Model and Data Reduction for Control, Identification and Compressed Sensing

Boris Krämer

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Mathematics

John A. Burns, Chair
Jeffrey T. Borggaard
Eugene Cliff
Serkan Gugercin
Lizette Zietsman

August 11, 2015
Blacksburg, Virginia

Keywords: Model and Data Reduction, Dynamical Systems, System Identification, Optimal Control, Compressed Sensing

Model and Data Reduction for Control, Identification and Compressed Sensing

Boris Krämer

(ABSTRACT)

This dissertation focuses on problems in design, optimization and control of complex, large-scale dynamical systems from different viewpoints. The goal is to develop new algorithms and methods, that solve real problems more efficiently, together with providing mathematical insight into the success of those methods. There are three main contributions in this dissertation.

In Chapter 3, we provide a new method to solve large-scale algebraic Riccati equations, which arise in optimal control, filtering and model reduction. We present a projection based algorithm utilizing proper orthogonal decomposition, which is demonstrated to produce highly accurate solutions at low rank. The method is parallelizable, easy to implement for practitioners, and is a first step towards a matrix free approach to solve AREs. Numerical examples for $n \geq 10^6$ unknowns are presented.

In Chapter 4, we develop a system identification method which is motivated by tangential interpolation. This addresses the challenge of fitting linear time invariant systems to input-output responses of complex dynamics, where the number of inputs and outputs is relatively large. The method reduces the computational burden imposed by a full singular value decomposition, by carefully choosing directions on which to project the impulse response prior to assembly of the Hankel matrix. The identification and model reduction step follows from the eigensystem realization algorithm. We present three numerical examples, a mass spring damper system, a heat transfer problem, and a fluid dynamics system. We obtain error bounds and stability results for this method.

Chapter 5 deals with control and observation design for parameter dependent dynamical systems. We address this by using local parametric reduced order models, which can be used online. Data available from simulations of the system at various configurations (parameters, boundary conditions) is used to extract a sparse basis to represent the dynamics (via dynamic mode decomposition). Subsequently, a new, compressed sensing based classification algorithm is developed which incorporates the extracted dynamic information into the sensing basis. We show that this augmented classification basis makes the method more robust to noise, and results in superior identification of the correct parameter. Numerical examples consist of a Navier-Stokes, as well as a Boussinesq flow application.

Acknowledgements

At first, I was a little afraid of taking on the daunting task of working towards a Ph.D., but as the last three and a half years taught me, it was a great, shaping, and ever rewarding time. Nonetheless, this would have taken much longer, and would have not been so enjoyable without the help, support, and advise of many, whom I am deeply indebted to.

Thanks goes first and foremost to my advisor Prof. John A. Burns for his continued support and encouragement. His reiteration of the famous "Why?" question helped me better understand my research and see the "big picture". Needless to say, his careful pinpointing (at what I would have called nuances until then...) helped improve my research and mathematical thinking. Moreover, I learned to pay close attention to the problems of science and engineering, and how good mathematics can provide valuable and practical solutions. Prof. Burns introduced me to optimal control theory, which fundamentally and sustainably shaped my research.

I am very grateful for having learned systems theory and reduced order modeling from Prof. Serkan Gugercin. A final class project turned into a research collaboration (Chapter 4), during which I learned a great deal about numerical linear algebra, Matlab hacks, and interpolation based model reduction. His detailed guidance and many fruitful discussions tremendously improved this thesis, and nourished my enthusiasm about reduced order modeling. Thanks, for always being so helpful, patient, and supportive during my years at Tech.

Big thanks goes to Prof. John R. Singler (Missouri S&T), for putting up with me during my Masters and Ph.D. endeavor as a collaborator and friend. Somehow, our paths have crossed often, starting from my Masters thesis on Burgers' equation, to the solution of large-scale matrix equations. John has been ever so helpful both with theory, as well as technical implementations, which have greatly improved Chapter 3. There has not been a single (out of thousands) question in the past years that you left unanswered!

I would also like to thank Prof. Jeff Borggaard for the many helpful discussions and help on reduced order modeling, control, coding and fluid dynamics. Thanks to Prof. Eugene Cliff for your continued presence at ICAM, where you shared your knowledge, from theory to implementations with me. On many days I stopped by knocking on the door, and your answers always allowed me to advance in my work. Thank you Prof. Lizette Zietsman for supporting me during my time in graduate school, in particular with the SIAM VT Chapter! Moreover, thanks to Prof. Peter Haskell for being such a supportive department head, and always having an open ear to listen to me.

In my final year of my graduate school, I was fortunate to be able to intern with Mitsubishi Electric Research Laboratories in Cambridge, MA, where much of Chapter 5 originates. Dr. Petros Boufounos introduced me to compressed sensing, an intriguingly powerful method in signal processing. Dr. Piyush Grover patiently discussed fluid dynamics and dynamical systems with me, and helped shape this research direction. Thanks to both of you, and MERL for giving me this great opportunity!

Lastly, this would have not been possible without being able to ask my fellow colleagues at VT and other institutions the “dumb” questions first; Thanks to: Nabil Chabaane, Weiwei Hu, Christopher Jarvis, Zhu Wang, Alan Lattimer, Saleh Nabi (MERL), and Dave Wells.

Maintaining a sane mind in graduate school clearly made this a success, so I would like to thank all my personal friends, my family and parents for helping me in this endeavor; for listening, for entertaining, and for being there when I needed you. My years at Virginia Tech would have not been so great and memorable without all of you!

Contents

List of Figures

List of Tables

Chapter 1

Motivation and Mathematical Background

1.1 Motivation

The desire to increase efficiency in industrial systems and to make accurate predictions of future states of such systems has driven applied mathematics for decades and continues to do so today. To improve today's highly-engineered systems, sophisticated mathematics has to be involved in the design and operation of industrial products. Modern computer architectures have vast amounts of memory and processors, as well as sophisticated software for computation and smart load distribution. This enables us to solve more and more complex problems, which in turn empower us to push the limits of predictive science further. Moreover, in the age of Big Data, an excessive amount of data is available from experiments as well as real-time recording, that can be used for optimization, control and observation of such systems.[1] How can we extract the relevant information needed for the task at hand from this shear amount of data? *Can we make better decisions by utilizing the available data, together with physics based models?*

On the other hand, different challenges arise where only limited computational resources are available, or where the time to execute a codes is a deciding factor.

[1] For instance, a project at Virgina Tech placed 241 sensors and accelerometers throughout a building to record structural health and vibrations of the building. The system generates 4GB of data every hour http://www.vt.edu/spotlight/innovation/2014-05-26-seb/sensors.html.

In engineering design, models have be simulated for multitudes of input parameters, which is impractical if each simulation takes multiple days or weeks. Moreover, on-board systems in cars, airplanes, air-conditioners, etc. have only limited computational power, and full system solves are prohibitive. The implementation of reduced order models in such applications is one way to address this challenge. These quickly and cheaply executable algorithms should also leverage the availability of data, and the advancements in physics based models.

We believe that incorporating data and model reduction methods into algorithms needed for optimization, design, control and estimation is highly relevant, and is the overall subject of this dissertation. With this motivation in mind, we develop algorithms and numerical techniques, that utilize the mathematical structure and physics of the problem, and can be applied to large-scale problems. Systems of large dimension ($n > 10^5$) arise in fluid dynamics - which is our particular focus - and more generally in Big Data problems in many areas of modern science and technology.

This dissertation consists of four parts, each of which being motivated by the optimization and control of large-scale, complex systems. To keep this work self-contained, the manuscript begins with background material on numerical linear algebra, ordinary differential equations, optimal control theory, various reduced order modeling techniques, as well as compressed sensing. For some readers this might be known, and those can skip to Chapter 2. Subsequent chapters are characterized by more novel work.

Chapter 2 introduces the optimal control problem for a one dimensional Burgers' equation coupled to a heat equation, which is used to mimic the Boussinesq approximation of thermally-driven flows. Reduced order controllers are obtained through "reduce-then-design", and special care is taken to preserve the physical structure of the system. The convergence of reduced order controllers to their high fidelity counterpart is demonstrated, and various POD implementation variants, such as the selection of the input ensemble used for computation of reduced order models, are discussed and numerically tested. Moreover, we investigate "out-of-sample" performance of the algorithms, when the underlying parameters of the dynamical system changed. This chapter raises important questions pertinent to reduced order models and control, illustrates the necessary discretization steps, and highlights many of the problems arising in more complex applications.

Optimal feedback control and filtering for large-scale systems is required in a wide range of applications, such as electrical circuits, fluid dynamics, design of integrated systems, etc. The solution of the optimality problem can be obtained by solving algebraic Riccati equations (ARE), which are nonlinear matrix equations. In Chapter 3 we provide a new algorithm to solve large-scale AREs via the method of proper orthogonal decomposition. The algorithm is particularly designed for practitioners in the field and provides a first step towards a matrix-free algorithm for solution of ARE. Compared to other state-of-the-art algorithms for Riccati equations, we were able to achieve high accuracy at low approximation rank.

In Chapter 4, we propose a new algorithm for system identification of multi-input, multi-output (MIMO) dynamical systems, which significantly reduces the computational cost of current methods. This research is motivated by using reduced order models for design and control of passive (or "smart") heating and ventilation systems, and system identification for complex dynamics in general. Specifically, when the model under consideration has a large number of both inputs and outputs, the computational burden for current system identification methods increases. To circumvent this problem, we interpolate the impulse responses along suitably chosen "tangential" directions and modify the conventional eigensystem realization algorithm by Kung [124] to recover the full input and output dimension at the final stage. The success of this algorithm is demonstrated on two applications, a mass-spring-damper system and a model of cooling steel profiles in a rolling mill. Using these test problems, we demonstrate that the computational cost decreases significantly, while maintaining a satisfactory level of accuracy. Importantly, we also give indicators where this algorithm can fail to produce accurate results, and therefore provide the practitioner with valuable guidance for the success of the method.

Finally, in Chapter 5, we design a new robust sensing algorithm for complex flows, with an application to airflow sensing in an indoor environment. The motivation is to design smart, passive heating, ventilation and cooling systems (HVAC) to reduce energy consumption in buildings. The incorporation of the physics and experimental data of thermally driven flows into control and sensing mechanisms can be essential to achieve this task. In this work, compressed sensing, a recently developed theory for efficient and robust signal sampling [56], is employed together with a reduced order surrogate model for the Navier-Stokes and Boussinesq equations. The goal is to use local (parametric) models for observation and control of such systems. To utilize the local models effectively, we sense noisy flow and temperature information, and classify the data into dynamic, parameter dependent regimes. The particular emphasis is on boundary sensing, which is deemed most practical for the application at hand. Once a reliable estimate of current operating conditions or flow patterns is obtained, controllers can make use of such information by selecting reduced order model suitable for this dynamic regime. The developed classification algorithm is based on a regime-library and is robust to noise. For high levels of noise, we suggest a new, even more robust "augmented" sensing basis, that incorporates the temporal dynamics of the flow regimes. With the purely data-based dynamic mode decomposition (DMD), we are able to extract dynamic features from the data, which approximate the underlying forward operator. With this method, we leverage the knowledge of time evolution of signals through reduced order models, and we are able to correctly identify the dynamic operating regime in more than 90% of the cases. To achieve the same classification performance without incorporating the dynamics, many more sensors would be needed, which is impractical.

Chapters 3–5 of this dissertation contain material that is either submitted for publication in the literature, or will be in the near future. In all chapters, I most directly developed the methods and derivations, worked on the numerical part, and did the writing. I developed and wrote the majority of the codes in this dissertation, minus some suggestions for improvement from co-authors and collaborators, which I carefully incorporated into the code.

1.2 Matrix Theory

Let $x \in \mathbb{C}^n$ be a vector and let $||x||_1 = \sum_{i=1}^n |x_i|$ and $||x||_2^2 = \sum_{i=1}^n |x_i|^2$ denote the 1 and 2 norm of x, respectively. The inner product of two vectors is defined by $(x, y)_{\mathbb{C}^n} = \sum_{i=1}^n x_i \bar{y}_i$ and the subscript is omitted where the dimensions are clear. Here, $\bar{y}_i$ is the complex conjugate of y_i. By $||x||_\infty = \max_i |x_i|$ we denote the infinity norm. The norms are related as

$$||x||_\infty \leq ||x||_2 \leq ||x||_1 \leq \sqrt{n}||x||_2.$$

Additionally, the notation

$$||x||_0 := \text{card}\{i : x_i \neq 0\}$$

denotes the number of non-zero elements in x. Note, that $||x||_0$ is not a norm, since it fails to satisfy the linearity assumption.

Let $A \in \mathbb{R}^{n \times m}$ be a matrix and denote by $A^T = [a_{ij}]^T = [a_{ji}]$ the *transpose* of a matrix; if $A \in \mathbb{C}^{n \times n}$, then $A^* = [a_{ij}]^* = [\bar{a}_{ji}]$ denotes the *conjugate transpose* of A. A real matrix is called *symmetric* if $A = A^T$; a complex matrix is called *self-adjoint*

(Hermitian)) if $A = A^*$. Sometimes, our notation is inspired by `Matlab`[2], so that

$$a_{:,i} := \ i\text{th column of } A, \qquad a_{j,:} := \ j\text{th row of } A,$$

and

$$A_{1:l,1:k} := \text{ first } l \text{ rows of the first } k \text{ columns of } A.$$

By $|A| = \det(A)$ we denote the *determinant* of A and it is well known that the following holds:

$$\{a_{:,1}, a_{:,2}, \ldots, a_{:,n}\} \text{ is linear independent } \quad \Leftrightarrow \quad \det(A) \neq 0.$$

In other words, the columns of A are linearly independent if and only if the determinant of A is nonzero, and in this case $Ax = b$ has a unique solution $x = A^{-1}b$.

Moreover, define the 2-induced norm

$$||A||_2 := \sup_{x \in \mathbb{R}^n} \frac{||Ax||_2}{||x||_2},$$

which satisfies

$$||A||_2 = \lambda_{max}^{1/2}(A^TA) = \lambda_{max}^{1/2}(AA^T) = \sigma_1(A),$$

where λ denotes eigenvalues and the σ_i are the singular values of A, as defined below. The Frobenius norm of a matrix is defined as

$$||A||_F^2 := \sum_k \sum_l |a_{kl}|^2,$$

and a convenient result allows for an economical calculation:

$$||A||_F^2 = \sum_{i=1}^{\min(n,m)} \sigma_i^2(A) = trace(A^TA).$$

For large problems, the Frobenius norm is still computable, as long as the product AA^T can be formed (it does not have to be stored though!). However, the matrix 2-norm relies on purely spectral information, and hence can become expensive for large matrices. We refer the reader to Golub [89, pp.57+71] for the previous equivalences and more details. Note, that the following simple equality holds $||A||_2^2 = 1 \Rightarrow$

[2] © 2015 The MathWorks, Inc. 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.

$||A||_F^2 = n$, which follows directly from the trace definition. Moreover, the Frobenius and 2-norm are invariant under orthogonal transformations. This is particularly helpful for practical computations, and is used in Chapter 3 for computation of the residual norm. In particular, if $A = QR$ is the standard QR-decomposition as defined below, then $||A||_F = ||R||_F$, and similarly $||A||_2 = ||R||_2$, which is used later to cheaply evaluate the norm of a residual matrix. Another useful property of the Frobenius and matrix 2-norm is that both satisfy the submultiplicativity $||AB|| \leq ||A|| ||B||$. In the following, we state matrix decompositions that are most frequently used in this thesis.

Theorem 1.2.1. *[89, Thm.5.2.1] Let $A \in \mathbb{R}^{m\times n}$ with $m \geq n$ be a matrix with full column rank. Then A has a QR-decomposition, where $A = QR$, $Q \in \mathbb{R}^{m\times m}$ is orthogonal, and $R \in \mathbb{R}^{m\times n}$ is upper triangular. The following holds:*

- $\text{span}\{a_{:,1}, \dots, a_{:,k}\} = \text{span}\ (q_{:,1}, \dots, q_{:,k}) \qquad k = 1, \dots, n.$
- $\text{ran}(A) = \text{ran}(Q_{1:m,1:n})$
- $\text{ran}(A)^\perp = \text{ran}(Q_{1:m,n+1:m})$
- $A = Q_{1:m,1:n}R_{1:n,1:n}$, (i.e., we can leave redundant columns out).

The QR-decomposition is therefore an alternative to Gram-Schmidt orthogonalization. In the particular case where A is "thin", i.e. $m \gg n$, the factorization can be implemented in an efficient manner, as done in `Matlab` with the command `qr`$(A, 0)$. This is our method of choice for orthogonalizing vectors in this thesis.

Theorem 1.2.2. *[89, Thm 3.2.1] The matrix $A \in \mathbb{R}^{n\times n}$ has an LU-factorization $A = LU$, where L is a lower triangular matrix (with ones in the diagonal) and U is an upper triangular matrix, if the leading $n-1$ submatrices of A are invertible. If the LU factorization exists and A is nonsingular, then* $\det(A) = u_{11} \cdots u_{nn}$.

The LU decomposition is frequently used when linear systems of the form $Ax = b_i$ have to be solved for $i = 1, \dots, k$. By precomputing the LU decomposition, the linear solve can be substantially simplified by first solving $Ly = b_i$, and subsequently $Ux = y$. Next, we consider both the symmetric and unsymmetric eigenvalue decompositions. The eigenvalue decomposition (and the related SVD) are at the heart of many model reduction and feature extraction algorithms.

Theorem 1.2.3. *[89, Thm.8.1.1] If $A \in \mathbb{R}^{n\times n}$ is symmetric, then there exists a real orthogonal matrix V such that*

$$V^T AV = \Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_n),$$

and the eigenpairs satisfy $Av_k = \lambda_k v_k$ for $k = 1, \dots, n$.

Theorem 1.2.4. *[89, Thm.7.1.9] If $A \in \mathbb{C}^{n\times n}$, then there exists a nonsingular V such that $V^{-1}AV = diag(J_1, \dots, J_t)$. The $J_i \in \mathbb{R}^{m_i \times m_i}$ are called Jordan blocks with λ_i on the diagonal and ones on the superdiagonal, and $\sum m_i = n$.*

The computation of the eigenvalue decomposition for nonsymmetric, large matrices is an active area of research. A general matrix A is called non-defective, if it has a distinct set of eigenvectors. In this case, A is diagonalizable, and the Jordan form is the standard eigenvalue decomposition. However, note that the eigenvectors do not have to be orthogonal! On another note, if the goal is to enforce unitary, eigenvectors, then one has to relax the "eigenvalue" matrix to become upper triangular.

Theorem 1.2.5. ***Schmidt-Eckardt-Young-Mirsky*** *[8, p.37]. Every matrix $A \in \mathbb{C}^{n\times m}$ with $m \leq n$ has a singular value decomposition (SVD). That is $A = U\Sigma V^*$, where U, V are unitary matrices, $UU^* = I_n, VV^* = I_m$ and $\Sigma_{ii} = \sigma_i \geq 0,\ i = 1, \dots, n$ are the singular values. The rank of the matrix is determined by the number of nonzero singular values. The best approximation error of A via a rank r matrix is given by*

$$\min_{rank(X)=r} ||A - X||_2 = \sigma_{r+1}(A), \tag{1.1}$$

and in the Frobenius norm

$$\min_{rank(X)=r} ||A - X||_F = \left(\sum_{i=r+1}^{n} \sigma_i^2(A) \right)^{1/2}, \tag{1.2}$$

provided that $\sigma_r < \sigma_{r+1}$. The (nonunique) minimizer to the approximation problem is given by the truncated singular value decomposition

$$X_* = U_r \Sigma_r V_r^T,$$

where U_r, V_r^T denote matrices of the first r columns or rows of U, V^T, respectively, and Σ_r is the leading $r \times r$ submatrix of Σ.

For symmetric matrices, the eigenvalue decomposition (EVD) and singular value decomposition are closely related, i.e. $|\lambda_i(A)| = \sigma_i(A)$ and the left eigenvectors and left singular vectors agree up to a sign change, see [8, p.37]. When computing the proper orthogonal decomposition of a dataset, one uses this property to have a more stable implementation of the algorithm, see the subsection on POD below.

Solving linear systems of the form $Ax = b$ is required in a large number of numerical algorithms for optimization, control, simulation, sensing, parameter estimation, and many more. However, the data b is often corrupted or inaccurate, and one would like to know the sensitivity of the solution with respect to errors in the data. This leads to the definition of the condition number.

Definition 1.2.6. The *condition number* of $A \in \mathbb{R}^{n\times n}$ is a measure for the sensitivity of $Ax = b$ to changes in the data b and is defined as

$$\kappa := ||A|| \cdot ||A^{-1}||,$$

and for the spectral norm, the norm is given as the ratio of the largest to smallest singular value:

$$\kappa = \frac{\sigma_1}{\sigma_n}.$$

Orthogonal matrices have perfect conditioning, i.e. $\kappa = 1$, and singular matrices have infinite condition number. Large condition numbers are concerning where linear solves are involved, since those can be extremely sensitive and therefore challenge every algorithmic implementation. The following result is standard and more detail can be found in [70, p.34]. To get a practical error bound, let $\hat{x} \in \mathbb{R}^n$ be any vector, for which we want to know its distance to the solution x of $Ax = b$, and denote it by $\Delta x = \hat{x} - x$. Therefore, define the residual $\epsilon = A\hat{x} - b$. Then, $\Delta x = A^{-1}(A\hat{x} - b)$ and consequently

$$||\hat{x} - x||_2 \leq ||A^{-1}||_2 \cdot ||\epsilon||_2.$$

In other words, the error in the solution depends on the error in the data, but is magnified by the condition number. This illustrates that small condition numbers are beneficial for accurately solving linear systems.

1.3 Partial Differential Equations and Approximation

In this dissertation, we are concerned with model reduction methods and algorithms for control, simulation and optimization of large systems. One instance, where such large models arise are after discretization of partial differential equations (PDE's). Here, we briefly introduce partial differential equations to illustrate the mathematical framework, and focus on some relevant results. By no means does this dissertation attempt to thoroughly address theoretical issues related to PDE's and infinite dimensional systems.

The quest to understand and control physical phenomena often starts with a model given by a partial differential equation (PDE). Those models arise from first principles such as conservation laws [128, §3.3] for mass, momentum, and/or energy. Partial differential equations are therefore widely considered good *models*[3] for physical phenomena, since a process can be modeled through various variables and dependencies, allowing for great flexibility. Partial differential equations model functional dependencies of quantities of interest, e.g., velocity in flows (Navier-Stokes equation), oscillation in electromagnetic fields (Helmholtz), a wave function in quantum mechanics (Schroedinger), potential fields (Laplace), and many more.

In Chapter 5, the Boussinesq equations are used to model the viscous, convective and buoyant forces for indoor-air behavior. The buoyancy arises due to temperature gradients in the fluid. The classical approach of conservation of mass and momentum leads to the Boussinesq equations

$$
\begin{aligned}
0 &= \nabla \cdot \mathbf{u} && (1.3)\\
\mathbf{u}_t &= \mu \Delta \mathbf{u} - (\mathbf{u} \cdot \nabla)\mathbf{u} - \nabla p - T + f && (1.4)\\
T_t &= k \Delta T - \mathbf{u} \cdot \nabla T, && (1.5)
\end{aligned}
$$

defined on $[0, T] \times \Omega$, where $T(t, x)$ is a scalar temperature function, $p(t, x)$ is a pressure field, $f(\cdot, \cdot)$ is a body force (e.g., gravity), $\mathbf{u}(t, x)$ is a vector-valued flow variable, and μ can be thought of a viscosity-like parameter. Moreover, we introduce square integrable initial conditions $T(0, x) = T_0(x)$ and $\mathbf{u}(0, x) = \mathbf{u}_0(x)$. A typical set of boundary conditions is to have a hot temperature on some portion of the boundary and a cold wall temperature otherwise. For the boundary of the flow field, we often consider prescribed inflows, or no-slip conditions. It is customary to

[3] We shall not forget though, that there is a *reality* beyond models.

eliminate the static pressure (hiding it somehow in T), so that one only has to solve for $\mathbf{u}$ and T. The Boussinesq equations are ultimately the model of choice, since they capture all relevant physical phenomena of wall bounded, thermally (natural convection) driven flow. Where temperature gradients are negligible, we use the Navier-Stokes equations instead.

1.3.1 The Finite Element Method

To illustrate the process of obtaining approximate ODE models from PDE's, we choose the finite element method (FEM) for its desirable mathematical features and wide applicability. For ease of presentation, assume that $T(t,x) = 0$, i.e. consider the incompressible Navier-Stokes (NS) equations

$$\begin{aligned} \mathbf{u}_t &= \mu\Delta\mathbf{u} - (\mathbf{u}\cdot\nabla)\mathbf{u} - \nabla p + f, && (1.6)\\ 0 &= \nabla\cdot\mathbf{u}, && (1.7)\\ \mathbf{u} &= 0 \text{ on } \partial\Omega, && (1.8)\\ \mathbf{u}(0,x) &= \mathbf{u}_0(x), && (1.9) \end{aligned}$$

where the variables have the same meaning as above. The book of Layton [128] gives an accessible introduction to the finite element method for viscous, incompressible flows and points to much of the existing literature for more in depth study. In the remainder, we assume that $\mu \gg 0$, since otherwise the dynamics of the NS-equations can produce turbulent flows, requiring more sophisticated discretization schemes. First, we define the notion of a solution, followed by a brief discussion of convergence to (analytical) solutions. Let $\mathbf{u}_i(\cdot)$ denote the i^{th} component of $\mathbf{u}$ and define

$$\mathcal{X}_{\mathbf{u}} = H_0^1(\Omega)^d := \{\mathbf{u}_i(\cdot) \in L^2(\Omega) \ : \ ||\nabla\mathbf{u}_i||_2 < \infty, \mathbf{u}_i(x) = 0 \text{ on } \partial\Omega\}^d$$

as the Hilbert space of functions with square integrable derivatives which vanish on the boundary. The notation $\{\}^d$ denotes the d-fold product space. Similarly, we define $L_0^2(\Omega)$ to be the space of square integrable functions which also vanish, almost everywhere, on the boundary. For notational convenience, let

$$\mathcal{X}_p := L_0^2(\Omega) = \{p(\cdot) \in L^2(\Omega) \ : \ \int_\Omega p(x)dx = 0\},$$

and the space of divergence free functions be

$$\mathcal{X}_{div=0} := \{\mathbf{u}(\cdot) \in \mathcal{X}_{\mathbf{u}} \ : \ (p, \nabla\cdot\mathbf{u}) = 0, \forall p \in \mathcal{X}_p\}.$$

Definition 1.3.1. [128, p.153] The pair $\{\mathbf{u}, p\}$ is a *strong solution* to (1.6) - (1.9) if $\mathbf{u} \in L^2(0,T;\mathcal{X}_{\mathbf{u}}) \cap L^\infty(0,T;L^2(\Omega))$ and

1. $\mathbf{u} : [0,T] \mapsto \mathcal{X}_{\mathbf{u}}$ is differentiable, $\mathbf{u}_t : (0,T] \mapsto \mathcal{X}^*_{\mathbf{u}}$ is integrable and $p : (0,T] \mapsto \mathcal{X}_{\mathbf{u}}$ is continuous.
2. For every $t \in (0,T]$, all $\mathbf{v} \in L^2(0,T;H^1_0(\Omega)) \cap L^\infty(0,T;L^2(\Omega))$, and all $q \in L^2(0,T;L^2_0(\Omega))$ we have

   $$\int_0^t (\mathbf{u}_t, \mathbf{v}) + (\mathbf{u} \cdot \nabla \mathbf{u}, \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) + \nu(\nabla \mathbf{u}, \nabla \mathbf{v}) dt = \int_0^t (f, \mathbf{v}) dt$$

   and

   $$\int_0^t (q, \nabla \cdot \mathbf{u}) dt = 0.$$

3. $\mathbf{u}_0 \in \mathcal{X}_{div=0}$ and $||\mathbf{u}(t,\cdot) - \mathbf{u}_0|| \to 0$ as $t \to 0$.
4. $\mathbf{u} \in L^4(0,T;\mathcal{X}_{\mathbf{u}})$.

