\operatorname{vec}(V^{j-1})}_{\operatorname{vec}(V^j)} + \cdots \\
&= \sum_{j=1}^{\infty} \operatorname{vec}(V^j), \qquad (6.2)
\end{aligned}$$

where $(\times)$ is only valid if $||(-I_r \otimes A - \Lambda \otimes E)^{-1}(\sum_{k=1}^{m} \tilde{N}_k{}^T \otimes N_k)||_2 < 1$ holds. In practice, the infinite sum is truncated after an appropriate number of additions. The columns of the summands V^j are now calculated without using any Kronecker products:

$$\begin{aligned}
V_i^1 &= (-\lambda_i E - A)^{-1} B \tilde{B}_i, \\
V_i^2 &= (-\lambda_i E - A)^{-1} \left(\sum_{k=1}^{m} N_k V^1 (\tilde{N}_k)_i\right),
\end{aligned}$$

$$\vdots$$

$$V_i^j = (-\lambda_i E - A)^{-1}\left(\sum_{k=1}^{m} N_k V^{j-1}(\tilde{N}_k)_i\right), \text{ for } i = 1,\ldots,r.$$

This calculation can be executed in the same way for vec(W). The same projection matrices are calculated using the Truncated BIRKA proposed by Flagg [34]. The large matrices $(-\lambda_i E - A)$ can be factorized by an LU-decomposition so that V_i^j can be calculated efficiently. In any case, approximating the Kronecker product as in (6.2) can lead to divergence if

$$||(-I_r \otimes A - \Lambda \otimes E)^{-1}(\sum_{k=1}^{m} \tilde{N}_k{}^T \otimes N_k)||_2 \geq 1.$$

It is advisable to check if this norm remains smaller than 1 during the execution of BIRKA, as divergence might lead to poor reduced order models. However, a direct calculation of the norm involves the inversion of $(-I_r \otimes A - \Lambda \otimes E) \in \mathbb{R}^{rn\times rn}$, which is not feasible for large systems due to high memory demands. Hence, the calculation of the Kronecker product has to be avoided. To this aim, we introduce the following norm estimation:

$$\begin{aligned}\left|\left|(-I_r \otimes A - \Lambda \otimes E)^{-1}\left(\sum_{k=1}^{m} \tilde{N}_k{}^T \otimes N_k\right)\right|\right|_2 &\leq ||(-I_r \otimes A - \Lambda \otimes E)^{-1}||_2 ||\sum_{k=1}^{m} \tilde{N}_k{}^T \otimes N_k||_2 \\ &\leq ||(-I_r \otimes A - \Lambda \otimes E)^{-1}||_2 \sum_{k=1}^{m} ||\tilde{N}_k{}^T \otimes N_k||_2 \\ &\underbrace{\leq}_{\text{see below}^1} ||(-I_r \otimes A - \Lambda \otimes E)^{-1}||_2 \sum_{k=1}^{m} ||\tilde{N}_k{}^T||_2 ||N_k||_2. \end{aligned} \tag{6.3}$$

If the last expression is smaller than 1, the algorithm is definitly usable. We have thus derived a sufficient condition.

The norm $||(-I_r \otimes A - \Lambda \otimes E)^{-1}||_2$ can be calculated without explicit inversion of the matrix. The following Lemmata (cf. [60] Chapter I.4 and [5] Chapter 3) will be used to establish the new result for the calculation of the corresponding norm in Proposition 6.1.3.

[1] $||M_1 \otimes M_2||_2 = ||M_1||_2 ||M_2||_2$ [44], Corollary 13.11.

Lemma 6.1.1. *For $M \in \mathbb{C}^{n\times n}$ nonsingular:*

$$||M^{-1}||_2 = \frac{1}{\min_{i=1\dots n}\sqrt{\lambda_i(M\overline{M}^T)}}.$$

Lemma 6.1.2. *For a normal matrix M:*

$$||M^{-1}||_2 = \frac{1}{\min_{i=1\dots n}|\lambda_i(M)|}.$$

By using the two Lemmata 6.1.1 and 6.1.2, we derive the following proposition, which will be used for the calculation of the norm of $(-I_r \otimes A - \Lambda \otimes E)^{-1}$.

Proposition 6.1.3. *For $A, E \in \mathbb{R}^{n\times n}$, symmetric, $D = \operatorname{diag}(d_1, \dots, d_r)$, $d_k \in \mathbb{C}$:*

$$||(-I_r \otimes A - D \otimes E)^{-1}||_2 = \frac{1}{\theta},$$

where

$$\theta = \min_{k=1\dots r} \begin{cases} |\lambda_{\min}(-A - d_k E)| & \text{for } \operatorname{Im}(d_k) = 0 \\ \sqrt{\lambda_{\min}((-A - d_k E)\overline{(-A - d_k E)}^T)} & \text{else} \end{cases}$$

Proof. The above matrix can be written as follows:

$$(-I_r \otimes A - D \otimes E) = \begin{bmatrix} -A - d_1 E & 0 & \dots & 0 \\ 0 & -A - d_2 E & \dots & 0 \\ \vdots & & & \vdots \\ 0 & \dots & 0 & -A - d_r E \end{bmatrix},$$

with $d_k \in \mathbb{C}$. For $d_k \in \mathbb{R}$ it is obvious that $(-A - d_k E)$ is normal due to A and E symmetric, and thus Lemma 6.1.2 can be used for calculating $\lambda_{\min}(-A - d_k E)$. For $d_k \in \mathbb{C}$ the eigenvalue $\lambda_{\min}((-A - d_k E)\overline{(-A - d_k E)}^T)$ is determined using Lemma 6.1.1. Taking the minimum of all calculated eigenvalues and inverting it concludes the proof. □

The calculation of $||(-I_r \otimes A - \Lambda \otimes E)^{-1}||_2$ can now be done by Proposition 6.1.3 using the MATLAB® [47] function `eigs`. For the estimation of the norm as given in equation (6.3), it remains to calculate the norms of N_k and $\tilde{N}_k$, which is done in MATLAB with the functions `normest` and `norm`, respectively.

For randomly chosen initial values, the norm estimate is possibly greater

than 1. However, as Λ and $\tilde{N}_k$ change towards their optimal values, the norm estimate improves. For this reason, at least two or three iterations should be performed to check if the norm is smaller than 1 with better approximations of Λ and $\tilde{N}_k$.

6.2. Stability

In contrast to the observations in [12], unstable systems have been encountered when applying BIRKA to industrial problems. Hence, a concept for stability preservation for the reduction with BIRKA is needed. Stability for linear and bilinear systems has been discussed in Sections 2.3.1.1 and 2.3.2.2. Whenever we speak of a linear stable system, we refer to a system with $\mathrm{Re}(\lambda_i(A,E)) < 0$ for the eigenvalues λ_i of a system.
For the special bilinear systems that result from parametric systems (cf. Section 5.3.2), it is possible to deduce a relation between the eigenvalues of the matrices A and $A+\sum_{k=1}^{m} u_k N_k$. As $N_k = 0$ for u_k resulting from the original linear inputs, only the inputs that are time independent will be taken into account and thus a comparison of the linear and bilinear eigenvalues is reasonable. In other words it holds (cf. Section 5.3.2):

$$\sum_{k=1}^{m} u_k N_k = \sum_{k=1}^{\overline{m}} u_k N_k,$$

and we use the latter for our comparison. Theorem 2.1.5 and Corollary 2.1.6, originally due to Bauer and Fike [38], allow us to show Proposition 6.2.1, providing results for the distance between the considered eigenvalues and the stability of the bilinear system in terms of the eigenvalues:

Proposition 6.2.1. *Let* $A = X\mathrm{diag}(\lambda_1, \ldots, \lambda_n)X^{-1}$ *with* $\mathrm{Re}(\lambda_i(A)) < -c < 0$ *for all* $i = 1, \ldots, n$. *If*

$$||u||_2 \sum_{k=1}^{\overline{m}} ||N_k||_2 < \frac{c}{\kappa_2(X)}, \tag{6.4}$$

then for any $j \in \{1, \ldots, n\}$, *there exists an* $i \in \{1, \ldots, n\}$ *such that*

$$|\lambda_i(A) - \lambda_j(A + \sum_{k=1}^{\overline{m}} u_k N_k)| < c.$$

In addition $\mathrm{Re}(\lambda_j(A + \sum_{k=1}^{\overline{m}} u_k N_k)) < 0$ *for* $j = 1, \ldots, n$.

Proof. With Corollary 2.1.6 one concludes:

$$\begin{aligned} |\lambda_i(A) - \lambda_j(A + \sum_{k=1}^{\overline{m}} u_k N_k)| &\leq \kappa_2(X)||\sum_{k=1}^{\overline{m}} u_k N_k||_2 \\ &\leq \kappa_2(X)||u||_2||\sum_{k=1}^{\overline{m}} N_k||_2 \\ &< c. \end{aligned}$$

Assume $\mathrm{Re}(\lambda_j(A + \sum_{k=1}^{\overline{m}} u_k N_k)) \geq 0$ for one fixed $j \in \{1, \ldots, n\}$. As $c < |\mathrm{Re}(\lambda_i(A))|$ for all $i = 1, \ldots, n$ and for j there exists i such that $|\lambda_i(A) - \lambda_j(A + \sum_{k=1}^{\overline{m}} u_k N_k)| < c$ one calculates:

$$\begin{aligned} c &< |\mathrm{Re}(\lambda_i(A))| \\ &\leq |\mathrm{Re}(\lambda_i(A))| + \mathrm{Re}(\lambda_j(A + \sum_{k=1}^{\overline{m}} u_k N_k)) \\ &= |\mathrm{Re}(\lambda_i(A)) - \mathrm{Re}(\lambda_j(A + \sum_{k=1}^{\overline{m}} u_k N_k)| \\ &\leq \sqrt{\left(\mathrm{Re}(\lambda_i(A)) - \mathrm{Re}(\lambda_j(A + \sum_{k=1}^{\overline{m}} u_k N_k))\right)^2 + \left(\mathrm{Im}(\lambda_i(A)) - \mathrm{Im}(\lambda_j(A + \sum_{k=1}^{\overline{m}} u_k N_k))\right)^2} \\ &< c, \end{aligned}$$

which leads to a contradiction. Therefore $\mathrm{Re}(\lambda_j(A + \sum_{k=1}^{\overline{m}} u_k N_k)) < 0$ holds. □

For systems with $E = I_n$ and sufficently small inputs u_k and matrices N_k (cf. (6.4)), the bilinear system remains stable and every eigenvalue of the bilinear system lies in a neighbourhood of an eigenvalue of the linear system. For E nonsingular, Proposition 6.2.1 remains valid for $E^{-1}A$ and $\sum_{k=1}^{m} u_k E^{-1} N_k$. Hence, it will be assumed that the eigenvalues of $E^{-1}A + \sum_{k=1}^{m} u_k E^{-1} N_k$ and $E^{-1}A$ are sufficiently close. This leads to the fact that stability preserving methods for the linear systems will be used, as we assume the pertubation in the eigenvalues of $E^{-1}A$ resulting from adding $\sum_{k=1}^{m} u_k E^{-1} N_k$ to be small.

6.2.1. Stability preservation using the systems Gramians. For linear systems (i.e. $N_k = 0$, $k = 1, \ldots, m$), stability can be preserved by using the following result due to Yousefi [70]. Basically, Villemagne and Skelton [66] have stated it even earlier, whereas Gugercin [39] used it in context of an interpolatory approach. Yousefi incorporated the fact that the eigenvalues of the reduced model will not exceed a certain value σ.

Proposition 6.2.2. *Given a linear stable system (A, B, C) with $\mathrm{Re}(\lambda_i(A)) < -\sigma < 0$ for $i = 1, \ldots, n$. Then for any arbitrary full row rank matrix $V \in \mathbb{R}^{n\times k}$ and $W = QV(V^TQV)^{-1}$, where $Q = Q^T > 0$ satisfies $A^TQ + QA + 2\sigma Q < 0$, the reduced model $(\hat{A}, \hat{B}, \hat{C})$ is stable and $\hat{A} = W^TAV$ satisfies $\mathrm{Re}(\lambda_i(\hat{A})) < -\sigma$ for $i = 1, \ldots, r$.*

For positive semidefinite Q, the proposition remains valid, if one assumes V^TQV to be invertible. We generalize this for a system with $E \neq I$, E nonsingular, Q positive semidefinite and $\hat{Q} = V^TE^TQEV$ nonsingular, which — up to the author's knowledge — has not been stated elsewhere.

Proposition 6.2.3. *Given a linear stable system (E, A, B, C) with E nonsingular and $\mathrm{Re}(\lambda_i(A, E)) < -\sigma < 0$ for $i = 1, \ldots, n$. Let $Q = Q^T \geq 0$ satisfy*

$$A^TQE + E^TQA + 2\sigma E^TQE \leq 0. \tag{6.5}$$

Then for any arbitrary full rank matrix $V \in \mathbb{R}^{n\times r}$ with $\hat{Q} = V^TE^TQEV$ nonsingular (and therefore $\hat{Q} > 0$), the reduced model $(\hat{E}, \hat{A}, \hat{B}, \hat{C})$ generated with

$$W = QEV(V^TE^TQEV)^{-1},$$

is stable and satisfies $\mathrm{Re}(\lambda_i(\hat{A}, \hat{E})) \leq -\sigma$ for $i = 1, \ldots, r$.

The proof of the Proposition follows exactly the proof of Yousefi (cf. Proposition 6.2.2). However, as we have introduced two generalizations — the presence of the E matrix and the non-strict Lyapunov inequality (cf. equation (6.5)) — we state it here for completeness.

Proof. Multiplying equation (6.5) with V^T and V and making use of

$$I_r = (V^T E^T Q E V)^{-T}(V^T E^T Q E V)^T = (V^T E^T Q E V)^{-1}(V^T E^T Q E V),$$

leads to:

$$\begin{aligned}
&V^T A^T Q E V + V^T E^T Q A V + 2\sigma V^T E^T Q E V \leq 0 \\
\Rightarrow &V^T A^T \underbrace{Q E V (V^T E^T Q E V)^{-T}}_{W} \underbrace{(V^T E^T Q E V)^T}_{\hat{Q}} \underbrace{(V^T E^T Q E V)^{-1}(V^T E^T Q}_{W^T} E V) \\
&+ V^T E^T \underbrace{Q E V (V^T E^T Q E V)^{-T}}_{W} \underbrace{V^T E^T Q E V}_{\hat{Q}} \underbrace{(V^T E^T Q E V)^{-1} V^T E^T Q}_{W^T} A V \\
&+ 2\sigma V^T E^T \underbrace{Q E V (V^T E^T Q E V)^{-T}}_{W} \underbrace{(V^T E^T Q E V)^T}_{\hat{Q}} \\
&\qquad \cdot \underbrace{(V^T E^T Q E V)^{-1} V^T E^T Q}_{W^T} E V \leq 0 \\
\Rightarrow &V^T A^T W \hat{Q} W^T E V + V^T E^T W \hat{Q} W^T A V + 2\sigma V^T E^T W \hat{Q} W^T E V \leq 0 \\
\Rightarrow &\hat{A}^T \hat{Q} \hat{E} + \hat{E}^T \hat{Q} \hat{A} + 2\sigma \hat{E}^T \hat{Q} \hat{E} \leq 0 \\
\Rightarrow &(\hat{A} + \sigma \hat{E})^T \hat{Q} \hat{E} + \hat{E}^T \hat{Q} (\hat{A} + \sigma \hat{E}) \leq 0.
\end{aligned}$$

Using the identity $\hat{E} = W^T E V = (V^T E^T Q E V)^{-T} V^T E^T Q E V = I_r$, let λ_i^r and v_i be any eigenvalue and eigenvector of $\hat{A} + \sigma I_r$, then:

$$\begin{aligned}
(\hat{A} + \sigma I_r)^T \hat{Q} + \hat{Q}(\hat{A} + \sigma I_r) \leq 0 \Rightarrow &v_i^* (\hat{A} + \sigma I_r)^T \hat{Q} v_i + v_i^* \hat{Q} (\hat{A} + \sigma I_r) v_i \leq 0 \\
\Rightarrow &\overline{\lambda_i^r} v_i^* \hat{Q} v_i + \lambda_i^r v_i^* \hat{Q} v_i \leq 0 \\
\Rightarrow &(\overline{\lambda_i^r} + \lambda_i^r) v_i^* \hat{Q} v_i \leq 0 \\
\Rightarrow &2\mathrm{Re}(\lambda_i^r) v_i^* \hat{Q} v_i \leq 0 \\
(v_i^* \hat{Q} v_i > 0) \Rightarrow &\mathrm{Re}(\lambda_i^r) \leq 0.
\end{aligned}$$

The eigenvalues of the reduced system are the eigenvalues of $\hat{A}$ as $\hat{E} = I_r$. Using $\lambda_i^r v_i = (\hat{A} + \sigma I_r) v_i$ this leads to $\hat{A} v_i = \lambda_i^r v_i - \sigma v_i = (\lambda_i^r - \sigma) v_i$. As $\mathrm{Re}(\lambda_i^r) \leq 0$ and $-\sigma < 0$, one can conclude that $\mathrm{Re}(\lambda_i^r - \sigma) < 0$ and therefore the reduced system is stable. □

The dual result is also true:

Proposition 6.2.4. *Given a linear stable system (E, A, B, C) with E nonsingular and $\mathrm{Re}(\lambda_i(A, E)) < -\sigma < 0$ for $i = 1, \ldots, n$, then for any arbitrary full row rank matrix $W \in \mathbb{R}^{n\times r}$ and $P = P^T \geq 0$ which satisfy*

$$APE^T + EPA^T + 2\sigma EPE^T \leq 0, \tag{6.6}$$

and nonsingular $\hat{P} = WEPE^TW^T$, the reduced model $(\hat{E}, \hat{A}, \hat{B}, \hat{C})$ generated with

$$V = PE^TW^T(WEPE^TW^T)^{-1},$$

is stable and satisfies $\mathrm{Re}(\lambda_i(\hat{A}, \hat{E})) \leq -\sigma$ for $i = 1, \ldots, r$.

Proof. The proof is analogous to the one of Proposition 6.2.3. □

For the calculation of the projection matrix W the following Lyapunov equation is solved:

$$(A + \sigma E)^T QE + E^T Q(A + \sigma E) = -C^T C \leq 0, \tag{6.7}$$

for a $\sigma < |\mathrm{Re}(\lambda_{\max}(A, E))|$. Hence one obtains

$$W = QEV(V^T E^T QEV)^{-1},$$

as in Proposition 6.2.3. The solution of the Lyapunov equation (6.7) is positive semidefinite, as the shifted system $(A+\sigma E, E)$ remains asymptotically stable.

Equation (6.7) can be solved by using the low rank ADI iteration (cf. for example [15, 57]) which generates a low rank factor Z, such that $Q \approx Z^T Z$. The calculated low rank matrix $\hat{Q} \approx V^T E^T Z^T ZEV$ can be singular. This always occurs if $\mathrm{rk}(Z) < \mathrm{rk}(V) = r$. Even if $\mathrm{rk}(V) \leq \mathrm{rk}(Z)$ one can not conclude that $V^T E^T QEV$ is nonsingular[2], but for $\mathrm{rk}(V)$ relatively small compared to $\mathrm{rk}(Z)$ it is often true.

Solving large Lyapunov equations is numerically demanding. For large systems ($n > 500,000$) it might be impossible — even with highly developed methods such as the ADI algorithm. Hence, this stability preserving method will reach its limitations when the system's dimensions get too large.

---

[2] $n = 4$, $E = I_4$, $V^T E^T Z^T = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

6.2.2. Stability preservation via one-sided projections. In the special case of symmetric matrices E,A and N_k and positive definite E, another possibility for preserving stability is to use only a single projection matrix (this is called one-sided method). The matrices of thermal systems provided in Section 4.1 have exactly these properties, and therefore this stability preservation approach is of interest.

Proposition 6.2.5 ([22]). *Given a linear system (i.e. $N_k = 0$) with A, E symmetric. If $E = E^T > 0$ and $A = A^T < 0$ then the system is asymptotically stable.*

Corollary 6.2.6 (4.4,[60]). *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix, $V \in \mathbb{R}^{n\times r}$ have orthonormal columns, and $\hat{A} = V^T AV$. Then*

$$\lambda_i(A) \geq \lambda_i(\hat{A}) \geq \lambda_{i+n-r}(A), \quad i = 1, \ldots, r. \tag{6.8}$$

Corollary 6.2.7 ([22]). *Given a linear system with $E = E^T > 0$ and $A = A^T < 0$, for $i = 1, \ldots, n$. Let $V \in \mathbb{R}^{n\times r}$ have orthonormal columns, $\hat{A} = V^T AV$ and $\hat{E} = V^T EV$, then the reduced system is asymptotically stable.*

Proof. With Corollary 6.2.6 one can conclude that the eigenvalues of the matrix $\hat{A} = V^T AV$ are negative. As $\hat{A} = \hat{A}^T$ and $V^T EV = \hat{E} = \hat{E}^T > 0$ one concludes with Proposition 6.2.5 that the reduced system is asymptotically stable. □

Hence for linear systems with A and E symmetric and E positive definite, stability can be preserved via one-sided projections. As shown in Proposition 6.2.1, the eigenvalues of a bilinear system, derived from a linear parametric system, can now be related to the eigenvalues of this linear system. Using Proposition 2.1.7 and Corollary 2.1.8, this leads to the following result:

Corollary 6.2.8. *Let $u_k \in \mathbb{R}$ for $k = 1, \ldots, \overline{m}$, $A \in \mathbb{R}^{n\times n}$ and $N_k \in \mathbb{R}^{n\times n}$ symmetric with eigenvalues $0 > \lambda_1(A) \geq \cdots \geq \lambda_n(A)$ and $\lambda_1(N_k) \geq \cdots \geq \lambda_n(N_k)$. Given that $V \in \mathbb{R}^{n\times r}$ has orthonormal columns and $\hat{A} = V^T AV$ and $\hat{N}_k = V^T N_k V$, then it holds*

$$|\lambda_i(\hat{A} + \sum_{k=1}^{\overline{m}} u_k \hat{N}_k) - \lambda_i(\hat{A})| \leq ||u||_2 \sum_{k=1}^{\overline{m}} ||N_k||_2. \tag{6.9}$$

Proof. Corollary 2.1.8 leads to

$$\begin{aligned} |\lambda_i(\hat{A} + \sum_{k=1}^{\overline{m}} u_k \hat{N}_k) - \lambda_i(\hat{A})| &\leq ||\sum_{k=1}^{\overline{m}} u_k \hat{N}_k||_2 \\ &\leq ||u||_2 \sum_{k=1}^{\overline{m}} ||\hat{N}_k||_2. \end{aligned}$$

As N_k and $\hat{N}_k$ are symmetric, they are normal and therefore fulfill

$$||\hat{N}_k||_2 = \max_{i=1,\dots,r} |\lambda_i(\hat{N}_k)| = \max\{|\lambda_1(\hat{N}_k)|, |\lambda_r(\hat{N}_k)|\},$$

and

$$||N_k||_2 = \max_{i=1,\dots,n} |\lambda_i(N_k)| = \max\{|\lambda_1(N_k)|, |\lambda_n(N_k)|\}.$$

With Corollary 6.2.6 one concludes $\lambda_1(N_k) \geq \lambda_i(\hat{N}_k) \geq \lambda_n(N_k)$. This leads to $||\hat{N}_k||_2 \leq ||N_k||_2$ and therefore equation (6.9) holds. □

If $||u||_2 \sum_{k=1}^{\overline{m}} ||N_k||_2$ is sufficiently small, one can assume that $\lambda_i(\hat{A} + \sum_{k=1}^{m} u_k \hat{N}_k) \approx \lambda_i(\hat{A})$ and therefore the reduced bilinear system is stable if the linear system is stable (cf. Corollary 6.2.7). In addition it holds:

Corollary 6.2.9. *Under the assumptions of Corollary 6.2.8 let $c \in \mathbb{R}^+$ with $c < |\lambda_{\max}(\hat{A})| = |\lambda_1(\hat{A})|$. If $||u||_2 \sum_{k=1}^{\overline{m}} ||N_k||_2 < c$ then $\lambda_i(\hat{A} + \sum_{k=1}^{\overline{m}} u_k \hat{N}_k) < 0$.*

Proof. Assume $\lambda_i(\hat{A} + \sum_{k=1}^{\overline{m}} u_k \hat{N}_k) \geq 0$ and calculate using equation (6.9):

$$\begin{aligned} c &< |\lambda_1(\hat{A})| \leq |\lambda_i(\hat{A})| \\ &\leq |\lambda_i(\hat{A})| + \lambda_i(\hat{A} + \sum_{k=1}^{\overline{m}} u_k \hat{N}_k) \\ &= |\lambda_i(\hat{A}) - \lambda_i(\hat{A} + \sum_{k=1}^{\overline{m}} u_k \hat{N}_k)| < c. \end{aligned}$$

This leads to a contradiction, so $\lambda_i(\hat{A} + \sum_{k=1}^{\overline{m}} u_k \hat{N}_k) < 0$. □

Remark 6.2.10. Using one-sided projections for the reduction of symmetric matrices, one can not only derive conditions for the eigenvalues of the bilinear rewritten parametric models (as given in 6.2.8 and 6.2.9), but also derive the BIBO stability preservation of general bilinear systems, as it has been done in Proposition 5.5.11.

6.2.3. Stability preservation - the workflow. As the reduced models that have been calculated with the stabilization process using the Gramians are in most cases better than those generated by a one-sided approach, the workflow in Figure 6.1 applies.