Having a mathematically sound definition of a solution, we next show that a suitable approximation scheme will guarantee convergence of solutions. To this end, we consider the finite dimensional approximation spaces

$$\mathcal{X}^h_{\mathbf{u}} \subseteq \mathcal{X}_{\mathbf{u}}, \quad \mathcal{X}^h_p \subseteq \mathcal{X}_p.$$

The approximations of the velocity and pressure function, $\mathbf{u}^h$ and p^h, in the above spaces need to satisfy the variational system

$$(\mathbf{u}^h_t, \mathbf{v}^h) + b^*(\mathbf{u}^h, \mathbf{u}^h, \mathbf{v}^h) + \nu(\nabla \mathbf{u}^h, \nabla \mathbf{v}^h) - (p^h, \nabla \cdot \mathbf{v}^h) = (f, \mathbf{v}^h), \tag{1.10}$$

$$(\mathbf{u}^h(0,\cdot) - \mathbf{u}_0, \mathbf{v}^h) = 0, \tag{1.11}$$

for all $\mathbf{v}^h \in \mathcal{X}^h_{div=0}$. Here, $b^*(\mathbf{u}, \mathbf{v}, \mathbf{w}) := \frac{1}{2}(\mathbf{u} \cdot \nabla \mathbf{v}, \mathbf{w}) - (\mathbf{u} \cdot \mathbf{w}, \mathbf{v})$ is the skew-symmetrized trilinear form. Additionally, the FEM approximating spaces need to satisfy the famous Ladyzhenskaya-Babuska-Brezzi $(\inf - \sup)$ condition, see [128, p.62]. It is well known, that under those conditions, the approximation problem is well defined, and a unique solution exists [128, Ch.9].

The construction of approximation spaces, i.e. the selection of proper basis functions (shape and polynomial order), goes beyond the scope of this introduction. One can think of basis functions in the approximation spaces as low order polynomials with support only over small subregions of the physical domain. A proper selection of the FE basis functions guarantees convergence at predefined rates. For one particular choice of a basis, namely the Taylor-Hood finite elements, an error estimate of the form

$$\sup_{0 \leq t \leq T} ||\mathbf{u} - \mathbf{u}^h||^2 + \nu \int_0^T ||\nabla \mathbf{u} - \nabla \mathbf{u}^h||^2 dt \leq C(\mathbf{u}, p, \nu) \cdot h^4$$

can be obtained, see [128, p.161]. A priori error bounds of this kind specify the asymptotic behavior of the error. They can, and should be, complemented by a posteriori error computations, to asses the actual error of the solution at hand.

The approximation (1.10) - (1.11) is a system of ordinary differential equations, since the spatial dependence is integrated out by virtue of the test functions. If one expands the dependent variable $\mathbf{u}$ in the same basis as the test functions (yielding a Galerkin system), i.e.

$$\mathbf{u}(t) = \sum_{i=1}^{n} a_i(t)\phi_i(x),$$

then the above system can be written as

$$\begin{aligned} E\dot{a}(t) &= Aa(t) + a^T N a + d + F(t), \\ a(0) &= a_0, \end{aligned}$$

where N is a tensor, and the terms $A, d, F(\cdot)$ are defined appropriately [128]. In essence, the goal is to show that spatial discretization of a PDE leads to a *system* of ODE's, which has to be solved using a time integrator.

The previous discussion about finite element methods for forward solution of the partial differential equation, gave a flavor of the solid foundation of FEM for simulation of complex systems. However, approximation schemes yielding finite dimensional models for optimization and control need to satisfy additional criteria and special care should be taken in this case.

Remark 1.3.2. In Subsection 1.5, the important system theoretic concepts of controllability and observability are introduced, and those ideas can be extended to infinite dimensional PDE systems as well. For an approximation scheme to retain e.g., controllability, additional conditions have to be imposed, see [51, §2, Prop.1.20]. Moreover, Ito [105] (see also the appendix) states that convergence combined with dual convergence and preservation of exponential stability of the associated semigroups guarantee that the computed control law converges to its infinite dimensional counterpart. In [17], Banks and Burns introduce the "AVE" scheme which can be used for discretization of control problems, as well as systems with delays. In short, care should be taken when using finite element schemes for optimization and control, as the "right" discretization for simulation could be the "wrong" discretization for control.

1.4 Ordinary Differential Equations

Ordinary differential equations (ODE) are at the heart of dynamical systems. They arise directly from modeling, or through semidiscretization of partial differential equations, as illustrated in the previous section. ODE's are well studied mathematical models, with an exhaustive theory ranging from existence and uniqueness and solution techniques (analytical and numerical) to asymptotic behavior and perturbation analysis. Most of the results required in this thesis are taken from [141], which is an excellent and concisely written introduction to the subject. Consider the general, nonlinear ordinary differential equation of the form

$$\dot{x}(t) = F(t, x(t)), \qquad t > 0 \tag{1.12}$$

$$x(0) = x_0, \tag{1.13}$$

where $F : \mathbb{R}^n \mapsto \mathbb{R}^n$ is a (generally) nonlinear function, and t denotes the independent time variable. For control and systems theory, the right hand side is often linear and time invariant, plus a control term (disturbance) is added such that

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad t > 0. \tag{1.14}$$

Such systems arise naturally through linear relations of quantities of interest, or through linearization of nonlinear dynamics.

1.4.1 Existence and Uniqueness

The existence and uniqueness of solutions to the initial value problem are first addressed, followed by a closed form solution expression.

Theorem 1.4.1. *[141, p.55] Let $F(\cdot,\cdot)$ be continuous on $\mathcal{D} = (0,\infty) \times \Omega$, where Ω is a connected domain in $\mathbb{R}^n$. Moreover, let F satisfy a Lipschitz condition in $\mathcal{D}$, i.e.*

$$||F(t,x) - F(t,y)|| \le L||x-y||, \qquad \forall (t,x),(t,y) \in \mathcal{D}$$

and some finite L. Then for any $(t_0, x_0) \in \mathcal{D}$, there exists as unique solution to (1.14)-(1.13) for all times $t \in (0, \infty)$.

The continuity condition on $F(\cdot, \cdot)$ guarantees existence, and the Lipschitz condition implies the uniqueness of solutions. A Lipschitz constant of $L = 1$ defines the usual notion of continuity. In the special case of

$$F(t, x(t)) = Ax(t) + Bu(t),$$

existence and uniqueness is straightforwardly verified. Moreover, there is a closed form solution, called the *Variation of parameters* or *Variation of constants* formula [141, p.98ff], given by

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-s)}Bu(s)ds, \quad \forall t \in (0, \infty). \tag{1.15}$$

The reader should note that there is a separate contribution to the solution from the propagation of the initial condition and from the external disturbance. The exponential matrix operator $\{e^{tA}\}_{t\geq 0}$ is more abstractly referred to as a C_0-semigroup. The concept of semigroups was originally formulated for infinite dimensional systems [67], and the reader is referred to the appendix for further information on operator semigroups. The above closed form solution is a convenient tool for analysis of systems of ODE's and can also be used to guide time integration (e.g., Rosenbrock-type) methods.

1.4.2 Stability

In this subsection, we introduce the notion of (Lyapunov) stability and give a perturbation result, known as Perron's theorem. This states that linear stability effects dominate "small" nonlinearities as $t \to \infty$, which in turn justifies linear control mechanisms.

Definition 1.4.2. ([141, p.183]) A matrix $A \in \mathbb{R}^{n\times n}$ is called *stable* or *Hurwitz* if all of its eigenvalues have negative real parts. If at least one of the eigenvalues has a positive real part, the matrix is called *unstable*.

This definition implies that the linear solution operator $\{e^{tA}\}_{t\geq 0}$ is uniformly exponentially bounded. Thus, there are constants $M_1 \geq 1$ and $\omega_1 > 0$ such that

$\|e^{tA}\|_2 \leq M_1 e^{-\omega_1 t}$. We shall next see, that for systems with linear and nonlinear part, stability can carry over from the linear part, as long as the nonlinearity is well behaved. This can be particularly helpful for systems that were linearized around an equilibrium solution.

Theorem 1.4.3 (Perron)**.** *([141, p.261]). Let $A \in \mathbb{R}^{n\times n}$ be stable and let $B(h) := \{x : ||x||_2 < h\}$ be the ball of radius h around the origin. Let $F : \mathbb{R}^+ \times B(h) \mapsto \mathbb{R}^n$ be continuous in (t,x) and satisfy*

$$F(t,x) = o(||x||) \quad \text{as } ||x|| \to 0,$$

uniformly in $t \in (0,\infty)$. Then the trivial solution of

$$\dot{x}(t) = Ax(t) + F(t,x)$$

is uniformly asymptotically stable.

1.4.3 Time Discretization and Stability

The finite element method and reduced order models yield (spatially) semi-discretized systems of ordinary differential equations, which have to be solved with time integration schemes. Here, we introduce the so called "θ-method" [93] for time discretization. For ease of presentation, consider the system of autonomous ordinary differential equations

$$\dot{x}(t) = F(x(t)), \qquad t \in [0,T], \tag{1.16}$$

where $x \in \mathbb{R}^n$, $F : \mathbb{R}^n \mapsto \mathbb{R}^n$. Let $\Delta t = T/s$ be a time step size and set $t_i = t_0 + i\Delta t$ for $i = 1,\dots s$. Let $x^m = x(t_m)$ and define the interpolation between x^m and x^{m-1} to be

$$x_\theta^m = \theta x^m + (1-\theta)x^{m-1}, \quad \theta \in [0,1].$$

We can then approximate equation (1.16) with the finite difference in time

$$\frac{x^m - x^{m-1}}{\Delta t} = F(x_\theta^m). \tag{1.17}$$

For $\theta = 0$, this yields the standard *forward Euler* method, for $\theta = 1$, the *backward Euler* scheme and for $\theta = 1/2$ the *Crank-Nicolson* method. The reader should observe that the nonlinearity needs information from both states x^m and x^{m-1} and so is not easy to implement. Thus, use

$$\frac{x^m - x^{m-1}}{\Delta t} = \frac{\theta x^m + (-\theta+1)x^{m-1} - x^{m-1}}{\theta\Delta t} = \frac{x_\theta^m - x^{m-1}}{\theta\Delta t}$$

and insert into equation (1.17) such that

$$\frac{x_\theta^m - x^{m-1}}{\theta \Delta t} = F(x_\theta^m). \tag{1.18}$$

Consequently, in every time step, solve the nonlinear algebraic equation

$$\frac{1}{\theta \Delta t} x_\theta^m - F(x_\theta^m) - \frac{1}{\theta \Delta t} x^{m-1} = 0 \tag{1.19}$$

for x_θ^m and compute the sought solution as

$$x^m = \frac{1}{\theta} x_\theta^m - \frac{1-\theta}{\theta} x^{m-1}.$$

Equation (1.19) is solved with a Newton-type method, where the initial guess at every time step is set to x_θ^{m-1}.

A time discretization scheme can give erroneous results, if the time step Δt is chosen too large. In fact, there is a rich theory of stability of time discretization schemes, both for standard ODE's, and PDE's in general, see [131]. Convergence of a finite difference scheme can be cast by the following definition.

Definition 1.4.4. [131, p.137] A one-step time discretization scheme is said to be *convergent*, if applying the method to any ODE of the form 1.16, with $F(\cdot)$ Lipschitz continuous, and with any set of starting values satisfying $x_0 = x(0)$, we obtain convergence in the sense that

$$\lim_{\substack{\Delta t \to 0 \\ n_t \Delta t = T}} x_{n_t} = x(T), \quad \forall T > 0,$$

where n_t denotes the number of time steps taken.

Convergence of a one-step time discretization scheme follows from stability and consistency of the method. For one-step schemes, stability is trivially satisfied, and consistency (that errors during one time step go to zero as $\Delta t \to 0$) needs to be shown. In general terms, the (explicit) forward Euler is unstable, unless a sufficiently small time step is taken. The (implicit) backward Euler scheme is unconditionally stable, with a first order convergence behavior. The Crank-Nicolson scheme is second order convergent, numerically stable, and hence often a good method to start with. More sophisticated methods, such as multi-step approaches, can be found in [131]. The reader should observe, that the above definition is an asymptotic expression, and of little use in practice. For an explicit time step Δt, one can find the regions of stability for the time integration methods, which involve the eigenvalues of the system matrix (linear case) or Jacobian (nonlinear case) [131, Ch.7]. The reader should also note, that the above ODE's often arise from a PDE discretization. In that respect, the size of the state space grows as the spatial grid is refined, adding additional complexity to the problem. In particular, the previously mentioned eigenvalues change with every mesh refinement, and satisfying the stability criterion leads to a relationship between the spatial mesh size h and the time step Δt, the well known *Courant-Friedrichs-Lewy* condition [131, §10.7]. In this thesis, the simulations were performed such that the CFL condition is met, or adaptive time stepping schemes are used that automatically adjust the time step so that the methods converge.

1.5 Control and Systems Theory

The field of systems theory finds widespread use in many areas of engineering, natural sciences and the life sciences. This is in part due to the maturity of the study of linear, time invariant (LTI) systems and related system features, see [126, 8] and the references therein. We shall mention the necessary concepts needed in this thesis and refer the interested reader to the excellent references given throughout for a more detailed study of systems theory.

1.5.1 Abstract Formulation of control problem

For a general framework, let $\Omega \in \mathbb{R}^d, d = 1, 2, 3$ be a domain, and let

$$Z = L^2(0, \infty; \Omega)$$

be the state space of the dynamical system. By $z(t, \cdot) \in Z$, one denotes the state variable of the dynamical system. The equivalence class $[z(t, \cdot)]$ is identified with a representative $z(t)$ meaning that for every fixed, positive and finite t, the function $z(t) \in Z$. After linearization of a possibly nonlinear dynamical system, define the linear operator

$$\mathcal{A} : \ \mathcal{D}(\mathcal{A}) \mapsto Z.$$

Further, let $U = \mathbb{R}$ be the space of control inputs. Similarly, we identify the control action $[\mathcal{B}u](\cdot)$ with $\mathcal{B}u \in Z$. The control operator $\mathcal{B} \in \mathcal{L}(U, Z)$ shall be

$$[\mathcal{B}u](t) = Bu(t).$$

When information about the entire state is not available or not of interest, one considers the sensed output

$$y(t) = \mathcal{C}z(t) \quad \in Y,$$

a Hilbert space. The operator $\mathcal{C} \in \mathcal{L}(Z, Y)$ is called the observation operator. The infinite dimensional linear dynamical system can then be written as

$$\dot{z}(t) = \mathcal{A}z(t) + \mathcal{B}u(t), \tag{1.20}$$

$$y(t) = \mathcal{C}z(t), \tag{1.21}$$

$$z(0) = z_0 \in Z. \tag{1.22}$$

With the variation of parameters formula, the solution to the system (1.20) - (1.22) is formally given by

$$z(t) = S(t)z_0 + \int_0^t S(t-s)\mathcal{B}u(s)ds,$$

where $S(t)$ generates an analytic semigroup on Z. To set up an optimal control problem, define the cost functional as

$$J(u, z) = \int_0^\infty \left\{ ||\mathcal{C}z(t)||_Y^2 + R||u(t)||_U^2 \right\} dt, \tag{1.23}$$

where $R \in \mathbb{R}$ represents a cost attributed to the control action.

Definition 1.5.1. Let $\mathcal{A} : \mathcal{D}(\mathcal{A}) \mapsto Z$ be the generator of an analytic semigroup and $\mathcal{B} \in \mathcal{L}(U, Z)$ a bounded control operator. The *Infinite Time Horizon Optimal Control Problem* consists of finding an optimal pair $(u^*(\cdot), z^*(\cdot))$ that minimizes the cost functional (1.23) subject to the dynamical system (1.20) - (1.22).

Definition 1.5.2. (1) The pair $(\mathcal{A}, \mathcal{B})$ is called stabilizable if there exists an operator $\mathcal{K} : \mathcal{L}(Z, U)$ such that $(\mathcal{A} - \mathcal{B}\mathcal{K})$ generates a uniformly exponentially stable semigroup on Z.
(2) The pair $(\mathcal{A}, \mathcal{C})$ is called detectable if there exists an operator $\mathcal{G} : \mathcal{L}(Y, Z)$ such that $(\mathcal{A} - \mathcal{G}\mathcal{C})$ generates a uniformly exponentially stable semigroup on Z.

Theorem 1.5.3. *([35],p.486) If $(\mathcal{A}, \mathcal{B})$ is stabilizable then the solution to the infinite time horizon optimal control problem is given by the linear operator $\mathcal{K} : Z \mapsto U$, called the* gain operator, such that

$$u^*(t) = -\mathcal{K}z(t) = -R^{-1}\mathcal{B}^*\Pi z(t),$$

where Π satisfies the operator Riccati equation

$$\mathcal{A}^*\Pi + \Pi\mathcal{A} - \Pi\mathcal{B}R^{-1}\mathcal{B}^*\Pi + \mathcal{C}^*\mathcal{C} = 0. \tag{1.24}$$

Here, $\mathcal{A}^*$ denotes the *adjoint operator* of $\mathcal{A}$ such that $(\mathcal{A}z, z) = (z, \mathcal{A}^*z)$ for all $z \in \mathcal{D}(\mathcal{A})$. By Riesz' representation theorem[4] there exists an integral kernel $k(x) \in Z$, called *functional gain*, such that the optimal control can be written as

$$u^*(t) = -\int_\Omega k(x)z(t)dx, \quad \forall t \in (0, \infty). \tag{1.25}$$

The functional gains are important for the placement of sensors and actuators [48, 47]. For more information on infinite dimensional control, the reader may consult [67]. The concepts which are introduced in the following section for finite dimensional systems can all be formulated in the infinite dimensional case as well, yet with great technical detail. In this work, we focus on the finite dimensional case, in part because certain problems, e.g. Chapter 5, are purely data-based. In the appendix, we give more detail about the infinite dimensional theory, in particular with respect to convergence of discretization. As an example, we formulate the coupled Burgers' equation in detail as an infinite dimensional control problem.

1.5.2 Finite Dimensional Linear Systems Theory

Consider a linear, time invariant system of the form

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{1.26}$$

$$y(t) = Cx(t) + Du(t), \tag{1.27}$$

where $x \in \mathbb{R}^n$ is the state variable, and $u(t)$ in $L^2[0, \infty)$ is the control function. Here, $A \in \mathbb{R}^{n\times n}$ is the system matrix, $B \in \mathbb{R}^{n\times m}$ is the control input matrix, $C \in \mathbb{R}^{p\times n}$ is the observation and $D \in \mathbb{R}^{p\times m}$ is the control-to-output mapping ("feed-through"). The above system can arise from discretization of a PDE, or can directly be obtained through modeling. The Variation of Parameters formula (1.15), provides a closed form of the input to output mapping in continuous time as

$$y(t) = Ce^{At}x_0 + \int_0^t Ce^{A(t-s)}Bu(s)ds, \quad \forall t \in (0, \infty).$$

[4] Since Z is a Hilbert space, so $Z = Z^*$, the mapping $\mathcal{K} : Z \mapsto U = \mathbb{R}$ can be represented as an inner product $(k(x), z)_Z$.

On the other hand, the Laplace transform of the state is given by $\mathcal{L}(x) := \hat{x}(s) = \int_0^\infty e^{-st}x(t)dt$ and it is easily seen that $\mathcal{L}(\dot{x}) = \int_0^\infty e^{-st}\frac{d}{dt}x(t)dt = s\mathcal{L}(x) - x(0)$. The Laplace transformed system reads as

$$\begin{aligned} s\hat{x}(s) - x(0) &= A\hat{x}(s) + B\hat{u}(s), \\ \hat{y}(s) &= C\hat{x}(s), \end{aligned}$$

which can then be rearranged into

$$\hat{y}(s) = C(sI - A)^{-1}B\hat{u}(s) + C(sI - A)^{-1}x(0).$$

Typically, one is interested in the system response to inputs, and hence we neglect the initial condition ($x(0) = 0$), so that the mapping from inputs to outputs in the frequency (Laplace) domain is given by the *transfer function* $G(s) := C(sI - A)^{-1}B$ as

$$\hat{y}(s) = G(s)\hat{u}(s) \tag{1.28}$$

The transfer function is the frequency domain analogue to the closed loop form of the output equation from above. Assume that the two different inputs $\hat{u}_1$ and $\hat{u}_2$ are given, so that the corresponding outputs can be bounded as

$$||\hat{y}_1(s) - \hat{y}_2(s)||_a \leq ||G(s)||_b \; ||\hat{u}_1 - \hat{u}_2||_a,$$

where the pair $a = 2, b = \mathcal{H}_\infty$ give a bound on the least squares error in the output and the norms $a = \infty, b = \mathcal{H}_2$ bound the maximum error in the output. Here,

$$||G||^2_{\mathcal{H}_2} = \frac{1}{2\pi}\int_{-\infty}^{\infty} trace(G^*(i\omega)G(i\omega))d\omega,$$

and

$$||G||_{\mathcal{H}_\infty} = \sup_{\omega\in\mathbb{R}} ||G(i\omega)||_2$$

are the corresponding system norms in the *Hardy spaces* [8, p.132+144]. Many model reduction methods are designed to minimize the above norms [91].

Another important concept in linear systems theory is the notion of observability and controllability. In short, these concepts help to determine states of a system that can be controlled, and observed, respectively. From a practical standpoint, controllability needs to be ensured by the design and location of actuators, and similarly for sensors in the concept of observability. From a model reduction point of view (see next section), it is often sufficient to retain states that are both observable and controllable.

Definition 1.5.4. [8, p.73] Given the system (A, B), a (nonzero) state $\bar{x}$ is *controllable* to the zero state, if there exists an input function $u(t)$ and a finite time T, such that $x(u, \bar{x}, T) = 0$. The controllable subspace of $\mathbb{R}^n$ is the set of all controllable states. The system (A, B) is *controllable*, if the controllable subspace is all of $\mathbb{R}^n$.

Theorem 1.5.5. *[126, p.55] The n-dimensional linear time invariant system $\dot{x}(t) = Ax(t) + Bu(t)$, with $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times p}$, is completely controllably of and only if the* controllability matrix

$$\mathcal{C} := [B,\ AB,\ A^2B, \ldots, A^{n-1}B] \tag{1.29}$$

has full column rank.

In the numerical linear algebra community the above space is called a Krylov subspace. Below, we give a simple example to illustrate the concept of controllability.

Example 1.5.6. Consider the two dimensional LTI system given by

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

The solution $x(t) = [x_1(t), x_2(t)]^T$ can be easily computed and $x_2(t) = e^{-t}$. By the definition above, the system is not controllable, since the is no *finite* time T such that $x_2(T) = 0$. Alternatively, one could also compute the controllability matrix (note, $n = 2$), which reads as

$$\mathcal{C} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix},$$

which does not have full column rank.

Definition 1.5.7. [126, p.65f] Let $y(t; t_0, x_0, u)$ be the output response of the LTI system (A, B, C) to the input $u(t)$. The system is called *completely observable* if for all t_1, there exists a t_0, such that if

$$y(t; t_0, x_0, u) = y(t; t_0, x_0', u) \quad \forall u(t) \quad \text{and} \quad t_0 \le t \le t_1,$$

then $x_0 = x_0'$. In other words, the initial state can be determined after a finite observation duration of the output.

Theorem 1.5.8. *[126, p.67] The n-dimensional LTI system (1.26) - (1.27) is completely observable if and only if the* observability matrix

$$\mathcal{O} := [C^T,\ C^TA,\ \ldots, C^TA^{n-1}]^T \tag{1.30}$$

has full rank n.

The reader should observe, that controllability and observability are dual concepts, see [126, §1.8]. In other words, the pair (A, B) is completely controllable, if and only if the pair (A^T, B^T) is completely observable.

Definition 1.5.9. [126, p.80] The *dual system* to (1.26) - (1.27) is the linear system

$$\begin{aligned}\dot{z}(t) &= A^T z(t) + C^T u(t),\\ y(t) &= B^T z(t).\end{aligned}$$

In the dual system, the role of the inputs and outputs is switched and the concept becomes important when considering the duality between controllers and observers of a system. In Chapter 3, we solve the algebraic Riccati equation based on the dual equations.

Controllability and observability are intricately related to the system gramians, as introduced below. Through those gramian matrices, the energy to steer a state to the zero state, as well as the energy needed to observe those states can be explicitly computed.

Definition and Theorem 1.5.10. [8, p.78f] For a stable LTI system (1.26) - (1.27), the *controllability gramian* Q and the *observability gramian* X are given by

$$Q := \int_0^\infty e^{At} B B^T e^{A^T t} dt, \qquad X := \int_0^\infty e^{A^T t} C^T C e^{At} dt. \tag{1.31}$$

The gramians satisfy the continuous time Lyapunov equations

$$AQ + QA^T + BB^T = 0, \qquad A^T X + XA + C^T C = 0. \tag{1.32}$$

The minimal energy to required to steer $x(t)$ from zero to x_f is given by $x_f^T Q x_f$ and the minimal energy produced by observing the output of the system whose initial state is x_0 is given by $x_0^T X x_0$.

The gramians provide the key ingredients for the *balancing transformation*, a change of basis that makes the states of the new system equally well controllable and observable. This leads to a model reduction technique, called balanced truncation [142], which guarantees to retain stability in the reduced order model. Balanced truncation uses the decay of the *Hankel singular values* defined as

$$\sigma_k^H = \sqrt{\lambda_k(XQ)}$$

as a truncation criterion. The Hankel singular values are system invariants and therefore important quantities for the study of linear systems.

1.5.3 Optimal Control

The goal of optimal control design is to minimize a given cost while obeying the underlying dynamics exactly or approximately. A key advantage of this theory is that the feedback is of linear type. However, one should keep in mind that the control law possesses zero robustness, as shown by J. C. Doyle in [75]. For the purpose of finite dimensional optimal control theory, we again assume a linear, time invariant system of the form (1.26) - (1.27).

Definition 1.5.11. The infinite time horizon optimal control problem on $\mathbb{R}^n$ is given as follows: Find the optimal control $u^*(\cdot) : [0, \infty) \mapsto U = \mathbb{R}^m$ and the corresponding optimal trajectory $x^*(\cdot) : [0, \infty) \mapsto \mathbb{R}^n$ that minimize the cost

$$J(u(\cdot), x(\cdot)) = \int_0^\infty \left\{ ||y||^2 + R||u(t)||^2 \right\} dt \tag{1.33}$$

subject to the dynamics in (1.26) - (1.27).

The above problem can be solved by standard constraint optimization methods. However, it turns out that there is an explicit solution to the above problem, which assumes the solution of a nonlinear matrix equation, as we shall see below. First, we give a definition and theorem about stabilizability.

Definition and Theorem 1.5.12. ([126, §1.3.4]) The single input control system (1.26),(1.27), (1.33) is called *stabilizable* if its uncontrollable poles are stable. In other words, there exists a $m \times n$ matrix K such that $[A - BK]$ is a stable matrix. Thus, all roots λ_i of the characteristic polynomial $det(A - BK - \lambda I) = 0$ satisfy $\Re(\lambda_i) < 0$. Moreover, there are constants $M \geq 1, \gamma > 0$ such that

$$|x(t)| \leq M e^{-\gamma t} |x_0| \quad \forall t > 0.$$

Conversely, if there is no such matrix K, then the system is not stabilizable and not controllable.

The following theorem relates the solution of the optimal control problem to solving the algebraic Riccati equation and explicitly states the cost of the control action.