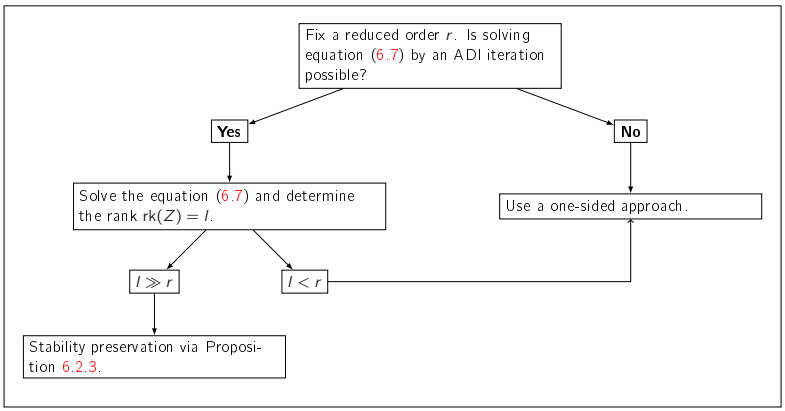


Figure 6.1. Proposed workflow for stabilization.

The reader should note that $l \gg r$ indicates the fact that the matrix $\hat{Q}$ (cf. Section 6.2.1) can still be singular, but for the case of $l \gg r$, it is more likely that $\hat{Q}$ is invertible.

Remark 6.2.11. For the reduction with these stability preserving methods, the matrix W originally given by BIRKA (cf. Algorithm 3) is not used within the reduction. Instead, either the matrix W given by Proposition 6.2.3 or simply $W = V$ (the one-sided approach) is used. This leads to the fact that the derived $\mathscr{H}_2$-optimality conditions as given in equations (5.43) to (5.47) or (5.50) to (5.53) are not completely fulfilled anymore. Only the

conditions (5.47) or (5.53) hold, as they only depend on the calculation of the matrix V.

6.2.4. Stabilization via mirroring of eigenvalues. Recently, Zeng, Chen and Lu [71] proposed a stability preservation for IRKA (cf. Algorithm 1). After reducing the model by a projection matrix generated during an IRKA step, the matrix $\hat{A} = S^{-1}\Lambda S$ (assume $\hat{E} = I_r$) is diagonalized and its unstable eigenvalues are mirrored:

$$\lambda_{\text{mir}}^j = -|\text{Re}(\lambda_j)| + \mathbf{i} \cdot \text{Im}(\lambda_j).$$

Finally, set $\hat{A} = S^{-1}\Lambda_{\text{mir}}S$ as the stable reduced matrix. In the bilinear case, this method can be used to obtain $\text{Re}(\lambda_i(\hat{A})) < 0$. However with this step, the BIBO stability will not be considered.
In Figure 6.2, results for the reduction with stabilization for different orders are compared. We reduce the simplified motor with $n = 2,952$ (cf. Section 4.3.2). However, we will not incorporate geometry variations and simply use one physical parameter (heat transfer coefficient) and three loads.
The original BIRKA (cf. Algorithm 3) is accurate for a reduced order of $r = 20$ (if a stable model has been obtained), whereas the reduction with the stabilization converges to a model, which — as it can be seen in the third output — is not a good approximation of the original. After increasing the order up to $r = 50$, BIRKA with the stabilization performs well.
If a stable reduced model is generated by a reduction, where V as given by BIRKA is used as one-sided projection, one obtains accurate results with $r = 100$. Hence, the stabilization via the mirroring of the eigenvalues can be sufficiently accurate with a smaller reduced order. Nevertheless, one needs to check if the reduced order model is accurate enough, as a reduction with the stabilization might lead to a convergence of the algorithm but still provides an inaccurate approximation of the original model.

Remark 6.2.12. This stability preservation only adresses the eigenvalues of the matrix $\hat{A}$. For a bilinear system, the BIBO stability might not be fulfilled. Hence, for the reduction of thermal models, we use the stability preservation via the one-sided projections (even if they result in larger reduced orders). They guarantee BIBO stable models, if the BIBO stability condition (as given in 2.3.24) is fulfilled for the original model. This result has been established in Proposition 5.5.11.

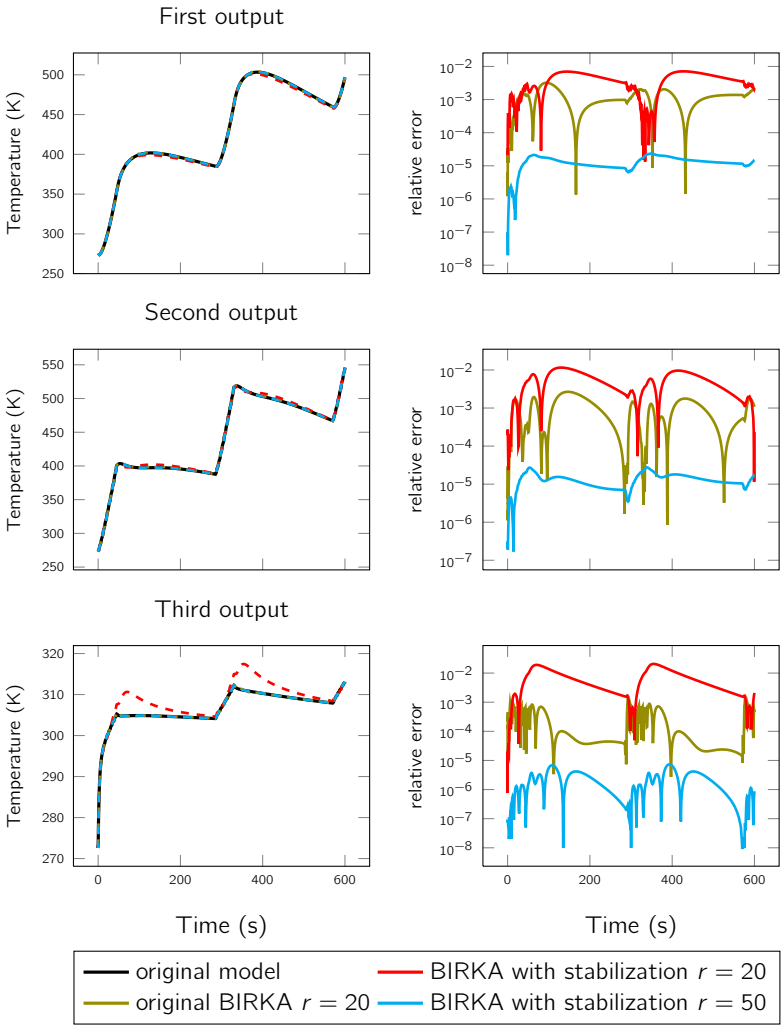


Figure 6.2. Reduction of the small motor model $n = 2,952$ using stabilization via mirroring of poles

6.3. Singular stiffness matrix A and large norm matrices N_k

6.3.1. Singular stiffness matrix A. The effect of thermal resistance between two parts can be modeled by a small gap between them (cf. Section 3.1.1). This can be done using Robin boundary conditions on the interface surfaces, and then modeling the resistance by a contact heat transfer coefficient h_c:

$$k_2 \frac{\partial T|_{l_1}(x,t)}{\partial \mathbf{n}} = -k_1 \frac{\partial T|_{l_2}(x,t)}{\partial \mathbf{n}} = h_c \left(T(x,t)\,|_{l_1} - T(x,t)|_{l_2} \right).$$

By constructing the parametrized heat equation as given in Section 4.1, this leads to the following parameter dependent stiffness matrix:

$$A + h_c N_c.$$

However, matrix A can be singular (but $A + h_c N_c$ is not) due to the following effect. Assume we are solving the heat equation for a model with two different parts, separated by a small gap (cf. Figure 3.1). In the case, where there is no heat flux between the two parts, the contact heat transfer coefficient is $h_c = 0$ and the boundary conditions become:

$$k_2 \frac{\partial T|_{l_1}(x,t)}{\partial \mathbf{n}} = -k_1 \frac{\partial T|_{l_2}(x,t)}{\partial \mathbf{n}} = 0.$$

Hence, the heat equations become

$$\rho C \frac{\partial T_1(x,t)}{\partial t} = k \Delta T_1(x,t) \text{ and } \rho C \frac{\partial T_2(x,t)}{\partial t} = k \Delta T_2(x,t),$$

on the two different parts, with $\frac{\partial T_1(x,t)}{\partial t} = \frac{\partial T_2(x,t)}{\partial t} = 0$ as the temperature is constant (i.e. $T_1(x,t) = T_1^{\text{const}}$ and $T_2(x,t) = T_2^{\text{const}}$) since no heat flux is present. The discretization of the heat equation with the boundary condition yields $A_1 T_1^{\text{const}} = 0$ and $A_2 T_2^{\text{const}} = 0$, which is only complied if A_1 and A_2 are singular matrices. In the case, where a heat flux between the two parts is present, a matrix N_c is included in the discretization and $A + h_c N_c$ is a nonsingular matrix — whereas A remains singular.

As BIRKA is not defined for systems with singular A matrix (and leads to inaccurate results when a reduction is performed), one needs to modify the original systems representation. As $A + h_c N_c$ is nonsingular, it is possible to use a shift s and obtain:

$$\begin{aligned} & A + h_c N_c \\ =& A + sN_c - sN_c + h_c N_c \\ =& \tilde{A} + \tilde{h}_c N_c \text{ where } \tilde{A} = A + sN_c \text{ and } \tilde{h}_c = h_c - s. \end{aligned}$$

One can now apply BIRKA, using the nonsingular $\tilde{A}$ instead of A. After the reduction, the calculation needs to be reversed: If $\hat{\tilde{A}}$ and $\hat{N}_c$ are the resulting reduced order matrices, one calculates: $\hat{A} = \hat{\tilde{A}} - s\hat{N}_c$. However, for a stable $\hat{\tilde{A}}$ the matrix $\hat{A}$ is not known to be stable, but one can connect the stability of $\hat{\tilde{A}}$ and $\hat{A}$ using Proposition 6.2.1, which leads to the following statement: Let $\mathrm{Re}(\lambda_i(\hat{\tilde{A}})) < -c$ and $X^{-1}\hat{\tilde{A}}X = \mathrm{diag}(\lambda_1, \ldots, \lambda_r)$. If $|s| \cdot ||\hat{N}_c||_2 < \frac{c}{\kappa_2(X)}$, then $\mathrm{Re}(\lambda_j(\hat{A})) < 0$ for all $j = 1, \ldots, r$.

6.3.2. Large norm matrices N_k. It is possible that BIRKA cannot be applied to a system where the norms of N_k are large. First of all, the Kronecker product approximation as given in Section 6.1

$$||(-I_r \otimes A - \Lambda \otimes E)^{-1}||_2 \sum_{k=1}^{m} ||\tilde{N}_k^{\,T}||_2 ||N_k||_2 < 1, \tag{6.10}$$

is not necessarily fulfilled. In addition, the BIBO stability condition[3] as given in Theorem 2.3.24

$$\sum_{k=1}^{m} ||E^{-1}N_k||_2 < \frac{\alpha}{M\beta}, \tag{6.11}$$

might not be fulfilled.

[3] With $\beta, \alpha \in \mathbb{R}$, $\beta > 0$ and $0 < \alpha \leq -\max_i(\mathrm{Re}(\lambda_i((A, E))))$ and $||e^{E^{-1}At}||_2 \leq \beta e^{-\alpha t}$, $t \geq 0$, $||u(t)|| = \sqrt{\sum_{k=1}^{m} |u_k(t)|^2} \leq M$.

One can then apply a simple scaling $g \in \mathbb{R}^+$ to the bilinear model and try to choose it such that (6.10) holds for the scaled matrices $\overline{N}_k = gN_k$.

$$\begin{aligned} & E\dot{x} = Ax + \sum_{k=1}^{m} N_k u_k x + Bu, \\ \Rightarrow \quad & E\dot{x} = Ax + \sum_{k=1}^{m} N_k \frac{g}{g} u_k x + B\frac{g}{g}u, \\ \Rightarrow \quad & E\dot{x} = Ax + \sum_{k=1}^{m} \overline{N}_k \overline{u}_k x + \overline{B}\overline{u}, \text{ with } \overline{N}_k = gN_k, \overline{B} = gB \text{ and } \overline{u} = \frac{1}{g} \cdot u. \end{aligned} \tag{6.12}$$

In addition, one might think of choosing the scaling such that the BIBO stability condition holds for the scaled model. However this is never the case:

Lemma 6.3.1. *If a bilinear system does not fulfill the BIBO stability condition* (6.11)*, the scaled system* (6.12) *does not fulfill the BIBO stability condition.*

Proof. Set $\Gamma = \sum_{k=1}^{m} ||E^{-1}N_k||_2$. It holds

$$\Gamma \cdot |g| = |g| \sum_{k=1}^{m} ||E^{-1}N_k||_2 = \sum_{k=1}^{m} ||E^{-1}gN_k||_2.$$

For the scaled input $\overline{u}$ one obtains:

$$||\overline{u}|| = ||\frac{1}{g}u|| = \frac{1}{|g|}||u|| \leq \frac{M}{|g|} := \overline{M}.$$

As the BIBO stability condition does not hold for the original system one obtains:

$$\begin{aligned} & \Gamma \geq \frac{\alpha}{M\beta}, \\ \Rightarrow \quad & \Gamma|g| \geq \frac{\alpha|g|}{M\beta} = \frac{\alpha}{\overline{M}\beta}. \end{aligned}$$

This shows, that the BIBO stability condition is not fulfilled for the scaled system as $\sum_{k=1}^{m} ||E^{-1}gN_k||_2 = \Gamma|g| < \frac{\alpha}{M\beta}$ does not hold. □

In our case, we mostly consider bilinear systems, that have been obtained by rewriting a parametric system (see Section 5.3.2) in the following

way:

$$E\dot{x} = (A + pN_1)x + Bu,$$
$$E\dot{x} = Ax + N_1\hat{u}_1x + N_2\hat{u}_2 + \cdots + N_{m+1}\hat{u}_{m+1} + \hat{B}\hat{u},$$

with $N_2 = \ldots = N_m = 0$, $\hat{B} = [0 \quad B]$, $B \in \mathbb{R}^{n\times m}$ and $\hat{u} = [p \quad u]^T$. Now the scaling g can be used in a slightly different way than for "originally" bilinear systems:

$$E\dot{x} = Ax + \overline{N}_1\overline{u}_1x + N_2\overline{u}_2 + \cdots + N_{m+1}\overline{u}_{m+1} + \overline{B}\overline{u},$$

with $\overline{N}_1 = gN_1$, $N_2 = \ldots = N_m = 0$, $[g\cdot 0 \quad B] = \overline{B} = \hat{B} = [0 \quad B]$ and $\overline{u} = \left[\frac{p}{g} \quad u\right]^T$. Hence the input u is only scaled in the entries which refer to N_1, and the matrix B is not scaled.

Using this scaling, one can not only try to scale in such a way that the Kronecker product approximation is fulfilled, but also that the BIBO stability condition is complied. This is possible if one assumes that

$$||\overline{u}|| = \sqrt{\frac{1}{|g|^2}\sum_{i=1}^{\overline{m}}|u_i|^2 + \sum_{i=\overline{m}+1}^{m}|u_i|^2} \leq \overline{M} \leq \frac{1}{|g|}||u||.$$

(In our example $\overline{m} = 1$.) Hence

$$\overline{M} \leq \frac{1}{|g|}M, \tag{6.13}$$

holds, and in addition one has

$$|g|\sum_{k=1}^{m}||E^{-1}N_k||_2 = \sum_{k=1}^{m}||E^{-1}gN_k||_2,$$

as $N_k = 0$ for $k > \overline{m}$. As the BIBO stability condition does not hold for the original model, one has:

$$\frac{\alpha|g|}{M\beta} \leq |g|\sum_{k=1}^{m}||E^{-1}N_k||_2.$$

But as (6.13) holds, it is possible that

$$|g|\sum_{k=1}^{m}||E^{-1}N_k||_2 < \frac{\alpha}{\overline{M}\beta},$$

is fulfilled (as $\frac{\alpha|g|}{\beta M} \leq \frac{\alpha}{\beta \overline{M}}$), if g is chosen in the right way.

When scaling the matrices N_k, we observed that the reduced orders increased. This effect can also be seen with linear models in the following way:

Remark 6.3.2. The scaling factor g might affect the reduced order, which has to be increased in order to obtain a good approximation of the original unscaled model.
For a further investigation of this behavior, we have introduced a scaling in a linear model (as $A = A_0 + gA_1$). A reduction of this scaled model was performed using a one-sided moment matching. The obtained matrix V_{scaled} has then been used to reduce the unscaled model ($A = A_0 + A_1$). By increasing the order of the scaled model, it has been possible to achieve a good approximation to the unscaled one.

In this chapter, several issues that occured, while applying BIRKA to large thermal models, have been examined. First, an approximation of the Kronecker product — originally due to Benner and Breiten [12] — has been presented.
Second, methods for the stability preservation of BIRKA have been derived. Assuming that the eigenvalues of the linear and the bilinear systems (obtained from parametric systems) are sufficiently close, one can use stability preservation methods for linear systems. First, a method using the system's Gramians has been transferred to systems with $E \neq I_n$ nonsingular and positive semidefinite Gramians (cf. Section 6.2.1). Second, the stability preservation using one-sided projections has been examined, and again stability preservation has been obtained for systems where the eigenvalues of the linear and bilinear/parametric system are sufficiently close. Recently, a stability preservation via mirroring of eigenvalues has been proposed by Zeng, Chen and Lu [71]. A short examination of this method has been added (cf. Section 6.2.4) — providing good results whenever the reduced order is sufficiently large. In addition one should note, as it has already been shown in Proposition 5.5.11, using one-sided projections for symmetric models leads to BIBO stable models.
Third, one needs to consider singular A matrices, which can be avoided by using shifts, and matrices N_k that have the same magnitude as the A matrix, which need to be scaled, in order to obtain good results. Results for these modifications will be presented in Chapters 7.2 and 8.

CHAPTER 7

Reduction of physically parametrized thermal models

In this chapter, we present results for the reduction of models where only physical properties are varied. This includes (contact) heat transfer coefficients (Robin boundary conditions) and fixed temperatures (Dirichlet boundary conditions). First, we consider the new bilinear $\mathscr{H}_2$-optimal algorithms derived in Section 5.5.4. They will be tested by reducing a bilinear heat equation model on a square with $n = 100$ degrees of freedom. Second, we will present results for the reduction with BIRKA (cf. Algorithm 3) and the modifications given in Chapter 6.

7.1. Results for the $\mathscr{H}_2$-optimal reduction on Grassmann manifolds

Results for the reduction with bilGFA (Algorithm 4), bilFGFA (Algorithm 5) and bilSQA (Algorithm 6) will be analyzed in this section. The derivation of the algorithms can be found in Section 5.5.4. Their main advantage is that they can preserve stability during reduction, if the original model is BIBO stable. To demonstrate their performance, the algorithms will be applied to a bilinear heat equation model on a square [26]:

$$\frac{\partial T}{\partial t} = \Delta T \qquad \text{in } \Omega = (0,1) \times (0,1),$$

$$\frac{\partial T}{\partial n} = 0.75 \cdot u_{1,2,3}(T-1) \qquad \text{on } \Gamma_1, \Gamma_2, \Gamma_3,$$
$$T = u_4 \qquad \text{on } \Gamma_4,$$

with $u_k(t) = \frac{1}{6}\cos(k\pi t)$ for $k = 1, \ldots, 4$ and one Dirichlet (on Γ_4) and three Robin boundary conditions ($\Gamma_{1,2,3}$). The discretization of the above differential equation leads to the following bilinear system:

$$\Sigma_{\text{bil}}: \quad \begin{cases} \dot{T}(t) = AT(t) + \sum_{k=1}^{4} N_k u_k(t) T(t) + Bu(t), \\ y(t) = CT(t), \end{cases} \tag{7.1}$$

with $A, N_k \in \mathbb{R}^{100\times 100}$, $B \in \mathbb{R}^{100\times 4}$ and $C \in \mathbb{R}^{100}$. We reduce the original model to order $r = 8$. The system is then BIBO stable, as the calculated A and N_k are symmetric, and $24.75 = \sum_{k=1}^{3} ||N_k||_2 < \frac{\alpha}{M\beta} = \frac{11}{\frac{1}{3}\cdot 1} = 33$ holds.
In every step of the algorithms, we are going to measure the error in the $\mathscr{H}_2$-norm as follows: First, we calculate the norm of the original model:

$$\mathcal{J}_{\text{o}} = ||\Sigma_{\text{bil}}||^2_{\mathscr{H}_2} = \text{tr}(CP_{11}C^T),$$

then after each step we calculate the $\mathscr{H}_2$-norm of the error system:

$$\mathcal{J}_{\text{err}} = ||\Sigma^{\text{err}}_{\text{bil}}||^2_{\mathscr{H}_2} = \text{tr}(CP_{11}C^T - 2CP_{12}\hat{C}^T + \hat{C}P_{22}\hat{C}^T).$$

The relative error of the system is the square root of the quotient of these norms:

$$\text{ERR}_{\text{rel}} = \sqrt{\frac{\mathcal{J}_{\text{err}}}{\mathcal{J}_{\text{o}}}}. \tag{7.2}$$

First, we apply bilFGFA (Algorithm 5). Second, we reduce with a bilinear version of the gradient flow algorithm (bilGFA, Algorithm 4). For the calculation of t_j, we use the adaptive stepsize for the linear case established by Yan and Lam [69], which turns out to be a good choice for the time stepping in our bilinear model. Third, we will compare the results with bilSQA (Algorithm 6).
We will initialize the algorithms with two different scenarios:

(I1) The matrix U_0 is obtained by generating a random matrix in $\mathbb{R}^{n\times r}$ followed by an SVD to orthogonalize the columns in order to fulfill the condition $U_0^T U_0 = I_r$. The relative $\mathscr{H}_2$-error of the starting model is $\text{ERR}_{\text{rel}} = 0.64524$.

(I2) The matrix U_0 is obtained by a reduction of the linear model (i.e. N_k are not considered) via a moment matching approach (cf. the book of Antoulas [5], Chapter 11) followed by taking only the first three columns of the projection matrix and filling the rest of the columns with basis vectors $e_1 = [1\ 0\ \dots\ 0]^T$ to e_{r-3}. Again, an orthogonalization is required to satisfy the condition $U_0^T U_0 = I_r$. The relative $\mathscr{H}_2$-error of the starting model is $\mathrm{ERR}_{\mathrm{rel}} = 0.29734$.

In addition to the initializations, we start the algorithms bilFGFA and bilSQA with different parameter choices:

(P1) $\epsilon = 0.5$, $\delta = 10^{-3}$ and $\gamma = 3, 100$.
(P2) $\epsilon = 0.9$, $\delta = 10^{-3}$, $\gamma = 420$, $c_1 = 10^{-12}$ and $c_2 = -10^{-7}$.

Remark 7.1.1. It should be noted that the choice of the parameters and of the initialization has a strong impact on the performance of the algorithms. During our analysis, several parameter choices and initializations have been tested (not only those presented here). Some of them lead to good results, others do not result in a descent of $\mathcal{J}(U)$ or require long simulation times until a minimum is reached.

After a user defined maximal number of iterations every algorithm stops. In addition, the following stopping criteria have been implemented: bilGFA stops after the 2-norm of the iterate Γ_j (cf. equation (5.82)) is smaller than a user defined tolerance, bilFGFA and bilSQA are stopped after the norm on the Grassmann manifold $||\nabla\mathcal{J}(U_j)|| = 2\mathrm{tr}(\nabla\mathcal{J}(U_j)^T\nabla\mathcal{J}(U_j))$ is smaller than a predefined tolerance. The results for the different initializations, parameter choices, stopping criteria and algorithms are summarized in Tables 7.1 and 7.2.

The results for the reduction with the initialization (I1) and different algorithms and parameter choices are shown in Figure 7.1.

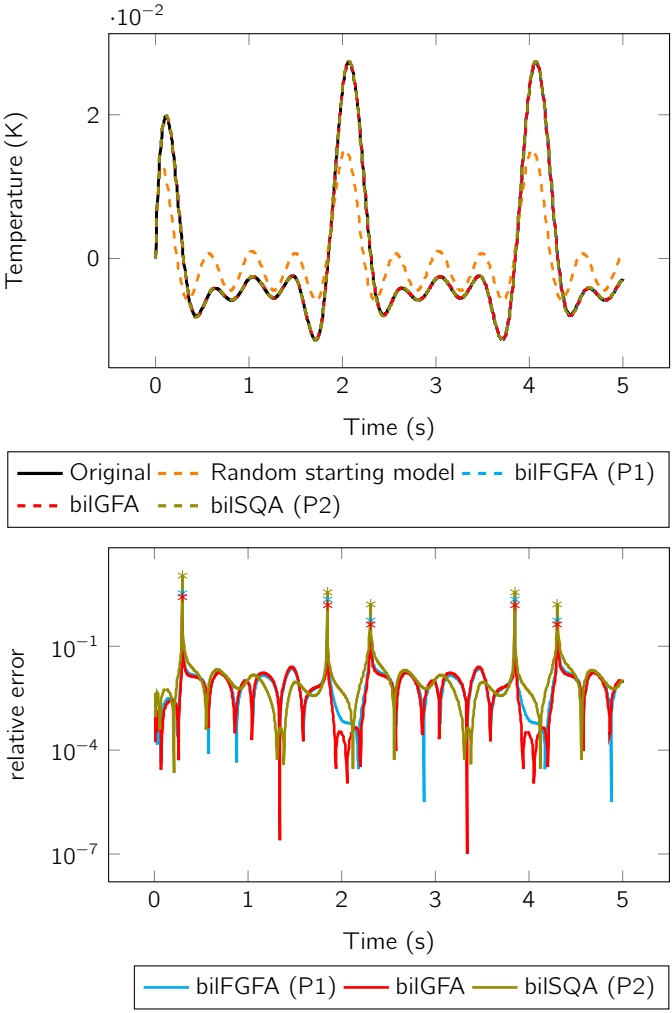


Figure 7.1. Reduction with bilGFA, bilFGFA and bilSQA for initialization (I1). Stopping criteria: $||\Gamma_j||_2 < 10^{-5}$, $||\nabla \mathcal{J}(U)|| < 10^{-9}$

The descent in the function $\mathcal{J}(U)$ for the reduction with initialization (I1) is plotted in Figure 7.2. One observes that bilSQA starts with the steepest descent — it is obtained by using Δ_j as a descent direction (cf. Section 5.5.4.4). However, after changing the direction to $-\nabla\mathcal{J}(U_j)$, the descent is smaller and can lead to large numbers of iterations depending on the stopping criterion used.

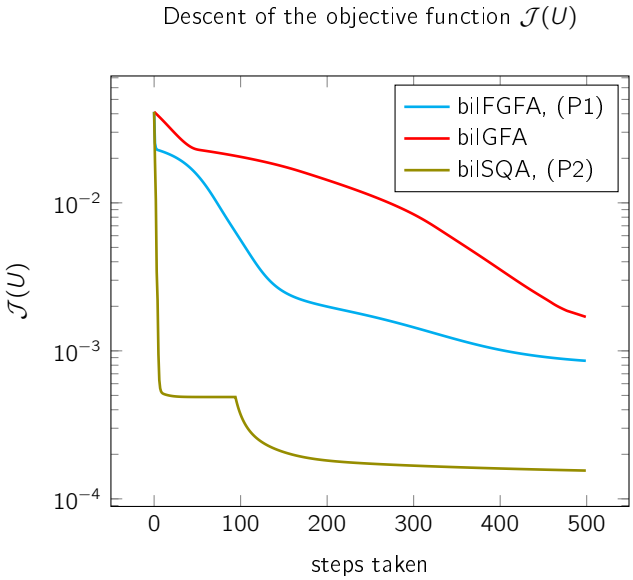


Figure 7.2. Descent in function $\mathcal{J}(U)$ for different algorithms using the initialization (I1)

Results for different stopping criteria with initialization (I1) are shown in Figures 7.3, 7.4 and 7.5. bilSQA performs best, which is consistent with Figure 7.2, where this algorithm shows the steepest descent. As given in Table 7.1 the corresponding relative $\mathscr{H}_2$-error for stopping criterion $||\nabla\mathcal{J}(U)|| < 10^{-6}$ is 0.04355. To reach comparable accuracy, more iterations and a smaller stopping criterion are required for the other two algorithms.

Table 7.1. Results using the different algorithms with initialization (I1) and different stopping criteria.

Algorithm and parameter choice	stopping criterion	number of iterations	approx. calculation time	relative $\mathscr{H}_2$-error of the final model
bilGFA	$\|\|\Gamma\|\|_2 < 10^{-2}$	49	6sec	0.484
bilGFA	$\|\|\Gamma\|\|_2 < 10^{-3}$	453	40sec	0.15143
bilGFA	$\|\|\Gamma\|\|_2 < 10^{-4}$	712	1min	0.088272
bilGFA	$\|\|\Gamma\|\|_2 < 10^{-5}$	3,415	5min	0.028826
bilFGFA (P1)	$\|\|\nabla\mathcal{J}(U)\|\| < 10^{-4}$	5	1sec	0.47982
bilFGFA (P1)	$\|\|\nabla\mathcal{J}(U)\|\| < 10^{-6}$	391	10sec	0.10271
bilFGFA (P1)	$\|\|\nabla\mathcal{J}(U)\|\| < 10^{-8}$	4,472	2min	0.037932
bilFGFA (P1)	$\|\|\nabla\mathcal{J}(U)\|\| < 10^{-9}$	9,821	4min	0.029936
bilSQA (P2)	$\|\|\nabla\mathcal{J}(U)\|\| < 10^{-4}$	8	1sec	0.075125
bilSQA (P2)	$\|\|\nabla\mathcal{J}(U)\|\| < 10^{-6}$	182	5sec	0.04355
bilSQA (P2)	$\|\|\nabla\mathcal{J}(U)\|\| < 10^{-8}$	1,771	40sec	0.037208
bilSQA (P2)	$\|\|\nabla\mathcal{J}(U)\|\| < 10^{-9}$	12,156	4min	0.035788

Table 7.2. Results using the different algorithms with initialization (I2) and different stopping criteria.

Algorithm and parameter choice	stopping criterion	number of iterations	approx. calculation time	relative $\mathscr{H}_2$-error of the final model
bilGFA	$\|\|\Gamma\|\|_2 < 10^{-2}$	9	1sec	0.27289
bilGFA	$\|\|\Gamma\|\|_2 < 10^{-3}$	107	9sec	0.14657
bilGFA	$\|\|\Gamma\|\|_2 < 10^{-4}$	776	1min	0.04425
bilGFA	$\|\|\Gamma\|\|_2 < 10^{-5}$	6,901	10min	0.028703
bilFGFA (P1)	$\|\|\nabla\mathcal{J}(U)\|\| < 10^{-4}$	4	1sec	0.26078
bilFGFA (P1)	$\|\|\nabla\mathcal{J}(U)\|\| < 10^{-6}$	316	7sec	0.10764
bilFGFA (P1)	$\|\|\nabla\mathcal{J}(U)\|\| < 10^{-8}$	3,637	1min30sec	0.041569
bilFGFA (P1)	$\|\|\nabla\mathcal{J}(U)\|\| < 10^{-9}$	5,864	2min	0.038746
bilSQA (P2)	$\|\|\nabla\mathcal{J}(U)\|\| < 10^{-4}$	5	0.2sec	0.091838
bilSQA (P2)	$\|\|\nabla\mathcal{J}(U)\|\| < 10^{-6}$	85	2sec	0.042356
bilSQA (P2)	$\|\|\nabla\mathcal{J}(U)\|\| < 10^{-8}$	1,899	40sec	0.036458
bilSQA (P2)	$\|\|\nabla\mathcal{J}(U)\|\| < 10^{-9}$	44,076	18min	0.030022

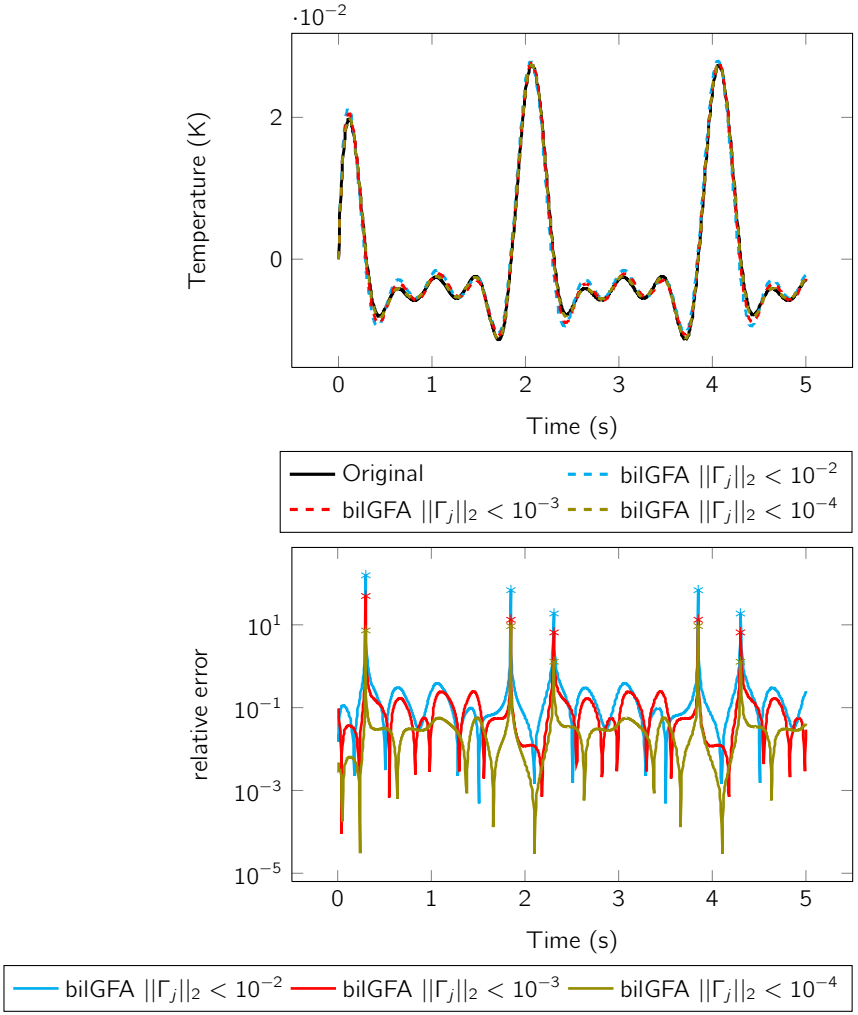


Figure 7.3. Reduction with biIGFA for initialization (I1) with different stopping criteria.

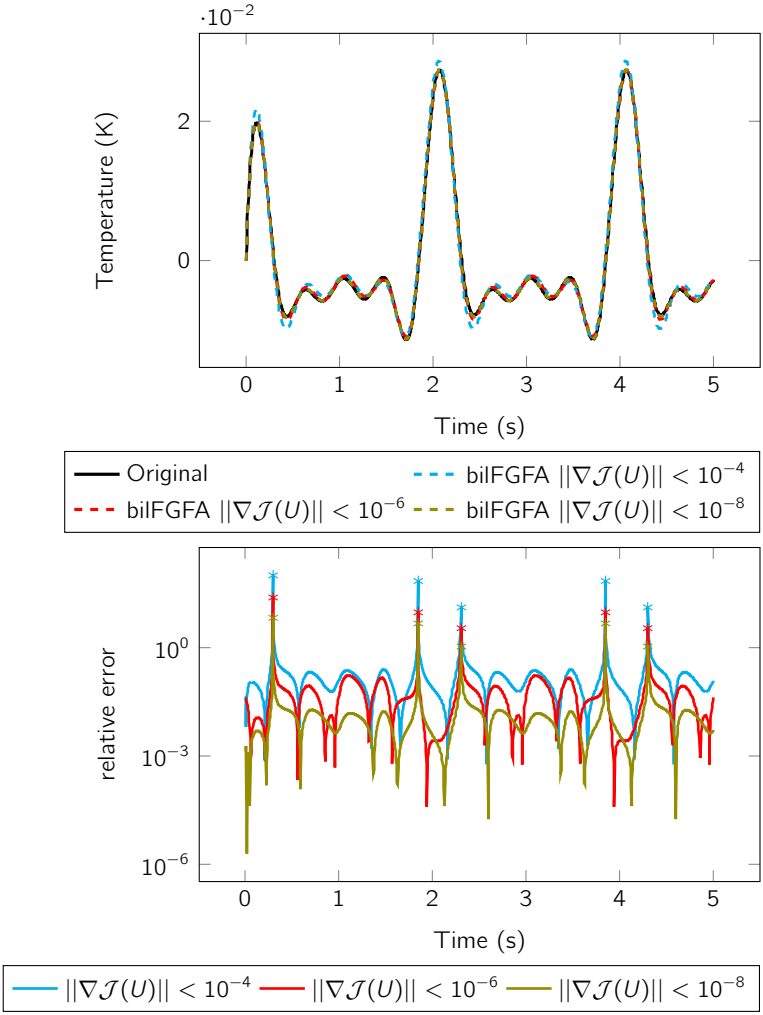


Figure 7.4. Reduction with bilFGFA for initialization (I1) with different stopping criteria.

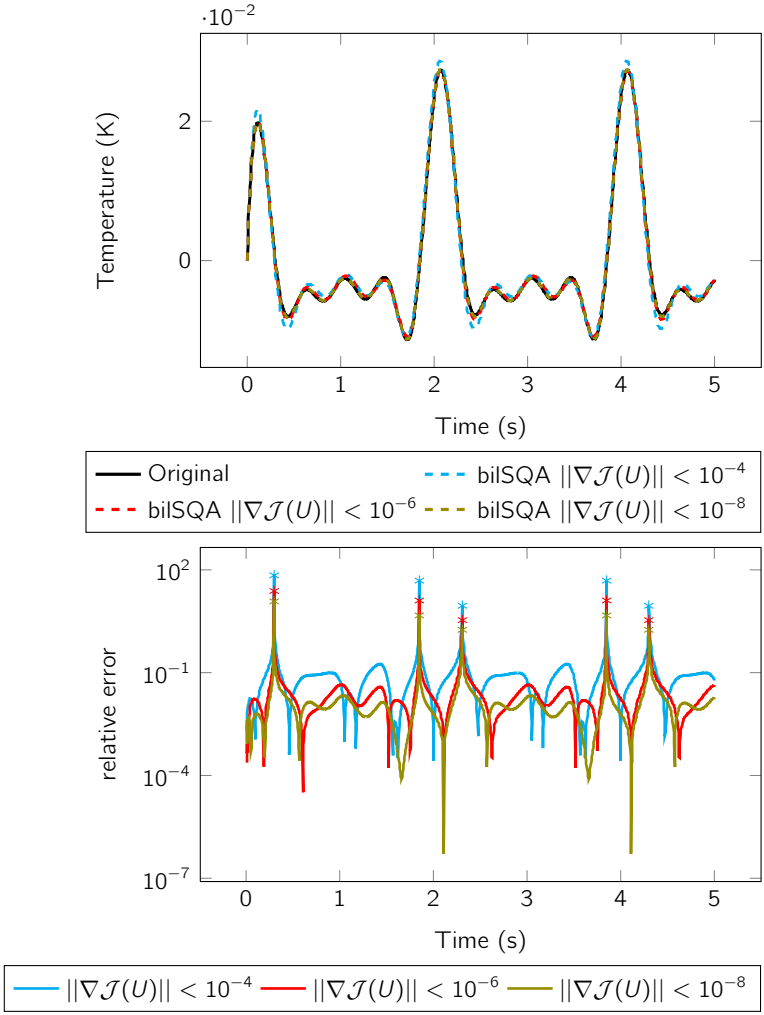


Figure 7.5. Reduction with bilSQA for initialization (I1) with different stopping criteria.

The algorithms (bilGFA, bilFGFA and bilSQA) perform well on this simple bilinear model. The quality of the resulting optimal models, however, depends on the selection of the initial matrix U_0, the stopping criteria and the optimization parameters $(\epsilon, \delta, \gamma, c_1, c_2)$. If they are not chosen carefully, it is possible that a large number of iterations is required. This can lead to long reduction times, if the algorithm is applied to larger models, even if one is able to solve the underlying Lyapunov and Sylvester equations (cf. (5.69) to (5.72)) in a reasonable amount of time.
The following open issues provide interesting opportunities for future research:

- The solution of the bilinear Lyapunov and Sylvester equations has been implemented directly. It remains open if it is possible to obtain reduced order models in a reasonable number of iterations (and hence time) using techniques for large systems (for example the ADI iteration presented among others in [57, 14]).
- For systems with symmetric A and N_k matrices, BIBO stability is preserved during the reduction, and the algorithm is converging. However, for systems where A and N_k are not symmetric it remains an open question if stability can be preserved in the reduced model.
- The derivation of an adaptive stepsize for the bilinear case might have an influence on the number of iterations and on the convergence behavior. For linear systems, Yan and Lam established Theorem 5.5.13 for their adaptive stepsize. As for bilinear systems an analogue stepsize is not yet known and the derivation of a similar theorem remains an open problem.
- In addition, one can think of finding a way to choose good optimization parameters $(\epsilon, \delta, \gamma, c_1, c_2)$ a priori. Or to update them in an efficient way during the reduction.
- The timestep in the algorithms is chosen using an Armijo condition. One might think of testing a different condition to chose the stepsize, for example a Wolfe condition on Grassmann manifolds (refer to Qi [55]).

7.2. Results for the reduction using BIRKA

In contrast to the bilinear fast gradient flow algorithm, which is not yet ready for the application to large bilinear models as shown in Section 7.1, BIRKA can be used in the context of large models. Nevertheless, several issues need to be adressed, such as stability preservation and the approximation of the Kronecker product. These issues have been discussed in Chapter 6. We will now present results for the reduction of a thermal model, where only physical properties are parametrized (cf. Section 3.3). The thermal analysis is carried out using Comsol Multiphysics®, version 3.5a [52]. By exporting several matrices from Comsol® and a thorough analysis of the underlying equations, it is possible to reconstruct a parametric model with variable parameters and loads of the form:

$$\Sigma_{\text{lin}}: \begin{cases} E\dot{T}(t) = \left(\tilde{A} + \sum_{i=1}^{q} h_i N_i + \sum_{k=q+1}^{v} (\tilde{h_c})_k N_k\right) T(t) + B \cdot \begin{bmatrix} h_1 T_\infty \\ \vdots \\ h_q T_\infty \\ T_0 \\ L(t) \end{bmatrix}, \\ y(t) = CT(t), \end{cases} \tag{7.3}$$

where q is the number of heat transfer coefficients h, and $v - q$ is the number of contact heat transfer coefficients h_c. If A is singular, it has been replaced by a non-singular matrix $\tilde{A}$ as described in Section 6.3.1.

In Figure 7.6, the modeled motor part is shown. One can see parts of stator, coil, housing and some insulation parts. The following loads and parameters need to be considered: On top of the housing a temperature T_0 is fixed to take a specified maximum temperature into account. The coil losses $L(t)$ are incorporated into the coil. Heat transfer by convection is modeled at seven different locations, for example on coil and housing, resulting in 7 heat transfer coefficients (i.e. $q = 7$). Thermal resistance is incorporated at six different locations, for example between insulation and stator or insulation and coil (i.e. $v = 13$). The size of the model is $n = 41,199$ and the original transient Comsol® simulation for one parameter setting takes about 90 minutes.

Two different models of the electrical motor have been examined. The first one considers only heat transfer coefficients as parameters and ignores the

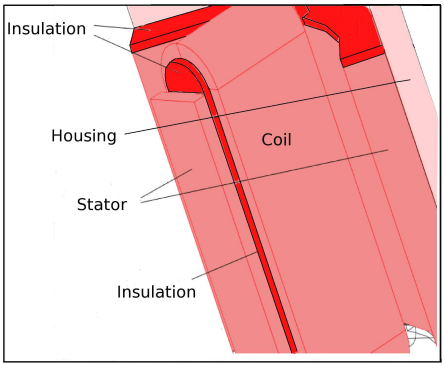


Figure 7.6. The Comsol® model for the heat transfer in a stator slice, without the rotor.

effects of thermal resistance between some parts of the motor. This leads to a model with 7 parameters and 4 loads. The second model additionally takes into account the thermal resistances and therefore contact heat transfer coefficients are considered, which leads to a model with 13 parameters and 4 loads. The temperatures at four different locations will be examined: at the front of the stator, at the coil and at two different points on the insulation.
Each of the resulting parametric systems (7.3) is reformulated as a bilinear system by following the procedure explained in [11] (cf. Section 5.3.2) and afterwards reduced using BIRKA (Algorithm 3). The calculation of the projection matrices V and W is performed as explained in (6.2) and the infinite sum is truncated after 10 summands.
The calculations were performed using MATLAB [47] on 12CPUs with 3GB RAM each.

7.2.1. Model 1 — no contact heat transfer coefficients.

7.2.1.1. *General results.* The stability of the original model is preserved by calculating the projection matrix W as described in Proposition 6.2.3. It required 16 iterations to finish the reduction, and the change in the eigenvalues between the last two iterations was less than 10^{-7}. The whole procedure

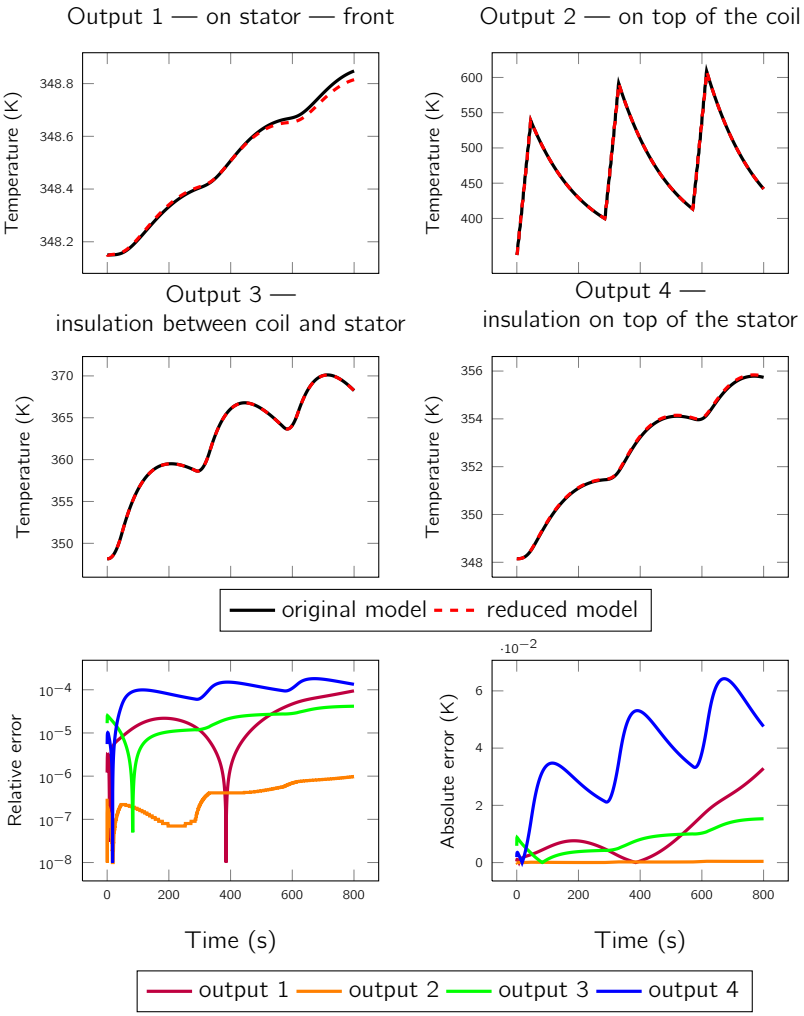


Figure 7.7. Temperatures at four locations of the motor - results of the original model compared to the reduced model and relative and absolute errors

took about 11 hours and resulted in a model of order $r = 50$ which can be simulated in 10 seconds. This is a speed-up of over 500 compared to the original simulation time of 90 minutes. Compared to the reduction time of 11 hours, the original model could have been simulated about 8 times.
When comparing the solution of the original model to the solution of the reduced model, one obtains only a small deviation, which can be seen in the error plots of Figure 7.7. The absolute error in temperature is smaller than 0.07 K, corresponding to a relative error of less than $2 \cdot 10^{-4}$. It is important to make sure that the reduced model gives reliable results over a wide range of parameter values and inputs. Simulations with the reduced model have been performed where the heat transfer coefficients are chosen from a range of 5 to 100, and the coil losses $L(t)$ and the ambient temperature T_∞ have been varied. For all these variations the reduced model gives an excellent approximation of the full model. In Figure 7.8, the behavior of the temperature for six different heat transfer coefficients on the coil is shown. The error plots on the right show that the relative and absolute errors are sufficiently small. In contrast to the standard pMOR methods (cf. Section 5.3) no training or interpolation in the parameters is required.

7.2.1.2. *Stability preserving — comparing the different approaches.* As explained in Sections 6.2.1 and 6.2.2, stability can be preserved by different procedures. Here, the following approaches will be examined:

- **gramianBIRKA:** The reduced model is calculated using V as in Algorithm 3 and equation (6.2), and the matrix W is calculated using Proposition 6.2.3. Results are shown in Figures 7.7 and 7.8.
- **BIRKA-tS:** The projection matrices V and W are calculated with Algorithm 3 and equation (6.2). Stability is not preserved. Hence, in every step of the iterative process the generated reduced system is saved, and a stable system is chosen from these systems. Such a stable system does not always exist, and even if it does, it is possibly not an optimal reduced system, as it is not necessarily the final reduced system.
- **BIRKA-oS:** The reduced model is calculated with Algorithm 3 and equation (6.2). Only in the last step a one-sided projection with V is used.
- **only V:** The projection matrix V is calculated as in Algorithm 3 and equation (6.2). In every step of the algorithm a one-sided projection is used to calculate the reduced model.

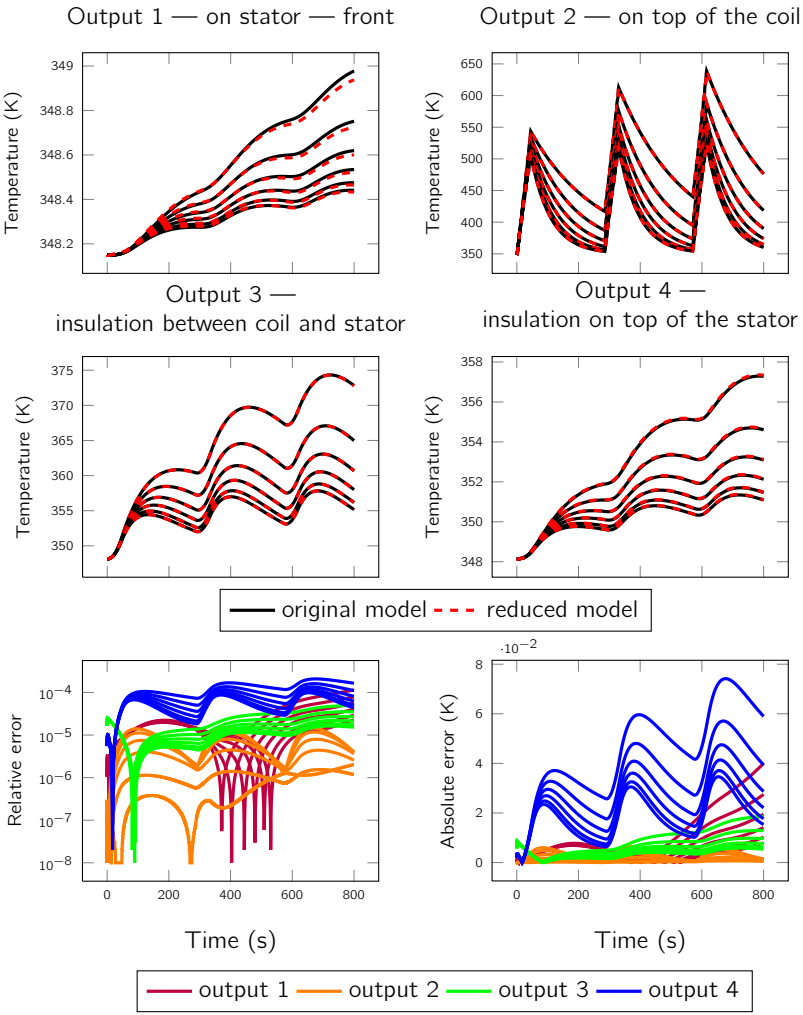


Figure 7.8. Temperature curve for six different values $(5, 25, 45, 65, 85, 100[W/m^2K])$ of the heat transfer coefficient on the coil together with the relative and absolute errors between original and reduced order models.