Theorem 1.5.13. *([126, p.237f]) If (A, B) is stabilizable, then the infinite time horizon optimal control problem from Definition 1.5.11, has a unique solution $(u_*(\cdot), x_*(\cdot))$. The linear feedback law reads as*

$$u_*(t) = -K x_*(t). \tag{1.34}$$

The gain matrix *is constant and given by*

$$K = R^{-1}B^T P_*, \tag{1.35}$$

where P_* is the unique non-negative definite solution to the time invariant Riccati *equation*

$$A^T P + PA - PBR^{-1}B^T P + C^T C = 0. \tag{1.36}$$

With this feedback control, the closed loop system in $\mathbb{R}^n$ reads as

$$\dot{x}_*(t) = [A - BK]x_*(t)$$

and is asymptotically stable. Moreover, the minimal cost can be computed as

$$J(u_*(\cdot), x_*(\cdot)) = \int_0^\infty \left\{ ||y_*||(t) + R||u_*(t)||^2 \right\} dt =< P_* x_0, x_0 > . \tag{1.37}$$

Thus, stabilizability guarantees the existence of a linear quadratic regulator. Systems are made stabilizable through design: Actuators should be able to influence the unstable modes, otherwise there is no hope for the system to eventually become stable. A similar statement can be made about the dual problem of observability.

Remark 1.5.14. The gain matrix K can give crucial insight into the placement of sensors. The columns of K have the infinite dimensional analogue of functional gains, which are square integrable functions in space. To illustrate the point, let $[x_1 \ x_2]^T$ be the state of a system and $K = [1 \ 0]$, which is equivalent to compact support of the functional gains in infinite dimensions. For control purposes, there is no need to sense/observe the component x_2, since by $u(t) = -Kx(t)$, this information would be useless. Generally, in spatial locations where K is "large" in some sense, sensors should give accurate information about the state, whereas where K is "small", less sensing effort can be spent.

In Chapter 3, we develop a new algorithm to solve the algebraic Riccati equation in high dimensions ($n \approx 100,000$), which leads to an accurately approximated controller. This approach is called "design-then-reduce", since the controller is first computed from the high fidelity model and then implemented through a low order representation. Where this is impractical, or where reduced order models are already at hand, an alternative is to proceed by "reduce-then-design", i.e. computing the control from reduced order models. The latter idea is illustrated in Chapter 2, where we use proper orthogonal decomposition for generation of a low dimensional model, and subsequently compute the feedback law. That chapter highlights the many computational choices involved in such an approach.

One should note, that linear controllers are frequently used for nonlinear systems, and the success of such an approach depends on the "order" of the nonlinearity. In particular, Theorem 1.4.3 (Perron) guarantees, that controllers computed from linear systems still stabilize weakly nonlinear dynamics.

1.6 Model Reduction

Model reduction is concerned with reducing the dimension and complexity of a (data or model based) system, so that the reduced order surrogate model is cheap to execute, while satisfying specified criteria, such as stability, accuracy, structure, etc. Therefore, model reduction is inherently problem dependent and goal oriented and one is well advised to consider the following questions for the given application:

- Which are the necessary features the reduced order model should have?
- Do we need to preserve nonlinearities, implement them efficiently, or is linearization appropriate?
- Is it necessary to preserve the structure of high fidelity model?
- What is the hardware environment on which it should be executed?

In summary, there is no "one-size-fits-all" technique in model reduction, since every problem and environment poses unique challenges. This is reflected in the variety of available model reduction techniques, where each method, in general, has known strength and weaknesses.

Proper Orthogonal Decomposition is often the method of choice for nonlinear systems, when the energy retained in the system is most important for the application. Dynamic mode decomposition [156, 162] in contrast is concerned with the dynamically most relevant modes, and therefore typically selects modes based on a frequency criterion. Moreover, methods such as balanced truncation [142], balanced POD [189, 155] or the eigensystem realization algorithm [124], construct reduced order models that retain the most controllable and observable modes of the system.

Other ROM methods accurately approximate the input-to-output behavior of the underlying system, rather than focusing on optimally representing the states. The iterative rational Krylov subspace algorithm [91] for linear and bilinear systems achieves this by optimally interpolating the transfer function of the system. In addition, there are many other model reduction techniques, such as LQG balanced truncation [26], reduced basis methods (good for parameter dependent systems) [121], as well as structure preserving model reduction techniques (for Port-Hamiltonian systems) [92].

1.6.1 Projection based model reduction

The model reduction techniques considered in this work are based on projections of the high dimensional dynamics onto a low number of modes that well capture certain features of the flow. Consider a general nonlinear dynamical system in $\mathbb{R}^n$ of the form

$$E\dot{x}(t) = F(x(t)) + Bu(t), \tag{1.38}$$

$$y(t) = Cx(t). \tag{1.39}$$

The goal of model reduction is to approximate the high dimensional state x in an appropriate bases $\{\phi_i\}_{i=1,\ldots,r}$ with $r \ll n$, so that

$$x(t) \approx x_r(t) = \sum_{i=1}^{r} a_i(t)\phi_i.$$

This is in essence achieved through a change of basis, by diverting from the nonphysical, high-dimensional basis to a data-informed basis. In the new, sparse basis, only a few functions are needed to capture the desired properties of the system. Rewriting the above approximation in convenient matrix-vector expressions yields

$$x(t) = \Phi a(t), \qquad \Phi = [\phi_1, \phi_2, \ldots, \phi_r] \in \mathbb{R}^{n\times r}.$$

Inserting the approximation into equations (1.38) -(1.39) and projecting onto the low dimensional subspace through Φ^T yields the r-dimensional system

$$E_r\dot{a}(t) = \Phi^T F(\Phi a(t)) + B_r u(t),$$

$$y(t) = C_r a(t),$$

with $E_r = \Phi^T E \Phi$, $A_r = \Phi^T A \Phi \in \mathbb{R}^{r \times r}$, the reduced control input matrix $B_r = \Phi^T B \in \mathbb{R}^{r \times p}$ and the sensing matrix $C_r = C\Phi \in \mathbb{R}^{m \times r}$. An apparent challenge in reduced order modeling is the treatment of the nonlinear term, since $F(\cdot)$ still needs to be evaluated at the full dimension n. The computational effort for the nonlinearity can be reduced by the empirical interpolation method (EIM) [22] and the discrete empirical interpolation method (DEIM) in the context of POD [61], which both interpolate the nonlinear function in a clever way. When dealing with projection methods, one needs to ensure that stability is preserved under projection. In particular, if A is a stable matrix, then A_r should also be stable.

Lemma 1.6.1. *(Stability Preservation under Projection)[59] Let (E, A, B, C) be a continuous time system with $\mathfrak{Re}(\lambda(A + A^T)) \leq 0$, where $\lambda(A)$ denotes any eigenvalue of A. Let $E = E^T > 0$. Then the projected reduced order model (E_r, A_r, B_r, C_r) is stable if $V \in \mathbb{R}^{n \times r}$ has full column rank.*

Note that as a special case, if $E = I$ and the matrix A is normal (symmetric, skew-symmetric or orthogonal matrices form a subset of normal matrices), then reduced order models obtained via projection as in Lemma 1.6.1 are stable. Care must be taken when Petrov-Galerkin projections are computed (different left and right projection), since stability can be lost in the process.

1.6.2 Proper Orthogonal Decomposition

Proper Orthogonal Decomposition (POD) is a data reduction technique, which extracts dominant structures from experimental or simulation data. In fact, the method determines the optimal basis for representing a given dataset with respect to the mean squared error. The method has been independently rediscovered many times, see Pearson (1901), Hotelling (1933), Loeve (1945) and Karhuenen (1946) and therefore is known under multiple names, such as Karhuenen-Loeve expansion, Principal Component Analysis and Hotelling transformation. Based on the extracted coherent structures, POD has been employed to produce reduced order models of Galerkin type, which are frequently used for simulation, design, and control. In this work, we are interested in fluid dynamical applications, where the success of POD for has been overwhelming, see [174, 13, 98, 135, 151, 186] and the references therein. Per definition, the POD basis depends on the data set from which it was computed. Extensions of POD to deal with parameter dependent systems can be found in [96, 108], where sensitivity information with respect to parameters is incorporated into the basis functions. While POD does not guarantee preservation of stability [12, 155, 7], employing stable model reduction techniques, such as balanced truncation, is computationally expensive and limited to linear systems. Extensions of balanced truncation to nonlinear systems were presented in [127, 189, 155]. In Chapter 2, we employ POD to generate reduced order surrogate models, which can be used for design and control. As a data based method, POD is not limited to a certain underlying model structure. As Volkwein [185, p.3] notes "the POD method is a universal tool that is applicable also to problems with time dependent coefficients and nonlinear systems.... This, and its ease of use makes POD very competitive in practical use, despite of a certain heuristic flavor". In this section, we introduce POD for general finite dimensional dynamical systems, and refer the reader interested in the infinite dimensional case to the excellent surveys [99, 185, 186].

The application of POD in this work is to use it as a model reduction technique, where PDE or ODE models are at hand. Therefore, we assume a dynamical system of the form

$$\dot{x} = F(x), \qquad x \in \mathbb{R}^n,$$

and solutions $x(t_k) = x_k$ for $k = 1, \ldots, s$. Proper orthogonal decomposition yields a basis that optimally represents the given solution data in the least squares sense. Thus, the data from the (approximate) solution is essential in finding a low order subspace. Proper orthogonal decomposition solves the optimization problem

$$\min_{\phi_i} \sum_{j=1}^{s} \left\| x_j - \sum_{i=1}^{r} (x_j, \phi_i)\phi_i \right\|_2^2 \quad \text{s.t.} \quad (\phi_i, \phi_j) = \delta_{ij}. \tag{1.40}$$

In the following, we derive the solution to the POD optimization problem and give a brief overview of error bounds. By orthogonality of the basis, one has for all i, j that

$$\begin{aligned} 0 &\leq \left\| x_j - \sum_{i=1}^{r} (x_j, \phi_i)\phi_i \right\|_2^2 \\ &= \left[x_j^T - \sum_{i=1}^{r} (x_j, \phi_i)\phi_i^T \right] \left[x_j - \sum_{i=1}^{r} (x_j, \phi_i)\phi_i \right] \\ &= x_j^T x_j - 2\sum_{i=1}^{r} (x_j, \phi_i)^2 + \sum_{i=1}^{r} (x_j, \phi_i)^2 \\ &= ||x_j||_2^2 - \sum_{i=1}^{r} (x_j, \phi_i)^2. \end{aligned}$$

Therefore, problem (1.40) is equivalent to

$$\max_{\phi_i} \sum_{j=1}^{s} \sum_{i=1}^{r} (x_j, \phi_i)^2 \quad \text{s.t.} \quad (\phi_i, \phi_j) = \delta_{ij}. \tag{1.41}$$

The above optimization problem is solved with the method of Lagrange multipliers, since the cost and constraints are continuously differentiable, and the cost function is convex. Let λ be the auxiliary variable and set the Lagrangian as

$$L(\phi, \lambda) = \sum_{j=1}^{s} (x_j, \phi)^2 + \lambda[1 - (\phi, \phi)],$$

which has the derivative

$$\frac{dL}{d\phi}(\phi, \lambda)|_{\phi=\hat{\phi}} = \sum_{j=1}^{s} 2(x_j, \hat{\phi})x_j - 2\lambda\hat{\phi}.$$

The derivative vanishes, whenever

$$\sum_{j=1}^{s} (x_j, \phi)x_j = \lambda\phi,$$

which we shall rewrite into an eigenvalue problem. The snapshots are conveniently stored in a matrix

$$X = [x_1 \; x_2 \ldots \; x_s] \; \in \mathbb{R}^{n\times s}, \tag{1.42}$$

which allows us to rewrite

$$\left(\sum_{j=1}^{s} (x_j, \phi)x_j\right)_k = \sum_{j=1}^{s} \sum_{l=1}^{n} X_{lj}\phi_l X_{kj} = \sum_{j=1}^{s} X_{kj} \sum_{l=1}^{n} X_{lj}\phi_l = (XX^T\phi)_k,$$

and hence the eigenvalue problem from above becomes

$$XX^T\phi_i = \lambda_i\phi_i. \tag{1.43}$$

The functions ϕ_i are called *POD modes* (or *POD basis functions*) and the λ_i are the *POD eigenvalues*. For details concerning the optimality of the basis and the derivation of the correlation matrix, see [98, Ch.3].

Method of Snapshots. In certain applications, such as in fluid dynamics, the state space dimension can be in the order of millions and only a few snapshots are available, so $s \ll n$. A computation of POD modes becomes prohibitively expensive, since it requires an eigenvalue decomposition of an $n \times n$ matrix. To remedy this computational burden, Sirovich [174] introduced the method of snapshots. The method essentially requires a singular value decomposition of a square matrix of size $s \times s$. To introduce the idea, let

$$X = \Phi\Sigma\Psi^T$$

be the singular value decomposition of the data, so that Φ, Ψ are orthogonal and its respective columns satisfy

$$X\psi_i = \sigma_i\phi_i, \quad X^T\phi_i = \sigma_i\psi_i, \quad i = 1, \ldots, rank(X).$$

It follows that

$$\begin{aligned} XX^T &= \Phi\Lambda\Phi^T \quad &\in \mathbb{R}^{n\times n}, \\ X^TX &= \Psi\Lambda\Psi^T \quad &\in \mathbb{R}^{s\times s}, \end{aligned} \tag{1.44}$$

where $\Lambda = \Sigma^2$ is the diagonal matrix containing the POD eigenvalues. Consequently, Ψ contains the eigenvectors of X^TX and the columns of Φ are the eigenvectors of XX^T. The method of snapshots first computes (1.44) either via eigenvalue decomposition or the more stable SVD. Since $X\Psi = \Sigma\Phi = \Lambda^{1/2}\Phi$ we have

$$\Phi = \Lambda^{-1/2}X\Psi,$$

and therefore $\Phi = [\phi_1,\ \phi_2,\ \ldots, \phi_s]$ contains the desired POD modes. The proper orthogonal modes can then be used to study flow features, or, which is the focus of this present work, to obtain reduced order models. In that case, POD is used in a Galerkin projection framework.

In some cases, such as when the scaling of entries in X varies by several magnitudes, the computation of the product X^TX or XX^T can introduce numerical errors. In this case, a direct singular value decomposition on X should be used. Alternatively, mean subtraction can remedy the scaling problem.

Mean Subtraction. It is customary to subtract the data mean from the snapshots before extracting the POD basis. This eliminates numerical errors when computing the coefficient of the leading POD mode, which often happens to be close to the mean, see [181, 107]. Let $\bar{x} = \frac{1}{s}\sum_{i=1}^{s} x_i$ be the data mean. Subtract $\bar{x}$ from the columns of X in (1.42) and proceed with the usual steps.

Selection of Modes. In most cases, not all POD modes are needed and one is only interested in some modes. Selection of the number r of kept POD modes is often based on an energy criterion of the form

$$\mathcal{E}(r) := \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{s} \lambda_i}.$$

The tolerance $\mathcal{E}_{min}$ depends on the complexity of the problem and user requirements. For instance, for simple models $\mathcal{E}_{min} = 99.99\%$ is common, yet for a complex 3D Boussinesq problem, $\mathcal{E}_{min} = 90\%$ already requires a large number of basis functions.

Optimality. Here, we give a brief heuristic of optimality in the finite dimensional setting and state the key theorem. One can reformulate (1.40) as a matrix approximation problem of the form

$$\min_{\text{rank}(P_r)=r} \|X - P_r X\|_2^2 \quad \text{s.t.} \quad P_r^2 = P_r \in \mathbb{R}^{n \times n},$$

where $P_r = \Phi_r \Phi_r^T$ and $\Phi_r \in \mathbb{R}^{n \times r}$. In other words, we are looking for the best rank r approximation to the data in the L_2 sense. The optimality result is hence a direct consequence of the Schmidt-Eckardt-Young-Mirsky Theorem 1.2.5, which reads as

$$\sum_{j=1}^{s} \left\| x_j - \sum_{i=1}^{r} (x_j, \phi_i) \phi_i \right\|_2^2 = \lambda_{r+1}$$

and in the Frobenius norm

$$\sum_{j=1}^{s} \left\| x_j - \sum_{i=1}^{r} (x_j, \phi_i) \phi_i \right\|_F^2 = \sum_{k=r+1}^{n} \lambda_k.$$

Recent advances in POD optimality results are found in [172].

Remark 1.6.2. POD can be applied to data in multiple space dimensions, such as a fluid evolving in three dimensional space, as follows. Let $\mathbf{u} = [u_x \; u_y \; u_z]$ consist of the components of the fluids velocity field in x, y, z directions. Note, that the quantities u_x, u_y, u_z itself are vectors of length n_x, n_y, n_z, respectively. These vectors contain the values of the velocity components at every point of the computational domain stacked into a vector. POD can then be applied to the vectorized data in the usual way.[5] However, when the data consists of solutions for multiple time variables, such as temperature and velocity in Boussinesq flows, special care must be taken when computing and constructing reduced order models, which is highlighted in Chapter 2.

Further Directions in POD. To conclude this overview, we would like to highlight three important directions of research about the POD method. First, POD extends to infinite dimensional systems, which lends itself naturally for application of PDE's, see [186, 185]. Second, the issue of snapshot selection to generate the dataset is an ongoing research topic. Iliescu and Wang [103] recently investigated POD error bounds when including snapshots of the difference quotients in the data set. They found that for optimal point-wise convergence in time, the snapshot difference quotients should be included in the generation of the POD basis, when utilizing a Galerkin approximation scheme to obtain a reduced order model. Thirdly, observe that POD is an input dependent model reduction technique, and therefore special care must be taken when selecting the input functions. One common approach is to excite the system through various choices of "rich" initial conditions, such as sine and cosine waves, and discontinuous step functions, which mimic a physical initial condition (e.g. in mixing flow fields). A second approach is to use external forcing, fed through either the boundary conditions or a control input. We demonstrate the success of those methods in Chapter 2.

1.6.3 Dynamic Mode Decomposition

Understanding the dominant features in dynamically evolving systems, such as the mechanisms triggering bifurcations and instability in flows, is important for both the construction of reduced order models and flow control. Dynamic Mode Decomposition (DMD) is yet another method to extract important features (modes) from flow data. DMD provides a set of complex modes and eigenvalues, which explains the data through spatial modes oscillating at a single frequency. Therefore, the modes are fundamentally different from POD modes, which are ranked by energy content. It has been demonstrated since its beginning in 2009 [156, 162], that the method can shed a different light onto structures in fluid dynamics, and it has found great success in this community. In [65], Rowley and co-authors addressed the issue of mode selection and provided an "optimized DMD" algorithm (mode selection is not

[5] For further reading, see the introduction with Matlab examples in [135].

as straightforward as in POD). Optimized DMD was found to outperform standard DMD in calculating the physically relevant frequencies. Mezić discusses the Koopman operator and its applications in the analysis of fluid flows in [140], and compares DMD to an alternative method to compute the spectrum of the Koopman operator. For actuated linear time invariant systems, it is still possible to extract the dynamic modes from the data, and additionally even the control actuator matrix B can be identified. This, and the relation to system identification can be found in [149]. Whenever low dimensional features are extracted from data, it is natural to build reduced order models, as done in [177] with a Galerkin projection. Moreover, the authors in [41] use compressive sampling to compute the dynamic mode decomposition for subsampled data.

The focus of this thesis is on data and model reduction techniques, and therefore we present DMD in the context of finite dimensional data. While the study of the Koopman operator and its application to PDE systems in infinite dimensions provides deep insight into the dynamics of solutions, we omit this topic herein and refer the interested reader to Mezić [140]. We start with a brief outline of DMD computation, following the work of Schmid [162]. In particular, the computationally more desirable implementation with a singular value decomposition and rank truncation is implemented. Note, that we do not employ the exact DMD, as proposed in [179].

Consider a nonlinear finite dimensional dynamical system

$$\dot{x}(t) = F(x), \qquad x(0) = x_0.$$

Assume that a collection of snapshots is available and stored in the matrices

$$\Psi_0 = \begin{bmatrix} | & | & & | \\ x_0 & x_1 & \cdots & x_{s-1} \\ | & | & & | \end{bmatrix}, \qquad \Psi_1 = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_s \\ | & | & & | \end{bmatrix}, \tag{1.45}$$

where $x(t_i) := x_i$ for $i = 1, \ldots, s$. The main hypothesis behind dynamic mode decomposition is that that there is a linear operator (thought of as Koopman operator in ergodic theory), that advances the snapshots by one time step. Since the data is finite dimensional, this operator is represented by a matrix $\boldsymbol{A}$, such that

$$\Psi_1 = \boldsymbol{A}\Psi_0. \tag{1.46}$$

The goal is to approximate the eigenvalues of $\boldsymbol{A}$ from data only, without relying on the system model. Therefore, DMD provides an entirely data based approach to extract dynamic information, i.e. spatial modes *and* their dynamics.

A second assumption for DMD is that beyond a critical number s, the snapshots become linear dependent. In other words, the rank of the data matrix Ψ_0 cannot increase beyond s. With this in mind, one cannot expect to resolve all n modes of the n-dimensional flow field, but aims to compute $r \approx s$ at best and then selects the ones which are dynamically most relevant. In a first step, compute the singular value decomposition of the data $\Psi_0 = U\Sigma V^T$. To eliminate rank deficiency or for a more stable numerical implementation (note, that this is not a significant truncation), one approximates

$$\Psi_0 \approx U_r \Sigma_r V_r^T,$$

where U_r, V_r^T are the first r columns of U, V, respectively and Σ_r is the leading $r \times r$ submatrix of the diagonal matrix Σ, containing the largest singular values in decreasing order. The reader should observe that the columns of U are nothing but the POD modes of the system. As is well known [8, p.37], $||\Psi_0 - U_r \Sigma_r V_r^T||_2 = \sigma_{r+1}$, so if the singular values decay rapidly, the truncation error is small. Inserting the approximation into (1.46) above yields

$$\Psi_1 \approx \boldsymbol{A} U_r \Sigma_r V_r^T,$$

where the right singular vectors are orthogonal, $V_r^T V_r = I_r$. Rewriting the above expression yields

$$U_r^T \Psi_1 V_r \Sigma_r^{-1} \approx U_r^T \boldsymbol{A} U_r$$

and we define $S_r := U_r^T \Psi_1 V_r \Sigma_r^{-1}$. The goal is to obtain the eigenmodes of $\boldsymbol{A}$ without ever computing $\boldsymbol{A}$. Therefore, one proceeds by taking the eigenvalue decomposition

$$S_r = Y \Lambda Y^{-1}.$$

If one has not performed a significant truncation up to this point, then $r \approx s$ and one can assume that $U_r U_r^T \approx I_n$. Then,

$$Y \Lambda Y^{-1} \approx U_r^T \boldsymbol{A} U_r \quad \Rightarrow \quad U_r Y \Lambda \approx \boldsymbol{A} U_r Y.$$

The dynamic modes are defined as $\Phi := U_r Y$, for they are approximate eigenmodes of the advance operator

$$\boldsymbol{A}\Phi \approx \Phi\Lambda. \tag{1.47}$$

In Schmid [162, Sec 2.5], convergence of the eigenvalues (i.e. growth and frequencies) of the dynamic modes is discussed and numerically verified by considering a linearized Navier-Stokes equation. Moreover, numerical examples for a linearized flow problem and an experimental data set demonstrate that DMD is capable of extracting relevant flow features, even from sub-domain data. As an interesting comment, Schmid notes that "if the sampling frequency is tuned to that of an oscillatory flow, the mapping from period to period will identify the Floquet matrix whose eigenvalues represent the Floquet multipliers."

Overall, the computation of the DMD involves a singular value decomposition of (economy) size $n \times s$, where $s \ll n$, and an eigenvalue decomposition of size $s \times s$ and is therefore computationally tractable for systems of large size. If the system dimension is yet too large, or only experimental data in a subregion is available, the modes for the full flow field can be reconstructed from drastically fewer measurements via compressed sensing, as shown in [41].

1.7 Compressed Sensing

Sensing information about complex systems from physical devices is relevant in many fields of science and engineering for predictive modeling and quantitative assessment of systems. In 1949, Shannon addressed the classical problem of reconstructing a signal from finite measurements and found that a signal has to be sampled at least at twice the maximum frequency present in the signal. In many cases, this requires sampling a large amount of data, which cannot be stored. For efficient transmission, the data is often compressed after acquisition, such as in the JPEG2000 image compression algorithm, or the MP3 format for video compression. Although it is obvious that this approach (sampling $2n$ frequencies, just to compress it to $k \ll n$ data points) is inherently inefficient, it has been a standard of an entire industry, for lack of a better method. Recent developments in the field of compressed sensing show promising results and are a hot topic in modern signal processing, see the book [85] and survey papers [73, 56]. We give a brief overview of this new theory and state relevant results for the work in this thesis. A discussion about applications and related work is deferred to Chapter 5.

Sensing and Sparsity. Compressed sensing aims to reconstruct a signal $x \in \mathbb{R}^n$ from $p \ll n$ measurements $y \in \mathbb{R}^p$ via

$$y = Cx,$$

where $C \in \mathbb{R}^{p \times n}$ denotes a sensing matrix. Clearly, this system is underdetermined and does have either infinitely many (in this case it is called consistent) solutions, or no solution at all. The situation is hopeless without further information about x. However, if one is interested in a specific solution of the problem, then, by imposing additional requirements on x (called regularization), a unique solution can be shown to exist. The assumption that the sensed signal has a certain structure is crucial to the success of compressed sensing.

Definition 1.7.1. A vector $x \in \mathbb{R}^n$ is called k-*sparse* if its support is of cardinality less or equal to k, i.e. $||x||_0 \leq k$. Moreover, a vector x is called k-*compressible* if it is well approximated by a k-sparse vector.

In this work, we are particularly interested in dynamical systems describing the motion of a fluid. It has been observed (theoretically by studying attractors and quantitatively through data analysis, see [99]) that the governing dynamics of fluids are often low dimensional. In other words, the signal can be represented in a non-trivial basis as

$$x = \Phi a,$$

where $\Phi \in \mathbb{R}^{n \times n}$ contains orthonormal basis vectors as columns and we assume that $a \in \mathbb{R}^n$ is k-sparse in this basis. In practice, the signal x is often only k-compressible, such as when Φ is the POD basis, as described in detail in §1.6.2. Another example is given by images, which can be sparsely represented by a wavelet basis. In light of the sparse or compressible representation of x, the reconstruction problem becomes

$$y = C\Phi a, \quad \text{where} \quad ||a||_0 \leq k.$$

The additional information that only k components of a are nonzero, or relevant (in the case of compressible signals) is key to the success of the method. For ease of derivation, let

$$\Theta := C\Phi \quad \Rightarrow \quad y = \Theta a, \tag{1.48}$$

where $\Theta \in \mathbb{R}^{p \times n}$. The situation is graphically depicted in Figure 1.1.

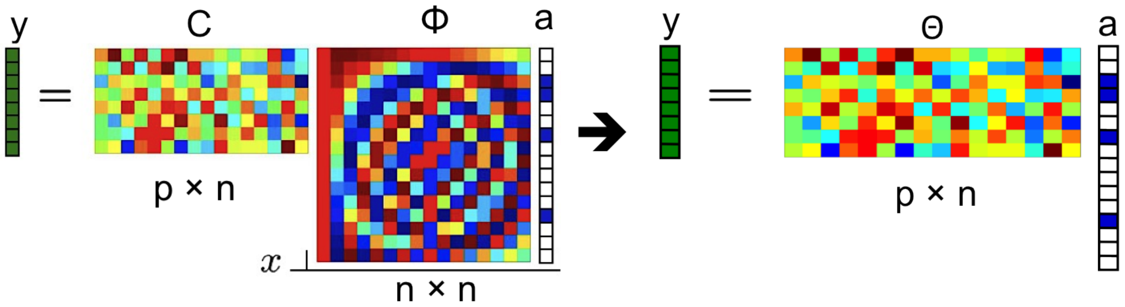

Figure 1.1: Illustration of sparse reconstruction. The idea behind compressed sensing is to choose the sensing matrix C and the sparsity basis Φ so that the matrix $\Theta = C\Phi$ has 'ideal' properties (as described below) for reconstruction of x, the full signal.

Traditionally, one of those solutions can be found through the pseudoinverse $a = (\Theta)^{\dagger}y$, which minimizes the l_2 norm of both vectors. However, this approach rarely finds a sparse solution. Therefore, certain methods use a posteriori thresholding techniques, that drop the $n-k$ smallest components of a to make it k-sparse. Unfortunately, this offers only an approximate solution, and the merit of compressed sensing is that one can do better. This raises several questions:

1. Under which conditions on C and Φ can we exactly (optimally) recover the k-sparse (k-compressible) solution of $y = C\Phi a$?

2. How large does p have to be?

3. How does noise present in the sensing influence the results?

4. Which algorithms can we use ?

To address the first question, one should note that $p \geq k$ is necessary. For a moment, let $\Phi_k \in \mathbb{R}^{n\times k}$ be the matrix of the k columns of Φ corresponding to nonzero entries in a. Pick $C = \Phi_k^T$ as the sensing matrix. Since Φ is orthonormal, we would then have $y = a$ and could sense the sparse signal coefficients directly with $p = k$ measurements! However, this approach is not practical. First, we seem to have a miraculous knowledge of the locations of the nonzero components in a, which is typically not the case. Second, from a practical perspective, $C = \Phi_k^T$ corresponds to sensing the variable of interest in the entire domain. While this is possible in imaging (see the single pixel camera http://dsp.rice.edu/cscamera), this is currently out of reach for fluids.

The developments in compressed sensing or compressive sampling provide answers to the design of C, while only assuming sparsity of the signal, but not any other assumption on the basis. In fact, we shall see that the sparsity significantly affects the acquisition process.

Reconstruction through Regularization. Ideally, one wishes to obtain the maximally sparse solution to the sensing problem, i.e.,

$$\min ||\hat{a}||_0 \qquad \text{subject to} \qquad y = \Theta \hat{a}.$$

The above problem is NP-hard and therefore intractable to solve. A major breakthrough in compresses sensing occured when Donoho [74, Thm. 2.4] showed that in many situations, the above solution is equivalent to the relaxed problem

$$\min ||\hat{a}||_1 \qquad \text{subject to} \qquad y = \Theta \hat{a}. \tag{1.49}$$

This problem can be solved via convex optimization and one can use Compressive Sampling Matching Pursuit (CoSaMP, [144]) algorithm to solve the ℓ_1-problem. The CoSaMP algorithm has been extended to include additional model assumptions [21], which is suitable for our work. Several other algorithms have been studied in [182].

We now turn to the question of how to choose C and which conditions need to be met in order for the optimization to yield a unique k-sparse solution, see [43, §2]. Let a_1, a_2 be two k-sparse solutions to (1.49), so that $\Theta(a_1 - a_2) = 0$ and $(a_1 - a_2)$ is at most $2k$-sparse. In order to guarantee uniqueness of a k-sparse solution, $2k$-sparse vectors cannot be in the nullspace of Θ, i.e., $null(\Theta) \cap \{a : ||a||_0 \leq 2k\} = \{0\}$. In other words, the matrix Θ needs to have a column rank of at least $2k$ to guarantee uniqueness. Finding a matrix C, such that this rank condition on $C\Phi$ is met is of combinatorial complexity and hence unfeasible for large n. Another key breakthrough in compressive sampling [58, 57, 73] brought about large classes of sensing matrices, for which the rank condition is met with high probability.

Definition 1.7.2. (Restricted Isometry Property) [43, Def.2.4]. A matrix $\Theta \in \mathbb{C}^{p \times n}$ satisfies the *restricted isometry property (RIP)* of order k, if there is some constant $0 < \delta_k < 1$ such that

$$1 - \delta_k \leq ||\Theta \hat{a}||_2^2 \leq 1 + \delta_k \tag{1.50}$$

for all k-sparse vectors $\hat{a} \in \mathbb{R}^n$ with $||\hat{a}||_2 = 1$.

Hence, the matrix Θ is close to an isometry, and therefore almost preserves distances. Put differently, the columns of Θ almost behave like an orthonormal system. We can now state an important result, which guarantees exact recovery if Θ satisfies the RIP.

Theorem 1.7.3. *([56, Thm. 3.1]). Assume that the vector $a \in \mathbb{R}^n$ is k-sparse and suppose that $\delta_{2k} + \delta_{3k} < 1$ for $\Theta = C\Phi$ in equation (1.50). Then the solution a^* to (1.49) is exact.*

In reality, not many signals are *exactly* k-sparse, but k-compressible. Nonetheless, we shall see that compressive sensing almost recovers the "best" solution. To illustrate the idea, let a be the sought solution of $y = \Theta a$ and a_k^* its best k-sparse approximation in the 2-norm, so that $a_k^* = \min_{||\hat{a}||_0 = k} ||a - \hat{a}||_2$.

Theorem 1.7.4. *[56, Thm. 3.2] Assume that a is k-sparse and suppose that $\delta_{3k} + \delta_{4k} < 2$ for $\Theta = C\Phi$ in equation (1.50). Then the solution a^* to (1.49) obeys*

$$||a - a^*||_2 \leq \gamma \cdot \frac{||a - a_k^*||_1}{\sqrt{k}},$$

where the constant is well behaved for reasonable values of δ_{4k}; e.g. $\gamma \leq 8.77$ for $\delta_{4k} = 0.2$. Under further assumptions on the RIP constants, we further have

$$||a - a^*||_1 \leq \gamma_2 ||a - a_k^*||_1.$$

The above theorem shows that even for k-compressible signals, the error of the solution obtained from the ℓ_1-reconstruction is not much worse than the optimal error, given that we knew a !

Next, we turn to the question of robustness of compressed sensing in the presence of noise. Therefore, assume that the output is inaccurate

$$y = \Theta a + \eta,$$

where η can be either stochastic or deterministic, and be bounded as $||\eta||_2 \leq \varepsilon$. Consequently, the optimization problem (1.49) has to be recast to account for the disturbance. Consider,

$$\min ||\hat{a}||_1 \quad \text{such that} \quad ||\Theta\hat{a} - y||_2 \leq \varepsilon. \tag{1.51}$$

Theorem 1.7.5. *[56, Thm. 4.1] Suppose that $a \in \mathbb{R}^n$ is arbitrary and $y = \Theta a + \eta \in \mathbb{R}^p$ is a corrupted measurement. Under the hypothesis of the previous theorem, the solution a^* to the noise aware optimization problem (1.51) obeys*

$$||a - a^*||_2 \leq c_1 \cdot \varepsilon + c_2 \cdot \frac{||a_0 - a_{0,k}||_1}{\sqrt{k}},$$

where the constants c_1, c_2 are again well behaved. Here, a_0 is the optimal solution to the noise free problem (1.49) and $a_{0,k}$ its best k-sparse approximation.

The above theorem is encouraging for practical applications, since the disturbance enters the error bound only linearly. Hence, for small noise levels, the ℓ_1 optimization problem almost recovers the true solution of the noise-free problem. Moreover, the fact that compressed sensing works even under the presence of noise and by removing the strict k-sparsity assumption is remarkable.

Choices for the Sensing Matrix. In light of the previous paragraph, the remaining problem is to find a sensing matrix C, such that $C\Phi$ satisfies the RIP. This problem is of combinatorial nature, with $\begin{pmatrix} n \\ k \end{pmatrix}$ choices, and hence hard to check. However, the theory of compressed sensing has unraveled large classes of matrices that satisfy the RIP (or a similar condition) with high probability, see [58, §1.5]. The following results are particularly interesting to us:

- *Gaussian measurements:* Assume that the entries of the matrix $C \in \mathbb{R}^{p \times n}$ are independent and identically distributed (i.i.d) with mean zero and variance $1/n$. Then C satisfies the RIP with probability $1 - \mathcal{O}(e^{-\epsilon n})$ for some $\epsilon < 1$, whenever
$$p \geq c \cdot k \log(n/k). \tag{1.52}$$
In particular, Gaussian matrices have the property that their product $C\Phi$ with a orthonormal matrix is again Gaussian. Therefore, they offer a perfect choice for compressed sensing.

- *Fourier measurements:* Let Φ be a Fourier matrix, i.e. $\Phi_{l,j} = \frac{1}{\sqrt{n}} \exp(-i\pi lj/n)$ and let C be the matrix that picks p rows of Φ uniformly at random. Then, it is conjectured that the same order of magnitude of measurements as in (1.52) is sufficient for reconstruction.

- *Incoherent measurements:* In cases where C does not have particular random structure, there is a computable measure to asses if the sparsity and sensing basis are suited for sparse recovery algorithms [55]. Assume that Φ is an orthonormal basis, that C is an orthogonal measurement system and define the *mutual coherence* as

  $$\mu(C, \Phi) := \max_{i,j} |(c_i, \phi_j)|.$$

  The mutual coherence quantizes the similarity between the measurement and sparsity basis and will take a value between 1 and $\sqrt{n}$. With this in mind, the ℓ_1-reconstruction (1.49) succeeds with overwhelming probability given that

  $$p \geq c \cdot \mu^2(C, \Phi) \cdot k \log(n).$$

  For low coherence pairs, only few measurements are necessary. If, however, $\mu = \sqrt{n}$, then the efficacy of compressed sensing is gone.

The results on compressed sensing are remarkable and initiated a whole body of research on compressed sensing and applications since its beginnings in 2005. In Chapter 5 we give more references to applications of compressed sensing and employ it to design an efficient classification algorithm for complex flows.

Chapter 2

Reduced Order Models for Feedback Control

A reduce-then-design approach via POD for feedback control of a coupled Burgers' equation is presented. The considered model is a hybrid partial differential equation, and therefore special care must be taken to retain the structure when deriving the finite element discretization and computing the reduced order model.

Reduced order controllers enjoy a great level of practicality, since they can be implemented on rather simple and cheap hardware. One way to compute the linear quadratic regulator (controller), is to solve the high dimensional nonlinear algebraic Riccati equation (1.36). This process is called "design-then-reduce", since the controller is designed at large-scale, and the system subsequently reduced. In Chapter 3, we develop an algorithm to solve AREs for large systems, and present numerical results of up to $n = 150,000$ variables. This contrasts the "reduce-then-design" approach, where ROM's (which are already in place for simulation and design) are used for control and optimization. In *certain* cases, this approach can fail to produce a convergent scheme, see [12, 155, 7]. In [63], the authors consider LTI systems, and give sufficient conditions for the reduce-then-design, and design-then-reduce approaches, depending on the reduction order r. Those results are not straightforwardly applicable to nonlinear systems, as considered herein. A discussion about the competing approaches can also be found in [11].

Often, the reduce-then-design approach is computationally cheaper, and cost and other feasibility reasons, motivate its use. Here, we numerically investigate this approach on a coupled Burgers' equation and have a close look at both performance and stability. This should convince the reader, that for this type of nonlinear problem, the reduce-then-design approach can give good results while saving computational effort.

Up to the authors knowledge, controlling a coupled Burgers' equation in the form given below has not yet been considered in the literature (for simulation results, see the authors previous work in [122]). Nonetheless, the employed computational methods are well known and the main goal is rather to emphasize the relevant steps and choices to arrive at a reduced order feedback controller for a hybrid system. In this sense, this chapter highlights important problems in control of distributed parameter systems and, by example, motivates much of the work for more complex systems in later parts of this thesis.

The PDE is discretized with a finite element (FE) method using piecewise linear basis functions and a reduced order model computed via proper orthogonal decomposition. Numerical results for the closed loop systems, both linear and nonlinear, are presented. Interestingly, the feedback gains efficiently control the system even when used at different parameters than where they were designed. Moreover, we shall see that the POD feedback acting on the FE system leads to similar results as full FE controller would produce. Moreover, the robustness of the controllers to changes in system parameters is investigated.

2.1 The Model - A hybrid 1D nonlinear PDE

The coupled Burgers' equation is a one dimensional model that incorporates many interesting questions related to thermal fluid dynamics, commonly modeled by the two or three dimensional Boussinesq equations. Herein, this model is used to illustrate computational aspects of feedback control and approximation of PDE's, and as a testbed for numerical studies. Here, we consider the coupled Burgers' equation

$$w_t(t,x) + w(t,x)w_x(t,x) = \mu w_{xx}(t,x) - \kappa T(t,x), \tag{2.1}$$

$$T_t(t,x) + w(t,x)T_x(t,x) = cT_{xx}(t,x) + b(x)u(t), \tag{2.2}$$

for $t > 0$ on the one dimensional domain $\Omega = (0,1)$ with boundary conditions

$$w(t,0) = 0 \quad w_x(t,1) = \epsilon, \tag{2.3}$$

$$T(t,0) = 0 \quad T(t,1) = 0, \tag{2.4}$$

for some $\epsilon > 0$ and initial conditions

$$w(0,x) = w_0(x) \quad \text{and} \quad T(0,x) = T_0(x). \tag{2.5}$$

Here, $w(\cdot,\cdot) \in H^2(0,\infty;\Omega)$ is a velocity-like function and $T(\cdot,\cdot) \in H^2(0,\infty;\Omega)$ is a temperature-like function. The parameter κ denotes the coefficient of the thermal expansion, c is the thermal diffusivity and $\mu = \frac{1}{Re}$ is the viscosity, the inverse of the Reynolds number. The function $b(x)$ denotes the location of the control action $u(t)$, and hence we have a distributed control action on the temperature. Of course, this adversely controls the velocity through the coupling. As a notational remark, $u(t)$ denotes an open loop control,disturbance or excitation, and by $u^*(t)$ we denote the unique optimal control to the quadratic optimization problem introduced below.

Burgers' equation has been studied intensively as nonlinear convection diffusion problem to investigate numerical algorithms. To the authors knowledge, the publication [60] first used proper orthogonal decomposition to obtain reduced order models for Burgers' equation. Kunisch and Volkwein [125] successfully applied POD to the LQR optimal control problem for Burgers' equation. Numerical issues related to finite precision arithmetic, sensitivity as well as a discussion of solutions for various boundary conditions are given in [46, 5, 4]. To improve the POD basis in the parameter space, [96] proposes two methods incorporating sensitivity analysis in the basis computation for POD. A numerical study of various sensitivity enhanced POD basis for the uncontrolled system can be found in [107]. A faster computation of the nonlinearity in Burgers' equation via POD, called 'group' POD, was proposed by Dickinson and Singler in [71].

In many practical applications, a first step to design a feedback controller involves the linearization of the nonlinear system; a rigorous theory for optimal control is available for linear dynamical systems [126]. The coupled Burgers' equation (2.1)-(2.2) is subsequently linearized around its steady state solution. From [5, 4] and the references therein it is known that the only equilibrium to Burgers' equation with homogeneous, mixed Dirichlet-Neumann boundary conditions is the zero solution. This solution is globally asymptotically stable. By imposing zero Dirichlet boundary conditions on the heat equation (2.2), the energy eventually dissipates and the system converges uniformly to the zero steady state, $T_{ss} \equiv 0$, independent of the initial condition. Thus, $w_{ss} = T_{ss} = 0$ is an equilibrium to (2.1)-(2.2). To this end, the velocity and temperature are decomposed into a steady state and fluctuation part as

$$w(t,x) = w_{ss}(x) + \tilde{w}(t,x) = \tilde{w}(t,x),$$
$$T(t,x) = T_{ss}(x) + \tilde{T}(t,x) = \tilde{T}(t,x).$$

It is assumed that the fluctuations are small in the $H^1(\Omega)$ norm, implying $\tilde{w} \cdot \tilde{w}_x \approx 0$. Thus, the linearized coupled Burgers' system is given by

$$\dot{\tilde{w}}(t,x) = \mu \tilde{w}_{xx}(t,x) - \kappa \tilde{T}(t,x), \tag{2.6}$$

$$\dot{\tilde{T}}(t,x) = c\tilde{T}_{xx}(t,x) + b(x)u(t), \tag{2.7}$$

and is also called the fluctuation system.

2.1.1 Abstract Formulation

Here, we give a concrete example of an abstract formulation of the linearized, coupled Burgers' equation. Let $z(t,\cdot) = [w(t,\cdot)\ T(t,\cdot)]^T \in L^2(0,\infty;\Omega) \times L^2(0,\infty;\Omega))$ be the state variable of the coupled system. The state space is defined as the Hilbert space

$$Z := L^2(0,\infty;\Omega) \times L^2(0,\infty;\Omega) = (L^2(0,\infty;\Omega))^2.$$

Moreover, define the Hilbert spaces

$$H^1_L(0,\infty;\Omega) := \{v \mid \frac{\partial}{\partial x} v \in L^2(0,\infty;\Omega), v(\cdot,0) = 0\},$$

and similarly

$$H^1_0(0,\infty;\Omega) := \{v \mid \frac{\partial}{\partial x} v \in L^2(0,\infty;\Omega), v(\cdot,0) = 0,\ v(\cdot,1) = 0\}.$$

The linear part of the dynamical system is given by the operator

$$\mathcal{A}:\ \mathcal{D}(\mathcal{A}) = (H^1_L(\Omega) \cap H^2(\Omega)) \times (H^1_0(\Omega) \cap H^2(\Omega)) \mapsto Z,$$

where $\mathcal{A}$ takes the specific form

$$[\mathcal{A}z](t) = \begin{bmatrix} \mu\frac{d^2}{dx^2} & -\kappa \\ 0 & \frac{d^2}{dx^2} \end{bmatrix} z(t).$$

As in §1.5, consider only a single control action, so that the space of controls is $U = \mathbb{R}$. The control operator $\mathcal{B} \in \mathcal{L}(U, Z)$ is then given by

$$[\mathcal{B}u](t) = \begin{bmatrix} 0 \\ b(x) \end{bmatrix} u(t).$$

The linearized system (2.6) - (2.7) takes the abstract form

$$\begin{aligned}\dot{z}(t) &= \mathcal{A}z(t) + \mathcal{B}u(t)\\ z(0) &= z_0\end{aligned}$$

on the state space Z. With the variation of parameters formula, the solution to the above system is formally given by

$$z(t) = S(t)z_0 + \int_0^t S(t-s)\mathcal{B}u(s)ds,$$

where $S(t)$ generates an analytic semigroup on Z.

Whenever linear outputs of the states are considered, they are given through

$$y(t) = \mathcal{C}z(t) \quad \in Y = H^1(0,\infty;\Omega).$$

The quadratic cost functional of interest for the optimal control problem is as before over an infinite time horizon, and takes the form

$$J(u,z) = \int_0^\infty \left\{||\mathcal{C}z(t)||_Y^2 + R||u(t)||_U^2\right\} dt, \tag{2.8}$$

where $R \in \mathbb{R}$ represents a cost attributed to the control action. Subsequently, we shall choose $\mathcal{C} = I_Z$ to be the identity operator in Z. This implies that one is interested in the entire state information as an input to the control problem. Therefore, the filtering problem does not have to be solved.

It can be shown that the operator $\mathcal{A}$ is stable, so that according to Theorem 1.5.3, the optimal control problem has a unique solution given by linear feedback $\mathcal{K}: Z \mapsto U$, where

$$u^*(t) = -\mathcal{K}z(t) = -\int_\Omega k^w(x)w(t,x)dx - \int_\Omega k^T(x)T(t,x)dx \quad \forall t \in (0,\infty),$$

which follows from Riesz' representation theorem. Here, $k^w(t,x)$ and $k^T(t,x)$ are called *feedback gains* for the velocity and temperature, respectively.

2.2 Approximation and LQR Control

For the purpose of spatial discretization of the infinite dimensional system, a group finite element method (GFEM) [84] is used for the linear and nonlinear high fidelity models and proper orthogonal decomposition is used to obtain the surrogate model. We briefly introduce those methods and refer the reader to [71, 122] for advantages and implementation of the GFEM and a group-POD approximation of the nonlinear coupled Burgers' equation (2.1) - (2.5).

2.2.1 Finite Element Method

Here, a brief derivation of the FE model for the linearized system (2.6) - (2.7) is given. For a detailed treatment of the nonlinear term, see [122]. Piecewise linear (PWL) finite element basis functions ("hat functions") are used to approximate the PDE solutions. The spatial domain $\Omega = (0, 1)$ is divided into $n + 1$ subintervals of equal length, with step size $h = \frac{1}{n+1}$. The finite element approximation spaces for the temperature and velocity are

$$\mathcal{X}_w^h(\Omega) := \{\xi_w \in PWL(\Omega) \mid \xi_w(0) = 0\},$$
$$\mathcal{X}_T^h(\Omega) := \{\xi_T \in PWL(\Omega) \mid \xi_T(0) = \xi_T(1) = 0\}.$$

Note, that the boundary conditions, equations (2.3) and (2.4), are built in to the FE approximation spaces. In particular, where a zero boundary condition is specified, one can omit having a basis function which assumes a nonzero value there. To be more specific, the temperature and velocity are approximated by

$$w(t, x) \approx w_n(t, x) = \sum_{i=1}^{n+1} \alpha_i(t)[\xi_w]_i(x) \quad \in \mathcal{X}_w^h,$$

and

$$T(t, x) \approx T_n(t, x) = \sum_{i=1}^{n} \beta_i(t)[\xi_T]_i(x) \quad \in \mathcal{X}_T^h.$$

To simplify notation, let $x(\cdot) = [\alpha(\cdot)^T \ \beta(\cdot)^T]^T \in \mathbb{R}^{2n+1}$ be the state variable of the FE system. We seek to retain the physical meaning of the two dependent variables in the derivation and the reduced order modeling. After linearization and discretization, the coupled Burgers' equation takes the standard, linear time invariant form with a mass matrix

$$E\dot{x}(t) = Ax(t) + Bu(t), \tag{2.9}$$
$$Ex(0) = x_0, \tag{2.10}$$

where the system matrix A has the block structure

$$A = \begin{bmatrix} \mu A_w & \kappa Q \\ 0 & c A_T \end{bmatrix} \in \mathbb{R}^{2n+1} \times \mathbb{R}^{2n+1}. \tag{2.11}$$

Similarly, the mass matrix assumes a block structure with

$$E = \begin{bmatrix} E_w & 0 \\ 0 & E_T \end{bmatrix} \in \mathbb{R}^{2n+1} \times \mathbb{R}^{2n+1}. \tag{2.12}$$

The individual matrices can be computed explicitly as

$$E_w = \frac{1}{6(n+1)} \begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 2 \end{bmatrix}, \quad A_w = (n+1) \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}.$$

Additionally, $E_T = [E_w]_{1:n,1:n}$, $A_T = [A_w]_{1:n,1:n}$ and $Q = [E_w]_{1:n+1,1:n}$ following the notation introduced earlier. The cost of the optimal control problem is given by equation (1.33) and we repeat it here for ease of presentation:

$$J(u(\cdot), x(\cdot)) = \int_0^\infty \left\{ ||x||^2 + R||u(t)||^2 \right\} dt.$$

The scalar R is chosen as a penalty/weighting term for the control action. The optimal control problem is then stated as:
Minimize $J(u, x)$ subject to the dynamic constraints (2.9) – (2.10).

The optimal control problem can be solved via the solution of the algebraic Riccati equation (1.36). Let $u_n(\cdot)$ be the control function for the n-th dimensional finite element system. The control function does not need to be discretized in space, but one should distinguish it from the control obtained from the surrogate model, denoted by $u_r(\cdot)$. In light of Theorem 1.5.13 the solution of the LQR problem is

$$u_n^*(t) = -Kx(t), \tag{2.13}$$

where the gain matrix $K \in \mathbb{R}^n$. Closing the feedback loop on the linearized system yields

$$E\dot{x}(t) = [A - BK]\, x(t). \tag{2.14}$$

With a linear controller, the closed loop nonlinear system reads as

$$E\dot{x}(t) = Ax(t) + F(x(t)) - BKx(t).$$

From perturbation theory (Perron's Theorem 1.4.3), this system is stabilizable via a linear controller, if the nonlinearity is "weak".

Remark 2.2.1. The optimal control $u_n(t)$ was designed using a classical linear quadratic regulator on the (large) finite element system. The theory for the convergence of the resulting feedback matrix to its infinite dimensional counterpart is well established. We refer the reader to the appendix, where several related results are stated, in particular Theorem A.1.7.

2.2.2 Reduced Order Modeling

Here, a reduce-then-design approach via POD is derived to compute the optimal control, and a convergence study is performed based on the size of the reduced order model. To assess performance, we are interested in both the robustness of the POD approximation with respect to parameter changes, as well as the ability of the reduced order controller to work on the full FE model. Ideally, one would like to see how the controllers from the reduced order models work in a physical setting. Due to the lack of an experimental environment,and after a thorough convergence study, we assume that the FE model is a good representation of the physical system, and henceforth call it the "truth" (or high fidelity) model.

The method of snapshots as outlined in §1.6.2 is used for POD computations, since the state space is larger than the number of time snapshots collected. To generate the solutions, simulate the fully nonlinear coupled Burgers' system (2.1) - (2.5) with open loop input as

$$E\dot{x}(t) = Ax(t) + F(x(t)) + Bu(t). \tag{2.15}$$

Let $x_i = x(t_i)$ for $i = 1, \ldots, n_s$ be snapshots from simulations of the above system. Then, the matrix of snapshots is defined, and partitioned, as

$$X := [x_0,\ x_1, \ldots, x_{n_s}] = \begin{bmatrix} X_w \\ X_T \end{bmatrix}.$$

The reader should observe that the functions $w(\cdot,\cdot)$ and $T(\cdot,\cdot)$ have different scaling and physical meanings, so the POD modes for both functions are computed separately. Let E_w and E_T be the mass matrices of the FE system. The singular value (or eigenvalue) decomposition of the correlation matrices are

$$\begin{aligned} [X_w]^T E_w X_w &= U_w \Sigma_w [U_w]^T, \\ [X_T]^T E_T X_T &= U_T \Sigma_T [U_T]^T. \end{aligned}$$

Let $[u_w]_i$ be the i^{th} column of U_w and $[\sigma_w]_i$ be the diagonal elements of Σ_w. The POD basis functions are then computed as

$$\begin{aligned} \phi_i &= \frac{1}{\sqrt{n_s [\sigma_w]_i}} X_w [u_w]_i, \quad i = 1, \ldots, r_1, \\ \psi_i &= \frac{1}{\sqrt{n_s [\sigma_T]_i}} X_T [u_T]_i, \quad i = 1, \ldots, r_2, \end{aligned}$$

where r_1 and r_2 are the dimensions of the POD spaces for the velocity and temperature, respectively. The POD basis vectors are stored in matrices

$$\begin{aligned} \Phi &= [\phi_1,\ \phi_2, \ldots, \phi_{r_1}] \in \mathbb{R}^{(n+1)\times r_1}, \\ \Psi &= [\psi_1,\ \psi_2, \ldots, \psi_{r_2}] \in \mathbb{R}^{n\times r_2}. \end{aligned}$$

The reduced order models are obtained through projection of the finite element spaces onto lower dimensional POD spaces. Hence,

$$A_r = \begin{bmatrix} \Phi & 0 \\ 0 & \Psi \end{bmatrix}^T \begin{bmatrix} \mu A_w & \kappa Q \\ 0 & c A_T \end{bmatrix} \begin{bmatrix} \Phi & 0 \\ 0 & \Psi \end{bmatrix}, \quad B_r = \begin{bmatrix} \Phi & 0 \\ 0 & \Psi \end{bmatrix}^T B, \quad C_r = C \begin{bmatrix} \Phi & 0 \\ 0 & \Psi \end{bmatrix}.$$

The mass matrix of the reduced order model is $E_r = I_r$, by virtue of orthogonality of the POD basis functions. Let $x_r(\cdot) = [\alpha_r^T(\cdot),\ \beta_r^T(\cdot)]^T$ be the state variable of the POD system. Then, the POD-ROM of the linear system (2.9),(2.10), is given by

$$\dot{x}_r(t) = A_r x_r(t) + B_r u_r(t), \tag{2.16}$$

with initial conditions

$$x_r(0) = x_{r,0} = \begin{bmatrix} \Phi & 0 \\ 0 & \Psi \end{bmatrix}^T x_0 \quad \in \mathbb{R}^{r_1+r_2}, \tag{2.17}$$

and cost function

$$J(x_r(\cdot), u_r(\cdot)) = \int_0^\infty \left\{ ||x_r||^2 + R||u_r(t)||^2 \right\} dt. \tag{2.18}$$

The temperature and velocity are approximated by

$$\begin{bmatrix} w(t,x) \\ T(t,x) \end{bmatrix} \approx \begin{bmatrix} \Phi\alpha_r(t) \\ \Psi\beta_r(t) \end{bmatrix}. \tag{2.19}$$

At this point, we shall mention that the model reduction as outlined above is preserving the structure of the problem; the reduced state has a clear separation of a temperature and velocity component. We believe this to be an important feature of the reduced order model.