For the outputs on top of the coil (output 2) and on the insulation between coil and stator (output 3), all approaches give a sufficient accuracy for a reduced order of $r = 40$. However, for the outputs on the stator-front (output 1) and on the insulation on top of the stator (output 4) the results differ. For the original BIRKA (**BIRKA-tS**), good results for all outputs are obtained for $r = 40$ if a stable model is found. The **gramianBIRKA** performs well for $r = 50$ (see Figures 7.7 and 7.8). For the two one-sided approaches, the order needs to be increased up to $r = 60$ for **BIRKA-oS** and up to $r = 100$ for **only V** to obtain accurate models (cf. Figure 7.9). The calculation in **BIRKA-tS** uses two projection matrices V and W, such that the optimality conditions hold. All important informations about the original model are provided by these matrices, and then transferred to the reduced order model. The three other methods will only use V in their reduction, whereas the information contained in W is lost. **BIRKA-oS** calculates matrices V and W in every step. In the last step V is used as a one-sided projection to obtain a stable reduced order model. Hence the information given by V and W is present during the calculation and gets lost only in the last step. The **gramianBIRKA** gets information not only from V as given by the original BIRKA, but also from the solution of the Lyapunov equation (6.7) whose solution Q is used for the calculation of W as given by Proposition 6.2.3. For this reason these methods perform well for $r = 60$ and $r = 50$ respectively. The method **only V** however, uses least information, as in every step of the original BIRKA only V is used for a one-sided reduction.

Table 7.3. Comparison of simulation times and reduction times for the second model

reduced order	approach	simulation time of reduced model	reduction time	speed-up
$r = 600$	**only V**	60s	3 days 3 hours	90
$r = 300$	**BIRKA-tS** after only V	15s	3d 3h + 12h	300
$r = 300$	**BIRKA-oS** after only V	15s	3d 3h + 12h	300

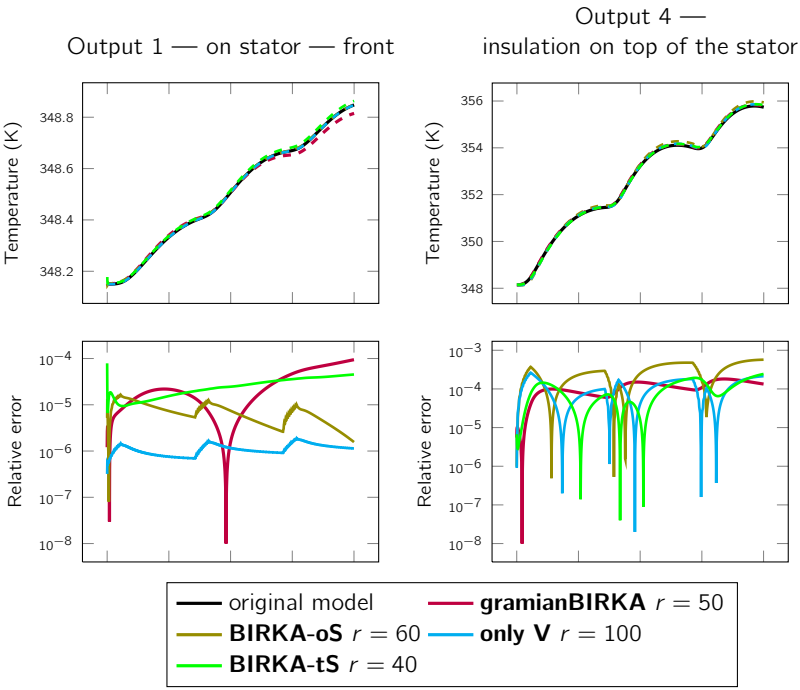


Figure 7.9. One-sided methods.

7.2.2. Model 2 — contact heat transfer coefficients. In the second model thermal resistance has been taken into account. Six additional contact heat transfer coefficients h_c are incorporated into the model. Their values range from $200\frac{W}{m^2K}$ up to $3,600\frac{W}{m^2K}$. These parameters can lead to a singular matrix A, and a shift s needs to be introduced to obtain a nonsingular matrix $\tilde{A} = A + sN$ as explained in Section 6.3.1. For every given $h_c \in [h_c^{min}, h_c^{max}]$, the center of the interval is chosen as a shift.
For this model, the stability preservation using Proposition 6.2.3 is not applicable, because the size of a reduced model will be larger than the rank of the low rank factor in the ADI iteration. Hence, stability can only be preserved using a one-sided projection. This leads to larger reduced orders

compared to an unmodified BIRKA.
For the reduction, the following approaches are used:

- **only V:** The projection matrix V is calculated as in Algorithm 3 and equation (6.2). In every step of the algorithm a one-sided projection is performed to calculate a reduced model.
- **BIRKA-tS** after only V: The projection matrices V and W are calculated with Algorithm 3 and equation (6.2) from a reduced model generated by **only V**. Stability is not preserved. Hence, in every step of the iterative process the generated reduced system is saved, and a stable system is chosen from these systems. This stable system does not always exist, and even if it does, it is possibly not an optimal reduced system, as it is not necessarily the final reduced system.
- **BIRKA-oS** after only V: The reduced model is calculated with Algorithm 3 and equation (6.2) out of a reduced model generated by **only V**. Only in the last step a one-sided projection with V is used.

The reduction was performed using the one-sided approach **only V** and took about 3 days and 3 hours. The reduced model has order $r = 600$ and can be simulated within 60 seconds, which corresponds to a speed-up of about 90 compared to the original simulation time of 90 minutes. This reduced model leads to a good approximation of the original model over the whole parameter range. This is illustrated for instance in Figure 7.10, where the variation of the heat transfer coefficient on the coil is shown. The two approaches **BIRKA-tS** and **BIRKA-oS** use the reduced model calculated with **only V** and reduce it again. This two step reduction has been done for the following reason: The larger the reduced order gets, the more unstable models are obtained within the reduction process. Hence choosing a stable model from the obtained reduced order models (as it is done in **BIRKA-tS**) is difficult, and stable models are in most cases not a good approximation to the original. In addition, a stabilization after the reduction (as it is done in **BIRKA-oS**) has the same problem – good approximations to the original model are rare. Hence, after this additional reduction process, which takes 12 hours, models of order $r = 300$ are obtained. These models can be simulated in 15 seconds, which corresponds to a speed-up of over 300 compared to the original simulation time. A summary of these results can

be found in Table 7.3. Figure 7.11 shows results for the original and the reduced models from the different approaches and the errors for output 3, which are the largest errors that occure. The reduced models generated with **only V** and **BIRKA-tS** show sufficient accuracy, whereas **BIRKA-oS** performs not accurate enough.

7.2.3. Discussion of the results. As given in Chapter 6, several issues were encountered when using BIRKA (Algorithm 3) for linear parametric models. The solution for the first issue — the approximation of the Kronecker product, cf. Section 6.1 — is used for all given reductions. In addition, the matrix A needs to be shifted to obtain a nonsingular $\tilde{A}$ (cf. Section 6.3.1) for the second model with contact heat transfer coefficients. The third issue had the largest effect on the reduction: The stability of the reduced order models needs to be preserved. Several strategies have been presented in Section 6.2 and examined on different models in this section. All stability preservation strategies can be used for the first model, whereas the strategy using the Gramian (Section 6.2.1) is not applicable for the second model.
It is found that with these strategies it is always possible to obtain stable reduced order models which give accurate results over a large parameter range (cf. Figures 7.8 and 7.10). This is possible without any sampling of the parameter space or interpolation between reduced order models, which is the standard approach for the reduction of parametric models (cf. Section 5.3). These small parametric models can therefore be used efficiently for optimization, where a large number of simulations for different parameter values is required.
For the second model, the model can be reduced down to an order of $r = 300$. This is, compared to the first model with orders from $r = 40$ to $r = 100$, relatively large. This might be due to the fact that the behavior in six additional parameters needs to be taken into account, and the matrix A needs to be shifted as well. In addition, the one-sided approach for the stabilization leads to higher reduced orders as observed also for the first model.

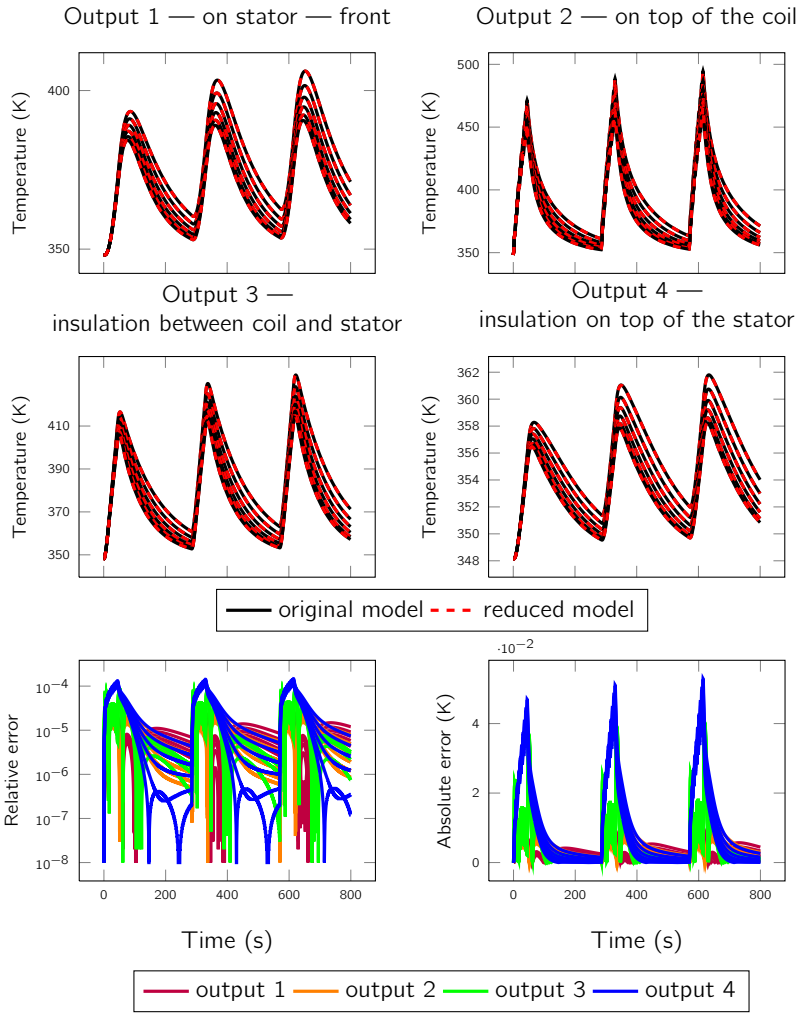


Figure 7.10. Temperature curves for six different values $(5, 25, 45, 65, 85, 100[W/m^2K])$ of the heat transfer coefficient on the coil, and the relative and absolute errors

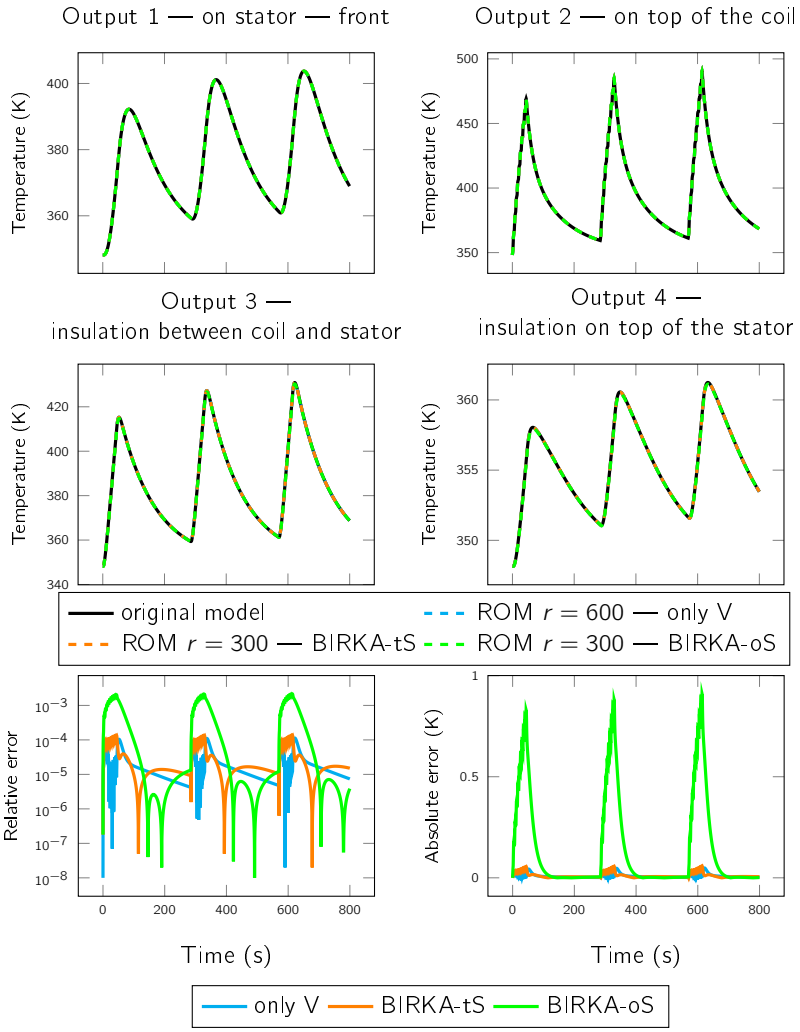


Figure 7.11. Results for reduction of model 2 with different approaches, together with the errors for output 3, the most sensitive of the outputs.

7.2.3.1. *Reduction times.* The main disadvantage of the approach are the long reduction times. This is due to the fact, that for every step of the reduction, several time-consuming calculations need to be performed.

In every step of the algorithm, the matrices V and W need to be calculated by the following formulas:

$$V_i^1 = (-\lambda_i E - A)^{-1} B \tilde{B}_i,$$
$$W_i^1 = (-\lambda_i E - A)^{-T} C^T \tilde{C}_i, \quad i = 1, \ldots, r,$$

and for $j = 2, \ldots,$ maxiterS (using the Kronecker product approximation, c.f. Section 6.1)

$$V_i^j = (-\lambda_i E - A)^{-1} \sum_{k=1}^{m} N_k V^{j-1} (\tilde{N}_k)_i,$$
$$W_i^j = (-\lambda_i E - A)^{-T} \sum_{k=1}^{m} N_k W^{j-1} (\tilde{N}_k)_i, \quad i = 1, \ldots, r.$$

The crucial point is that V^{j-1} and W^{j-1} are required in the calculation of V^j and W^j and have to be calculated a priori, so the inversions of $(-\lambda_i E - A)$ and $(-\lambda_i E - A)^T$ need to be performed $r \cdot (\text{maxiterS} + 1)$ times.

For the calculations presented in this chapter, the inversion of the matrices $(-\lambda_i E - A)$ and $(-\lambda_i E - A)^T$ was done using an LU-factorization. In every step of the algorithm, r LU-factorizations are performed, and all the matrices L_i and U_i are stored. The columns of the matrices V and W are obtained in the following way: Calculate the r columns of V^1 and W^1 by:

$$V_i^1 = (-\lambda_i E - A)^{-1} B \tilde{B}_i = U_i^{-1} L_i^{-1} B \tilde{B}_i,$$
$$W_i^1 = (-\lambda_i E - A)^{-T} C^T \tilde{C}_i = L_i^{-T} U_i^{-T} C^T \tilde{C}_i. \quad i = 1, \ldots, r$$

For all $j = 2, \ldots,$ maxiterS the r columns of V^j and W^j are then calculated by:

$$V_i^j = U_i^{-1} L_i^{-1} \sum_{k=1}^{m} N_k V^{j-1} (\tilde{N}_k)_i,$$
$$W_i^j = L_i^{-1} U_i^{-T} \sum_{k=1}^{m} N_k W^{j-1} (\tilde{N}_k)_i, \quad i = 1, \ldots, r.$$

However using MATLAB, the LU-factorization is not the fastest possibility for the calculation of $A^{-1}b$ (if b is a vector). We will now compare reduction times for the approach using the LU-factorization, and the direct calculation of $A^{-1}b$ via the "backslash" (written as $A\backslash b$) functionality in MATLAB.

We consider the model with $n = 41,199$ and 13 parameters and 4 loads and the following assumptions:

1) The sum $V = \sum_{j=1}^{\infty} \mathrm{vec}(V^j)$ (c.f. 6.2) is truncated at $j = \mathrm{maxiterS} = 10$ (W is handled in the same way).
2) The algorithm is assumed to converge after 15 steps and the reduced order is $r = 300$.

Using the LU-factorization, one observes for the calculation of one step (results may differ depending on the memory and CPUs available):

- 300 LU-factorizations need to be calculated and saved. Each LU-factorization, takes about 6 seconds, and hence in total 30 minutes.
- The $2r \cdot \mathrm{maxiterS}$ columns of V and W need to be calculated using the matrices L_i and U_i. For one column, this takes 0.7 second and hence for $2r \cdot \mathrm{maxiterS} = 6000$ this takes $6000 \cdot 0.7\mathrm{sec} = 70\mathrm{min}$.

So the total calculation time for one step is approximately 100min. After convergence (15 steps, i.e. $15 \cdot 100\mathrm{min} = 25\mathrm{h}$) this leads to an overall calculation time of more than a day[1].

Using the "backslash" implemented in MATLAB, one observes for the calculation of one step (results may differ depending on the memory and CPUs available):

- The $2r \cdot \mathrm{maxiterS}$ columns of V and W need to be calculated. If one column requires 0.4sec, one obtains: $2 \cdot 300 \cdot 10 \cdot 0.4 = 2400\mathrm{sec} = 40\mathrm{min}$.

Hence the total calculation time for one step is approximately 40min. Until convergence (15 steps) on needs $15 \cdot 40\mathrm{min} = 600\mathrm{min} = 10\mathrm{h}$ of time. For this example, the "backslash" functionality implemented in MATLAB

[1] The large reduction times of more than 3 days mentioned in table 7.3 depend on the following: First, extra LU-factorizations for $(-\lambda_i E - A)^T$ have been calculated, which are not necessary as those of $(-\lambda_i E - A)$ can be used. Second, more steps than the described 15 steps have been used. Third, the reduction order used is $r = 600$, which leads to more inversions.

only needs 40% of the reduction time, than the calculation with the LU-factorization.

However, for larger models (around $n > 100,000$), where loading the matrices L_i and U_i is faster than the calculation of $A\backslash b$, it can be beneficial to use the LU-factorization. All calculations for the small models ($n = 2,952$) in the upcoming sections are done using the "backslash" functionality in MATLAB.

Opportunities for further improvement open up for the parallelization of the calculation of the LU-factorizations and the columns, as each factorization and column can be calculated independently from the others. Depending on the number of available parallel slots, several factorizations and columns can be calculated at the same time, hence the overall process of the reduction can be sped up.

CHAPTER 8

Reduction of thermal models with geometric variations

In this chapter, two models of an electrical motor with geometric variations will be considered. The first one is a large model with $n = 71,978$ degrees of freedom, the second one — with a less complex geometry for the ease of presentation — is a smaller model with $n = 2,952$ degrees of freedom (cf. Section 4.3.2). The geometric variations are described by using affine parameters μ and θ (scaling of flange and housing in z-direction), and non-affine parameters γ and ρ (scaling of housing and stator in (x, y)-plane)[1]. One physical parameter — a heat transfer coefficient h on the housing — will be considered here (for more details on the model see Section 4.3). As in the previous chapter, the reduction will be performed using BIRKA, and stability preservation is obtained by using a one-sided approach (cf. Section 6.2 and Section 7.2 as **only V**). Due to the geometric variations, the parametric linear models have a different structure than models

[1]Strictly speaking, γ and ρ are non-affine for the dependency in A (cf. Section 4.3.2), but not for the other matrices. For the ease of presentation and calculation, we refer to and treat them as non-affine parameters.

with only physical parameters. Two different approaches for reformulating linear parametric models as bilinear models (cf. 5.3.2) will be introduced in this chapter. In addition to the reformulation step, it will be necessary to interpolate the bilinear models as the dependence on the parameters stays present. This will be done by using standard interpolation methods known from the pMOR literature as described in Section 5.3.1.

8.1. Reformulation of the linear parametric as bilinear systems

The models of the electrical motors with geometric variations can be described by the following linear parametric system:

$$\Sigma_{\text{lin,p}} : \begin{cases} E(\theta,\mu,\gamma,\rho)\dot{x}(t) \\ \qquad = \left(A(\theta,\mu,\gamma,\rho) + hA_h(\theta,\gamma,\rho)\right)x(t) + B(\theta,\mu,\gamma,\rho)u(t), \\ y(t) = Cx(t). \end{cases} \tag{8.1}$$

The parameters are: μ and θ (scaling of flange and housing in z-direction), γ and ρ (scaling of housing and stator in radial direction), and a heat transfer coefficient h (on the housing). The parameter dependent matrices are:

$$\begin{aligned} E(\theta,\mu,\gamma,\rho) &= E_0(\gamma,\rho) + \theta E_\theta(\gamma,\rho) + \mu E_\mu(\gamma,\rho), \\ A(\theta,\mu,\gamma,\rho) &= A_0(\gamma,\rho) + \frac{1}{1+\theta}A_{\frac{1}{1+\theta}}(\gamma,\rho) + \theta A_\theta(\gamma,\rho) \\ &\qquad + \frac{1}{1+\mu}A_{\frac{1}{1+\mu}}(\gamma,\rho) + \mu A_\mu(\gamma,\rho), \\ A_h(\theta,\gamma,\rho) &= A_{h0}(\gamma,\rho) + \theta A_{h\theta}(\gamma,\rho), \\ B(\theta,\mu,\gamma,\rho) &= \Bigg[\frac{1}{1+\mu}B_{\frac{1}{1+\mu}}(\gamma,\rho) + \mu B_\mu(\gamma,\rho) + B_0(\gamma,\rho) \\ &\qquad B_{h0}(\gamma,\rho) + \theta B_{h\theta}(\gamma,\rho) \quad (1+\theta)B_S(\gamma,\rho)\Bigg], \\ u(t) &= \begin{bmatrix} T_0 & hT_\infty & S(t) \end{bmatrix}^T. \end{aligned} \tag{8.2}$$

These equations show, that the parameters θ and μ (resulting from originally linear scalings in the model cf. Section 4.3) are affine, where as the parameters γ and ρ are not (resulting originally from non-linear scalings cf. Section 4.3). This parametrized linear model can now be reformulated as a bilinear model in two different ways.

8.1.1. Complete reformulation as a bilinear model (R1). We want to make use of the special structure that allows us to reduce a parametric model as a bilinear model. Here, the structure (cf. (8.1) and (8.2)) is however slightly different from the one described in Section 5.3.2. The matrix E depends on the parameters and not all parameters are affine. Hence, we can only rewrite the system as a bilinear system with a parameter dependency in $E(\theta, \mu, \gamma, \rho)$ and non-affine dependencies (parameters γ, ρ) in the other matrices. For our first approach, we will fix h and consider only the parameter dependency in geometry:

$$\begin{aligned}(E_0(\gamma,\rho) + \theta E_\theta(\gamma,\rho) + \mu E_\mu(\gamma,\rho))\,\dot{x}(t) \\ = A_0(\gamma,\rho)x(t) + \sum_{k=1}^{m} N_k(\gamma,\rho)u_k(t)x(t) + B(\gamma,\rho)u(t), \\ y(t) = Cx(t),\end{aligned}$$

with

$$\begin{aligned}u(t) = \Bigg[&\frac{1}{1+\theta} \quad \theta \quad \frac{1}{1+\mu} \quad \mu \quad T_0 \\ &\frac{1}{1+\mu}T_0 \quad \mu T_0 \quad T_\infty \quad \theta T_\infty \quad (1+\theta)S(t)\Bigg]^T,\end{aligned}$$

$$\begin{aligned}A_0(\gamma,\rho) &= A_0(\gamma,\rho) + hA_{h0}(\gamma,\rho), \\ N_1(\gamma,\rho) &= A_{\frac{1}{1+\theta}}(\gamma,\rho), \qquad N_2(\gamma,\rho) = A_\theta(\gamma,\rho) + hA_{h\theta}(\gamma,\rho), \\ N_3(\gamma,\rho) &= A_{\frac{1}{1+\mu}}(\gamma,\rho), \qquad N_4(\gamma,\rho) = A_\mu(\gamma,\rho), \\ N_5(\gamma,\rho) &= \cdots = N_{10}(\gamma,\rho) = 0,\end{aligned}$$

$$\begin{aligned}B(\gamma,\rho) = \Bigg[&\begin{matrix}0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0\end{matrix} \quad B_0(\gamma,\rho) \quad B_{\frac{1}{1+\mu}}(\gamma,\rho) \quad B_\mu(\gamma,\rho) \\ &hB_{h0}(\gamma,\rho) \quad hB_{h\theta}(\gamma,\rho) \quad B_S(\gamma,\rho)\Bigg].\end{aligned}$$

Throughout this chapter, we will refer to this reformulation as reformulation one **(R1)**.