Solving the LQR problem for the POD system (2.16)-(2.18) as outlined in Theorem 1.5.13 yields the linear feedback law

$$u_r^*(t) = -K_r x_r(t) \tag{2.20}$$

which exponentially stabilizes the POD-ROM. As before, the gain matrix is constructed from the solution of an algebraic Riccati equation as

$$K_r = R^{-1}[B_r]^T P_r,$$

where P_r is the solution to the algebraic Riccati equation

$$P_r A_r + [A_r]^T P_r - P_r B_r R^{-1}[B_r]^T P_r + C_r^T C_r = 0. \tag{2.21}$$

The equation (2.21) is of reduced order $r_1 + r_2 \ll n$. This reduces much of the computational effort to compute the controller and hence to close the loop in the dynamical system. The gain matrix can be injected into the finite element space via

$$K_r^n := [K_w^T \; K_T^T]^T = \begin{bmatrix} \Phi & 0 \\ 0 & \Psi \end{bmatrix} K_r.$$

The closed-loop FE system is given by

$$E\dot{x}(t) = [A - BK_r^n]\, x(t) + F(x(t)). \tag{2.22}$$

The function $u_r^*(t) = -K_r x_r(t)$ provides an optimal control for the reduced order model, yet there is no guarantee about its performance on the finite element system. Therefore, we say that $u^*(\cdot) = -K_r^n x(t)$ provides a "suboptimal" feedback control for the full order model and investigate its performance in the numerical section below.

Remark 2.2.2. An alternative would be to not separate the data, and compute the POD from the data matrix X directly. By using the method of snapshots, the resulting correlation matrix reads as

$$X^T E X = \left(\begin{bmatrix} w \\ T \end{bmatrix} (t_i), \ \begin{bmatrix} w \\ T \end{bmatrix} (t_j) \right)_{i,j=1,\dots,n_s} = (w(t_i), w(t_j))_{\mathbb{R}^{n+1}} + (T(t_i), T(t_j))_{\mathbb{R}^n}.$$

Consequently, the projection matrix resulting from the POD modes will have the structure $\Phi := \begin{bmatrix} \Phi_w \\ \Phi_T \end{bmatrix}$. Note, that the Φ_w, Φ_T will not be the same as computed earlier! Using Φ to project the linear system (2.9)-(2.10), the resulting ROM becomes

$$E \frac{d}{dt} a_r(t) = A_r a_r(t) + \Phi^T F(\Phi a_r(t)),$$

where the approximation of the solution in n dimensions is

$$\begin{bmatrix} w(t) \\ T(t) \end{bmatrix} \approx \begin{bmatrix} \Phi_w \\ \Phi_T \end{bmatrix} a_r(t).$$

The "states" of the ROM are then $a_r(t) = [\Phi_w]^T w(t) + [\Phi_T]^T T(t)$. Here, the physical meaning of the states $a_r(t)$ is unclear, and moreover it is not possible to allow for different sizes of basis functions in Φ_w and Φ_T.

2.3 Numerical Results

We present numerical experiments to investigate the performance of "suboptimal" feedback, computed from reduced order models, on the full finite element model. We investigate the convergence of $K_r^n \to K$ as the reduced basis size increases. Moreover, the POD feedback gains are used to control the nonlinear FE system. Following, the robustness of the controllers to parameter changes is studied. The parameters for the simulation are given in Table 2.1. The reader should observe that the system matrix, and, correspondingly, the feedback gains are parameter dependent:

$$E\dot{x}(t) = [A(\mu_2) - BK_r^n(\mu_1)]x(t) + F(x(t)), \tag{2.23}$$

where $\mu = \frac{1}{Re}$ is the viscosity parameter.

Parameter	Name	Value
Thermal conductivity	c	0.01
Coeff. of the thermal expansion	κ	1.0
Reynolds number	Re	120
Distributed control location	$b(x)$	x
Domain	Ω	$[0,1]$
Final time	T	5s
Snapshots to compute POD	n_s	300
Cost on control	R	0.1

Table 2.1: Parameters for the numerical experiments for the coupled Burgers' equation.

2.3.1 Dependence of Gains on Snapshot Data

Proper orthogonal decomposition extracts the most energetic modes from experimental or simulation data of a dynamical system. The nonlinear coupled Burgers' equation (2.15) is excited with various inputs $u(t) \approx \delta(\hat{t} - t)$ at some $\hat{t} > 0$, and n_s snapshots collected equidistantly in time. In numerical experiments, we found that impulse excitations provide a richer dataset than exciting the system with nonzero initial conditions. More precisely, an impulse shortly after start is modeled as

$$u(t) = c\lceil \sin(\pi(5t - 0.1)) \rceil H(1 - (5t - 0.1)), \tag{2.24}$$

where $H(\cdot)$ is the Heaviside function, i.e. $H(t) = 0,$ if $t < 0$ and $H(t) = 1,$ if $t \geq 0$, and c is a constant. A plot of $u(t)$ is given in Figure 2.1. As previously mentioned, we restrict ourselves to a scalar input $u(t)$, so $B \in \mathbb{R}^n$ is a column vector.

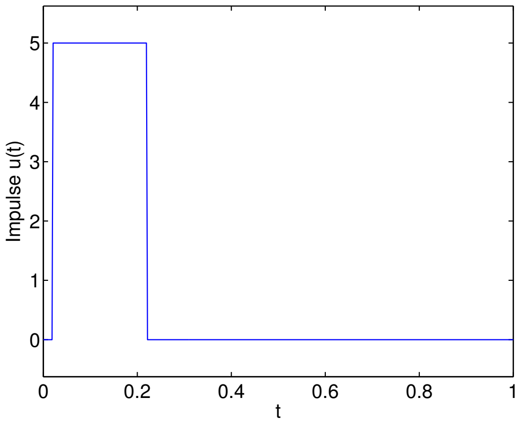

Figure 2.1: Impulse function $u(t)$ with $c = 5$ in equation (2.24).

The response $x(t)$ over five seconds to the impulse $u(t)$ with $c = 5$ is plotted in Figure 2.2. The impulse excitation quickly tends to zero, due to the dissipation of the system and the zero boundary conditions on the temperature.

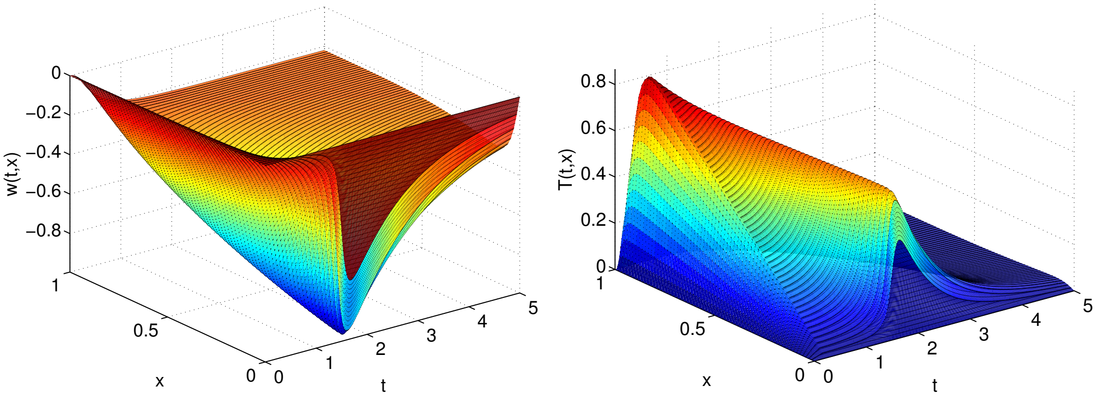

Figure 2.2: Impulse response of the FE system with $n = 64$ to $u(t)$ with $c = 5$; velocity (left) and temperature (right).

The convergence of the feedback gains as n, the discretization parameter of the FE model, increases, is considered first. Theoretically, the question of convergence of the FE feedback gains to their infinite dimensional counterparts has been settled by

Ito [105], see the appendix for more information. A numerical convergence history of the finite element gains with respect to n is given in Figure 2.3, and Table 2.2. Convergence of the gains is established by both convergence in norm, as well as decreasing differences between subsequent gain functions. For comparison purposes, the gains in Table 2.2 were interpolated onto the finest resolution of $n = 128$.

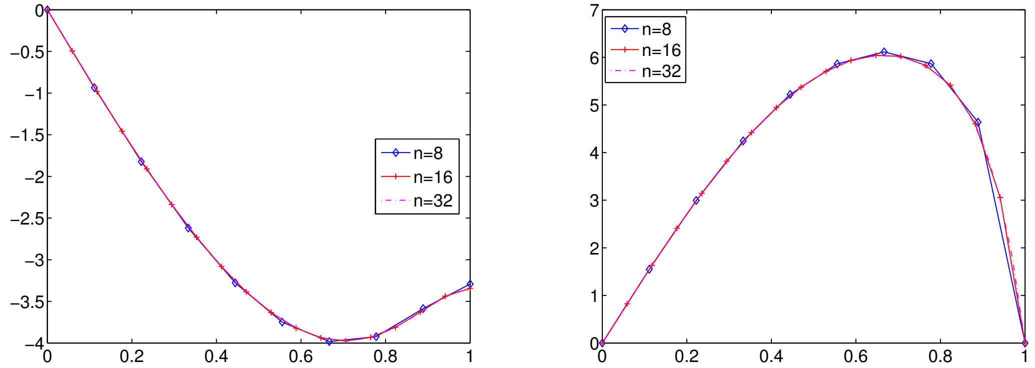

Figure 2.3: Convergence of finite element feedback gains with respect to the mesh size n; velocity gain (left) and temperature gain (right).

Model size	$\|\|K_w^n\|\|_2$	$\|\|K_T^n\|\|_2$	$\frac{\|\|K_w^n - K_w^{n/2}\|\|_2}{\|\|K_w^{n/2}\|\|_2}$	$\frac{\|\|K_T^n - K_T^{n/2}\|\|_2}{\|\|K_T^{n/2}\|\|_2}$
$n = 8$	36.86	55.85	–	–
$n = 16$	35.88	53.29	0.064	0.088
$n = 32$	35.35	51.82	0.037	0.063
$n = 64$	35.09	51.04	0.020	0.040
$n = 128$	34.95	50.64	0.010	0.023

Table 2.2: Convergence of the feedback gains for velocity K_w^n and temperature K_T^n as the FE approximation is refined.

2.3.2 Convergence of Gains as Basis Increases

Having confirmed that the finite element gains converge, we now turn our focus to the convergence behavior of the feedback gains computed from the reduced order surrogate model. The training data is generated from the nonlinear finite element model with $n = 64$, excited with an impulse $u(t)$ as in equation (2.24), with $c = 5$. It should soon become clear, that using a different number of basis functions for both the temperature and velocity gives great flexibility in achieving good convergence results with a minimal number of cumulative basis functions. Figure 2.4 shows the convergence of the gains computed from the low order POD matrices E_r, A_r, Br to the FE feedback gains computed from E, A, B. The gains from the ROM show good agreement with the finite element gains, when $r_1 + r_2 = 8$ overall basis functions are used. Apparently, more velocity bases are needed than temperature bases. This is often observed in hybrid systems, where functions can have different scales and complexity.

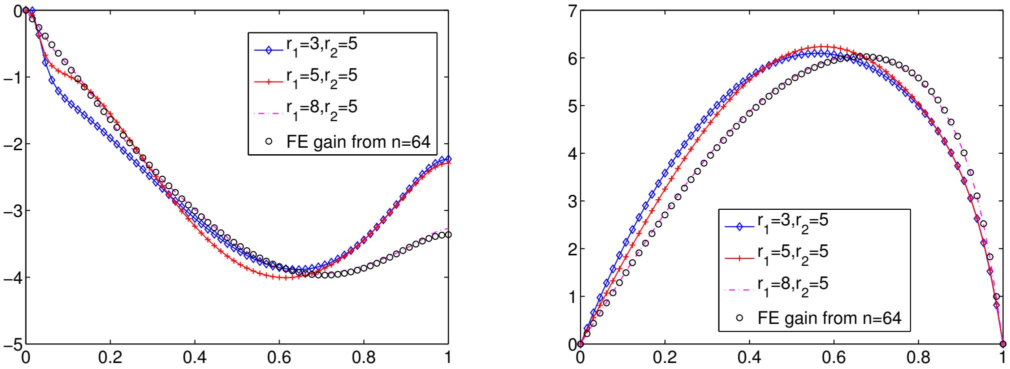

Figure 2.4: Convergence of POD to FE gains ($n = 64$) when increasing the POD basis size; velocity gain (left) and temperature gain (right).

We briefly demonstrate, that the value of c in the definition of the impulse influences the richness of the simulated data. Since $u(t)$ is an approximation of the delta distribution, it has unit integral, which implies that $c = 5$. Choosing larger values of c to generate the training data significantly changes the shape and convergence rate of the feedback gains. In Figure 2.5, the convergence of the feedback gains computed from the data generated from a pulse with $c = 15$ is shown. The left plot shows the velocity feedback gains and the right plot contains the gains for the temperature. By pure visual inspection, the gain functions do not converge as quickly as in Figure 2.4, where the correct impulse is used to excite the data. With this simple example, we intend to demonstrate that care must be taken in the generation of the training data. When the reduced order model is used for computation of the feedback gains only, then an ensemble of high order gains K, computed from various parameter settings,

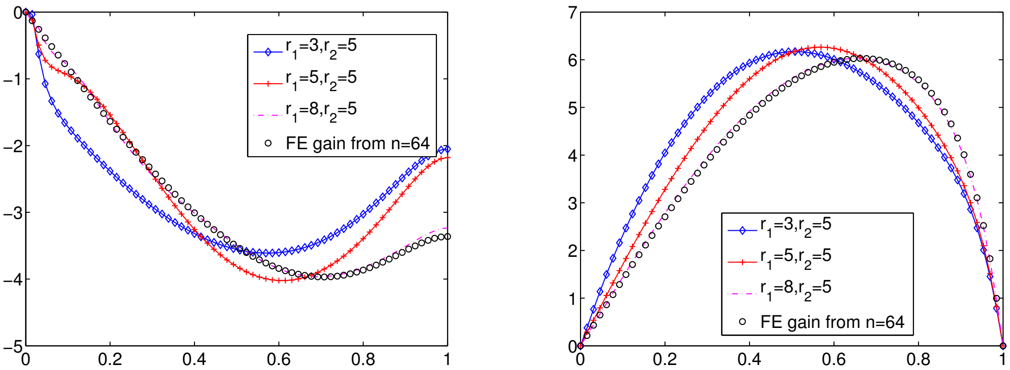

Figure 2.5: POD feedback gains for velocity (left) and temperature (right). The snapshots were generated with a pulse $u(t)$ and $c = 15$.

and initial conditions can be beneficial, see [12].

Next, a more in depth study of the suboptimal POD feedback gains is provided. For $r_1 = 5, r_2 = 3$, we compare the sub-optimally controlled nonlinear finite element system ($n = 64$) with the open loop system. Figure 2.6 shows the open loop dynamics of the nonlinear FE system over the first five seconds. The initial conditions are $T_0(x) = 5\sin(\frac{1}{2}x)$ and $w_0(x) = g(x) - g(1-x)$, where $g(x) = H(x - .3) - (1 - H(\frac{1}{5} - (x - .3)))$. Due to the zero Dirichlet boundary conditions on the temperature, even the open loop system decays to zero quickly. For a better comparison, the average temperature over time is plotted in Figure 2.7. The controller, as expected, speeds up the transient to the zero steady state.

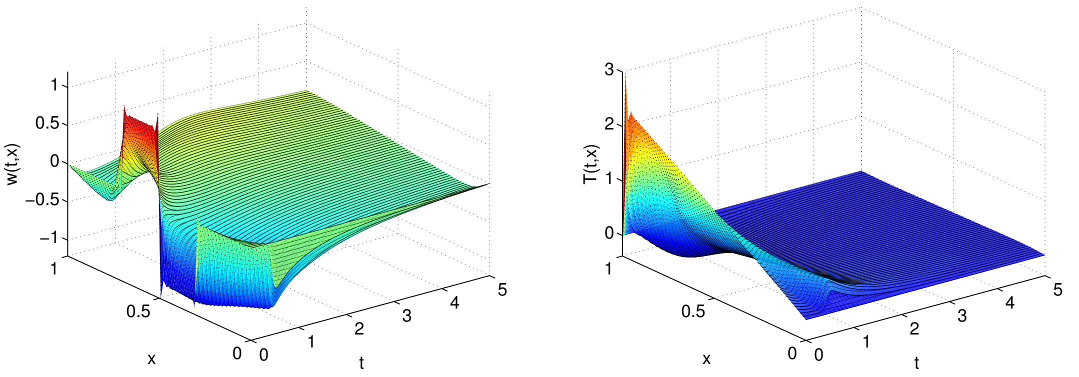

Figure 2.6: Simulation of the nonlinear open loop FE system with $n = 64$; velocity (left) and temperature (right).

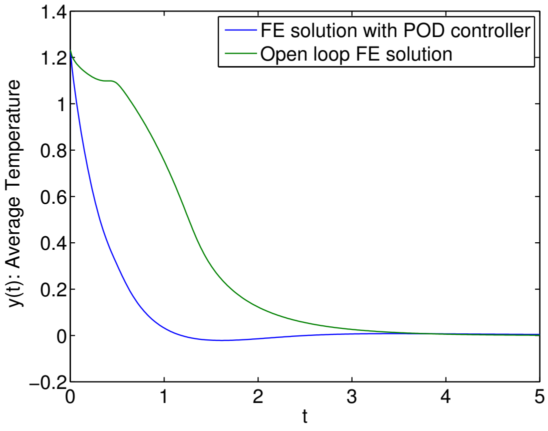

Figure 2.7: Average temperature versus time. Open and closed loop nonlinear FE system ($n = 64$) with POD controller, computed from $r_1 = 5,\ r_2 = 3$ basis functions.

Stability of the closed loop system is arguably the most important property a controller should achieve. Given a stabilizable dynamical system, the resulting closed loop dynamical system with optimal control is exponentially asymptotically stable. This means that all eigenvalues of the closed loop system have negative real part. Recall, that control via POD gains only produces a suboptimal feedback for the FE system of the form

$$E\dot{x}(t) = (A - BK_r^n)x(t),$$

and stability of the closed loop system therefore needs to be investigated. The above problem leads to a generalized eigenvalue formulation

$$EV = \tilde{A}V\Lambda,$$

where Λ is a diagonal matrix containing the eigenvalues, and $\tilde{A} \in \{A, (A-BK_r^n), (A-BK)\}$. The generalized eigenvalues of the open loop FE system at $n = 64$, and two closed loop systems are plotted in Figure 2.8. The eigenvalues were computed with `eigs` in `MATLAB`. The reader should observe, that closing the feedback loop indeed moves the eigenvalues to the left side, as desired. The real parts of the first three eigenvalues of the open loop FE, closed loop FE and closed loop FE system with a POD controller are given in Table 2.3.

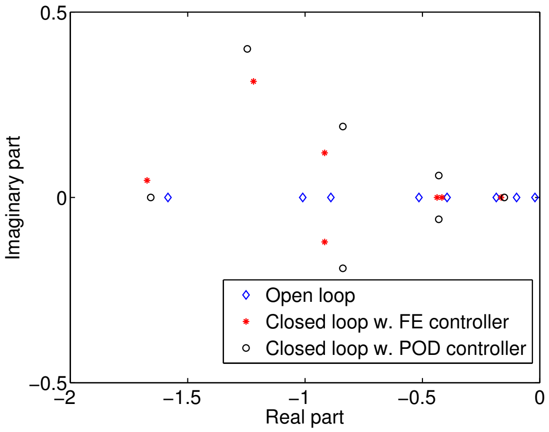

Figure 2.8: Eigenvalues of the open loop matrix A, and closed loop FE matrices, $A - BK$ and $A - BK_r^n$, where $n = 64$.

System	λ_1	λ_2	λ_3
Open loop FE	-0.021	-0.099	-0.186
Closed loop w. FE controller	-0.165	-0.417	-0.437
Closed loop w. POD controller	-0.151	-0.430	-0.430

Table 2.3: Real part of the three eigenvalues closest to the imaginary axis; linear open and closed loop systems at $n = 64$.

The above numerical experiments show that a POD controller applied to the FE system yields similar results compared to using the FE gains. However, solving the LQR problem is much cheaper for the ROM, see Table 2.4. The spatial integration is more expensive for the POD model, since the basis functions are dense in $\mathbb{R}^n$, whereas they are sparse for the FE model ("hat functions"), hence they can be quickly integrated.

Model	dimension	Model assembly	Solving LQR
FE	128	0.03s	1.40s
POD	5+3	0.53s	0.02s

Table 2.4: CPU time [s] to build (spatial integration) the FE and POD system, and to solve the Riccati equation for the feedback gain.

2.3.3 Off-Design Use

In many physical environments, conditions and model parameters change with time. For instance, when controlling indoor-air environments, the underlying system parameters, boundary conditions, etc., change during the application of the controller. Thus, one would like for the controller to be robust to small changes in the parameters and operating environment. As noted earlier, the closed loop systems, (2.14) and (2.22), explicitly depend on parameters:

$$\dot{x}(t) = [A(\mu_2) - BK(\mu_1)]x(t) + F(x(t)), \tag{2.25}$$

where μ_1 and μ_2 are the reciprocals of two different Reynolds numbers. Note, that the feedback gain is computed using the system matrices A, B, C and the penalty term R. For $A = A(\mu)$ depends on the Reynolds number, the functional gain is computed for a specific Reynolds number, so we emphasize $K = K(\mu)$. Below, we investigate the performance of a gain $K(\mu_1)$ on a dynamical system under varying Reynolds numbers.

Figures 2.9 – 2.10 show the performance of the POD and FE controllers when the Reynolds number of the open loop system is varied. The full nonlinear FE model at a baseline Reynolds number of $Re_2 = 120$ is simulated with an impulse $u(t)$, as given in equation (2.24). Then, the POD-ROM as well as the POD gains are computed by solving a low order Riccati equation. Subsequently, the Reynolds number Re_1 that influences the dynamics of the system is changed. A comparison of the relative errors in the gains $||K(\mu_2) - K(\mu_1)||/||K(\mu_1)||$ and the propagation of these errors to the solutions $||x(\mu_1, K(\mu_2), t) - x(\mu_1, K(\mu_2), t)||/||x(\mu_1, K(\mu_1), t)||$, as computed from equation (2.25) is given. Figure 2.9 depicts the result for the controller acting on the nonlinear system, while Figure 2.10 shows the results for the controlled linear system.

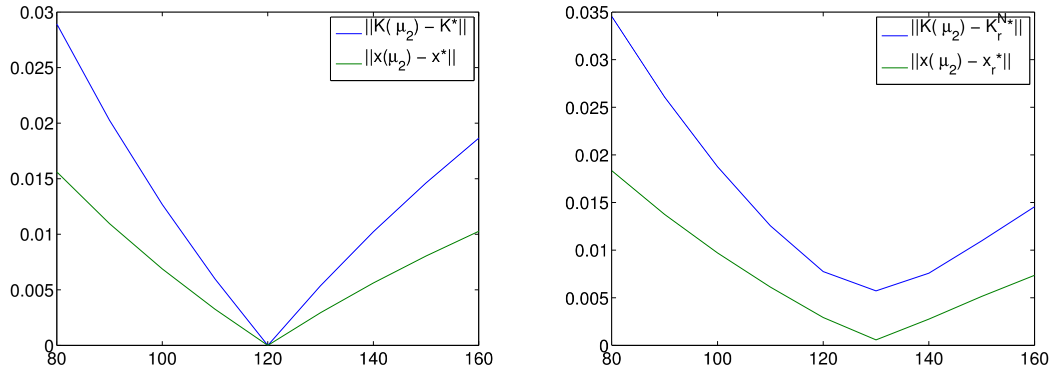

Figure 2.9: Nonlinear FE system with $n = 64$: Relative errors in feedback gains and solutions as the Reynolds number ($Re = \frac{1}{\mu}$) is varied. (left): The effect on the FE system; (right): POD system.

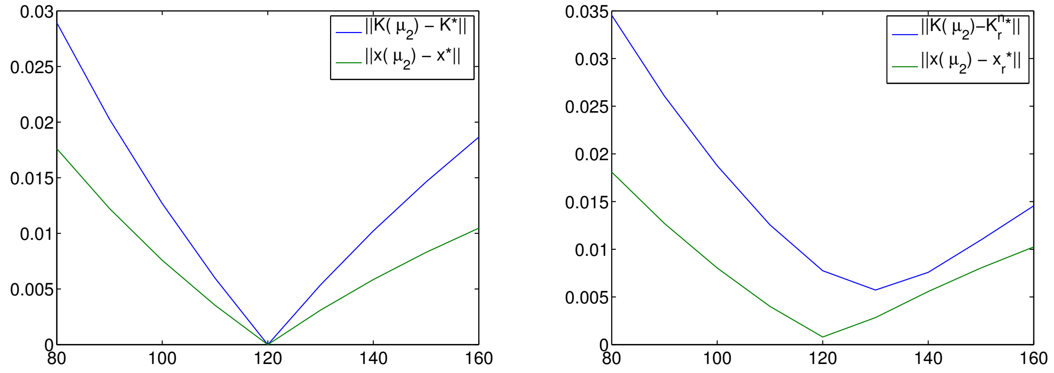

Figure 2.10: Linear FE system with $n = 64$: Relative errors in feedback gains and solutions as the Reynolds number ($Re = \frac{1}{\mu}$) is varied. (left): The effect on the FE system; (right): POD system.

Figure 2.11 shows the eigenvalues $\lambda(A(\mu_2) - BK(\mu_1))$ of the closed loop system for $\mu_1 = \frac{1}{120}$ and $\mu_2 = \frac{1}{80}$. Note, that all eigenvalues are bound away from zero in the open left half plane. However, the eigenvalues did move slightly closer towards zero, as the feedback is only sub-optimal.

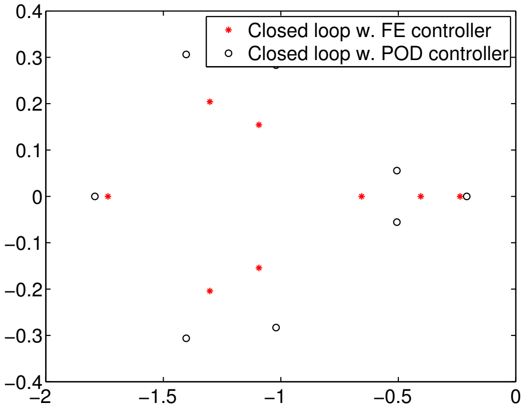

Figure 2.11: Eigenvalues of the closed loop linear FE system at $\mu_2 = \frac{1}{80}$ with controller computed at $\mu_1 = \frac{1}{120}$.

2.4 Conclusions

In this part of the thesis, a coupled Burgers' equation as a simplified model of thermal fluid dynamics, and numerical testbed, was considered. We illustrated and derived the computational scheme for designing LQR controllers via reduced order models. We addressed data generation options (which excitation), and finding a sufficient size of the ROM. Moreover, we provided a careful, structure-preserving derivation of the reduced order model, and discussed an alternative, and its drawbacks. An in depth numerical study comparing performance, robustness and computational effort to design controllers via proper orthogonal decomposition reduced order models. The closed loop POD controllers satisfactory controlled the high fidelity model, remarkably so for off-design parameters. In our assessment, the reduced order controllers performed fairly well in parametric neighborhoods of the training sets from which they were generated. In particular, the controllers provided a robust stabilization with respect to parametric changes. As we mentioned in the introduction, the reduce-then-design approach can fail for certain systems. However, for a system like the coupled Burgers' equation considered here, it gave satisfactory performance, while allowing computational savings at the online stage of the controller.

In the next chapter, we approach the optimal control problem from a different perspective, namely the design-then-reduce approach. There, we develop a method which works on large-scale systems, is also based on POD, and provides a sound framework for solution of ARE through simulations of linear systems. The algebraic Riccati equations in the present chapter were computed via rather slow Schur-form solvers in `Matlab`, which is unfeasible for large systems.

Chapter 3

Solution of Large-Scale Riccati Equations

The solution of large-scale matrix algebraic Riccati equations (AREs) is important for instance in control design and model reduction and remains an active area of research. We propose a projection method to obtain low rank solutions of AREs based on simulations of linear systems coupled with proper orthogonal decomposition (POD). The method can take advantage of existing (black box) simulation code to generate the projection matrices. Furthermore, simulation methods such as parallel computation, domain decomposition, and multi-grid methods can be exploited to increase accuracy and efficiency and to solve large-scale problems. We present numerical results demonstrating that the proposed approach can produce highly accurate approximate solutions. We also briefly discuss making the proposed approach completely data-based so that one can use existing simulation codes without accessing system matrices. Furthermore, a comparison with an extended Krylov subspace method shows that the proposed approach can give higher accuracy at a lower approximate solution rank.

3.1 Introduction

Riccati equations play an important role in a variety of problems, including optimal control and filtering. For instance, the solution to the algebraic Riccati equation (ARE) determines the optimal feedback control solving the linear quadratic regulator problem. Such feedback control laws are used to stabilize a dynamical system and to steer the dynamics to desired equilibrium states. Moreover, the problem of optimal state estimation from given measurements of a linear dynamical system also involves a solution to an algebraic Riccati equation. For details about control and estimation, see [126, Chapter 4] and [45, Chapter 12] and the many references therein. Solutions of AREs are also important for certain model reduction algorithms, such as LQG balanced truncation, [8, §7.5] and [183, 26, 28]. Furthermore, optimizing sensor and actuator locations in optimal control problems can require the solution of many AREs; see, e.g., [62, 69, 117, 64]. In this chapter, we consider nonlinear matrix algebraic Riccati equations of the form

$$A^T PE + E^T PA - E^T PBB^T PE + C^T C = 0. \tag{3.1}$$

Here, $A, E \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{p\times n}$ are given matrices, and $P \in \mathbb{R}^{n\times n}$ is the unknown matrix. As can be seen from Theorem 1.5.13, if (E, A, B) is stabilizable, the exact solution P of the ARE (3.1) exists and is symmetric positive semidefinite. This chapter is concerned with efficiently solving the above equation, and to provide a first step for a completely matrix free way of finding the solution P, without having explicit access to A, B, C, but maybe only to its weak forms, or their action on a vector, which is suitable for many commercial codes.

We are concerned with approximating the solution of the ARE (3.1) for large-scale systems, i.e., when $n > 10,000$.[1] When deriving a spatial discretization of a partial differential equation in more than one dimension using, e.g., finite element methods (§1.3), the resulting systems are inevitably large. Dimensions of $n \gg 10^6$ are rather common for such applications. Even if the matrices E and A are sparse and there are only few inputs and outputs, $m, p \ll n$, the exact solution P of the ARE is a dense $n \times n$ matrix; therefore obtaining or even storing the exact solution is problematic. Fortunately, the solution P is often of low numerical rank when $p, m \ll n$ [32, 146] and many recent solution approaches exploit this by constructing factored low rank approximate solutions of the form $P \approx ZZ^T$.

Over the past 50 years many methods and techniques have been developed to efficiently solve small or moderate size nonlinear matrix equations of Riccati type; see, e.g., [37] for an overview. A large amount of recent research has been devoted to the development and analysis of algorithms for large-scale AREs; see the recent survey

[1] The definition of "large" is clearly user dependent. Our definition is motivated by the fact that a great amount of engineering, design and control both in the corporate and academic world is still performed on standard desktop computers. For a standard desktop computer, matrix decompositions or solving matrix equations of order $n > 10,000$ is challenging.

[32]. Many of the approaches are inspired by computational linear algebra methods. The prevalent methods include invariant subspace methods [94, 44, 30], spectral projection methods [154, 53, 14], Krylov subspace methods in a projection framework [158, 106, 109, 110, 97, 168], Kleinman-Newton methods [120, 25, 52, 83, 31], and subspace iterations [132, 27].

The reader should note, that the solution of the algebraic Riccati equation is, for some applications, merely a detour to obtain the gain matrices $K = B^T PE$. Where the goal is to obtain feedback gain matrices, solving the Chandrasekhar equations [116, 133, 50, 49] provides an elegant alternative, since it circumvents the solution of the algebraic Riccati equation and directly computes the gain matrices through integration of linear systems. Traditionally, convergence of solutions to the Chandrasekhar is rather slow, yet a hybrid method involving Proper Orthogonal Decomposition for faster computation has been developed by Borggaard and Stoyanov [39].

There has also been interest in developing data-based algorithms that approximate the solution of the ARE. Such approaches do not require direct access to the matrices (E, A, B, C). This can be important if one has an existing (possibly complex) simulation code for which it is difficult or impossible to access the relevant matrices. In such a data-based setting, researchers have not typically focused on approximating the solution P of the Riccati equation; instead, researchers have primarily focused on approximating quantities that depend on the Riccati solution. For example, one can attempt to approximate the feedback gain matrix needed for optimal feedback control problems, or one can attempt to approximate the optimal control $u(t)$ directly, without computing the feedback gains. Direct methods to compute the feedback function for large-scale systems were developed in [148] and successfully applied to a linearized Navier-Stokes equation, see [163]. However, those methods do not reveal the structure and shape of K, which can be crucial in the placement of sensors and actuators.

Other data-based algorithms include approaches based on the aforementioned Chandrasekhar equations [39, 2] and a vast variety of model reduction methods (see, e.g., [12, 11, 19, 18, 130]). All of these approaches have been used successfully on a variety of problems, but they can have drawbacks. For example, the iterative optimization algorithms only provide the optimal control which cannot be used for feedback purposes. Most notably, these methods have typically not aimed to provide highly accurate approximations of the Riccati solution P.

We propose a new projection based method in §3.4 to solve AREs based on simulations of linear systems coupled with proper orthogonal decomposition (POD) and

Galerkin projection. The proposed approach is a first step towards an accurate, completely data-based projection method for solution of AREs, and is particularly formulated for researchers familiar with POD. We refer the readers to §1.6 for an introduction to projection based methods and POD. After describing the POD-projection approach, we present numerical results in §3.5 indicating that the proposed method can be used to accurately compute the feedback gain matrices. The method provides an accurate low rank approximate solution of the ARE, which can be used for model reduction applications, such as LQG balanced truncation, etc.

We note that a data-based POD approach has been previously coupled with a Kleinman-Newton iteration in [173] to approximately solve AREs. However, the recent work [168] indicates that projection methods outperform approaches based on the Kleinman-Newton iteration.

To relate our numerical findings to state-of-the-art low rank Riccati solution methods for large-scale problems, we choose an extended Krylov subspace projection method [168]. In §3.5, we find that the proposed POD approach gives high accuracy for a fixed approximation rank and is tractable for large-scale applications. The proposed method generally is not as computationally and storage efficient as Krylov methods or other computational linear algebra approaches. However, the primary goal of this work is not to produce the most efficient solver; instead, our goal is to move toward an efficient, accurate, completely data-based method that can take advantage of existing simulation codes. In addition, when the matrices come from discretizations of PDEs, efficient domain decomposition and multigrid techniques can be used to solve the linear systems, giving the method additional flexibility. Moreover, for systems with multiple outputs, the required simulations can be run in parallel. The numerical results suggest that it may be possible to develop a convergence theory for low rank solutions of algebraic Riccati equations.

3.2 Background on AREs

Throughout this paper, we consider linear time invariant systems (E, A, B, C) given by

$$E\dot{x}(t) = Ax(t) + Bu(t), \tag{3.2}$$

$$y(t) = Cx(t), \tag{3.3}$$

where $E, A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$ and $m, p \ll n$. Systems arising from discretization of of PDE's with overlapping basis functions have $E \neq I$. We assume the matrix E is invertible, so we explicitly do not consider descriptor systems.

For certain systems, such as those arising in the spatial discretization of the linearized Navier-Stokes equations (e.g., [6]), the system (3.2)-(3.3) is not stable in the sense that $E^{-1}A$ has positive eigenvalues. Recall, that a system is called asymptotically stable if all eigenvalues of $E^{-1}A$ have negative real part. Stabilizing unstable systems is one important problem in feedback control applications. Whenever the system is unstable, the observability gramian given by representation (3.6) below does not exist and certain ad hoc solution strategies for ARE can have poor performance. To make $E^{-1}A$ stable, one first finds a stabilizing feedback gain K_1, for instance via the algebraic Bernoulli equations [6] or by integrating the Chandrasekhar equations until a stabilizing feedback gain is obtained [20]. The new matrix $E^{-1}(A - BK_1)$ is then stable.

Henceforth, we assume that $E^{-1}A$ is stable. In light of the preceding discussion, we emphasize that this is not a necessary assumption. In fact, the assumption of stability of $E^{-1}A$ solely implies that the stabilizing feedback has already been applied.

Many algorithms to solve Riccati equations exploit the intrinsic connection to the linear Lyapunov equation

$$A^TXE + E^TXA + C^TC = 0. \tag{3.4}$$

One should note that (3.4) is a linear matrix equation obtained from the algebraic Riccati equation (3.1) by ignoring the nonlinear term. The connection between the Riccati and Lyapunov equation can have important implications when devising algorithms to solve AREs. In this chapter, we generate a subspace spanned by the singular vectors of an approximate Lyapunov solution to compute approximate solutions for AREs via projection.

Assuming that $E^{-1}A$ is stable, the solution to Lyapunov equation admits a closed form. To see this, rewrite the Lyapunov equation (3.4) above as

$$\tilde{A}^T\tilde{X} + \tilde{X}\tilde{A} + \tilde{C}^T\tilde{C} = 0, \tag{3.5}$$

where

$$\tilde{X} = E^TXE, \quad \tilde{A} = E^{-1}A, \quad \tilde{C} = C.$$

It is well known that the solution of the transformed Lyapunov equation (3.5) has the following integral representation:

Theorem 3.2.1. *([8, Proposition 4.27]) If $\tilde{A} = E^{-1}A$ is a stable matrix, then the Lyapunov equation (3.5) has a unique solution $\tilde{X}$, called the* ***observability***

gramian, *which has the representation*

$$\tilde{X} = \int_0^\infty e^{t\tilde{A}^T} \tilde{C}^T \tilde{C} e^{t\tilde{A}} dt. \tag{3.6}$$

The solution of the original Lyapunov equation (3.4) can be expressed in a similar form.

Corollary 3.2.2. *If $\tilde{A} = E^{-1}A$ is a stable matrix, then the Lyapunov equation (3.4) has a unique solution X given by*

$$X = \int_0^\infty e^{tE^{-T}A^T} E^{-T} C^T C E^{-1} e^{tAE^{-1}} dt.$$

Proof. The above theorem gives

$$X = \int_0^\infty E^{-T} e^{t(E^{-1}A)^T} C^T C e^{tE^{-1}A} E^{-1} dt.$$

Since A is stable, the power series expansion of the exponential matrix converges, and we have

$$e^{tE^{-1}A}E^{-1} = \sum_{i=0}^\infty (tE^{-1}A)^i E^{-1} = \sum_{i=0}^\infty E^{-1}(tAE^{-1})^i = E^{-1}e^{tAE^{-1}}$$

by virtue of $(E^{-1}A)^i E^{-1} = E^{-1}(AE^{-1})^i$, for all i. Similarly, one can see that $E^{-T}e^{t(E^{-1}A)^T} = e^{tE^{-T}A^T}E^{-T}$, which completes the proof. □

In a similar fashion, one can rewrite the Riccati equation (3.1) as

$$\tilde{A}^T \tilde{P} + \tilde{P}\tilde{A} - \tilde{P}\tilde{B}\tilde{B}^T\tilde{P} + \tilde{C}^T\tilde{C} = 0, \tag{3.7}$$

where now

$$\tilde{X} = E^T X E, \quad \tilde{A} = E^{-1}A, \quad \tilde{B} = E^{-1}B, \quad \tilde{C} = C.$$

Theorem 3.2.3. *[66] If $(\tilde{A}, \tilde{B})$ is stabilizable, then the closed form of the unique positive semidefinite solution to Riccati equation (3.7) is given by the implicit integral formulation*

$$\tilde{P} = \int_0^\infty e^{t\tilde{A}^T} \left(\tilde{C}^T\tilde{C} - \tilde{P}\tilde{B}\tilde{B}^T\tilde{P} \right) e^{t\tilde{A}} dt.$$

This is an important representation of the solution both for analysis and to gain crucial insight for the design of the approximation scheme.

Corollary 3.2.4. *If $(\tilde{A}, \tilde{B})$ is stabilizable, then the solution to the Riccati equation (3.1) is given by*

$$P = \int_0^\infty e^{tE^{-T}A^T} \left(E^{-T}C^TCE^{-1} - PBB^TP\right) e^{tAE^{-1}} dt.$$

Proof. The proof follows the technique of the previous corollary, i.e.

$$\begin{aligned} P &= \int_0^\infty E^{-T} e^{t(E^{-1}A)^T} (C^TC - E^TPEe^{-1}BB^TE^{-T}E^TPE) e^{tE^{-1}A} E^{-1} dt \\ &= \int_0^\infty e^{t(AE^{-1})^T} (E^{-T}C^TC - PBB^TP) e^{tAE^{-1}} dt \end{aligned}$$

□

The reader should note, that the above results are of theoretical nature. In practice, computing E^{-1} should be avoided at all cost. Instead, alternative approaches, such as a change of variables, or successive solution of linear systems should be implemented.

In some problems, it is more natural to work with weighted norms on $\mathbb{R}^n$. For example, in many PDE systems the natural state spaces are the Hilbert spaces L^2 or H^1. Thus, when the system (3.2)-(3.3) arises from a standard finite element spatial discretization of a partial differential equation system, the matrix E is symmetric positive definite and the E-weighted norm of a vector equals the L^2 norm of the corresponding finite element function. In general, for a symmetric positive definite matrix $W \in \mathbb{R}^{n\times n}$, let $(x, y)_W = y^TWx$ denote the W-weighted inner product and let $\|x\|_W = (x, x)_W^{1/2} = (x^TWx)^{1/2}$ denote the W-weighted norm.

Remark 3.2.5. An alternative approach to deal with the matrix E that is used in many works is to perform the following change of variables. Since E is positive definite, a Cholesky decomposition $E = LL^T$ exists and one can transform the system to

$$\dot{z}(t) = \hat{A}z(t) + \hat{B}u(t), \quad y(t) = \hat{C}z(t), \tag{3.8}$$

where

$$z(t) = L^Tx(t), \quad \hat{A} = L^{-1}AL^{-T}, \quad \hat{B} = L^{-1}B, \quad \hat{C} = CL^{-T}. \tag{3.9}$$

Of course, if E is the identity matrix, then the new system is identical to the original system. For large-scale systems, the transformed matrices $(L^{-1}C)^T$ and $L^{-1}B$ can be formed explicitly using linear solves. However, the matrix $\hat{A} = L^{-1}AL^{-T}$ may or may not be computed explicitly, depending on the structure of the problem. With a standard Cholesky factorization of E, sparsity of A is lost, which is undesirable for the methods consider in this work. However, with a reordering, a minimal fill in Cholesky factor can be computed.[2] Many methods require that sparse linear solves can be computed efficiently. In that case, we never need the full matrix $\hat{A}$, but only its action on a vector z. Therefore, one can compute $\hat{A}z = L^{-1}(A(L^{-T}z))$ which only requires sparse solves and sparse matrix vector multiplications.

3.3 Direct Projection Methods

Projection methods have been demonstrated to be efficient methods to compute solutions of Riccati and Lyapunov equations [106, 109, 110, 111, 166, 97, 168], and can be divided into two steps. The first step is the computation of the approximate solution of the ARE via a specific algorithm. The second step is a computation of a matrix residual norm (or other criteria) to test the accuracy of the approximate solution obtained in the first step. Below, we give an overview of the projection approach, discuss the choice of a projection matrix, and review residual norms and approximation errors.

3.3.1 Projection Framework for ARE

For a symmetric positive definite weight matrix $W \in \mathbb{R}^{n\times n}$, assume we have a matrix V_r with full column rank such that

$$V_r = [v_1,\ v_2, \dots, v_r] \in \mathbb{R}^{n\times r}, \qquad V_r^T W V_r = I_r, \tag{3.10}$$

where $v_i \in \mathbb{R}^n$ for each i and $r \ll n$ is the reduced order dimension. The matrix V_r is called a *projection matrix*. In analogy to the projection based model reduction in §1.6, but with slightly different notation, one obtains a reduced order model (E_r, A_r, B_r, C_r) of the system (E, A, B, C) via

$$E_r = V_r^T E V_r, \qquad A_r = V_r^T A V_r, \qquad B_r = V_r^T B, \qquad C_r = C V_r. \tag{3.11}$$

[2] In `Matlab`, the command `[L,p,S]=chol(E,'lower','matrix')` computes the minimal fill in Cholesky factor, so that $E = SLL^TS^T$, where S is a permutation matrix.

The reduced order matrices give rise to solving the projected ARE

$$A_r^T \Pi_r E_r + E_r^T \Pi_r A_r - E_r^T \Pi_r B_r B_r^T \Pi_r E_r + C_r^T C_r = 0 \quad \in \mathbb{R}^{r\times r}, \tag{3.12}$$

which can alternatively be obtained by imposing a Galerkin condition on the residual matrix. Assuming the low order ARE is well posed, the solution $\Pi_r \in \mathbb{R}^{r\times r}$ can be computed using the well developed methods for moderate sized AREs; e.g., the direct solver `care` in `Matlab`. Having solved the low order ARE (3.12), one defines a low rank approximate solution to the large-scale ARE (3.1) as

$$P_r := V_r \Pi_r V_r^T \approx P. \tag{3.13}$$

Projection methods by definition yield low rank factored solutions. As noted earlier, the solution to the low rank algebraic Riccati equation Π_r is symmetric positive semidefinite. Thus, the eigenvalue decomposition $\Pi_r = U_r S_r U_r^T$ is used to define the low rank factor

$$Z_r = V_r U_r S_r^{1/2}, \tag{3.14}$$

which in turn gives $P_r = Z_r Z_r^T$. In practice, only the low rank factors are stored and used where the solution P_r would be needed.

Steps (3.10)-(3.14) are common to all Galerkin projection methods. The distinctive feature of a method is the generation of the projection matrix V_r. As we see later, if the columns of V_r are in the span of the solution to the Lypunov equation (3.4) then the projection based method for solving an ARE can be very accurate. A discussion about measures of accuracy of approximate solutions to ARE is given in §3.3.3 below. Recall from Lemma 1.6.1, that stability of A under projection is preserved under fairly simple conditions.

3.3.2 The Choice of the Projection Matrix

Below, we give a brief overview of available methods for computing V_r, as they distinguish the various projection methods. In the next chapter, we propose a new POD based algorithm to compute the projection matrix.

Regular Krylov Methods

Direct projection methods for solving a Lyapunov equation of the form (3.4) with $E = I$ were first considered in [158] and variations, including a GMRES method on the matrix residual can be found in [106]. Recall, that the explicit solution (3.6) with $E = I$ to the Lyapunov equation contains $e^{tA^T}C^T$ in the integrand. The idea is that a good approximation of the integrand combined with a suitable quadrature rule should be sufficient for convergence of the gramian. Therefore, the authors in [158] used the standard Krylov subspace

$$\mathcal{K}_r(A^T, C^T) := \text{span}\{C^T, A^T C^T, \ldots, (A^T)^{r-1}C^T\} \tag{3.15}$$

to achieve $e^{A^T}C^T \approx p_{r-1}(A^T)C^T$, where $p_{r-1}(A^T)$ is a matrix polynomial of maximum degree $r-1$. Note, that this does not necessarily imply that the matrix exponential is well approximated by the matrix polynomial, but only imposes a lower bound of the form

$$\|e^{A^T}C^T - p_{r-1}(A^T)C^T\|_2 \leq \|e^{A^T} - p_{r-1}(A^T)\|_2 \|C^T\|_2.$$

More sophisticated quadrature rules are considered in [90], where it was shown that under rather mild assumptions a quadrature with ‘sinc’ quadrature points and appropriate weights yields good low rank approximations of X. It has been widely noted that the projection matrix V_r has to be of high rank for those methods to converge.

Extended Krylov Subspace Method

We next describe an improved Krylov subspace method which is widely used for solving large AREs and Lyapunov equations. Thus, consider the extended Krylov space

$$\mathbb{K}_r(A^T, C^T) := \mathcal{K}_{r/2}(A^T, C^T) + \mathcal{K}_{r/2}(A^{-T}, C^T), \tag{3.16}$$

with $E = I$ and $W = I$, and $A^{-T} := (A^{-1})^T$. For notational convenience, we assume that r is an even integer. In [78], it was shown that the enriched Krylov subspaces yield more accurate approximations for $p_{r-1}(A^T)C^T$ than the standard Krylov space $\mathcal{K}_r(A^T, C^T)$ in (3.15). This result has been exploited in the design of an iterative method for the solution of the Lyapunov equation, see [166]. The proposed Extended Krylov Subspace Method (EKSM) was found to outperform other methods in terms of CPU-time and memory requirements.

The direct projection method using the extended Krylov subspace above has been used to solve Riccati equations in [97]. For certain examples, the projection based EKSM was found to be better in terms of CPU-time and memory requirement than the Cholesky factorized-ADI method. A block Arnoldi algorithm was employed to generate V_r from the matrices A^T and C^T. Recall, that the Arnoldi iteration yields

$$A^T V_r = V_r H_r + T_r, \qquad \text{where} \qquad V_r, T_r \in \mathbb{R}^{n\times r},\ H_r \in \mathbb{R}^{r\times r}. \tag{3.17}$$

The matrix H_r is then upper Hessenberg and the residual T_r is orthogonal to the columns of V_r, so $V_r^T T_r = 0$, for $r = 1, 2, \ldots, n$. Additionally, the columns of V_r are mutually orthonormal, so $V_r^T V_r = I_r$. The extended block Arnoldi procedure is summarized in Algorithm 1. Note, that this algorithm is built for multi-input multi-output (MIMO) systems where $m, p > 1$. In every step of the iteration, a "thin" QR-decomposition [89, page 230] has to be computed. Essentially, this means that only a few n-dimensional vectors have to be orthonormalized via a modified Gram-Schmidt procedure. In `Matlab` one should call `qr(F,0)` for the thin QR decomposition of a tall matrix F.

Algorithm 1 : Extended Block Arnoldi (EBA) Algorithm ([97])

```
Input: $A^T \in \mathbb{R}^{n\times n}$, $C^T \in \mathbb{R}^{n\times p}$ and an integer $r$.
Output: $V_r \in \mathbb{R}^{n\times r}$, an orthogonal projection matrix.
Compute the QR-decomposition of $[C^T, A^{-T}C^T]$, i.e. $[C^T, A^{-T}C^T] = V_1\Lambda$.
Set $\mathcal{V}_0 = \{\ \}$.
for $j = 1, 2, \ldots, r$ do
    Set $V_j^{(1)}$: first $p$ columns of $V_j$ and $V_j^{(2)}$: second $p$ columns of $V_j$.
    $\mathcal{V}_j = [\mathcal{V}_{j-1}, V_j]$; $\hat{V}_{j+1} = [A^T V_j^{(1)}, A^{-T} V_j^{(2)}]$.
    Orthogonalize $\hat{V}_{j+1}$ with respect to $\mathcal{V}_j$ to get $V_{j+1}$, i.e.,
    for $i = 1, 2, \ldots, j$ do
        $H_{i,j} = V_i^T \hat{V}_{j+1}$ ;
        $\hat{V}_{j+1} = \hat{V}_{j+1} - V_i H_{i,j}$;
    end for
    Compute the QR-decomposition of $\hat{V}_{j+1}$, i.e. $\hat{V}_{j+1} = V_{j+1} H_{j+1,j}$.
end for
```

At every iteration of Algorithm 1, $2p$ new columns are added to V_r. The inversion is implemented by precomputing the decomposition $A = LU$, where L, U are lower and upper triangular matrices, respectively. Hence, the inversion only requires solving triangular systems successively. Other options, including iterative methods and preconditioning, can be used for inverting A. In [97], Algorithm 1 is used within the general projection framework outlined above. The authors constructed a computationally cheap evaluation of the Riccati equation matrix residual, which serves as a stopping criterion for the algorithm. To achieve this, the Arnoldi recurrence turned out to be crucial. With this step, only matrices of size r are required to compute the stopping criterion.

When the matrix E is not the identity, there are (at least) two approaches to modifying the above algorithm. The first approach is based on a change of variables, as mentioned in Remark 3.2.5, and equation (3.9). Using this transformation in the ARE (3.1) leads to the transformed ARE

$$\hat{A}^T\hat{P} + \hat{P}\hat{A} - \hat{P}\hat{B}\hat{B}^T\hat{P} + \hat{C}^T\hat{C} = 0.$$

The above extended Krylov approach can be applied to obtain an approximate solution $\hat{P}_r$ of the transformed ARE, and an approximation $P_r = L^{-T}\hat{P}_rL^{-1}$ to the solution P of the original ARE (3.1) can be recovered by inverting the change of variables.

In an implementation, the matrix $\hat{A}$ is never explicitly formed; instead applications of $\hat{A}$ and $\hat{A}^{-1}$ to a matrix or vector can be accomplished by a sequence of matrix multiplications and linear solves. It may also be possible to implement this approach implicitly, i.e., the change of variables is only performed at the last stage of the computation (c.f. [70, §6.6.5]).

As Benner and Saak note in [32, page 37], this change of variables approach is required in order to obtain a fast computation of the residual. Also, if E is not symmetric positive definite, an LU decomposition of E can be used for the change of variables (see [32, page 37]). Furthermore, note that if A is symmetric, then the symmetry is retained in the changed variables.

The second approach does not require symmetry of E and simply replaces (A^T, C^T) with $(E^{-T}A^T, E^{-T}C^T)$ in the extended Krylov approach. Again, the matrix $E^{-T}A^T$ and its inverse can be applied to another matrix or vector by a sequence of matrix multiplications and linear solves.

We can also enforce orthogonality with respect to a weighting matrix W with the following modifications of Algorithm 1:

- Replace the QR decomposition in steps 1 and 10 with a W-weighted stabilized Gram-Schmidt procedure. (See §3.4 for details.)
- Replace step 7 by the W-weighted version $H_{i,j} = V_i^TW\hat{V}_{j+1}$.

In the second approach, it appears that one loses the fast computation of the residual (again, see [32, page 37]); instead, the (slower) general purpose QR algorithm for the residual discussed in §3.3.3 below can be used.

The authors are not aware of a thorough comparison of these two approaches in the literature. We tested both approaches on the convection diffusion PDE example discussed in §3.5. We found that the second approach gave much greater accuracy when approximating the solution P of the ARE (3.1) in the original variables.

Remark 3.3.1. The presence of the matrices $(E^{-T}A^T, E^{-T}C^T)$ in the second approach can be motivated as follows. First, recall when $E = I$, the matrices (A^T, C^T) correspond to the presence of $e^{tA^T}C^T$ in the solution representation of the associated Lyapunov equation (see Theorem 3.2.1). When $E \neq I$, the matrices $(E^{-T}A^T, E^{-T}C^T)$ now correspond to the solution representation of the associated Lyapunov equation in Corollary 3.2.2.

Gramian Based Projection

The relationship between the ARE (3.1) and the Lyapunov equation (3.4), motivates the following projection matrix. The solution of the Lyapunov equation can be computed with `lyap` in `Matlab` as

$$X = \texttt{lyap}(A^T, C^T C, [\,], E^T).$$

`Matlab` uses the SLICOT SB03MD routine for this problem. At first, the Schur decomposition of A^T is computed and then the new system solved by forward substitution. The algorithm is backward stable and requires $O(n^3)$ operations, therefore becoming unfeasible for large n. We next compute the singular value decomposition of the observability Gramian

$$X = V\Sigma W^T,$$

where the columns of $V = [v_1\ v_2\ \ldots v_n]$ span the range space of X. Truncation of V after r columns yields the projection matrix as $V_r = [v_1\ v_2\ \ldots v_r]$. Finally, the projected Riccati equation (3.12) is solved. We include this method for testing on small problems since it is closely related to the POD projection method proposed in §3.4.