Using this reformulation, it is possible that the norms of the matrices N_1 to N_4 are of the same magnitude as the norm of A_0. This can lead to the fact that the BIBO stability condition (cf. Theorem 2.3.24) is not fulfilled, which means that the system is possibly not BIBO stable. In addition, for the reduction of the system with BIRKA it is crucial that the system fulfills the condition (cf. Section 6.1)

$$||(I_r \otimes A_0 - \Lambda \otimes E)^{-1} \left(\sum_{k=1}^{m} \tilde{N}_k^T \otimes N_k \right) ||_2 < 1,$$

as the Kronecker product needs to be approximated. If the norm is larger than one, the algorithm might show no convergence behavior. To overcome these difficulties, the N_k can be scaled with an appropriate scaling factor g (cf. Section 6.3.2). This leads to the reduction of the following system:

$$\begin{aligned}(E_0(\gamma,\rho)+\theta E_\theta(\gamma,\rho)+\mu E_\mu(\gamma,\rho))\dot{x}(t) \\ = A_0(\gamma,\rho)x(t) + \sum_{k=1}^{m} g N_k(\gamma,\rho) u_k^g(t) x(t) + B(\gamma,\rho)u^g(t), \\ y(t) = Cx(t),\end{aligned}$$

with matrices A_0, N_k and B given as above, and

$$u^g(t) = \left[\frac{1}{g(1+\theta)} \quad \frac{\theta}{g} \quad \frac{1}{g(1+\mu)} \quad \frac{\mu}{g} \quad T_0 \right. \\ \left. \frac{1}{1+\mu} T_0 \quad \mu T_0 \quad T_\infty \quad \theta T_\infty \quad (1+\theta)S(t) \right]^T.$$

8.1.2. Incomplete reformulation as bilinear model (R2). For the second approach, the transformation into a bilinear model will only be conducted for the physical parameter h, whereas the dependency on the geometry will be regarded as a parameter dependency in a bilinear model. This leads to

the following bilinear, parametric system:

$$\Sigma_{\text{bilin}}(\mathfrak{p}): \begin{cases} E(\mathfrak{p})\dot{x}(t) = A(\mathfrak{p})x(t) + \sum_{k=1}^{4} N_k(\mathfrak{p})u_k(t)x(t) + B(\mathfrak{p})u(t), \\ y(t) = Cx(t), \end{cases} \tag{8.3}$$

where $\mathfrak{p} = (\theta, \mu, \gamma, \rho)$. The matrices are as follows:

$$\begin{aligned} E(\mathfrak{p}) &= E_0(\gamma,\rho) + \theta E_\theta(\gamma,\rho) + \mu E_\mu(\gamma,\rho), \\ A(\mathfrak{p}) &= A_0(\gamma,\rho) + \frac{1}{1+\theta} A_{\frac{1}{1+\theta}}(\gamma,\rho) + \theta A_\theta(\gamma,\rho) \\ &\qquad + \frac{1}{1+\mu} A_{\frac{1}{1+\mu}}(\gamma,\rho) + \mu A_\mu(\gamma,\rho), \\ N_1(\mathfrak{p}) &= A_{h0}(\gamma,\rho) + \theta A_{h\theta}(\gamma,\rho), \\ N_2(\mathfrak{p}) &= \cdots = N_4(\mathfrak{p}) = 0, \\ B(\mathfrak{p}) &= \begin{bmatrix} 0 & \frac{1}{1+\mu} B_{\frac{1}{1+\mu}}(\gamma,\rho) + \mu B_\mu(\gamma,\rho) + B_0(\gamma,\rho) \\ & B_{h0}(\gamma,\rho) + \theta B_{h\theta}(\gamma,\rho) \quad (1+\theta) B_S(\gamma,\rho) \end{bmatrix}, \\ u(t) &= \begin{bmatrix} h & T_0 & hT_\infty & S(t) \end{bmatrix}^T. \end{aligned}$$

Throughout this chapter, we will refer to this reformulation as reformulation two **(R2)**. A short summary for both reformulation methods can be found in Table 8.1.

8.2. Methods for the interpolation of the reduced models

For both of the two reformulations, the bilinear models will be reduced with a one-sided version of BIRKA (cf. Algorithm 3, Section 7.2) at different sampling points $\mathfrak{p}_j = (\theta_j, \mu_j, \gamma_j, \rho_j), j = 1, \ldots, J$, in the parameter space. In these points, reduced order models $\hat{E}(\mathfrak{p}_j), \hat{A}(\mathfrak{p}_j), \hat{N}_k(\mathfrak{p}_j), \hat{B}(\mathfrak{p}_j)$ and projection matrices $V(\mathfrak{p}_j)$ will be obtained. In the upcoming sections, we compare different interpolation strategies to construct reduced models at other parameter points $\mathfrak{p}_{\text{new}} = (\theta_{\text{new}}, \mu_{\text{new}}, \gamma_{\text{new}}, \rho_{\text{new}})$. We will give a short overview here, for a more detailed presentation, the reader is referred to Section 5.3.1.

Table 8.1. Two reformulation methods – short summary.

(R1)	**(R2)**
Complete reformulation	Incomplete reformulation
Dependence in physical parameters h will be ignored. All affine parameters on the right hand side of (8.1) (see also (8.2)) will be shifted to the input, whereas matrix E, still depends on them.	Reformulation of the model is only conducted for the physical parameter h. All matrices still depend on the parameters in geometry.

The interpolation methods, that will be used can be arranged into two different classes: One-step methods and two-step methods.

One-step methods (see Section 5.3.1)**:**

After the reduced order models in different points $\mathfrak{p}_j$ have been obtained one needs to

1) *Adjust the reduced order bases.*
Different reduced order models do not lie in the same state space and hence a transformation to the same state space is needed. One needs to find a reference subspace R_V and transformations M_j and T_j such that the states can be transferred to the reference subspace. One obtains:

$$\begin{aligned}
\overline{E}_j &= M_j^T \hat{E}(\mathfrak{p}_j) T_j, \\
\overline{A}_j &= M_j^T \hat{A}(\mathfrak{p}_j) T_j, \\
\overline{N}_{kj} &= M_j^T \hat{N}(\mathfrak{p}_j)_k T_j, \\
\overline{B}_j &= M_j^T \hat{B}(\mathfrak{p}_j), \\
\overline{C}_j &= \hat{C}(\mathfrak{p}_j) T_j, \text{ for } j = 1, \ldots, J.
\end{aligned}$$

2) *Choose the interpolation manifold and interpolation method.*
Interpolate the matrices $\overline{E}_j, \overline{A}_j, \overline{N}_{kj}, \overline{B}_j$ and $\overline{C}_j$ to obtain the reduced order model at $\mathfrak{p}_{\text{new}}$.

Four different methods will be used to conduct the adjusting of the bases and the interpolation – we will refer to them as the **one-step methods**:

(P1) This approach was developed by Panzer et al. [53]. The reference subspace R_V is given by a SVD of the matrices $V(\mathfrak{p}_j)$. As transformations one uses $T_j = M_j = (R_V^T V(\mathfrak{p}_j))^{-1}$. After the transformation to the reference coordinate system, a linear interpolation is used to obtain a reduced model at the interpolation point $\mathfrak{p}_{\text{new}}$. (No special manifold is chosen.)

(P2) Like **(P1)**, just use a *weighted* SVD of the matrices $V(\mathfrak{p}_j)$.

(A1) This approach was introduced by Amsallem et al. [3]. The reference subspace is obtained by choosing the projection matrix of a reference model $R_V = V(\mathfrak{p}_{j_0})$ from the given reduced models. In our case, this will be the nearest model with respect to the new parameter point $\mathfrak{p}_{\text{new}}$. The matrix $T_j = U_j Z_j^T$ is given by the SVD of $V(\mathfrak{p}_j)^T R_V = U_j \Sigma_j Z_j^T$, and the matrix M_j is obtained as $M_j = \hat{E}(\mathfrak{p}_j)^{-T} = \left(V(\mathfrak{p}_j)^T E(\mathfrak{p}_j) V(\mathfrak{p}_j)\right)^{-1}$. Hence it holds $\overline{E}_j = I_r$ after the transformation. Now, for every matrix $\overline{A}_j, \overline{N}_{kj}, \overline{B}_j$ and $\overline{C}_j$ a manifold for the interpolation needs to be chosen. Here, we choose the manifold of real $n \times n$ matrices for the interpolation of $\overline{A}_j$ and $\overline{N}_{kj}$, the manifold of real $n \times m$ matrices for the interpolation of $\overline{B}_j$ and the manifold of real $p \times n$ matrices for the interpolation of $\overline{C}_j$. The interpolation is now conducted on the tangential space to the matrix in the reference point. (I.e. in $\mathcal{T}_{\overline{A}_{j_0}} \mathcal{M}$ for the interpolation of the matrices $\overline{A}_j$.) A linear interpolation between the matrices is used. Details for the choice of the manifold are given in Section 5.3.1.2.

(A2) Like **(A1)**, just use the manifold of the non-singular $n \times n$ matrices for the interpolation of the matrices $\overline{A}_j$.

Two-step methods:
The second class of methods will be called **two-step methods**. They can be used only if at least one affine parameter is present.

- **First step:** First, the non-affine parameters are fixed in one point $\hat{J}$ and only the affine parameters will be varied, i.e. $(\theta_k, \mu_l, \gamma_{\hat{J}}, \rho_{\hat{J}})$, $k = 1, \ldots, K, l = 1, \ldots, L$. A global projection matrix is calculated by using a SVD:

$$V_{\text{global},\hat{J}} = \text{svd}\left([V(\theta_1, \mu_1, \gamma_{\hat{J}}, \rho_{\hat{J}}) \quad V(\theta_1, \mu_2, \gamma_{\hat{J}}, \rho_{\hat{J}}) \quad \ldots \quad V(\theta_K, \mu_L, \gamma_{\hat{J}}, \rho_{\hat{J}})]\right).$$

  The global projection matrix is calculated such that $V_{\text{global},\hat{J}} \in \mathbb{R}^{n \times r}$ with the same reduced order r as for the matrices $V(\theta_k, \mu_k, \gamma_{\hat{J}}, \rho_{\hat{J}})$. In a new parameter point $(\theta_{\text{new}}, \mu_{\text{new}}, \gamma_{\hat{J}}, \rho_{\hat{J}})$ the reduced model can now easily be obtained. For example for the reduced mass matrix E:

$$\begin{aligned}\hat{E}((\theta_{\text{new}}, \mu_{\text{new}}, \gamma_{\hat{J}}, \rho_{\hat{J}})) &= V_{\text{global},\hat{J}}^T E_0(\gamma_{\hat{J}}, \rho_{\hat{J}}) V_{\text{global},\hat{J}} \\ &+ \theta_{\text{new}} V_{\text{global},\hat{J}}^T E_\theta(\gamma_{\hat{J}}, \rho_{\hat{J}}) V_{\text{global},\hat{J}} + \mu_{\text{new}} V_{\text{global},\hat{J}}^T E_\mu(\gamma_{\hat{J}}, \rho_{\hat{J}}) V_{\text{global},\hat{J}}. \quad (8.4)\end{aligned}$$

  The calculation of a global projection matrix is now done for all points (γ_j, ρ_j), and results in reduced models where θ_{new} and μ_{new}, the affine parameters, are already fixed. Hence for the affine parameters in $\mathfrak{p}_{\text{new}}$ no interpolation needs to be done, it remains only to interpolate the non-affine parameters.
- **Second step:** The interpolation of the non-affine parameters, i.e. matrices $\hat{E}((\theta_{\text{new}}, \mu_{\text{new}}, \gamma_{\hat{j}}, \rho_{\hat{j}}))$, $\hat{A}((\theta_{\text{new}}, \mu_{\text{new}}, \gamma_{\hat{j}}, \rho_{\hat{j}}))$, $\hat{N}_k((\theta_{\text{new}}, \mu_{\text{new}}, \gamma_{\hat{j}}, \rho_{\hat{j}}))$, $\hat{B}((\theta_{\text{new}}, \mu_{\text{new}}, \gamma_{\hat{j}}, \rho_{\hat{j}}))$ and $\hat{C}((\theta_{\text{new}}, \mu_{\text{new}}, \gamma_{\hat{j}}, \rho_{\hat{j}}))$, $j = 1, \ldots, J$ is done using the interpolation methods stated during the explanation of the one-step methods.

We will refer to this approach as **(Af-A1)**, **(Af-A2)**, **(Af-P1)** or **(Af-P2)** depending on the method that is used for the interpolation in the second step. In the case where all parameters are affine and only the first step needs to be done we call the method **(Af)**.

For a quick reference, all methods are summarized in Tables 8.2 and 8.3.

Table 8.2. One-step methods for the interpolation of reduced order models.

	(P1)	**(P2)**	**(A1)**	**(A2)**
reference subspace	$R_V = \text{svd}([V(\mathfrak{p}_1), \ldots, \ldots, V(\mathfrak{p}_K)])$, SVD of the projection matrices	$R_V = \text{svd}([\omega_1 V(\mathfrak{p}_1), \ldots, \ldots, \omega_K V(\mathfrak{p}_K)])$, weighted SVD of the projection matrices	$R_V = V(\mathfrak{p}_{j_0})$, projection matrix of chosen reference model	$R_V = V(\mathfrak{p}_{j_0})$, projection matrix of chosen reference model
transformation matrices	$T_j = M_j = (R_V^T V(\mathfrak{p}_j))^{-1}$	$T_j = M_j = (R_V^T V(\mathfrak{p}_j))^{-1}$	$T_j = U_j Z_j^T$ is given by the SVD of $V(\mathfrak{p}_j)^T R_V = U_j \Sigma_j Z_j^T$, and the matrix M_j is obtained as $M_j = \left(V(\mathfrak{p}_j)^T E(\mathfrak{p}_j) V(\mathfrak{p}_j)\right)^{-1}$	$T_j = U_j Z_j^T$ is given by the SVD of $V(\mathfrak{p}_j)^T R_V = U_j \Sigma_j Z_j^T$, and the matrix M_j is obtained as $M_j = \left(V(\mathfrak{p}_j)^T E(\mathfrak{p}_j) V(\mathfrak{p}_j)\right)^{-1}$
manifolds for interpolation	no manifold is chosen	no manifold is chosen	the manifold of real $n \times n$, $n \times m$ and $p \times n$ matrices, depending on which matrix to interpolate	the manifold of real $n \times n$, $n \times m$ and $p \times n$ matrices, depending on which matrix to interpolate - for $\hat{A}(\mathfrak{p}_j)$ the manifold of the non-singular matrices is chosen

Table 8.3. Two-step methods for the interpolation of reduced order models.

	(Af-P1)	**(Af-P2)**	**(Af-A1)**	**(Af-A2)**	**(Af)**
First step	Calculation of reduced order models for the affine parameters in the new parameter point $\mathfrak{p}_{\text{new}}$ see 8.2.				
Second step – interpolation method used	**(P1)**, see Table 8.2	**(P2)**, see Table 8.2	**(A1)**, see Table 8.2	**(A2)**, see Table 8.2	no interpolation necessary – only affine parameters

8.3. Reduction and interpolation using reformulation one

To simplify the presentation, we fix the parameters μ, γ, ρ, so only one affine parameter θ remains. After the reformulation **(R1)** and a scaling of the matrices N_1 and N_2 as explained in Section 8.1.1, the following system is obtained:

$$\Sigma_{\text{bilin}}(\theta): \begin{cases} (E_0 + \theta E_\theta)\,\dot{x}(t) = A_0 x(t) + \sum_{k=1}^{6} g N_k u_k^g(t) x(t) + B u^g(t), \\ y(t) = C x(t), \end{cases} \tag{8.5}$$

with

$$\begin{aligned} u^g(t) &= \left[\tfrac{1}{g(1+\theta)} \quad \tfrac{\theta}{g} \quad T_0 \quad T_\infty \quad \theta T_\infty \quad (1+\theta)S(t)\right]^T, \\ A_0 &= A_0(\gamma,\rho) + hA_{h0}(\gamma,\rho) + \frac{1}{1+\mu} A_{\frac{1}{1+\mu}}(\gamma,\rho) + \mu A_\mu(\gamma,\rho), \\ N_1 &= A_{\frac{1}{1+\theta}}(\gamma,\rho), \\ N_2 &= A_\theta(\gamma,\rho) + hA_{h\theta}(\gamma,\rho), \\ N_3 &= \cdots = N_6(\gamma,\rho) = 0, \end{aligned}$$

$$B = \begin{bmatrix} 0 & 0 & & \\ \vdots & \vdots & \frac{1}{1+\mu} B_{\frac{1}{1+\mu}}(\gamma,\rho) + \mu B_{\mu}(\gamma,\rho) + B_0(\gamma,\rho) & \\ 0 & 0 & & \end{bmatrix}$$

$$hB_{h0}(\gamma,\rho) \quad hB_{h\theta}(\gamma,\rho) \quad B_S(\gamma,\rho)\Big].$$

Now the results for the large model with $n = 71,978$ degrees of freedom from Section 4.3 are discussed. As noted before, the N_k are large and need to be scaled before a reduction of the scaled system (8.5) can be performed.
Using BIRKA as given in Algorithm 3 and the Kronecker product approximation (cf. Section 6.1), we reduce the model as given by equation (8.5) at five different sampling points $\theta \in \{0, 0.5, 1, 1.5, 2\}$ to a reduced order of $r = 700$. After the reduction, stable models are obtained by using a one sided projection V in the last model (cf. **BIRKA-oS** in Section 7.2.1.2).
The interpolation between the reduced models at the sampling points is conducted using methods **(P2)**, **(A1)** and **(Af)** from Section 8.2.
We examine the temperature distribution at four different points in the model: At the bottom of the housing, on the coil, in the upper bearing and at the bottom of the rotor. Results for the interpolated models at two different parameter points $\theta_{\text{new}} \in \{0.45, 1.65\}$ for two $\{0, 2\}$, three $\{0, 1, 2\}$ and five $\{0, 0.5, 1, 1.5, 2\}$ sampling points can be found in the Figures 8.1 and 8.2, for the first and the fourth output, respectively.

The quality of the approximation improves with increasing the number of sampling points. When using five sampling points, the interpolated reduced models for $\theta_{\text{new}} \in \{0.45, 1.65\}$ yield good results for the first three outputs. It seems however difficult to approximate the fourth output, which — even with five sampling points — only leads to good models for the approach via a global projection matrix **(Af)**, as it can be seen in Figure 8.2. This might be related to the fact that this output lies on the bottom of the rotor and is not directly attached to the stator (as main heat source). Hence the heat can only be transferred via housing and flange.

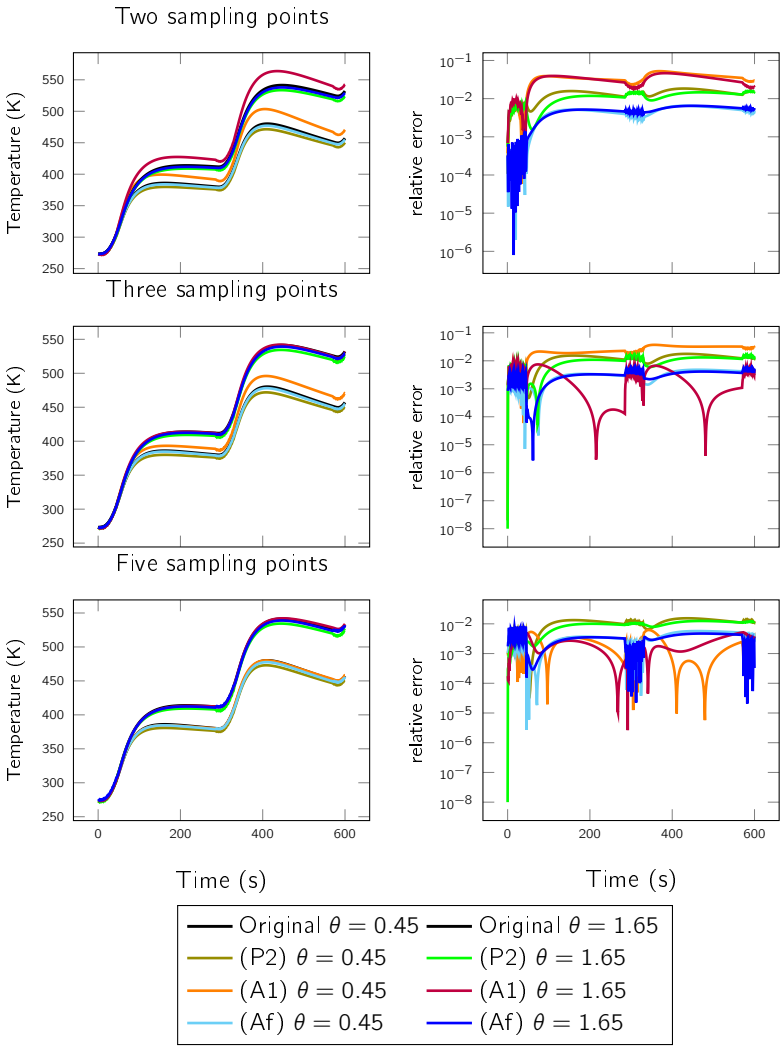


Figure 8.1. First output (bottom of the housing), interpolation of reduced order models (r=700) in a different number of sampling points, with results in different interpolation points.

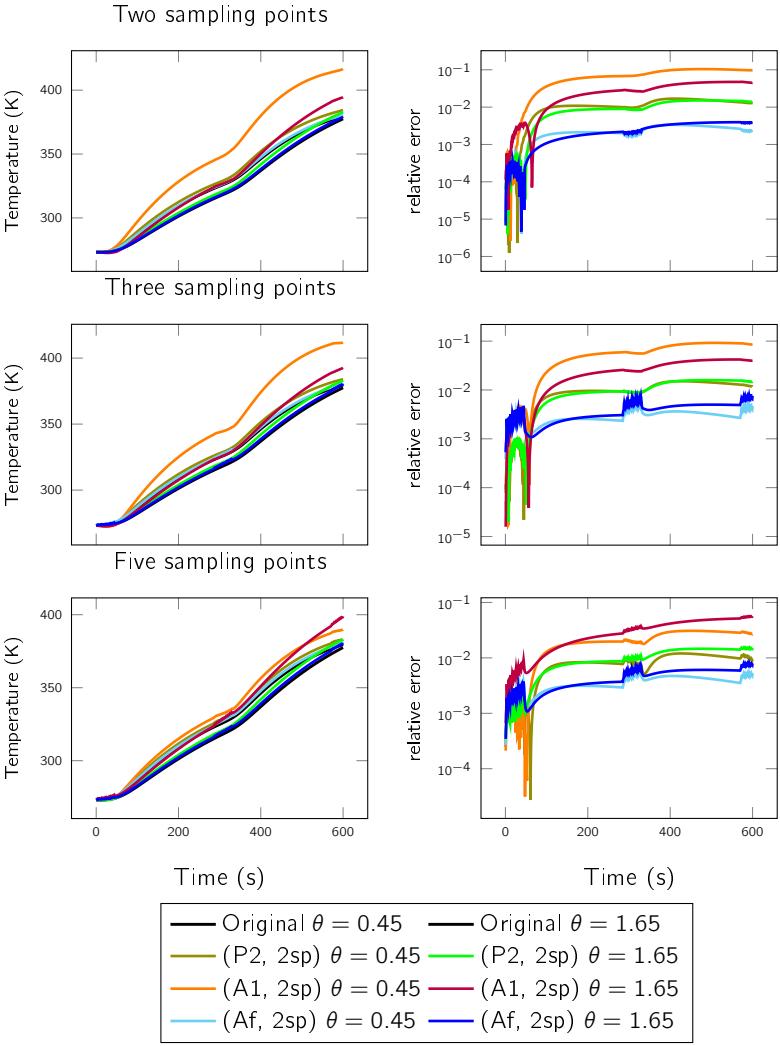


Figure 8.2. Fourth output (bottom of the rotor), interpolation of reduced order models in a different number of sampling points, with results in different interpolation points.

Table 8.4. Costs for the reduction and interpolation of the model with one affine parameter.

Method	Costs
Offline – Reduction in one parameter point	1 week per sampling point
Online – Interpolation with **(A1)** or **(A2)**	20-25min
Online – Interpolation with **(P1)** or **(P2)**	10-15min
Offline – Calculation of the global projection matrix	20min
Online – Assembling of the model in the new parameter point	<1min

As we have considered a model in one affine parameter, it was possible to use the method via a global projection matrix **(Af)** and no (additional) interpolation between the reduced order models. This method always leads to good results, and hence it can be recommended whenever the parameter dependency is affine and the calculation of the SVD of all matrices $V(\theta_j)$ does not exceed the computational capacity. Method **(A1)** outperforms **(P2)** in approximation of the first output (five sampling points), whereas **(P2)** performs better for the outputs two to four. Hence, one cannot state that one interpolation method is better than the other.
The reduction of the large model for one sampling point required up to one week on 12 CPUs with 3GB RAM each. So sampling in more than one parameter will easily exceed the available resources or lead to extremely long simulation times [2]. Hence, the interpolation methods will now be tested on the smaller model with $n = 2,969$ degrees of freedom from Section 4.3. In addition, we will change the reformulation method, and use the second reformulation (cf. Section 8.1.2, **(R2)**), as there will be no need to scale the models prior to the reduction, as we have noted that a scaling in the N_k increases the reduced order (cf. Remark 6.3.2).

Costs for the reduction and interpolation can be found in Table 8.4. Except for the reduction that has been performed on 12CPUs with 3GB RAM each, the calculations have been performed on visualization nodes

[2] A discussion explaining the long simulation times can be found in Section 7.2.3.

that are used simultaneously by different users. Depending on the memory demands and the loads of the other users, the calculation times can differ.

8.4. Reduction and interpolation using the second reformulation

For the presentation of the results obtained by using the second reformulation **(R2)** (cf. Section 8.1.2), the model with $n = 2,969$ will be used. It has been presented in Section 4.3 and is shown in Figures 4.7 and 4.8. For three different points the temperature profile is monitored: On the bottom of the housing (output 1), on the stator (output 2) and on the upper part of the rotor (output 3).

To obtain stable reduced order models the one-sided approach **only V** (cf. Chapter 7.2) is chosen. This leads to larger reduced orders as an original BIRKA – however stability is preserved automaticly, which is crucial for the interpolation steps. For every sampling point $\mathfrak{p}_j = (\theta_j, \mu_j, \gamma_j, \rho_j)$ the original model was reduced to an order of $r = 100$. The parameters (θ_j, μ_j) are affine, and the parameters (γ_j, ρ_j) are non-affine, hence our explained two-step approach applies. The sampling points are given as:

2sp: $\theta_j, \mu_j \in \{0, 2\}$ and $\gamma_j, \rho_j \in \{1, 3\}$; 2^4 sampling points
3sp: $\theta_j, \mu_j \in \{0, 1, 2\}$ and $\gamma_j, \rho_j \in \{1, 2, 3\}$; 3^4 sampling points
5sp: $\theta_j, \mu_j \in \{1, 0.5, 1, 1.5, 2\}$ and $\gamma_j, \rho_j \in \{1, 1.4122, 2, 2.5878, 3\}$; 5^4 sampling points

where $\{1.0489, 1.4122, 2, 2.5878, 2.9511\}$ are the Chebychev points within $[1, 3]$. We use 1 and 3 instead of 1.0489 and 2.9511 as each of the parameters is in the closed interval $[1, 3]$.
For the interpolation of the models, we will use four different methods. First, an interpolation in all four parameters $(\theta_j, \mu_j, \gamma_j, \rho_j)$ will be performed directly (one-step approach) by using the two interpolation methods **(A1)** and **(P2)**. In addition, a two-step approach will be applied by using the methods **(Af-P2)** and **(Af-A1)** - see Section 8.2.

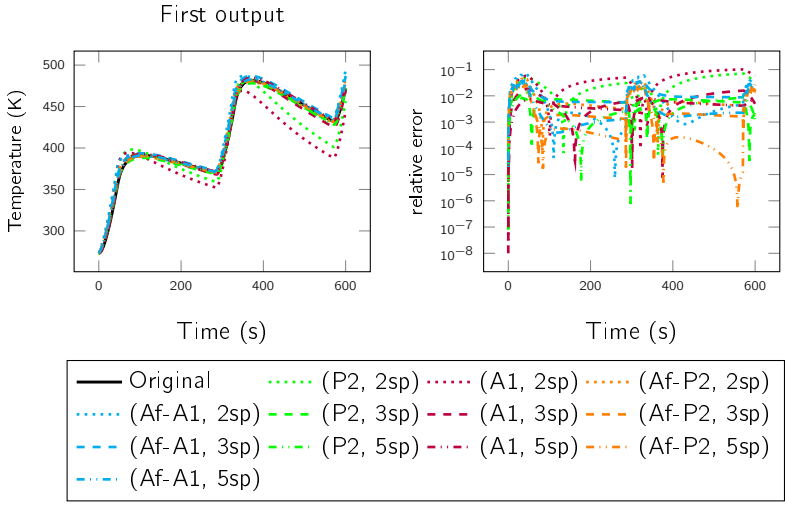


Figure 8.3. Temperature curves from reduced models obtained by interpolation with different methods and numbers of sample points in point $\theta = 1.67, \mu = 1.78, \gamma = 2.36, \rho = 1.22$.

In Figure 8.3 the results for two, three and five sampling points in the first output for the interpolation point

$$\mathfrak{p}_{\text{new}_0} = (\theta = 1.67, \mu = 1.78, \gamma = 2.36, \rho = 1.22),$$

and reduced order $r = 100$ are shown. For two sampling points (dotted lines) the two-step methods (i.e. **(Af-P2)** and **(Af-A1)**) lead to better results than the one-step methods (i.e. **(P2)** and **(A1)**). For three sampling points (dashed lines), the one-step methods get better in general, and for five sampling points (dashdotted lines), the approximation using the one-step methods is sufficiently accurate — especially for the approach **(A1)**.

Considering three other interpolation points

$$\mathfrak{p}_{\text{new}_1} = (\theta = 1.67, \mu = 1.78, \gamma = 2.976, \rho = 2.73),$$

$$\mathfrak{p}_{\text{new}_2} = (\theta = 1.56, \mu = 1.2, \gamma = 1.47, \rho = 1.634),$$
$$\text{and } \mathfrak{p}_{\text{new}_3} = (\theta = 0.34, \mu = 0.13, \gamma = 1.134, \rho = 1.22),$$

the results for the interpolated models can be found in Figures 8.4 to 8.6. One observes that one obtains good results for five sampling points in all four different interpolation points $\mathfrak{p}_{\text{new}_i}$. There are however differences in the quality of the approximation. The point $\mathfrak{p}_{\text{new}_1}$ is for example not perfectly approximated by the approaches **(Af-P2)** and **(Af-A1)**. In addition, one can observe oscillations in the approximations by **(Af-P2)** and **(Af-A1)**. They occur whenever there is a significant change in the dynamics of the model.

In general: For few sampling points, the two-step methods **(Af-A1)** and **(Af-P2)** (i.e. using a global projection matrix for the affine parameter dependency and then interpolating the non-affine parameters) lead to better results than a direct interpolation. However, as the number of sampling points increases, the approaches with direct interpolation (i.e. **(A1)**, **(P2)**) perform as good as the ones with a global projection matrix for the affine parameters, or even better. Hence, if the reduction in one sampling point is time consuming (as it is using BIRKA — cf. Section 7.2.3), it is desirable to sample as few points as possible. If the calculation of a global projection matrix in the affine parameters is not too time consuming, using few sampling points and one of the two-step methods (**(Af-A1)** and **(Af-P2)**) yields satisfactory results.

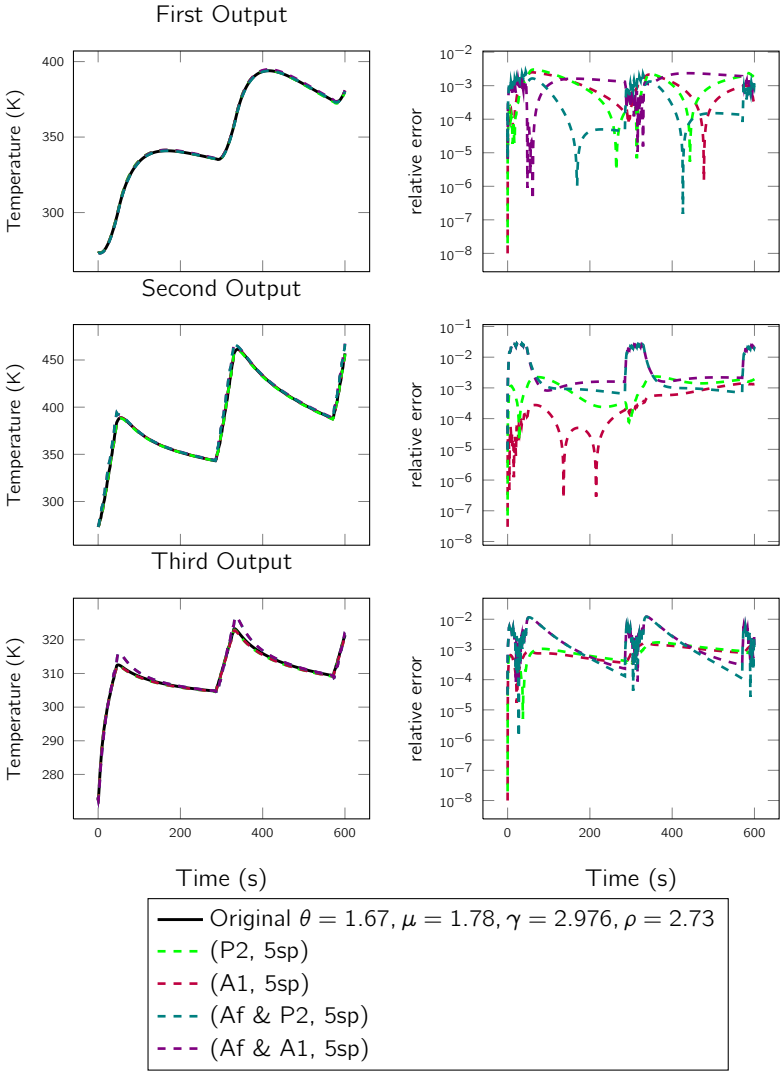


Figure 8.4. Interpolation of reduced order models in five sampling points at $\mathfrak{p}_{\text{new}_1} = (\theta = 1.67, \mu = 1.78, \gamma = 2.976, \rho = 2.73)$.

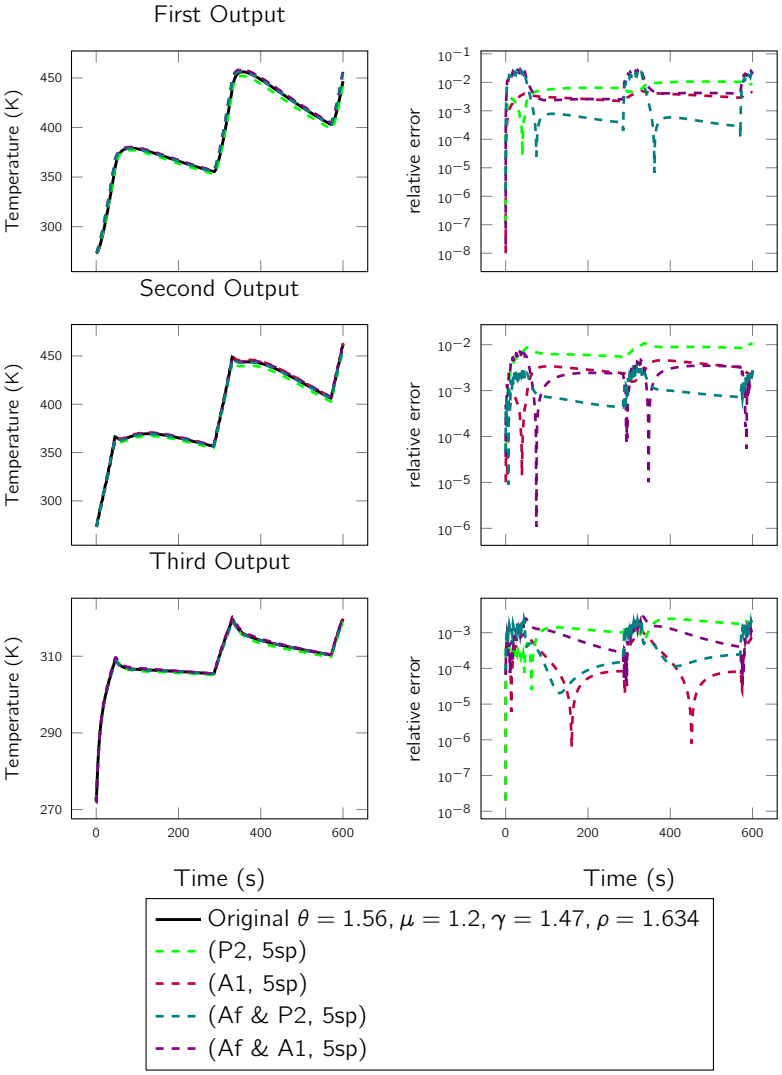


Figure 8.5. Interpolation of reduced order models in five sampling points at $\mathfrak{p}_{\text{new}_2} = (\theta = 1.56, \mu = 1.2, \gamma = 1.47, \rho = 1.634)$.

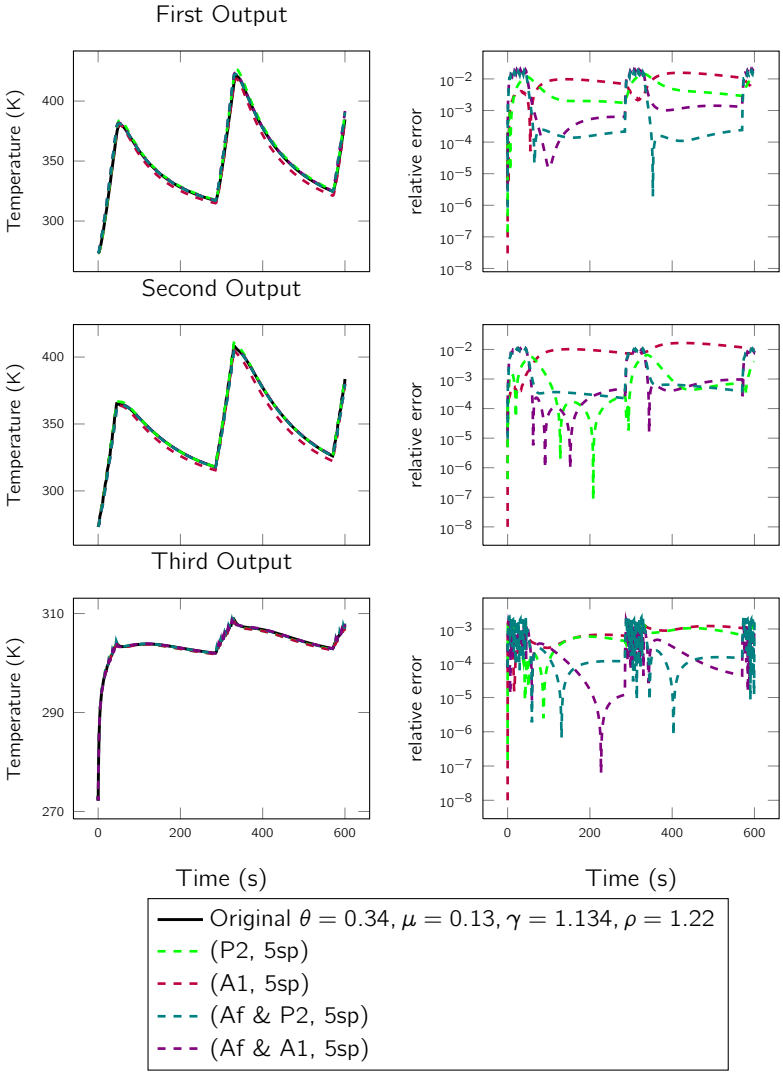


Figure 8.6. Interpolation of reduced order models in five sampling points at $\mathfrak{p}_{\text{new}_3} = (\theta = 0.34, \mu = 0.13, \gamma = 1.134, \rho = 1.22)$.

Table 8.5. Costs for the reduction and interpolation of the model with two affine and two non-affine parameters

reduction in sampling points (sp)	offline	reduction in one parameter point: $\approx$ 30min. for 2 sp: $2^4 \cdot$ 30min= 8h, for 3 sp: $3^4 \cdot$ 30min$\approx$ 1.7days, for 5 sp: $5^4 \cdot$ 30min$\approx$ 13days.
one-step method	online	Interpolation with **(A1)** $<$ 10min, **(A2)** $<$ 15min, **(P1)** $<$ 10min, **(P2)** $<$ 5min.
two-step method	offline	one global projection matrix for fixed non-affine parameters $(\theta_k, \mu_l, \gamma_{\hat{j}}, \rho_{\hat{j}})$: $\approx$ 1min, for 2 sp: $2^2 \cdot$ 1min= 4min, for 3 sp: $3^2 \cdot$ 1min$\approx$ 6min, for 5 sp: $5^2 \cdot$ 1min$\approx$ 25min.
	online	interpolation of non-affine parameters with: **(A1)** $<$5s, **(A2)** $<$10s, **(P1)** $<$20s, **(P2)** $<$5s.

In Table 8.5 approximate costs for the reduction and interpolation are summarized. Again the calculations have been performed on visualization nodes used simultaneously by different users. The calculation times are therefore only approximations depending on load and available memory on the nodes. It is not surprising, that the interpolation using all parameter points is slower that the one, where only the non-affine parameters need to be interpolated. In general, **(A1)** is faster than **(A2)** and **(P2)** is faster than **(P1)**. This is due to the following behavior: The reduction in **(P2)** is performed using a weighted SVD. We use the weights that will be used for the linear interpolation of the models afterwards. As only the nearest models with respect to the new parameter point are used in the interpolation, only the projection matrices $V(\mathfrak{p}_j)$ from these models are used for the calculation of the reference subspace R_V. In contrast, all matrices $V(\mathfrak{p}_j)$ are used for the SVD in **(P1)**. This explains longer calculation times. During the

execution of **(A1)** and **(A2)**, the interpolation is done on tangential spaces of matrix manifolds. The matrices need to be mapped to these spaces by using different logarithms (see Table 5.1). Whereas the manifold of $n \times m$ matrices only involves a subtraction, the manifold of nonsingular matrices requires an inversion and a matrix logarithm. This leads to longer calculation times.

8.4.0.1. *Interpolation methods (A2) and (P1).* So far, only results for the interpolation methods **(A1)** and **(P2)** have been presented. This is due to the fact that the obtained results for the approaches **(A2)** and **(P1)** are in most cases not as good as the results for the other approaches. A comparison of the approaches **(P1)** and **(P2)** for the interpolation point

$$\mathfrak{p}_{\text{new}_0} = (\theta = 1.67, \mu = 1.78, \gamma = 2.36, \rho = 1.22),$$

can be found in Figure 8.7, and results for the approach **(A2)** for the interpolation point $\mathfrak{p}_{\text{new}_3}$ are shown in Figure 8.8. Whereas the method **(P1)** usually gives reasonable results, the method **(A2)** has significant problems in the approximation of the third output of the model.

Method **(A2)** fails to provide a reasonable approximation. This might be related to the interpolation procedure. First, all matrices in the sampling points $A(\mathfrak{p}_j)$ (which belong to the manifold $\mathcal{M}$ of the non-singular matrices) need to be transferred to the tangential space regarding the reference model $\mathcal{T}_{A(\mathfrak{p}_{j_0})}\mathcal{M}$, then a "classic" interpolation — in our case linear interpolation — is performed on these elements of $\mathcal{T}_{A(\mathfrak{p}_{j_0})}\mathcal{M}$. It is not clear, that the "classic" interpolation stays in the tangential space, and hence the interpolated matrix $A(\mathfrak{p}_{\text{new}})$ might lead to inaccurate results.

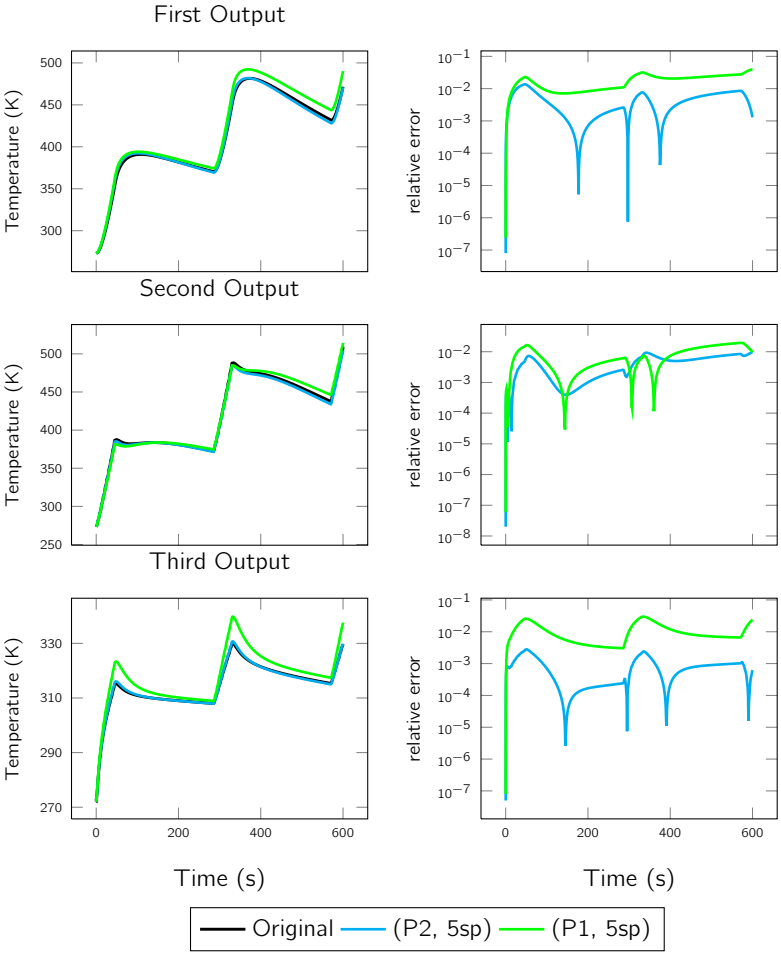


Figure 8.7. Interpolation using the approaches **(P1)** and **(P2)** in five sampling points at interpolation point $\theta = 1.67, \mu = 1.78, \gamma = 2.36, \rho = 1.22$.

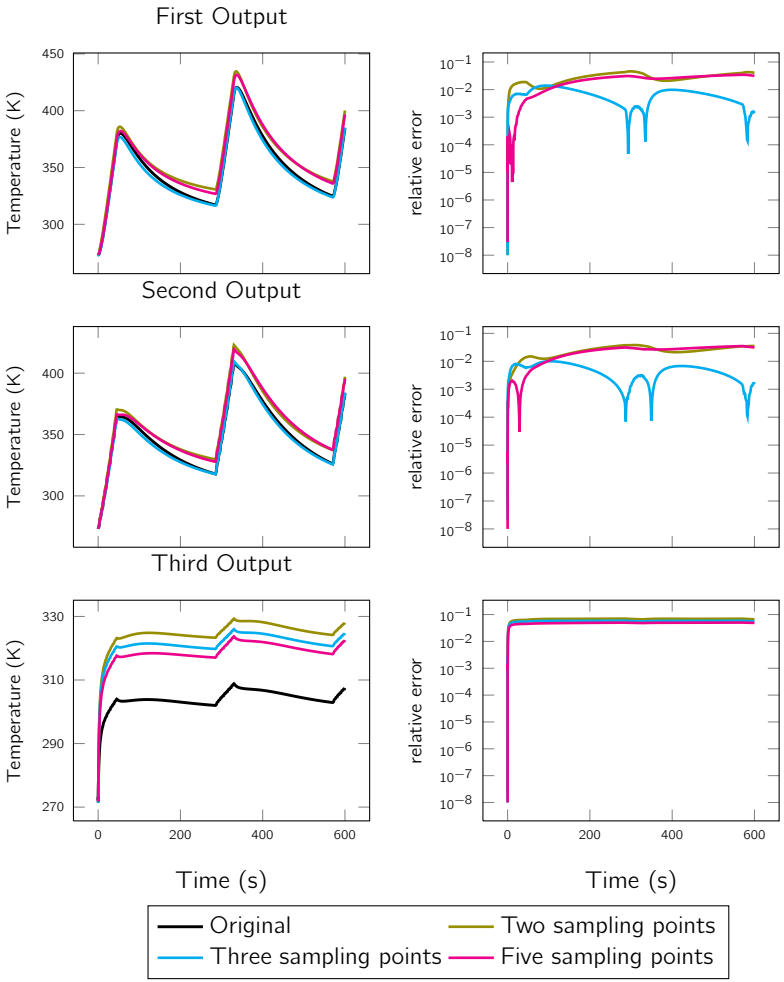


Figure 8.8. Interpolation of reduced order models in two,three and five sampling points at interpolation point $\theta = 0.34, \mu = 0.13, \gamma = 1.134, \rho = 1.22$ for **(A2)**.

8.4.1. Discussion of results. In this chapter, results for the reduction and interpolation of thermal models with geometric variations have been presented. Linear parametric models have been reformulated as bilinear models in two different ways (cf. Section 8.1.1 and 8.1.2) and then reduced using BIRKA with one-sided projections (cf. Sections 6.2.2 and 7.2.1.2). First, results for the first reformulation **(R1)** (cf. Section 8.1.1), for a model with $n = 71,978$ and one geometrical parameter have been shown (cf. Figures 8.1 and 8.2). An additional preprocessing step was necessary to avoid problems resulting from the fact, that the norms of N_k and A are of the same magnitude. A scaling was introduced and lead to a large reduced order $r = 700$. The second reformulation **(R2)**, Section 8.1.2, does not require this preprocessing. Due to high computational demands (cf. Section 7.2.3), all results for the second reformulation and four parameters have been presented for a smaller model with $n = 2,969$. Interpolation of this model using different numbers of sampling points and interpolation methods (cf. Section 8.2) have been performed. In general, all methods give reasonable results. However, the method **(P2)** — using a weighted SVD to obtain the reference subspace — usually outperforms the method **(P1)** — the non-weighted SVD. In addition, it was not possible to obtain good results for the interpolation method on tangential spaces of non-singular matrices **(A2)**, whereas the interpolation on tangential spaces of $\mathbb{R}^{k\times l}$ leads to good results **(A1)**. The two approaches **(A1)** and **(P2)** usually give comparable results, hence it is not possible to favor one method over the other. Having two affine and two non-affine parameters, it is recommended to use a two-step method — first calculate a global projection matrix for the affine parameters and then interpolate the reduced order models in the non-affine parameters. For few sample points these methods yield usually better results than the one step methods.