3.3.3 Residual Computations and Approximation Errors

In order to set stopping criteria or to compare various methods regarding accuracy, it is necessary to define a quality measure for approximate solutions. A relative residual is often used as a stopping criteria to select the rank of the approximate solution. Many existing approaches to solving the ARE (3.1) are linked with specialized algorithms that rapidly compute (or estimate) the residual. We do not attempt to review these algorithms here; see the survey paper [32] for some details and references. Below, we present a QR algorithm from [29] to compute the residual that is applicable for many low rank ARE solution methods. We use this approach for the computations in this chapter. The method is not as computationally cheap as the specialized algorithms mentioned above; however, it is still computationally tractable and scales to large problems. We also discuss the relationship of the residual to the actual approximation error.

Let $P_r \in \mathbb{R}^{n\times n}$ be a symmetric positive semidefinite approximate solution of the ARE (3.1). The residual is defined as

$$\mathcal{R}(P_r) := A^T P_r E + E^T P_r A - E^T P_r B B^T P_r E + C^T C. \tag{3.18}$$

We assume P_r has a low rank factorization $P_r = Z_r Z_r^T$, where $Z_r \in \mathbb{R}^{n\times r}$ and $r \ll n$. Then the residual can be rewritten as

$$\mathcal{R}(P_r) = FGF^T, \quad F = \begin{bmatrix} C^T & A^T Z_r & E^T Z_r \end{bmatrix}, \quad G = \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & I \\ 0 & I & -(Z_r B)(Z_r B)^T \end{bmatrix}.$$

We have $F \in \mathbb{R}^{n\times(2r+p)}$ and $G \in \mathbb{R}^{(2r+p)\times(2r+p)}$, where $C \in \mathbb{R}^{p\times n}$. Let $F = QR$ be the thin QR decomposition of F, see [89, page 230]. Since $Q^T Q = I$, we have

$$\|\mathcal{R}(P_r)\|_2 = \|RGR^T\|_2, \quad \|\mathcal{R}(P_r)\|_F = \|RGR^T\|_F.$$

Therefore, computing the norm of the residual matrix $\mathcal{R}(P_r)$ (e.g., of size $n \gtrsim 10^5$) can be reduced to the norm computation of a small square matrix (e.g., of size $2r + p \approx O(10)$ or $O(100)$) using a thin QR decomposition.

Next, we review the connection between the residual and the approximation error. Let P_r be any approximation to the exact solution of the ARE (3.1). Ideally, one should use the actual approximate error

$$\varepsilon = \|P - P_r\|, \tag{3.19}$$

to assess the accuracy of the approximate solution, where $\|\cdot\|$ either denotes the matrix 2-norm or some other matrix norm. Of course, the exact solution is almost never available and, as mentioned above, the residual is often used instead. To link the residual norm to the error in the solution, the authors in [119, 175] gave error bounds relating (3.19) to the residual when $E = I$. We state these theorems below. First, define the linear operator $\Omega_{P_r} : \mathbb{R}^{n\times n} \mapsto \mathbb{R}^{n\times n}$ by

$$\Omega_{P_r}(X) := (A - BB^T P_r)^T X + X(A - BB^T P_r).$$

This operator has the structure of the "Lyapunov operator" in equation (3.4). Moreover, if $(A - BB^T P_r)$ is stable, then Theorem 3.2.1 implies that Ω_{P_r} is invertible (i.e. the Lyapunov equation has a unique solution), which is crucial for proving the next theorem.

Theorem 3.3.2. *([119, Theorem 2'']) Let $P_r \geq 0$ be an approximation to the unique symmetric positive semidefinite solution P to the ARE (3.1). If $(A - BB^T P_r)$ is stable and*

$$\|P - P_r\|_2 \leq \frac{1}{3\|\Omega_{P_r}^{-1}\|_2 \|B\|_2^2} \qquad \text{and} \qquad 4\|\Omega_{P_r}^{-1}\|_2^2 \; \|\mathcal{R}(P_r)\|_2 \|B\|_2^2 < 1,$$

then

$$\|P - P_r\|_2 \leq \frac{2\|\Omega_{P_r}^{-1}\|_2 \|\mathcal{R}(P_r)\|_2}{1 + \sqrt{1 - 4\|\Omega_{P_r}^{-1}\|_2^2 \; \|\mathcal{R}(P_r)\|_2 \|B\|_2^2}} \leq 2\|\Omega_{P_r}^{-1}\|_2 \|\mathcal{R}(P_r)\|_2.$$

Rewriting the above result yields

$$\|P - P_r\|_2 \leq c(P_r)\|\mathcal{R}(P_r)\|_2,$$

where $c(P_r) = 2\|\Omega_{P_r}^{-1}\|_2$ depends on the approximate solution P_r. Theoretically, this inequality bounds the error in the approximate solution by the matrix residual. The term $c(P_r)$ can be viewed as a condition number of the algebraic Riccati equation. However, computing the quantity $\|\Omega_{P_r}^{-1}\|_2 = \sup_{X\neq 0}(\|\Omega_{P_r}^{-1}(X)\|/\|X\|)$ is not straightforward and only computable upper bounds were stated in [119, §3]. Furthermore, Sun [175] sharpened the above error bound while simultaneously relaxing the assumptions. A statement of those results is beyond the scope of this paper. For the extended Krylov approach discussed above, a similar error bound is given in [97, Theorem 3.2].

3.4 POD Projection Method

We propose using POD in the projection framework to approximate solutions to AREs in an accurate and computationally efficient manner. For a review of POD, refer to §1.6.2 in the background material. First, a POD method is employed to approximate the dominant eigenvectors of the observability gramian X solving the Lyapunov equation (3.4). Those vectors are used to construct a projection matrix V_r.

Willcox and Peraire [189] proposed a snapshot based approach to approximate solutions of the Lypunov equation in n dimensions (see also [158, 155]). In particular, they suggested using snapshots of simulations of linear systems to compute the observability gramian (3.6). In [169], this idea was extended to infinite dimensional Lyapunov equations and a rigorous convergence theory was presented. Specifically, error bounds and convergence of the low rank, finite dimensional solution to the infinite dimensional gramian were obtained. We follow this approach and first compute an approximation of the observability gramian X and subsequently project the ARE using the approximate dominant left singular vectors of X. The dimension of the required singular value decomposition is limited by $\min(n, ps)$, with s being the number of snapshots collected during the simulation of a linear system.

In control and systems theory, the dual equations of the underlying optimal control problem are

$$E^T \dot{z}_i(t) = A^T z_i(t), \tag{3.20}$$

$$E^T z_i(0) = c_i^T, \tag{3.21}$$

for all $i = 1, \ldots, p$, where $C^T = [c_1^T, c_2^T, \ldots, c_p^T]$. By the theory of ordinary differential equations, the unique solution to (3.20)-(3.21) is given by $z_i(t) = e^{tE^{-T}A^T} E^{-T} c_i^T$, for $i = 1, \ldots, p$. Due to the representation of the Lyapunov solution in Corollary 3.2.2, the authors in [189] thus suggested to use simulations of the dual equations to approximate the solution of the observability gramian X.

When $E^{-1}A$ is stable, the gramian can be rewritten as follows:

$$\begin{aligned} X &= \int_0^\infty e^{tE^{-T}A^T} E^{-T} C^T C E^{-1} e^{tAE^{-1}} dt \\ &= \int_0^\infty e^{tE^{-T}A^T} E^{-T} [c_1^T, \ldots, c_p^T] \begin{bmatrix} c_1 \\ \vdots \\ c_p \end{bmatrix} E^{-1} e^{tAE^{-1}} dt \\ &= \int_0^\infty e^{tE^{-T}A^T} E^{-T} [c_1^T, \ldots, c_p^T] \left(e^{tE^{-T}A^T} E^{-T} [c_1^T, \ldots, c_p^T] \right)^T dt \\ &= \int_0^\infty [z_1(t), \ldots, z_p(t)][z_1(t), \ldots, z_p(t)]^T dt \\ &= \int_0^\infty [Z(t)][Z(t)]^T dt. \end{aligned}$$

We approximate the observability gramian by the finite time integral

$$X \approx X_T := \int_0^T [Z(t)][Z(t)]^T dt,$$

where T specifies a final time, chosen so that a good approximation of the infinite integral is obtained. The finite time integral can be approximated by quadrature, such that

$$X \approx X_{T,\delta} := \sum_{i=1}^s \delta_i [Z(t_i)][Z(t_i)]^T, \tag{3.22}$$

using positive weights δ_i and a time grid $0 = t_1 < t_2 < \ldots < t_s = T$. Here, let $s_i = t_{i+1} - t_i$ be the step size for the numerical integration scheme at the i^{th} time step. In matrix form, the approximation reads as

$$X \approx X_{T,\delta} := Z \Delta Z^T,$$

where Δ is the diagonal matrix of weights and Z contains snapshots of simulations, as outlined below. The method of snapshots [174] is used for the POD computations, as briefly reviewed in §1.6.2. We refer the reader to, e.g., [186, Chapters 2–3] for more detail. We note that the POD computations can be performed using other approaches; see, e.g., [81, 24]. Some technical details for the implementation of the POD snapshot based approach to approximate the gramian are listed below.

- For accurate simulation of the dual system (3.20)-(3.21), a proper set of time steps has to be chosen a priori, or adaptively during the time stepping. In this work, we simulated the test problems with Matlab's adaptive time stepping solvers `ode45` and `ode23s`, with default absolute and relative error tolerances. In most cases, the snapshots are selected at the time steps chosen by the adaptive solver.

- In case of highly stiff problems, the time steps s_i are small, which results in a larger set of snapshots than is needed for computation of V_r. In this case, a subset of snapshots from the previous step is selected, and the singular value decomposition computed from this smaller set of vectors. In practice, we found this approach to not lose significant accuracy, compared to keeping all time snapshots.

- The final time T is the only parameter that needs to be fixed for the POD based approach. One possible approach is to choose T so that the norms of each solution $z_i(t)$ are below a certain tolerance. (Solutions of exponentially stable systems must tend to zero for large enough time.) For certain simulation codes, it is possible to choose the tolerance and the simulation will determine the time T. Also, we note that different final times can be taken for each simulation, but for simplicity we use one final time T.

- In this work, the weights for the approximation of the integral are chosen by the trapezoidal rule, which yielded high accuracy in the projection framework as demonstrated in §3.5.

The remainder of this section focuses on the construction of the projection matrix V_r so that it is orthogonal with respect to a W-weighted inner product, i.e., $V_r^T W V_r = I$, where W is a symmetric positive definite matrix. For a given initial condition $E^{-T}c_i^T$, simulate system (3.20)-(3.21) and assemble the time snapshots in a matrix

$$Z_i = [z_i(t_1)\ z_i(t_2) \dots z_i(T)] \in \mathbb{R}^{n \times s}.$$

Simulations starting with every column of $E^{-T}C^T$ are concatenated in the matrix

$$Z = [Z_1,\ Z_2, \dots, Z_p] \in \mathbb{R}^{n \times ps}.$$

Further, the approximate observability gramian in the new variables can be factored as

$$X_{T,\delta} = YY^T, \quad Y = Z\Delta^{1/2} \in \mathbb{R}^{n \times ps}.$$

However, we refrain from ever forming the gramian explicitly. The projection method only requires the W-orthogonal eigenvectors of $X_{T,\delta}$ to construct V_r, so there is no need to form the approximate gramian explicitly, and we can work with the factor Y instead.

For large systems in which the state space dimension exceeds the number of snapshots, the well known method of snapshots to compute the eigenvectors proceeds as follows. First, compute the eigenvalue decomposition

$$Y^TWY = \Phi\Lambda\Phi^T \quad \in \mathbb{R}^{ps\times ps}, \tag{3.23}$$

and rescale (if necessary) the eigenvectors so that they are orthonormal with respect to the standard inner product, i.e., $\Phi^T\Phi = I$. Here, $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots)$, and the eigenvalues are ordered so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$. The projection matrix V_r is given by the matrix consisting of the first r eigenvectors of $X_{T,\delta}$, which is given by

$$V_r = Y\Phi_r\Lambda_r^{-1/2}, \tag{3.24}$$

where $\Phi_r \in \mathbb{R}^{n\times r}$ denotes the matrix consisting of the first r columns of Φ, and $\Lambda_r = \mathrm{diag}(\lambda_1, \ldots, \lambda_r)$. The procedure is summarized in pseudocode in Algorithm 2. Note that the loop computation can be executed in parallel.

Algorithm 2 : POD method to compute projection matrix

```
Input: $E, A, W \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m},\ C \in \mathbb{R}^{p\times n}$, final time $T$, maximal order $r_{max}$.
Output: Projection matrix $V_{r_{max}}$.
for i = 1, 2, . . . , p do
    Simulate $E^T\dot{z}_i = A^Tz_i,\ E^Tz_i(0) = c_i^T$ and place time snapshots of solutions in the matrices $Z_i = [z_i(t_1^i), z_i(t_2^i), \ldots, z_i(t_{s_i}^i = T)]$.
    Compute quadrature weights (see (3.22)), and let $\Delta_i = \mathrm{diag}(\delta_1^i, \delta_2^i, \ldots, \delta_s^i)$.
end for
$Y = [Z_1, \ldots, Z_p] \cdot \mathrm{diag}(\Delta_1, \ldots, \Delta_p)^{1/2}$.
Compute the singular value decomposition $Y^TWY = \Phi\Lambda\Phi^T$ .
$V_{r_{max}} = Y\Phi_{:,1:r_{max}}\Lambda_{1:r_{max},1:r_{max}}^{-1/2}$.
```

In exact arithmetic, the projection matrix V_r is orthonormal with respect to the W-weighted inner product whenever $\lambda_r > 0$. This can be seen by direct computation: since $\Phi_r^T\Phi_r = I$, we have

$$V_r^TWV_r = \Lambda_r^{-1/2}\Phi_r^T(Y^TWY)\Phi_r\Lambda_r^{-1/2} = \Lambda_r^{-1/2}\Phi_r^T(\Phi_r\Lambda_r\Phi_r^T)\Phi_r\Lambda_r^{-1/2} = I.$$

However, as is well known, in finite precision arithmetic accuracy can be lost due to forming the required matrix products [87, 157]. In the large-scale problem considered in §3.5 below, we found that $V_r^TWV_r$ began to deviate significantly from the identity matrix as r increased. To deal with the loss of W-orthogonality, we use a W-weighted stabilized Gram-Schmidt procedure with reorthogonalization to form a new W-orthonormal matrix V_r with the same span as the V_r matrix constructed above. To be complete, we present a simple implementation in Algorithm 3 below..

Algorithm 3 : W-Weighted Stabilized Gram-Schmidt with Reorthogonalization

Input: $V_r = [v_1, v_2, \ldots, v_r] \in \mathbb{R}^{n\times r},\ W \in \mathbb{R}^{n\times n}$.
Output: $V_r \in \mathbb{R}^{n\times r}$ such that span(V_r) remains unchanged and $V_r^T W V_r = I \in \mathbb{R}^{r\times r}$.

```
for $\ell = 1, 2$ (reorthogonalization) do
    for $i = 1, \ldots, r$ do
        $v_i = v_i/(v_i^T W v_i)^{1/2}$.
        for $j = i+1, \ldots, r$ do
            $v_j = v_j - (v_j^T W v_i)\, v_i$.
        end for
    end for
end for
```

3.4.1 Parallels to Standard and Rational Krylov Subspaces

Rational Krylov subspace techniques (and the related ADI iteration) have been demonstrated to be efficient for solving Lyapunov equations; see [79] and the recent survey [167]. Moreover, in [168] rational Krylov subspaces were compared with the EKSM as direct projection methods for the ARE and it was found that both methods show a similar convergence behavior. In this short section, we highlight some of the similarities and differences between the projection spaces obtained from the POD method and rational Krylov subspace methods. A thorough analysis is left for future work.

Recall, the integral representation of the solution of the Lyapunov equation in Corollary 3.2.2 motivated us to consider approximating the integral via quadrature using snapshots of the dual differential equation (3.20)-(3.21). Consider a simple variable step backward Euler scheme for time discretization with a time grid $t_{k+1} = t_k + s_k,\ k = 1, \ldots, n_s - 1$, where s_i are the time steps. Then, one has

$$E^T \frac{x_{k+1} - x_k}{s_k} \approx E^T \dot{x}_{k+1} = A^T x_{k+1}.$$

Rearranging terms, one has the simple recursion

$$x_{k+1} = \big(I - s_k E^{-T} A^T\big)^{-1} x_k = s_k^{-1}\big(s_k^{-1} I - E^{-T} A^T\big)^{-1} x_k.$$

The initial condition is given by $x_0 = E^{-T} C^T$, so that the following sequence of snapshots is collected:

$$\mathcal{S}_{BE} = \left\{ x_0,\ s_1^{-1}\big(s_1^{-1} I - E^{-T} A^T\big)^{-1} x_0, \ldots, \prod_{i=1}^{n_s-1} s_i^{-1}\big(s_i^{-1} I - E^{-T} A^T\big)^{-1} x_0 \right\}$$

of dimension n_s. Note that $\operatorname{span}(\mathcal{S}_{BE}) = \operatorname{span}(\mathcal{K}_{n_s}(E^{-T}A^T, x_0, \boldsymbol{s}^{-1}))$ is a rational Krylov subspace, and $\boldsymbol{s}^{-1} = [s_1^{-1}, s_2^{-1}, \ldots, s_{n_s-1}^{-1}]$ is the vector of inverted time steps.

If we arrange the columns of $\mathcal{S}_{BE}$ in a matrix X and take its singular value decomposition, then the left singular vectors form an orthonormal basis for the range of X. The POD approach only uses the r dominant left singular vectors to generate V_r, and therefore the span of V_r would only approximate the span of $\mathcal{S}_{BE}$. Alternatively, one can perform a block Arnoldi algorithm on the columns of $\mathcal{S}_{BE}$ to obtain V_r. Thus, for the same set of parameters $\boldsymbol{s}$, both the POD based approach as well as the rational Krylov subspace method yield similar results. In other words, the POD approach with a stable backward Euler scheme approximates a rational Krylov subspace.

However, there are significant differences between the two approaches in practice:

- One would likely use time stepping schemes with higher accuracy than backward Euler in the POD approach. Certain time stepping schemes might possibly lead to snapshot sets $\mathcal{S}_{BE}$ of a type not usually considered in standard rational Krylov subspace approaches.
- The POD approach generates the large "Krylov subspace" $\mathcal{S}_{BE}$ first and then builds V_r by extracting the dominant singular vectors of the matrix with the same columns as $\mathcal{S}_{BE}$. In contrast, a rational Krylov approach increases the rank of the Krylov subspace and V_r iteratively; a residual is used to stop updating the construction.

The last item shows that, in general, the POD approach requires more computational cost and storage. However, adaptive time stepping algorithms can often achieve high accuracy with a low number of snapshots. In Examples 2 and 3 in §3.5 the POD approach with $n_s \approx c \cdot r$, where $c \lesssim 5$, gave high accuracy. For instance, in Example 3, $n_s = 130$ and at order $r = 40$ the residual is approximately 10^{-13}. Plus, the shift selection is performed by adaptive time stepping routines, which yield accurate simulation results for linear systems. Thus, other than fixing a final time, $T = t_{n_s}$, there are no parameters needed for the POD based approach. Moreover, the matrices A, C are often not explicitly needed for simulations, and weak formulations of PDE's can be used in the spatial integration routine. Therefore, the POD based projection method has an additional level of flexibility and we plan to extend it to a completely matrix free algorithm.

It is interesting to note that with a forward Euler scheme we obtain another rational Krylov subspace. In fact, a forward Euler approximation yields the recurrence

$$x_{k+1} = (I + s_i E^{-T} A^T) x_k,$$

so that the collection of snapshots is

$$\mathcal{S}_{FE} = \left\{ x_0,\ (I + s_1 E^{-T} A^T) x_0, \ldots, \prod_{i=1}^{n_s - 1} (I + s_i E^{-T} A^T) x_0 \right\}.$$

One can easily see that

$$\operatorname{span}(\mathcal{S}_{FE}) = \operatorname{span} \left\{ x_0, (E^{-T} A^T) x_0, \ldots, (E^{-T} A^T)^{n_s - 1} x_0 \right\},$$

which is the span of the standard Krylov subspace $\mathcal{K}_{n_s}(E^{-T} A^T, x_0)$. Interestingly, forward Euler frequently requires small time steps for stability, so $\mathcal{S}_{FE}$ would be rather large. Analogously, large rational Krylov subspaces are needed for the projection framework to give accurate results.

3.5 Numerical Results

In this section, we present numerical results for four different test problems. The first and second are benchmark problems from structural dynamics, whereas third and fourth problems arise from spatial discretization of heat transfer and fluid dynamical problems. The latter two problems have symmetric positive definite mass matrices E.

Table 3.1 contains details and parameters of the test models. By $\lambda_{max}(A, E)$ we denote the (approximate) largest generalized eigenvalue of A. Also, $\omega(A, E)$ denotes the (approximate) generalized numerical abscissa of A, which is the largest generalized eigenvalue of $(1/2)(A + A^T)$. These quantities were computed in `Matlab` with `eigs(·,1,'LR')` for the generalized eigenvalue problems $Ax = \lambda Ex$ and $(A+A^T)x = \lambda Ex$, respectively.

Problem	n	m	p	$\lambda_{max}(A,E)$	$\omega(A,E)$	$A = A^T$?	$E = I$?
space station	270	3	3	$-4.8 \cdot 10^{-2}$	$+7.7 \cdot 10^{-9}$	no	yes
space station	1412	3	3	$-2.1 \cdot 10^{-3}$	$+9.9 \cdot 10^{-18}$	no	yes
diffusion	9900	1	1	$-6.2 \cdot 10^{-1}$	$-6.2 \cdot 10^{-1}$	yes	no
conv.-diff.	122150	1	1	$-1.6 \cdot 10^{-1}$	$+7.5 \cdot 10^{-1}$	no	no

Table 3.1: Parameters of the four test models for solution of ARE.

All of the examples are stable, however the values of the numerical abscissa indicate that the uncontrolled problems are quite different from each other. For the first two problems, half of the values $\omega(A,E)$ are on, or very close to the imaginary axis. The numerical abscissa indicates whether solutions of the system (3.2)-(3.3) with no control (i.e., $u(t) = 0$) can experience transient growth before decaying to zero; this transient growth is possible only if the numerical abscissa is positive; e.g., [178, §14 and 17]. Also see [16] for more about the numerical abscissa and related matrix Lyapunov equations.

Recall, the first step for solving ARE with projection methods consists of generating a projection matrix $V_r \in \mathbb{R}^{n\times r}$. Next, the reduced order matrices (E_r, A_r, B_r, C_r) are computed via projection and the Riccati equation (3.12) is solved in r dimensions with `care` in `Matlab`. Since the projection is constructed to be orthogonal in the weighted inner product $(\cdot,\cdot)_E$, one has that $E_r = I$. The direct solution routine solves the Hamiltonian eigenvalue problem associated with ARE and is further described in [10]. The low rank approximation of P is then given by $P_r = Z_r Z_r^T$. A general projection algorithm following the steps in §3.3 is used to compare the methods, see Algorithm 4. In all cases, following [166, 97] and the remark in [32, p.9], $tol = 10^{-12}$ was used for truncation of the singular values of the low rank solution Π_r. This additional rank reduction at the reduced order level targets numerical inaccuracies.

Remark 3.5.1. We also briefly tested setting $tol = 0$ in two versions of Example 3 below with smaller values of n (generated using coarser finite element meshes). Of course, there is no further rank reduction in this case. In these tests, the matrix residual for the POD approach (not shown here) levels off at order 10^{-15} instead of order 10^{-13} when we use the above tolerance. However, this increases the rank of P_r.

Algorithm 4 : Projection based Riccati solver with residual computation

Input: $E, A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m},\ C \in \mathbb{R}^{p\times n},\ V_r \in \mathbb{R}^{n\times r}$, a tolerance $tol > 0$.
Output: The low rank factor Z_l of $P_l = Z_l Z_l^T$, and residual norm vector $[\gamma_1, \dots, \gamma_r]$.

```
for $l = 1, 2, \dots r$ do
    Let $V_l = V_r(:, 1:l)$. Compute $E_l = V_l^T E V_l,\ A_l = V_l^T A V_l,\ B_l = V_l^T B,\ C_l = C V_l$.
    Solve $A_l^T \Pi_l E_l + E_l^T \Pi_l A_l - E_l^T \Pi_l B_l B_l^T \Pi_l E_l + C_l^T C_l = 0$.
    Compute (svd or evd): $\Pi_l = U \Sigma U^T$, where $\Sigma = \mathrm{diag}[\sigma_1, \dots, \sigma_l]$ and $\sigma_1 \geq \dots \geq \sigma_l$.
    Determine $k$ such that $\sigma_{k+1} < tol < \sigma_k$; set $\Sigma_k = \mathrm{diag}[\sigma_1, \dots, \sigma_k],\ U_k = U_{:,1:k}$ and compute $S_l = U_k \Sigma_k^{1/2}$.
    $Z_l = V_l S_l$, i.e., the low rank factor of $P_l = Z_l Z_l^T$.
    Compute $\gamma_l = \|\mathcal{R}(P_l)\|$ as in §3.3.3.
end for
```

Since the first problem is small, for comparison purposes we compute the solution P via a direct solver (`care`) and compute the actual error in the solution, $\|P - P_r\|$, for increasing r. Moreover, we examine the convergence of the feedback gains using the relative error $\|K - K_r\|/\|K\|$, where $K = B^T P E$ and $K_r = B^T P_r E$. For large problems, we consider the convergence behavior of the residual norm $\|\mathcal{R}(P_r)\|$ and the relative change in the feedback gains $\|K_r - K_{r-1}\|/\|K_r\|$. Note that the full matrix P_r is never explicitly used or stored, and we only work with the low rank factor Z_r from equation (3.14). In particular, the residual computation is performed as in §3.3.3 and the feedback gain matrices are evaluated as $K_r = B^T P_r E = (B^T Z_r)(Z_r^T E)$.

The first two problems were computed on a 2010 MacBook Pro with a 2.66 GHz Intel Core i7 Processor and 4GB RAM. `Matlab` was used as a software in the version of `R2012b`. The convection diffusion problem was solved on a computer cluster with two 6-core Intel Xeon X5680 CPU's at 3.33GHz. The cluster has a Random-Access Memory of 48GB and runs on Scientific Linux 6.4 with `Matlab` in the Version of 2013b. Machine precision on both machines is in accordance with IEEE double precision standard, $eps = 2.210^{-16}$.

3.5.1 Example 1: ISS1R Flex Model

This model describes a structural subsystem of the International Space Station (ISS) and is taken from [8]. During the assembly process of the ISS many international partners are involved and robust stability and performance of all stages has to be certified. Many flexible structures, operational modes, and control systems result in a complex dynamical system. For robustness and performance assessment, it is critically important to identify the potential for dynamic interaction between the flexible structure and the control systems. As more components are added to the space station, the original symmetry gets lost, which poses additional simulation challenges. The subsystem 1R is a flex model of the Russian Service Module. For simulation purposes, the goal is to obtain a reduced order model that matches the frequency spectrum of the structures well. Solving the algebraic Riccati equation for this model is important for model reduction, such as LQG balanced truncation, as well as optimal control design for the overall plant.

The system matrix A is non-symmetric and dense with 63843 nonzero elements, i.e. 88% fill-in. The mass matrix is the identity [3], $E = I$. The condition number of A is $1.0 \cdot 10^2$. For $r = 2, 4, \dots, 60$, we computed low rank Riccati solutions with each of the described methods and plotted the norm of the matrix residual in Figure 3.1. For the POD based projection method, the dual system was simulated from $t = 0$ to $T = 15s$ and snapshots were taken every $0.02s$. This amounts to a combined 2253 snapshots of the three simulations of the system. The residual norm for the solution P produced by the direct `care` solver is $1.8 \cdot 10^{-12}$. The reader should observe that not even the direct solver achieves machine precision for the residual norm.