CHAPTER 9

Conclusions and Outlook

9.1. Summary and Conclusions

The main objective of this work was to investigate the use of bilinear $\mathscr{H}_2$-optimal methods in parametric Model Order Reduction. As shown by Benner and Breiten [11], it is possible to reformulate a certain class of linear parametric systems as bilinear systems (cf. Section 5.3.2). The parameters can then be considered as inputs and the reduction can be performed without any sampling and interpolation in the parameter space, as most of the other methods for pMOR do [53, 3, 37, 13]. After obtaining a bilinear model, one can make use of bilinear Model Order Reduction. In this work, we focused on two methods for bilinear $\mathscr{H}_2$-optimal Model Order Reduction, which are described in Chapter 5. BIRKA (cf. Algorithm 3), originally obtained by Benner and Breiten [12], is stated and new algorithms for the bilinear $\mathscr{H}_2$-optimal reduction have been developed. These algorithms use optimization on Grassmann manifolds and — as a main advantage — can preserve stability. We have proven the stability preservation for symmetric, bilinear systems and analyzed the convergence behavior of the algorithms.

In addition to these theoretical results, several models for the thermal analysis of electrical motors have been built using Comsol® 3.5a (cf. Chapter 3). Linear parametric systems have been exported from Comsol® by an analysis of the underlying equations (cf. Chapter 4). For industrially relevant problems, both physical and geometric parameters need to be considered and the parameter dependency after the reduction must be preserved. As the resulting models are usually large (in our case $n = 41,199$, $n = 71,978$, and $n = 2,969$), the bilinear $\mathscr{H}_2$-optimal reduction methods have to be capable of dealing with these large systems.

The newly developed methods for the reduction using optimization on Grassmann manifolds are, however, not yet ready (cf. Section 7.1) for the use with these large systems, but results for the reduction of a heat equation on a square have been stated. BIKRA (cf. Section 5.5,[12]) is capable of reducing the large models, but several problems have been identified. In some cases, the stiffness matrix A is singular, the magnitude of the N_k is too large and a scaling needs to be introduced. Also unstable models have been obtained after the reduction. All these issues haven been examined and solutions have been proposed (cf. Chapter 6).

Numerical results for the reduction of two different types of models have been obtained. On one hand, a part of an electrical motor model, incorporating physical parameters, has been considered. These models are parametrized with physical parameters and have a structure that easily allows to reformulate them as a bilinear model. Reduction with BIRKA yields good results, not only in a certain parameter interval, but globally in the whole parameter range (cf. Chapter 7.2, Figure 7.8). The second type of models are electrical motor models, that in addition to the physical parameters use parameters that describe changes in geometry. This leads to models with a structure that can not easily be rewritten as a bilinear system. Hence one can reformulate the model as a bilinear model for certain parameters and interpolate the other parameters (cf. Chapter 8). For the interpolation, several well known methods from pMOR have been used (cf. [53, 3, 37]), which generally lead to good results. There are, however, differences in the quality of the approximation. For models with an affine parameter dependence in certain parameters, using a global projection matrix for the affine

parameter dependence leads to good results and can outperform a direct interpolation, especially for few sampling points.

9.2. Future research

Based on the work that has been presented in this thesis, several opportunities for future research have been identified:

- The new methods for the bilinear $\mathscr{H}_2$-optimal MOR using optimization methods on the Grassmann manifold as developed in Sections 5.5.4 and 7.1 still require some investigation:
  - The Algorithms bilGFA, bilFGFA and bilSQA have not yet been tested on large problems, due to the fact that one needs to solve large bilinear Sylvester equations. In the future, low-rank approximations to the solutions should be applied such as the ADI iteration (cf. [57, 14]), to allow treatment of large systems.
  - Convergence and the stability preservation for the Algorithms bilGFA,bilFGFA and bilSQA have not yet been established for bilinear systems with non-symmetric A and N_k.
  - For the optimization, one needs to correctly set several parameters to ensure a descent in the objective function. It would be an advantage to identify robust criteria based on which these parameters can be chosen.
- The reduction of the large parametric thermal models has been done using BIRKA [12]. The reduction times for our large models are within the range of several hours to a few days for 12 CPUs with 3GB RAM (see Section for a discussion 7.2.3). However, the structure of BIRKA would allow a parallelization, which could significantly reduce the reduction time.
- One interpolation approach by Amsallem [3] shows weak performance for some models (cf. Section 8.4.0.1). This could be caused by the fact that our used interpolation method does not preserve the membership in the tangential space. This behavior requires a development of interpolation procedures that do stay on the corresponding manifold.
- The interpolation methods used for the reduction of the parametric models require the reduction at several sampling points. The

number of sampling points has a strong impact on the computational demands, so it is worthwhile to explore methods to systematically and optimally sample the parameter space, e.g. using sparse grids [10] or latin hypercube sampling [4, 20].

APPENDIX A

Derivation of the bilinear $\mathscr{H}_2$-optimal conditions

A.1. Wilson conditions

We start by differentiating the norm

$$\begin{aligned}\mathcal{J} =&||\Sigma_{\text{bil}}^{\text{err}}||_{\mathscr{H}_2}^2 = \text{tr}(\begin{bmatrix}C & -\hat{C}\end{bmatrix} P^{\text{err}} \begin{bmatrix}C^T \\ -\hat{C}^T\end{bmatrix})\\ =&\text{tr}(P^{\text{err}} \begin{bmatrix}C^T \\ -\hat{C}^T\end{bmatrix} \begin{bmatrix}C & -\hat{C}\end{bmatrix})\\ =&\text{tr}(P^{\text{err}}\mathcal{C}),\end{aligned} \tag{5.42}$$

as given by Zhang and Lam [72] with respect to a parameter γ:

$$\frac{\partial \mathcal{J}}{\partial \gamma} = \text{tr}(\frac{\partial P^{\text{err}}}{\partial \gamma}\mathcal{C}) + \text{tr}(P^{\text{err}}\frac{\partial \mathcal{C}}{\partial \gamma}).$$

First, we insert the following Lyapunov equation in the derived norm:

$$(A^{\text{err}})^T Y^{\text{err}} E^{\text{err}} + (E^{\text{err}})^T Y^{\text{err}} A^{\text{err}} + \sum_{k=1}^{m} (N_k^{err})^T Y^{\text{err}} N_k^{\text{err}} + \underbrace{(C^{\text{err}})^T C^{\text{err}}}_{=\mathcal{C}} = 0, \tag{A.1}$$

and obtain:

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial \gamma} =&\text{tr}\left(\frac{\partial P^{\text{err}}}{\partial \gamma}\left(-(A^{\text{err}})^T Y^{\text{err}} E^{\text{err}}\right.\right.\\ &\left.\left.-(E^{\text{err}})^T Y^{\text{err}} A^{\text{err}} - \sum_{k=1}^{m} (N_k^{\text{err}})^T Y^{\text{err}} N_k^{\text{err}}\right)\right) + \text{tr}(P^{\text{err}}\frac{\partial \mathcal{C}}{\partial \gamma}).\end{aligned} \tag{A.2}$$

Second, we will derive the other Lyapunov equation of the error system:

$$A^{\text{err}}P^{\text{err}}(E^{\text{err}})^T + E^{\text{err}}P^{\text{err}}(A^{\text{err}})^T + \sum_{k=1}^{m} N_k^{\text{err}}P^{\text{err}}(N_k^{\text{err}})^T + \underbrace{B^{\text{err}}(B^{\text{err}})^T}_{=\mathcal{B}} = 0, \tag{A.3}$$

and multiply it from the left by $Y^{\text{err}}(= (E^{\text{err}})^{-1}Q^{\text{err}}E^{\text{err}})$:

$$\begin{aligned} &2\text{tr}(\frac{\partial A^{\text{err}}}{\partial \gamma}P^{\text{err}}(E^{\text{err}})^T Y^{\text{err}}) + 2\text{tr}(A^{\text{err}}\frac{\partial P^{\text{err}}}{\partial \gamma}(E^{\text{err}})^T Y^{\text{err}}) \\ &+2\text{tr}(A^{\text{err}}P^{\text{err}}\frac{\partial (E^{\text{err}})^T}{\partial \gamma}Y^{\text{err}}) + 2\text{tr}(\sum_{k=1}^{m}\frac{\partial N_k^{\text{err}}}{\partial \gamma}P^{\text{err}}(N_k^{\text{err}})^T Y^{\text{err}}) \\ &+\text{tr}(\sum_{k=1}^{m} N_k^{\text{err}}\frac{\partial P^{\text{err}}}{\partial \gamma}(N_k^{\text{err}})^T Y^{\text{err}}) + \text{tr}(\frac{\partial \mathcal{B}}{\partial \gamma}Y^{\text{err}}) = 0. \end{aligned} \tag{A.4}$$

Adding (A.4) to the derived norm (A.2) leads to the following equation:

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \gamma} =& 2\text{tr}(\frac{\partial A^{\text{err}}}{\partial \gamma}P^{\text{err}}(E^{\text{err}})^T Y^{\text{err}}) + 2\text{tr}(\frac{\partial E^{\text{err}}}{\partial \gamma}P^{\text{err}}(A^{\text{err}})^T Y^{\text{err}}) \\ &+\sum_{k=1}^{m} 2\text{tr}(\frac{\partial N_k^{\text{err}}}{\partial \gamma}P^{\text{err}}(N_k^{\text{err}})^T Y^{\text{err}}) + \text{tr}(\frac{\partial \mathcal{B}}{\partial \gamma}Y^{\text{err}}) + \text{tr}(P^{\text{err}}\frac{\partial \mathcal{C}}{\partial \gamma}). \end{aligned} \tag{A.5}$$

Differentiating by the reduced matrices leads to:

$$\frac{\partial \mathcal{J}}{\partial \hat{a}_{ij}} = 2\text{tr}(\frac{\partial A^{\text{err}}}{\partial \hat{a}_{ij}}P^{\text{err}}(E^{\text{err}})^T Y^{\text{err}}) = 2\text{tr}(\frac{\partial \hat{A}}{\partial \hat{a}_{ij}}(P_{12}^T E^T Y_{12} + P_{22}\hat{E}^T Y_{22})).$$

As an optimal reduced model would fulfill $\frac{\partial \mathcal{J}}{\partial \hat{a}_{ij}} = 0$ for all i, j one can conclude

$$P_{12}^T E^T Y_{12} + P_{22}\hat{E}^T Y_{22} = 0. \tag{A.6}$$

One obtains for the derivative with respect to the e_{ij}:

$$\frac{\partial \mathcal{J}}{\partial \hat{e}_{ij}} = 2\text{tr}(\frac{\partial E^{\text{err}}}{\partial \hat{e}_{ij}}P^{\text{err}}(A^{\text{err}})^T Y^{\text{err}}) = 2\text{tr}(\frac{\partial \hat{E}}{\partial \hat{e}_{ij}}(P_{12}^T A^T Y_{12} + P_{22}\hat{A}^T Y_{22})),$$

and again, this leads to:

$$P_{12}^T A^T Y_{12} + P_{22}\hat{A}^T Y_{22} = 0. \tag{A.7}$$

For the matrices N_k one derives:

$$\frac{\partial \mathcal{J}}{\partial (\hat{n}_k)_{ij}} = 2\text{tr}(\frac{\partial N_k^{\text{err}}}{\partial (\hat{n}_k)_{ij}}P^{\text{err}}(N_k^{\text{err}})^T Y^{\text{err}}) = 2\text{tr}(\frac{\partial \hat{E}}{\partial (\hat{n}_k)_{ij}}(P_{12}^T N_k^T Y_{12} + P_{22}\hat{N}_k^T Y_{22})),$$

for all $k = 1, \ldots, m$. One obtaines:

$$P_{12}^T N_k^T Y_{12} + P_{22} \hat{N}_k^T Y_{22} = 0, \quad k = 1, \ldots, m. \tag{A.8}$$

The equations for B and C involve more complicated calculations:

$$\begin{aligned}
\frac{\partial \mathcal{J}}{\partial \hat{b}_{ij}} &= tr\left(\frac{\partial \mathcal{B}}{\partial \hat{b}_{ij}} Y^{\text{err}}\right) = tr\left(\begin{bmatrix} 0 & Be_j e_i^T \\ e_i e_j^T B^T & e_i e_j^T \hat{B}^T + \hat{B} e_j e_i^T \end{bmatrix} \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix}\right) \\
&= \text{tr}(Be_j e_i^T Y_{12}) + \text{tr}(e_i e_j^T B^T Y_{12} + e_i e_j^T \hat{B}^T Y_{22} + \hat{B} e_j e_i^T Y_{22}) \\
&= \text{tr}(B^T Y_{12} e_i e_j^T) + \text{tr}(\hat{B}^T Y_{12} e_i e_j^T) + \text{tr}(e_i e_j^T B^T Y_{12}) + \text{tr}(e_i e_j^T \hat{B}^T Y_{22}) \\
&= 2\text{tr}(e_i e_j^T (B^T Y_{12} + \hat{B}^T Y_{22})),
\end{aligned}$$

This yields:

$$B^T Y_{12} + \hat{B}^T Y_{22} = 0. \tag{A.9}$$

Whereas

$$\begin{aligned}
\frac{\partial \mathcal{J}}{\partial \hat{c}_{ij}} &= tr\left(P^{\text{err}} \frac{\partial \mathcal{C}}{\partial \hat{c}_{ij}}\right) \\
&= tr\left(\begin{bmatrix} -P_{12} e_j e_i^T C & -P_{11} C^T e_i e_j^T + P_{12} e_j e_i^T \hat{C} + P_{12} \hat{C}^T e_i e_j^T \\ -P_{22} e_j e_i C & -P_{12}^T C^T e_i e_j^T + P_{22} e_j e_i^T \hat{C} + P_{22} \hat{C}^T e_i e_j^T \end{bmatrix}\right) \\
&= \text{tr}(-P_{12} e_j e_i^T C) + \text{tr}(-P_{12}^T C^T e_i e_j^T) + \text{tr}(P_{22} e_j e_i^T \hat{C}) + \text{tr}(P_{22} \hat{C}^T e_i e_j^T) \\
&= 2\text{tr}((-P_{12}^T C^T + P_{22} \hat{C}^T) e_i e_j^T) = 0,
\end{aligned}$$

yields

$$-P_{12}^T C^T + P_{22} \hat{C}^T = 0. \tag{A.10}$$

A.2. Derivation of the optimality conditions by Benner and Breiten

Following Benner and Breiten [12], the representation of the $\mathscr{H}_2$-norm will be derived with respect to the eigenvalues of the reduced system $\hat{\lambda}_i$ and $\tilde{N}_k$, $\tilde{B}$, $\tilde{C}$:

$$\begin{aligned}\mathcal{J} = \operatorname{vec}(I_{2p})^T &\left(\begin{bmatrix} C & -\tilde{C}\end{bmatrix} \otimes \begin{bmatrix} C & -\tilde{C}\end{bmatrix}\right) \\ \times \Bigg(&- \begin{bmatrix} E & \\ & I_r\end{bmatrix} \otimes \begin{bmatrix} A & \\ & \Lambda\end{bmatrix} - \begin{bmatrix} A & \\ & \Lambda\end{bmatrix} \otimes \begin{bmatrix} E & \\ & I_r\end{bmatrix} \\ &- \sum_{k=1}^{m} \begin{bmatrix} N_k & \\ & \tilde{N}_k{}^T\end{bmatrix} \otimes \begin{bmatrix} N_k & \\ & \tilde{N}_k{}^T\end{bmatrix}\Bigg)^{-1} \\ \times &\begin{bmatrix} B \\ \tilde{B}^T\end{bmatrix} \otimes \begin{bmatrix} B \\ \tilde{B}^T\end{bmatrix} \operatorname{vec}(I_{2m}).\end{aligned}$$

We will need the following lemma, originally given by Benner and Breiten [12]:

Lemma A.2.1. *Let $C(x) \in \mathbb{R}^{p\times n}$, $A(y), E, N_k \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ with $x, y \in \mathbb{R}$. Let*

$$\mathcal{L}(y) = \left(-A(y) \otimes E - E \otimes A(y) - \sum_{k=1}^{m} N_k \otimes N_k\right),$$

and assume that C and A are differentiable with respect to x and y. Then

$$\begin{aligned}&\frac{\partial}{\partial x}\left[\operatorname{vec}(I_p)^T (C(x) \otimes C(x))\mathcal{L}(y)^{-1}(B \otimes B)\operatorname{vec}(I_m)\right] \\ = \;& 2\operatorname{vec}(I_p)^T \left(\frac{\partial}{\partial x} C(x) \otimes C(x)\right) \mathcal{L}(y)^{-1}(B \otimes B)\operatorname{vec}(I_m),\end{aligned}$$

and

$$\begin{aligned}&\frac{\partial}{\partial y}\left[(vec(I_p)^T (C \otimes C)\mathcal{L}(y)^{-1}(B \otimes B)\, vec(I_m)\right] \\ = \;& 2\left[(vec(I_p)^T (C \otimes C)\mathcal{L}(y)^{-1} \left(\frac{\partial \mathcal{A}(y)}{\partial y} \otimes E\right) \mathcal{L}(y)^{-1}(B \otimes B)\, vec(I_m)\right].\end{aligned}$$

Proof. The proof given by Benner and Breiten shows this result for $E = I_n$. The case $E \neq I_n$ is a straight forward generalization of the proof, which will therefore be omitted here. □

In addition, we will need the following matrix:

$$M := \begin{bmatrix} I_r \otimes \begin{bmatrix} I_n \\ 0\end{bmatrix} & I_r \otimes \begin{bmatrix} 0^T \\ I_r\end{bmatrix}\end{bmatrix},$$

where $0 = \text{zeros}(r, n)$. It holds for M:

$$M^T \left(\tilde{N}_k^T \otimes \begin{bmatrix} N_k & \\ & \hat{N}_k \end{bmatrix} \right) M = \begin{bmatrix} \tilde{N}_k{}^T \otimes N_k & \\ & \tilde{N}_k{}^T \otimes \hat{N}_k{}^T \end{bmatrix},$$

as well as $MM^T = I_{r^2 n}$. We will now start with the differentiation of the norm with respect to $\tilde{C}$ by making use of Lemma A.2.1:

$$\begin{aligned}
\frac{\partial \mathcal{J}}{\partial \tilde{C}_{ij}} =& 2(\text{vec}(I_{2p}))^T \left(\begin{bmatrix} 0 & -e_i e_j^T \end{bmatrix} \otimes \begin{bmatrix} C & -\tilde{C} \end{bmatrix} \right) \\
& \times \left(- \begin{bmatrix} E & \\ & I_r \end{bmatrix} \otimes \begin{bmatrix} A & \\ & \Lambda \end{bmatrix} - \begin{bmatrix} A & \\ & \Lambda \end{bmatrix} \otimes \begin{bmatrix} E & \\ & I_r \end{bmatrix} \right. \\
& \qquad \left. - \sum_{k=1}^{m} \begin{bmatrix} N_k & \\ & \tilde{N}_k{}^T \end{bmatrix} \otimes \begin{bmatrix} N_k & \\ & \tilde{N}_k{}^T \end{bmatrix} \right)^{-1} \\
& \times \begin{bmatrix} B \\ \tilde{B}^T \end{bmatrix} \otimes \begin{bmatrix} B \\ \tilde{B}^T \end{bmatrix} \text{vec}(I_{2m}) \\
\underbrace{=}_{\text{see footnote}} & 2(\text{vec}(I_{2p}))^T \left(\begin{bmatrix} 0 & -e_i e_j^T \end{bmatrix} \otimes \begin{bmatrix} C & -\hat{C} \end{bmatrix} \right) \\
& \times \left(- \begin{bmatrix} E & \\ & I_r \end{bmatrix} \otimes \begin{bmatrix} A & \\ & \hat{A} \end{bmatrix} - \begin{bmatrix} A & \\ & \Lambda \end{bmatrix} \otimes \begin{bmatrix} E & \\ & I_r \end{bmatrix} \right. \\
& \qquad \left. - \sum_{k=1}^{m} \begin{bmatrix} N_k & \\ & \tilde{N}_k{}^T \end{bmatrix} \otimes \begin{bmatrix} N_k & \\ & \hat{N}_k \end{bmatrix} \right)^{-1} \\
& \times \begin{bmatrix} B \\ \tilde{B}^T \end{bmatrix} \otimes \begin{bmatrix} B \\ \hat{B}^T \end{bmatrix} \text{vec}(I_{2m}) \\
=& 2(\text{vec}(I_{2p}))^T \left(-e_i e_j^T \otimes \begin{bmatrix} C & -\hat{C} \end{bmatrix} \right) \\
& \times \left(MM^T \left(-I_r \otimes \begin{bmatrix} A & \\ & \hat{A} \end{bmatrix} - \Lambda \otimes \begin{bmatrix} E & \\ & I_r \end{bmatrix} \right. \right. \\
& \qquad \left. \left. - \sum_{k=1}^{m} \tilde{N}_k{}^T \otimes \begin{bmatrix} N_k & \\ & \hat{N}_k \end{bmatrix} \right) MM^T \right)^{-1} \\
& \times \tilde{B}^T \otimes \begin{bmatrix} B \\ \hat{B}^T \end{bmatrix} \text{vec}(I_{2m})
\end{aligned}$$

$$
\begin{aligned}
=&2(\mathrm{vec}(I_{2p}))^T \left(-e_i e_j^T \otimes \begin{bmatrix} C & -\hat{C} \end{bmatrix}\right) \\
&\times \left(M \left(- \begin{bmatrix} I_r \otimes A & \\ & I_r \otimes \hat{A} \end{bmatrix} - \begin{bmatrix} \Lambda \otimes E & \\ & \Lambda \otimes I_r \end{bmatrix} \right.\right. \\
&\qquad \left.\left. - \sum_{k=1}^{m} \begin{bmatrix} \tilde{N}_k^T \otimes N_k & \\ & \tilde{N}_k^T \otimes \hat{N}_k \end{bmatrix} \right) M^T \right)^{-1} \\
&\times \tilde{B}^T \otimes \begin{bmatrix} B \\ \hat{B}^T \end{bmatrix} \mathrm{vec}(I_{2m}) \\
=&2(\mathrm{vec}(I_{2p}))^T \begin{bmatrix} -e_i e_j^T \otimes C & e_i e_j^T \otimes \hat{C} \end{bmatrix} \\
&\times \left(- \begin{bmatrix} I_r \otimes A & \\ & I_r \otimes \hat{A} \end{bmatrix} - \begin{bmatrix} \Lambda \otimes E & \\ & \Lambda \otimes I_r \end{bmatrix} \right. \\
&\qquad \left. - \sum_{k=1}^{m} \begin{bmatrix} \tilde{N}_k^T \otimes N_k & \\ & \tilde{N}_k^T \otimes \hat{N}_k \end{bmatrix} \right)^{-1} \\
&\times \begin{bmatrix} \tilde{B}^T \otimes B \\ \tilde{B}^T \otimes \hat{B}^T \end{bmatrix} \mathrm{vec}(I_{2m}) \\
=&-2\mathrm{vec}(I_p)^T (e_i e_j^T \otimes C) \\
&\cdot \left(-I_r \otimes A - \Lambda \otimes E - \sum_{k=1}^{m} \tilde{N}_k^T \otimes N_k \right)^{-1} (\tilde{B}^T \otimes B)\mathrm{vec}(I_m) \\
&+2\mathrm{vec}(I_p)^T (e_i e_j^T \otimes \hat{C}) \\
&\cdot \left(-I_r \otimes \hat{A} - \Lambda \otimes I_r - \sum_{k=1}^{m} \tilde{N}_k^T \otimes \hat{N}_k \right)^{-1} (\tilde{B}^T \otimes \hat{B})\mathrm{vec}(I_m).
\end{aligned}
$$

The differentiation with respect to the eigenvalues $\hat{\lambda}_i$ is done as follows. First, we use Lemma A.2.1:

[1] Using $\Lambda = S^{-1}\hat{A}S$, $\tilde{N}_k^T = S^{-1}\hat{N}_k S$, $\tilde{B}^T = S^{-1}\hat{B}$, $\tilde{C} = \hat{C}S$.

$$
\begin{aligned}
\frac{\partial \mathcal{J}}{\partial \lambda_i} &= 2\mathrm{vec}(I_{2p})^T \left(\begin{bmatrix} C & -\tilde{C} \end{bmatrix} \otimes \begin{bmatrix} C & -\tilde{C} \end{bmatrix} \right) \\
&\quad \times \left(- \begin{bmatrix} E & \\ & I_r \end{bmatrix} \otimes \begin{bmatrix} A & \\ & \Lambda \end{bmatrix} - \begin{bmatrix} A & \\ & \Lambda \end{bmatrix} \otimes \begin{bmatrix} E & \\ & I_r \end{bmatrix} \right. \\
&\qquad \left. - \sum_{k=1}^{m} \begin{bmatrix} N_k & \\ & \tilde{N}_k{}^T \end{bmatrix} \otimes \begin{bmatrix} N_k & \\ & \tilde{N}_k{}^T \end{bmatrix} \right)^{-1} \left(\begin{bmatrix} 0 & 0 \\ 0 & e_i e_i^T \end{bmatrix} \otimes \begin{bmatrix} E & \\ & I_r \end{bmatrix} \right) \\
&\quad \times \left(- \begin{bmatrix} E & \\ & I_r \end{bmatrix} \otimes \begin{bmatrix} A & \\ & \Lambda \end{bmatrix} - \begin{bmatrix} A & \\ & \Lambda \end{bmatrix} \otimes \begin{bmatrix} E & \\ & I_r \end{bmatrix} \right. \\
&\qquad \left. - \sum_{k=1}^{m} \begin{bmatrix} N_k & \\ & \tilde{N}_k{}^T \end{bmatrix} \otimes \begin{bmatrix} N_k & \\ & \tilde{N}_k{}^T \end{bmatrix} \right)^{-1} \\
&\quad \times \begin{bmatrix} B \\ \tilde{B}^T \end{bmatrix} \otimes \begin{bmatrix} B \\ \tilde{B}^T \end{bmatrix} \mathrm{vec}(I_{2m}) \\
&= 2\mathrm{vec}(I_2 p)^T \left(\begin{bmatrix} C & -\tilde{C} \end{bmatrix} \otimes \begin{bmatrix} C & -\hat{C} \end{bmatrix} \right) \\
&\quad \times \left(- \begin{bmatrix} E & \\ & I_r \end{bmatrix} \otimes \begin{bmatrix} A & \\ & \hat{A} \end{bmatrix} - \begin{bmatrix} A & \\ & \Lambda \end{bmatrix} \otimes \begin{bmatrix} E & \\ & I_r \end{bmatrix} \right. \\
&\qquad \left. - \sum_{k=1}^{m} \begin{bmatrix} N_k & \\ & \tilde{N}_k{}^T \end{bmatrix} \otimes \begin{bmatrix} N_k & \\ & \hat{N}_k \end{bmatrix} \right)^{-1} \\
&\quad \times \left(\begin{bmatrix} I_n & \\ & I_r \end{bmatrix} \otimes \begin{bmatrix} I_n & \\ & S \end{bmatrix} \right) \left(\begin{bmatrix} 0 & 0 \\ 0 & e_i e_i^T \end{bmatrix} \otimes \begin{bmatrix} E & \\ & I_r \end{bmatrix} \right) \\
&\quad \times \left(\begin{bmatrix} I_n & \\ & I_r \end{bmatrix} \otimes \begin{bmatrix} I_n & \\ & S^{-1} \end{bmatrix} \right) \\
&\quad \times \left(- \begin{bmatrix} E & \\ & I_r \end{bmatrix} \otimes \begin{bmatrix} A & \\ & \hat{A} \end{bmatrix} - \begin{bmatrix} A & \\ & \Lambda \end{bmatrix} \otimes \begin{bmatrix} E & \\ & I_r \end{bmatrix} \right. \\
&\qquad \left. - \sum_{k=1}^{m} \begin{bmatrix} N_k & \\ & \tilde{N}_k{}^T \end{bmatrix} \otimes \begin{bmatrix} N_k & \\ & \hat{N}_k \end{bmatrix} \right)^{-1} \\
&\quad \times \begin{bmatrix} B \\ \tilde{B}^T \end{bmatrix} \otimes \begin{bmatrix} B \\ \hat{B} \end{bmatrix} \mathrm{vec}(I_{2m})
\end{aligned}
$$

$$
\begin{aligned}
&= 2\mathrm{vec}(I_p)^T \left(-\tilde{C} \otimes \begin{bmatrix} C & -\hat{C} \end{bmatrix}\right) \\
&\times \left(MM^T(-I_r \otimes \begin{bmatrix} A & \\ & \hat{A} \end{bmatrix} - \Lambda \otimes \begin{bmatrix} E & \\ & I_r \end{bmatrix} \right. \\
&\qquad \left. - \sum_{k=1}^{m} \tilde{N}_k{}^T \otimes \begin{bmatrix} N_k & \\ & \hat{N}_k \end{bmatrix}) MM^T \right)^{-1} \\
&\times \left(e_i e_i^T \otimes \begin{bmatrix} E & \\ & I_r \end{bmatrix} \right) \\
&\times \left(MM^T(-I_r \otimes \begin{bmatrix} A & \\ & \hat{A} \end{bmatrix} - \Lambda \otimes \begin{bmatrix} E & \\ & I_r \end{bmatrix} \right. \\
&\qquad \left. - \sum_{k=1}^{m} \tilde{N}_k{}^T \otimes \begin{bmatrix} N_k & \\ & \hat{N}_k \end{bmatrix}) MM^T \right)^{-1} \\
&\times \tilde{B}^T \otimes \begin{bmatrix} B \\ \hat{B} \end{bmatrix} \mathrm{vec}(I_m) \\
&= -2\mathrm{vec}(I_p)^T (\tilde{C} \otimes C) \left(-I_r \otimes A - \Lambda \otimes E - \sum_{k=1}^{m} \tilde{N}_k{}^T \otimes N_k \right)^{-1} \\
&\times (e_i e_i^T \otimes E) \left(-I_r \otimes A - \Lambda \otimes E - \sum_{k=1}^{m} \tilde{N}_k{}^T \otimes N_k \right)^{-1} (\tilde{B}^T \otimes B)\mathrm{vec}(I_m) \\
&+ 2\mathrm{vec}(I_p)^T (\tilde{C} \otimes \hat{C}) \left(-I_r \otimes \hat{A} - \Lambda \otimes I_r - \sum_{k=1}^{m} \tilde{N}_k{}^T \otimes \hat{N}_k \right)^{-1} \\
&(e_i e_i^T \otimes I_r) \left(-I_r \otimes \hat{A} - \Lambda \otimes I_r - \sum_{k=1}^{m} \tilde{N}_k{}^T \otimes \hat{N}_k \right)^{-1} (\tilde{B}^T \otimes \hat{B})\mathrm{vec}(I_m).
\end{aligned}
$$

The conditions for the differentiation with respect to $\tilde{N}_k$ and $\tilde{B}$ can be derived in exactly the same manner, hence they will be omitted here. Setting the derived equations to zero leads to the optimality conditions stated in Section 5.5.2.

A.3. Proof of Theorem 5.5.4

We demonstrate the following result:

Theorem A.3.1 ([12]). *Assume Algorithm 2 converges. Then* $\hat{E}^{\text{opt}}$, $\hat{A}^{\text{opt}}$, $\hat{N}_k^{\text{opt}}$, $\hat{B}^{\text{opt}}$ *and* $\hat{C}^{\text{opt}}$ *fullfil the Wilson optimality conditions (5.43)-(5.47).*

Proof. We denote by $\overline{E}$, $\overline{A}$, $\overline{N}_k$, $\overline{B}$, $\overline{C}$ the matrices corresponding to the step before the last step. A state space transformation can be used to transform this model to the optimal model, due to the convergence of the algorithm:

$$\overline{E} = T^{-1}\hat{E}^{\text{opt}}T, \quad \overline{A} = T^{-1}\hat{A}^{\text{opt}}T, \quad \overline{N}_k = T^{-1}\hat{N}_k^{\text{opt}}T, \overline{B} = T^{-1}\hat{B}^{\text{opt}},$$
$$\overline{C} = \hat{C}^{\text{opt}}T,$$

By the orthogonalization step in the Algorithm 2, we know that

$$V^{\text{opt}} = X^{\text{opt}}F, \quad W^{\text{opt}} = Y^{\text{opt}}G,$$

with $F, G \in \mathbb{R}^{r\times r}$ nonsingular. The following two Sylvester equations hold:

$$AX^{\text{opt}}\overline{E}^T + EX^{\text{opt}}\overline{A}^T + \sum_{k=1}^{m} N_k X^{\text{opt}}\overline{N}_k^T + B\overline{B}^T = 0, \tag{A.11}$$

$$A^TY^{\text{opt}}\overline{E} + E^TY^{\text{opt}}\overline{A} + \sum_{k=1}^{m} N_k^TY^{\text{opt}}\overline{N}_k - C^T\overline{C} = 0. \tag{A.12}$$

The first equation (A.11) is multiplied with $\left(W^{\text{opt}}\right)^T$ from the left, and the expressions for $\overline{E}$, $\overline{A}$, $\overline{N}_k$, $\overline{B}$, $\overline{C}$ are inserted:

$$\left(W^{\text{opt}}\right)^T AX^{\text{opt}}FF^{-1}\overline{E}^T + \left(W^{\text{opt}}\right)^T EX^{\text{opt}}FF^{-1}\overline{A}^T$$
$$+ \left(W^{\text{opt}}\right)^T \sum_{k=1}^{m} N_k X^{\text{opt}}FF^{-1}\overline{N}_k^T + \left(W^{\text{opt}}\right)^T B\overline{B}^T = 0,$$
$$\Rightarrow \left(W^{\text{opt}}\right)^T A\overbrace{X^{\text{opt}}F}^{V^{\text{opt}}}F^{-1}T^T(\hat{E}^{\text{opt}})^TT^{-T}$$
$$+ \left(W^{\text{opt}}\right)^T E\overbrace{X^{\text{opt}}F}^{V^{\text{opt}}}F^{-1}T^T(\hat{A}^{\text{opt}})^TT^{-T}$$
$$+ \left(W^{\text{opt}}\right)^T \sum_{k=1}^{m} N_k \underbrace{X^{\text{opt}}F}_{V^{\text{opt}}}F^{-1}T^T(\hat{N}_k^{\text{opt}})^TT^{-T} + \left(W^{\text{opt}}\right)^T B\hat{B}^{\text{opt}}T^{-T} = 0.$$

By multiplying with T^T from the right this leads to the following Lyapunov equation:

$$\hat{A}^{\text{opt}}F^{-1}T^T(\hat{E}^{\text{opt}})^T+\hat{E}^{\text{opt}}F^{-1}T^T(\hat{A}^{\text{opt}})^T+\sum_{k=1}^{m}\hat{N}_k^{\text{opt}}F^{-1}T^T(\hat{N}_k^{\text{opt}})^T + \hat{B}^{opt}(\hat{B}^{opt})^T = 0.$$

Under the assumption that the reduced order system is stable this equation has an unique solution and hence $P_{22} = F^{-1}T^T$. We multiply the second Sylvester equation (A.12) with $\left(V^{\text{opt}}\right)^T$ from the left and insert the given expressions, which leads to:

$$(\hat{A}^{\text{opt}})^T G^{-1}T^{-1}\hat{E}^{\text{opt}}+(\hat{E}^{\text{opt}})^T G^{-1}T^{-1}\hat{A}^{\text{opt}}+\sum_{k=1}^{m}(\hat{N}_k^{\text{opt}})^T G^{-1}T^{-1}\hat{N}_k^{\text{opt}} + (\hat{C}^{opt})^T\hat{C}^{opt} = 0.$$

Multiplying this equation with -1 gives the solution $Y_{22} = -G^{-1}T^{-1}$ and as Y_{22} is a symmetric matrix this leads to: $Y_{22} = -T^{-T}G^{-T}$. Inserting the expressions for the overlined matrices into the Sylvester equations (A.11) and (A.12) yields to the following equations:

$$AX^{\text{opt}}T^T(\hat{E}^{\text{opt}})^T + EX^{\text{opt}}T^T(\hat{A}^{\text{opt}})^T + \sum_{k=1}^{m}N_kX^{\text{opt}}T^T(\hat{N}_k^{\text{opt}})^T + B(\hat{B}^{\text{opt}})^T = 0,$$

$$A^TY^{\text{opt}}T^{-1}\hat{E}^{\text{opt}} + E^TY^{\text{opt}}T^{-1}\hat{A}^{\text{opt}} + \sum_{k=1}^{m}N_k^TY^{\text{opt}}T^{-1}\hat{N}_k^{\text{opt}} + C^T\hat{C}^{\text{opt}} = 0.$$

hence one obtains $P_{12} = X^{\text{opt}}T^T$ and $Y_{12} = Y^{\text{opt}}T^{-1}$. The Wilson conditions can now be proven:

$$\begin{aligned}&Y_{12}^TEP_{12} + Y_{22}\hat{E}^{\text{opt}}P_{22}\\ &= T^{-T}(Y^{\text{opt}})^TEX^{\text{opt}}T^T - T^{-T}G^{-T}(W^{\text{opt}})^TEV^{\text{opt}}F^{-1}T^T\end{aligned}$$

$$= T^{-T}(Y^{\text{opt}})^T E X^{\text{opt}} T^T - T^{-T} G^{-T} G^T (Y^{\text{opt}})^T E X^{\text{opt}} F F^{-1} T^T = 0.$$

with similar calculations for conditions (5.44) and (5.45). For the other conditions one obtains:

$$Y_{12}^T B + Y_{22} \hat{B}^{\text{opt}} = T^{-T}(Y^{\text{opt}})^T B - T^{-T} G^{-T} (W^{\text{opt}})^T B = 0$$
$$\hat{C}^{\text{opt}} P_{22} - C P_{12} = \hat{C}^{\text{opt}} F^{-1} T^T - C X^{\text{opt}} T^T = 0.$$

□

Bibliography

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008.

[2] S. A. Al-Baiyat and M. Bettayeb. A new model reduction scheme for k-power bilinear systems. In *Proceedings of the 32nd IEEE Conference on Decision and Control*, volume 1, pages 22–27, 1993.

[3] D. Amsallem and C. Farhat. An online method for interpolating linear parametric reduced-order models. *SIAM J. Sci. Comput.*, 33(5):2169–2198, 2011.

[4] D. Amsallem, M. Zahr, Y. Choi, and C. Farhat. Design optimization using hyper-reduced-order models. *submitted for publication*. Available from http://www.stanford.edu/~amsallem/HROM_Opt.pdf.

[5] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, Philadelphia, 2005.

[6] A. C. Antoulas, C. Beattie, and S. Gugercin. Interpolatory Model Reduction of Large-Scale Dynamical Systems. In J. Mohammadpour and K. M. Grigoriadis, editors, *Efficient Modeling and Control of Large-Scale Systems*, pages 3–58. Springer US, 2010.

[7] H.-D. Baehr and K. Stephan. *Heat and Mass Transfer*. Springer Berlin Heidelberg, 2006.

[8] Z. Bai and D. Skoogh. A projection method for model reduction of bilinear dynamical systems. *Linear Algebra Appl.*, 415(2–3):406–425, 2006.

[9] U. Baur, C. A. Beattie, P. Benner, and S. Gugercin. Interpolatory projection methods for parameterized model reduction. *SIAM J. Sci. Comp.*, 33(5):2489–2518, 2011.

[10] U. Baur and P. Benner. Modellreduktion für parametrisierte Systeme durch balanciertes Abschneiden und Interpolation (Model Reduction for Parametric Systems Using Balanced Truncation and Interpolation). *at-Automatisierungstechnik*, 57(8):411–420, 2009.

[11] P. Benner and T. Breiten. On $\mathscr{H}_2$-model reduction of linear parameter-varying systems. *Proc. Appl. Math. Mech.*, 11:805–806, 2011.

[12] P. Benner and T. Breiten. Interpolation-Based $\mathscr{H}_2$-Model Reduction of Bilinear Control Systems. *SIAM J. Matrix Anal. Appl.*, 33(3):859–885, 2012.

[13] P. Benner, S. Gugercin, and K. Willcox. A Survey of Model Reduction Methods for Parametric Systems. MPI Magdeburg Preprint MPIMD/13-14, August 2013. Available from http://www.mpi-magdeburg.mpg.de/preprints/.

[14] P. Benner and P. Kürschner. Computing Real Low-rank Solutions of Sylvester equations by the Factored ADI Method. MPI Magdeburg Preprint MPIMD/13-05, May 2013. Available from http://www.mpi-magdeburg.mpg.de/preprints/.

[15] P. Benner, J.-R. Li, and T. Penzl. Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems. *Numer. Linear Algebra Appl.*, 15(9):755–777, 2008.

[16] D. Braess. *Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*. Springer, 5. edition, 2013.

[17] T. Breiten and T. Damm. Krylov subspace methods for model order reduction of bilinear control systems. *Syst. Control Lett.*, 59:443–450, 2010.
[18] I. N. Bronstein, K. A. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik*. Verlag Harry Deutsch, 2001.
[19] C. Bruni, G. Dipillo, and G. Koch. On the mathematical models of bilinear systems. *Ricerche di Automatica*, 2:11–26, 1971.
[20] 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.
[21] A. Bunse-Gerstner, D. Kubalińska, G. Vossen, and D. Wilczek. $\mathcal{H}_2$-norm optimal model reduction for large scale discrete dynamical MIMO systems. *J. Comput. Appl. Math.*, 233(5):1202–1216, 2010.
[22] R. Castané Selga. *The Matrix Measure Framework for Projection-based Model Order Reduction*. PhD thesis, Technische Universität München, 2011.
[23] M. Condon and R. Ivanov. Krylov subspaces from bilinear representations of nonlinear systems. *COMPEL*, 26(2):399 – 406, 2007.
[24] R.S. Dahiya. Laplace transform pairs of n-dimensions. *Internat. J. Math. and Math. Sci.*, 8(3):449–454, 1985.
[25] P. D'Alessandro, A. Isidori, and A. Ruberti. Realization and structure theory of bilinear dynamic systems. *SIAM J. Control Optim.*, 3(12):517–535, 1974.
[26] T. Damm. Direct methods and ADI-preconditioned Krylov subspace methods for generalized Lyapunov equations. *Numer. Linear Algebra Appl.*, 9:853–871, 2008.
[27] 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 Des. Integr. Circuits Syst.*, 5:678–693, 2004.
[28] W. Demtröder. *Experimentalphysik 1: Mechanik und Wärme*. Springer, 3. edition, 2004.
[29] J. Donea, A. Huerta, J.-Ph. Ponthot, and A. Rodriguez-Ferran. *Arbitrary Lagrangian-Eulerian Methods*. John Wiley & Sons, Ltd, 2004.
[30] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, 1998.
[31] R. Eid. *Time Domain Model Reduction By Moment Matching*. PhD thesis, Technische Universität München, 2009.
[32] O. Farle, V. Hill, P. Ingelstroem, and R. Dyczij-Edlinger. Multi-parameter polynomial order reduction of linear finite element models. *Math. Comput. Model. Dyn. Syst.*, 14(5):421–434, 2008.
[33] L. Feng and P. Benner. A robust algorithm for parametric model order reduction. *Proc. Appl. Math. Mech.*, 7(1):1021501–1021502, 2007.
[34] G. M. Flagg. *Interpolation Methods for the Model Reduction of Bilinear Systems*. PhD thesis, Virginia Polytechnic Institute and State University, 2012.
[35] K. Gallivan, A. Vandendorpe, and P. Van Dooren. Model Reduction of MIMO Systems via Tangential Interpolation. *SIAM J. Matrix Anal. Appl.*, 26(2):328–349, 2004.
[36] K. Gallivan, A. Vandendorpe, and P. Van Dooren. Sylvester equations and projection-based model reduction. *J. Comput. Appl. Math.*, 162(1):213 – 229, 2004.
[37] M. Geuss, H. Panzer, and B. Lohmann. On parametric model order reduction by matrix interpolation. In *Proc. Eur. Control Conf. (ECC '13)*, pages 3433–3438, 2013.
[38] G. Golub and C. Van Loan. *Matrix Computations*. John Hopkins University Press, 4. edition, 2013.
[39] S. Gugercin. An iterative SVD-Krylov based method for model reduction of large-scale dynamical systems. In *Proc. 44th IEEE Conf. Decis. Control & Eur. Control Conf. (CDC-ECC '05)*, pages 5905 – 5910, 2005.
[40] S. Gugercin and A. C. Antoulas. A survey of model reduction by balanced truncation and some new results. *Internat. J. Control*, 77(8):748–766, 2004.

[41] S. Gugercin, A. C. Antoulas, and C. Beattie. $\mathscr{H}_2$ Model Reduction for Large-Scale Linear Dynamical Systems. *SIAM J. Matrix Anal. Appl.*, 30(2):609–638, 2008.
[42] C. Hartmann, A. Zueva, and B. Schäfer-Bung. Balanced model reduction of bilinear systems with applications to positive systems. Matheon Preprint 720, March 2010. Available from http://opus4.kobv.de/opus4-matheon/frontdoor/index/index/docId/706.
[43] D. Kubalinska. *Optimal interpolation-based model reduction*. PhD thesis, Universität Bremen, 2008.
[44] A. J. Laub. *Matrix Analysis for Scientists and Engineers*. SIAM, 2005.
[45] Y. Lin, L. Bao, and Y. Wei. Order reduction of bilinear MIMO dynamical systems using new block Krylov subspaces. *Comput. Math. Appl.*, 58(6):1093 – 1102, 2009.
[46] J. Lunze. *Regelungstechnik 2: Mehrgrößensysteme, Digitale Regelung*. Springer, 2010.
[47] MATLAB. *Version 7.10.0 (R2010a)*. ©2010 The MathWorks, Inc. Natick, Massachusetts. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.
[48] L. Meier and D. G. Luenberger. Approximation of linear constant systems. *IEEE Trans. Automat. Control*, 12(5):585–588, 1967.
[49] R. R. Mohler. *Nonlinear Systems - Volume II - Applications to Bilinear Control*. Prentice Hall, Englewood Cliffs, New Jersey, first edition, 1991.
[50] C. Moosmann. *ParaMOR — Model Order Reduction for parameterized MEMS applications*. PhD thesis, Universität Freiburg, 2007.
[51] COMSOL Multiphysics. Modeling Guide. *Version 3.5a*, 1998-2008.
[52] COMSOL Multiphysics. *Version 3.5a*. ©COMSOL AB, 2008.
[53] H. Panzer, J. Mohring, R. Eid, and B. Lohmann. Parametric Model Order Reduction by Matrix Interpolation. *at – Automatisierungstechnik*, 58(8):475–484, 2010.
[54] J. R. Phillips. Projection-based approaches for model reduction of weakly nonlinear, time-varying systems. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 22(2):171–187, 2003.
[55] C. Qi. *Numerical Optimization Methods On Riemannian Manifolds*. PhD thesis, Florida State University, 2011.
[56] W. J. Rugh. *Nonlinear system theory: the Volterra/Wiener approach*. Johns Hopkins series in information sciences and systems. Johns Hopkins University Press, 1981.
[57] J. Saak. *Efficient Numerical Solution of Large Scale Algebraic Matrix Equations in PDE Control and Model Order Reduction*. PhD thesis, Technische Universität Chemnitz, 2009.
[58] B. Salimbahrami and B. Lohmann. Krylov Subspace Methods in Linear Model Order Reduction: Introduction and Invariance Properties. Sci. Rep., Inst. of Automation, Uni. Bremen, 2002.
[59] T. Siu and M. Schetzen. Convergence of Volterra series representation and BIBO stability of bilinear systems. *Int. J. Syst. Sci.*, 12(22):2679–2684, 1991.
[60] G. W. Stewart and J.-G. Sun. *Matrix Perturbation Theory (Computer Science and Scientific Computing)*. Academic Press, 1990.
[61] T. Stykel. *Analysis and Numerical Solution of Generalized Lyapunov Equations*. PhD thesis, Technische Universität Berlin, 2002.
[62] R. Tahir-Kheli. *General and Statistical Thermodynamics*. Graduate Texts in Physics. Springer, 2011.
[63] H. L. Trentelman, Stoorvogel A. A., and M. Hautus. *Control Theory for Linear Systems*. Springer, 2001.
[64] P. Van Dooren, K. A. Gallivan, and P.-A. Absil. $\mathscr{H}_2$-optimal model reduction of MIMO systems. *Appl. Math. Lett.*, 21:1267–1273, 2008.
[65] A. Vandendorpe. *Model Reduction of Linear Systems, an Interpolation point of View*. PhD thesis, Université Catholique De Louvain, 2004.

[66] C. De Villemagne and R. E. Skelton. Model reductions using a projection formulation. *Internat. J. Control*, 46(6):2141–2169, 1987.

[67] D. S. Weile, E. Michielssen, E. Grimme, and K. Gallivan. A method for generating rational interpolant reduced order models of two-parameters linear systems. *Appl. Math. Lett.*, 12:93–102, 1999.

[68] Y. Xu and T. Zeng. Fast Optimal $\mathscr{H}_2$-Model Reduction Algorithms based on Grassmann Manifold Optimization. *Int. J. Numer. Anal. Model.*, 10(4):972–991, 2013.

[69] W.-Y. Yan and J. Lam. An Approximate Approach to $\mathscr{H}_2$-Optimal Model Reduction. *IEEE Trans. Automat. Control*, 44(7):1341–1357, 1999.

[70] A. Yousefi. *Preserving Stability in Model and Controller Reduction with application to embedded systems*. PhD thesis, Technische Universität München, 2006.

[71] T. Zeng, J. Chen, and C. Lu. A Tangential Interpolation Algorithm for Optimal $\mathscr{H}_2$-Model Reduction with Stability Guarantee. In *Proc. Fifth International Conference on Computational and Information Sciences (ICCIS)*, pages 955–958, 2013.

[72] Z. Zhang and J. Lam. On $\mathscr{H}_2$-model reduction of bilinear Systems. *Automatica*, (38):205–216, 2002.

Parts of this dissertation (see in particular Chapters 6 and 7.2) have been published in the following work:

A.Bruns and P.Benner. Parametric model order reduction of thermal models using the bilinear interpolatory rational Krylov algorithm. *Mathematical and Computer Modelling of Dynamical Systems*, Vol. 21, Iss. 2, 103-129, 2015.

Ehrenerklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; verwendete fremde und eigene Quellen sind als solche kenntlich gemacht.
Ich habe insbesondere nicht wissentlich:

- Ergebnisse erfunden oder widersprüchliche Ergebnisse verschwiegen,
- statistische Verfahren absichtlich missbraucht, um Daten in ungerechtfertigter Weise zu interpretieren,
- fremde Ergebnisse oder Veröffentlichungen plagiiert oder verzerrt wiedergegeben.

Mir ist bekannt, dass Verstöße gegen das Urheberrecht Unterlassungs- und Schadenersatzansprüche des Urhebers sowie eine strafrechtliche Ahndung durch die Strafverfolgungsbehörden begründen kann.
Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

(Ort, Datum)

(Unterschrift)