Figure 3.1, left, shows the residual norm $\|\mathcal{R}(P_r)\|_F$ and the actual error in the solution $\|P - P_r\|_F$ for the EKSM, POD based method, and the gramian based approach. Computing the gramian with a direct solver and then projecting performs best in terms of accuracy. As mentioned earlier, this approach is only feasible for small scale problems. The POD based method shows a rather monotone convergence behavior of the residual norm, yet it is several orders of magnitude larger than the gramian based approach. The extended Krylov subspace method shows oscillations in the residual norm and has the largest residual norms of the three methods. A monotone decreasing error is desirable, for it guarantees that extra work for computing an

[3] The mass matrix of the benchmark model is the identity. The authors are not certain how this model was obtained, e.g. by finite element/difference or lumped parameter modeling. Therefore, it is not clear if the modeling introduced an identity mass matrix, or if the nontrivial mass matrix was inverted.

increased size reduced order model is not wasted. The reader should observe that in this example the residual is a good indicator of the accuracy of solutions. In general, this does not have to be true; see the discussion in [190, §5].

Figure 3.1, right, shows the convergence of the feedback gain matrices. The gains show a similar convergence behavior compared to the residual norm and solution error, since $K_r = B^T P_r E$. Note that the convergence in both plots is slow for the POD and EKSM method, which indicates the numerical rank of P is rather large. In fact, the rank of P is 267 (computed with `rank` in `Matlab`) and the decay of the singular values for determination of the numerical rank is plotted in Figure 3.3 and commented on below.

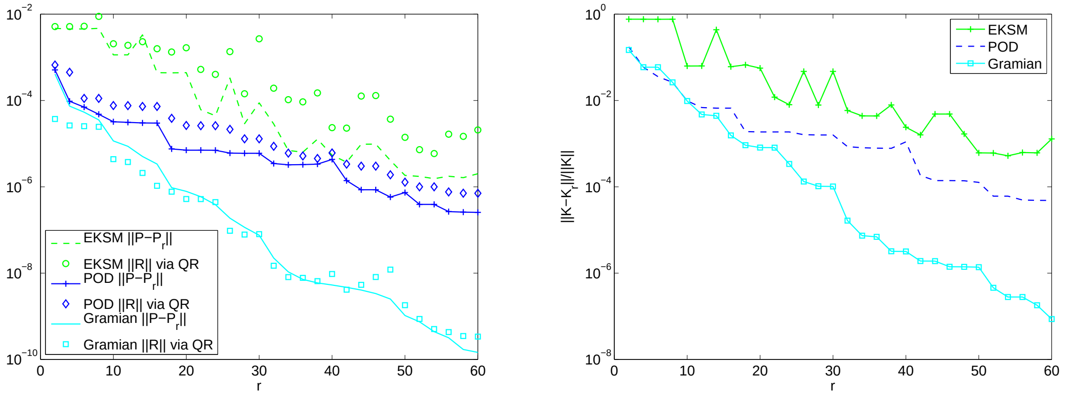

Figure 3.1: ISS1R model. (left): Residual norm $\|\mathcal{R}(P_r)\|_F$ and solution error $\|P - P_r\|_F$. (right): Convergence of feedback gains $\|K - K_r\|_2 \backslash \|K\|_2$.

Recall that the POD method approximates the observability gramian by a numerical quadrature. The number and location of the time samples t_j and weights δ_j is important for the quadrature to be accurate. Increasing both the final time T when terminating simulations and the number of snapshots should yield a more accurate approximation of the gramian. Figure 3.2, left, plots the effect of increasing the final time for simulation of the dual system. Terminating the simulations at $T = 5s$ does not yield enough data to approximate the gramian X properly and therefore the method does not perform well. The output of the `Matlab` simulations were 2184 snapshots. However, $T = 15s$ yields a significant improvement compared to the previous result and 4903 snapshots were generated. Lastly, simulating the system for $25s$ yields the richest projection subspace, generated from 6426 snapshots.

Note that only the singular value decomposition of $YY^T \in \mathbb{R}^{n\times n}$ is needed for the projection, apart from the multiplication, the cost of computing the SVD does not increase with additional snapshots. Figure 3.2, right, shows the effect of increasing the time samples, while fixing the final time at $T = 15s$. With the coarse snapshot collection of $0 : 0.1 : 15$ to compute V_r, the residual shows several jumps, and does not converge monotone. As we increase the snapshots, the residual convergence becomes monotone. Note, that the adaptive snapshot selection (by the solver), denoted by [015] performs almost identical to the equidistant snapshot selection, labeled as $0 : 0.01 : 15$.

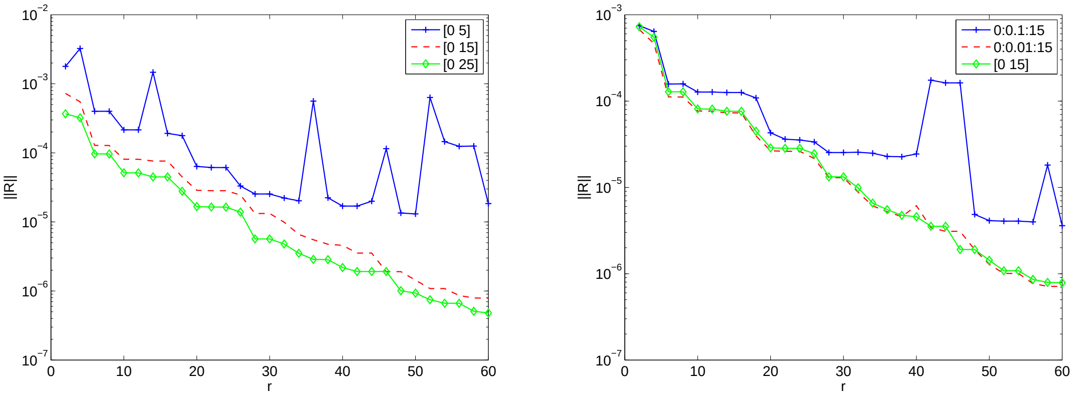

Figure 3.2: ISS1R model, effect on residual norm $\|\mathcal{R}(P_r)\|_F$ for the POD method when time grid is changed. (left): Increasing the final time T for approximation of the integral in equation (3.22). (right): Fixed final time $T = 15s$, and more time snapshots are taken.

As we have seen above, the gramian approach with the `lyap` solver as described in §3.3.2 performs very well. In other words, the left singular vectors of the observability gramian provide a rich subspace for solving ARE with a projection method. To further investigate this, Figure 3.3, left, shows a plot of the singular values of the Lyapunov and Riccati solution, obtained using `lyap` and `care`. The singular values of both solutions are close to each other and only separate at approximately 10^{-10}. This could explain the excellent behavior of the gramian based approach.

Figure 3.3, right plots the quotient of actual error in the approximation, $\|P - P_r\|_F$, and residual, $\|\mathcal{R}(P_r)\|_F$. From theoretical results in §3.3.3, it is known that $\|P - P_r\| \leq c \cdot \|\mathcal{R}(P_r)\|$, where $c = \|\Omega_{P_r}^{-1}\|$ depends on the current solution. From a theoretical perspective, it is sufficient for c to be bounded from above. However, when the residual is used to judge about the accuracy of a computation, the magnitude of the constant does play a crucial role. For instance,let $c = 1000$, so the actual error could be up to 1000 times higher than the residual would suggest. Therefore, we view the residual as an error "indicator" and are aware of the fact that the actual error can be nominally higher than the residual norm. It is demonstrated in [190, §5] that residuals can sometimes fail to be a good indicator of the accuracy of solutions for Lyapunov equations.

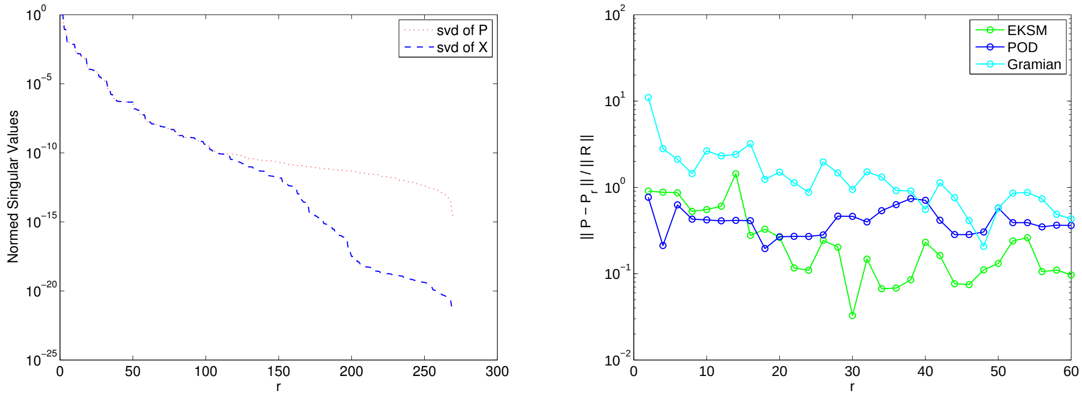

Figure 3.3: ISS1R model. (left): Normed singular values of the observability gramian X and Riccati solution P using direct solvers `lyap` and `care`. (right): The relation between residual and actual approximation error, $\|P - P_r\|_F / \|\mathcal{R}(P_r)\|_F$.

3.5.2 Example 2: ISS12A Flex Model

This model describes an advanced stage of the international space station. The matrix A is non-symmetric and has 2118 non-zero entries, so A is sparse. Moreover, the mass matrix is the identity, $E = I$.

For the POD based algorithm, the dual system was simulated for all three initial conditions c_i^T from $t = 0s$ to $T = 20s$ with a time stepping of $0.02s$. This amounts to a collection of $s = 3003$ snapshots of simulations. Since only the left singular vectors of Y with size 1412x3003 are needed, we instead compute the singular value decomposition of YY^T of size 1412x1412 for efficiency reasons. The direct computation of the Riccati solution took $65.3s$, whereas the gramian approach took $19.1s$, the POD method 4.9s and the EKSM only 0.4s.

The behavior of the norm of the residual is compared to the actual error in the solutions in Figure 3.4, left. This should give an idea how well the residual can inform the user about the actual convergence of the Riccati solution. All three projection based methods converge rather slowly, considering that the residual norm for the exact solution P with the direct `'care'` solver is $4.3 \cdot 10^{-12}$. The EKSM shows larger residuals, while both the POD and gramian method have two orders of magnitude smaller residual norms. Figure 3.4, right plots the quotient of actual error in the approximation, $\|P - P_r\|_F$, and residual, $\|\mathcal{R}(P_r)\|_F$. Recall, that from theoretical results in §3.3.3, it is known that $\|P - P_r\| \leq c\|\mathcal{R}(P_r)\|$, where $c = \|\Omega^{-1}\|$. However, the "constant" does depend on the current solution, so $c = c(P_r)$, and it is not at all obvious if the constant is well behaved.

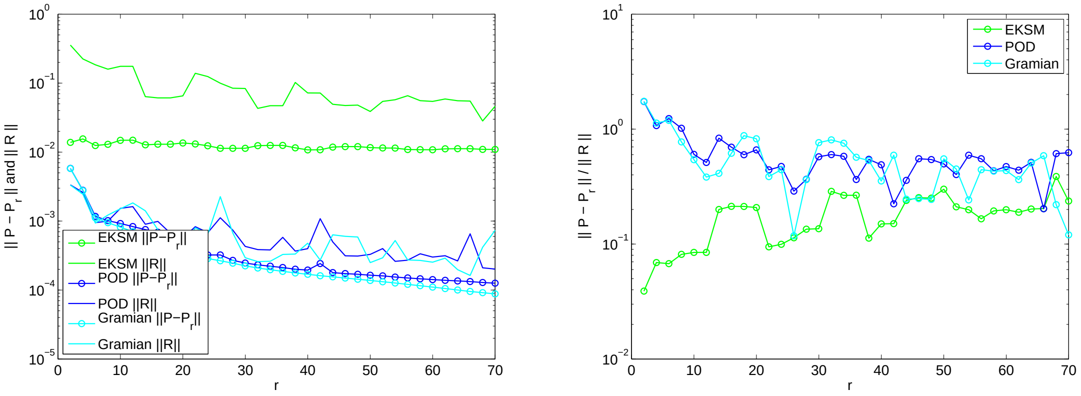

Figure 3.4: ISS12A model. (left): Residual norm $\|\mathcal{R}(P_r)\|_F$ and solution error $\|P - P_r\|_F$. (right): The quotient $\|P - P_r\|_F / \|\mathcal{R}(P_r)\|_F$ provides insight into the actual information the contains about the convergence to the true solution.

A plot of the singular values for the solution of the Riccati equation and the Lyapunov equation is given in Figure 3.5, left. The singular values of both solutions are once more very similar and only separate closer to machine precision. Consequently, the gramian approach performs very well.

At this point, we have a closer look into convergence of the gain matrix $K_r \to K$, which is important for control design. From a simple, yet crude, triangular inequality it can be seen that

$$\|K_r - K\|_2 \leq \|B\|_2 \|P_r - P\|_2,$$

where it becomes clear that for large $\|B\|_2$, those errors can be orders of magnitude apart. For the gains to converge, we need both the norm of the gains $||K_r||_2 \to const.$, as well as the difference $||K_{r,i-1} - K_{r,i}||_2 \to 0$. Alternatively, since we have the "true" gain K, it is sufficient to have $||K - K_r||_2 \mapsto 0$. For the POD and gramian approach, the *norm* of the functional gain matrix converged quickly, after $r = 4$ to the "true" norm of the optimal gain $2.22 \cdot 10^{-4}$ (computed via `care`). EKSM showed oscillations around that value, and converged only at $r = 16$. Figure 3.5, right, shows the convergence $\|K - K_r\|_2$ of the approximated gains, $K_r = B^T P_r$ to the gain K which is directly computed via `care`. The POD and gramian methods converge almost identically.

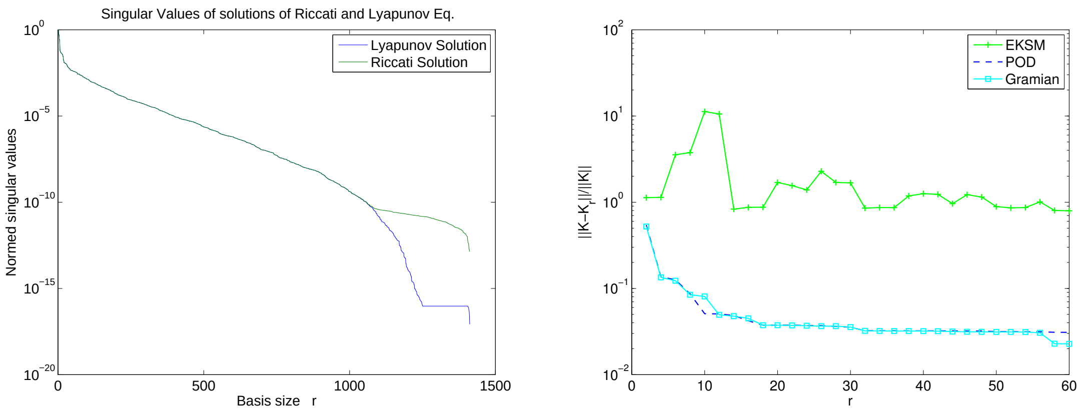

Figure 3.5: ISS12A model. (left): Singular values of the observability gramian X and Riccati solution P using `lyap` and `care`. (right): Convergence of the feedback gains.

3.5.3 A diffusion problem in 2D

We consider the diffusion problem in 2D:

$$w_t = \mu(w_{xx} + w_{yy}) + b(x, y)u(t)$$

on $\Omega = [0, 1] \times [0, 1]$ with Dirichlet boundary conditions on the bottom, right and top walls:

$$w(t, x, 0) = 0, \quad w(t, 1, y) = 0, \quad w(t, x, 1) = 0,$$

and Neumann boundary condition on the left wall:

$$w_x(t,0,y) = 0.$$

We choose $b(x, y) = 5$ if $x \geq 1/2$ and $b(x, y) = 0$ otherwise. The outputs are taken to be

$$\eta(t) = \int_\Omega 5w(t,x,y)dxdy.$$

After semi-discretization by a standard piecewise bilinear finite element method, a finite dimensional system with representation (E, A, B, C) is obtained, where $n = 9900$, $m = p = 1$. The diffusion parameter is $mu = 0.05$. In this example, the system matrix A is symmetric and has a condition number of $6.17 \cdot 10^3$. The dual system was simulated for $15s$ using `ode23s` in `Matlab` with default error tolerances and options `mass` and `jacobian` set as E and A^T, respectively. The norm of the final snapshot was $3.8 \cdot 10^{-4}$. The adaptive time stepping routine returned $n_s = 104$ snapshots of solutions.

As mentioned earlier, the E-weighted norm of a vector equals the L^2 norm of the corresponding finite element function. The L^2 norm of a function is a natural measure of magnitude for such a problem; therefore, we use the E-weighted inner product and norm for this example. Then, the projection matrix V_r (approximately) satisfies $V_r^T E V_r = I$, and so we simply use an identity matrix in place of the projected matrix $E_r = V_r^T E V_r$.

Figure 3.6, left, shows the residual norm of solutions to ARE, using the EKSM and POD based method, respectively. The residual norm for the POD method decreases quickly and then levels off at 10^{-13} when $r = 35$. As a point of comparison, at this projection order the residual norm for the extended Krylov space method is at the order of 10^{-9}. Figure 3.6, right, shows the relative refinement of the feedback gains with increasing projection order. A similar convergence history as previously observed before can be seen. Note, that the feedback gain vectors continue to refine past $r = 35$, although the residual matrix stagnates.

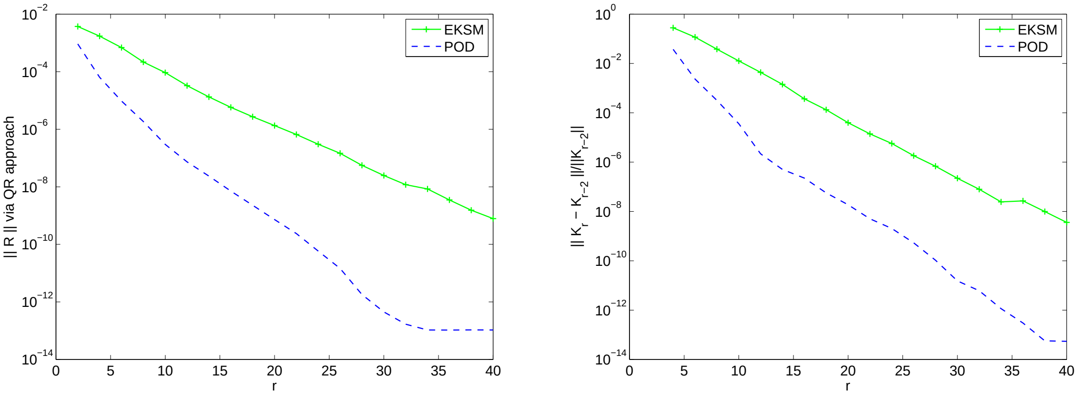

Figure 3.6: Diffusion model. (left): Residual norm $\|\mathcal{R}(P_r)\|_F$ as the rank of P_r increases. (right): Convergence of the feedback gain vectors for increasing r.

3.5.4 A convection diffusion problem in 2D

With this example, we consider a convection diffusion equation, which is a common partial differential equation model arising in fluid dynamics and a variety of other application areas. The problem is given by

$$w_t = \mu(w_{xx} + w_{yy}) - c_1(x,y)w_x - c_2(x,y)w_y + b(x,y)u(t),$$

where $c_1(x,y) = x\sin(2\pi x)\sin(\pi y)$ and $c_2(x,y) = y\sin(\pi x)\sin(2\pi y)$. The diffusion parameter is $\mu = 0.05$. The boundary conditions, the function $b(x,y)$, and the output are chosen as in the previous example. This time, the semi-discretization by a standard piecewise bilinear finite element method yields a non-symmetric A matrix with a condition number of $1.0 \cdot 10^6$. The state dimension is $n = 122150$, and $m = p = 1$. The dual system was simulated using `ode23s` in `Matlab` with default error tolerances and options `mass` and `jacobian` set as E and A^T, respectively.

Since the state space dimension is large for this example, the gramian approach as well as the computation of the Riccati solution with direct methods are not feasible on a standard desktop computer. In fact, it was not even feasible on our computer cluster with specifications given above. Note that for this model, the largest real part of the eigenvalues of A is approximately $-1.6 \cdot 10^{-6}$, so simulations were performed from $t = 0$ to $T = 50s$. The generalized eigenvalues are listed in Table 3.1 above. The norm of the final snapshot was $8.8 \cdot 10^{-3}$. The adaptive ODE solver returned $n_s = 130$

snapshots of the solution. The mass matrix E is well conditioned, $\text{cond}(E) \approx 14.1$, however the same is false for the matrix A since $\text{cond}(A) \approx 10^6$.

Figure 3.7, left, shows the residual norm of the ARE solutions obtained from both the POD based method and the extended Krylov subspace method. The residual norm for the POD method again decreases quickly and then levels out at order 10^{-13} when the projection order is $r = 40$. For comparison, at this projection order the residual norm for EKSM is at the order of 10^{-7}. Figure 3.7, right, shows the relative change in the gain vectors computed for every additional two basis vectors, i.e., $\|K_r - K_{r-2}\|_2/\|K_{r-2}\|_2$. When using $r = 50$ modes for the POD projection method, the relative change in the gains is approximately 10^{-13}. Again, the relative change in the gains shows similar trends to the matrix residual convergence.

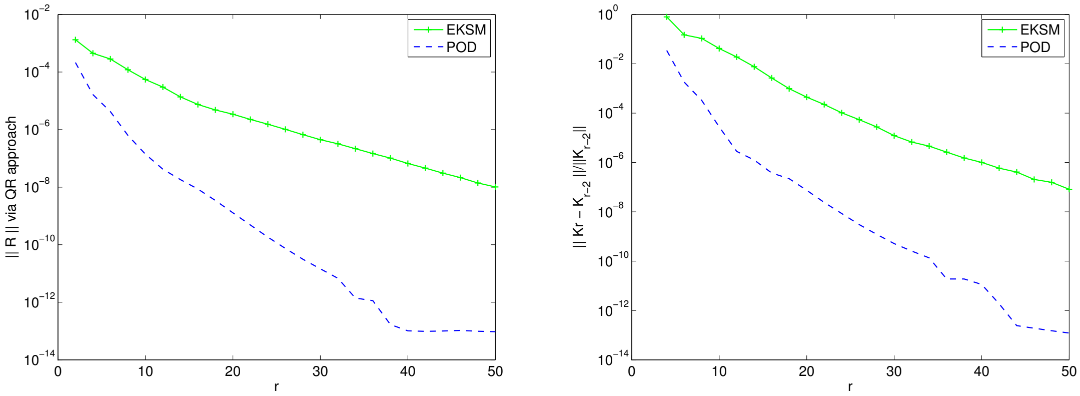

Figure 3.7: Convection diffusion model. (left): Residual norm $\|\mathcal{R}(P_r)\|_F$ as the rank of P_r increases. (right): Convergence of the feedback gain vectors for increasing r.

3.6 Conclusion

We presented a new POD based projection approach to compute solutions of algebraic Riccati equations. The method relies on proper orthogonal decomposition to compute an approximation of the solution to the related Lyapunov equation via the algorithm in [189, 169]. The resulting dominant left singular vectors are used in the projection framework to solve algebraic Riccati equations. Numerical results demonstrate that this POD basis is sufficiently rich for the projection approach to produce accurate solutions at low solution rank. Currently, no convergence theory is available for this method and this will be part of future work, since the numerical results are very promising.

To put our numerical results in perspective to state-of-the-art low rank ARE solution methods, we compared this POD projection method to the extended Krylov subspace method [97]. The POD approach will generally require more computational effort and storage compared to EKSM, ADI and other similar approaches. It was demonstrated that the POD projection approach can be efficiently computed and can give high accuracy at a low solution rank. Again, we emphasize that our primary goal is not to develop the most efficient solver, but to move toward an efficient, highly accurate, completely data-based algorithm. We also note that the proposed approach may be naturally implemented by many researchers in a variety of fields who are already familiar with POD computations.

In our numerical experiments, we used an adaptive ODE solver in `Matlab` to approximate the solutions of the required ordinary differential equations. Adaptive time stepping methods can be found in other existing simulation packages, such as IFISS [165]. The POD approach will often still be tractable for large-scale systems even if an adaptive ODE solver is not used; in the literature, POD computations are frequently performed using ODE solvers with a constant time step. The computational effort and required storage will usually be larger when the time steps are not chosen adaptively.

POD-based approaches can often be directly applied to systems governed by partial differential equations. For such problems, one can use existing simulation codes without accessing the matrix approximations of the operators; see, e.g., the discussion in [169]. A rigorous convergence theory for POD-based approximations to infinite dimensional Lyapunov solutions is available; see [169]. We believe the ideas in [169] coupled with the present approach will enable direct approximation of operator Riccati equations arising from partial differential equation systems. It may be possible to extend the existing theory to obtain rigorous error estimates for Riccati solutions.

It may also be possible to exploit the parallels between the POD method and rational Krylov subspace techniques (as discussed in §3.4.1) to obtain a thorough analysis of the method.

As mentioned in the introduction, for certain applications it is of interest to make the proposed approach completely data-based so that access to system matrices is not required. This can sometimes be done directly for AREs arising from parabolic partial differential equations (such as the second and third example problems considered in §3.5). Briefly, assume the simulation code for the parabolic PDE is based on a bilinear form as in finite element methods and other Galerkin methods. Then one only needs to be able to use the existing simulation code to evaluate the bilinear form acting on the relevant POD modes to project the A operator (see [170, §3.1]). Also, it may be possible to modify the techniques from [170] to make the proposed algorithm completely data-based when the bilinear form is not available or the ARE does not arise from a parabolic PDE. We intend to explore these issues in more detail elsewhere.

Chapter 4

Subspace Based System Identification

The Eigensystem Realization Algorithm (ERA) is a commonly used data-driven method for system identification and model reduction of dynamical systems. The main computational difficulty in ERA arises when the system under consideration has large number of inputs and outputs, thus requiring to compute a full SVD of a large-scale dense Hankel matrix. In this work, we present an algorithm that aims to resolve this computational bottleneck via tangential interpolation. This involves projecting the original impulse response sequence onto suitably chosen directions. Numerical examples demonstrate that the modified ERA algorithm with tangentially interpolated data produces accurate reduced order models. At the same time, computational cost and memory requirements compared to standard ERA are reduced significantly.

4.1 Introduction

Control of complex systems can be achieved by incorporating low dimensional surrogate models, which approximate the input-output behavior of the original system well. Where mathematical models are not at hand, data-driven techniques are used to approximate the system response to external inputs. The field of subspace based system identification (SI) provides powerful tools for fitting a linear time invariant (LTI) system to given input-output responses of the measured system. Applications of subspace based system identification arise in many engineering disciplines, such as in aircraft wing flutter assessment [101, 152], vibration analysis for bridges [72], structural health analysis for buildings [54], modeling of indoor-air behavior of energy efficient buildings [38], flow control [102, 136, 115, 187], seismic imaging [139] and many more. In all applications the identification of LTI systems was crucial for analysis and control of the plant. An overview of applications and methods for subspace based system identification can be found in [184] and more recently in [150, 153].

The eigensystem realization algorithm (ERA) by Kung [124] offers one solution to the system identification problem, while simultaneously involving a model reduction step. The algorithm uses discrete time impulse response data to construct reduced order models via a singular value decomposition. Importantly, the obtained reduced order LTI models retain stability, see §4.2.

Starting with Kung's work, many variants have been proposed in the literature. [139] gives a slight modification of Kung's original algorithm and successfully apply ERA to seismic imaging problems. In [114], the authors investigated criteria for modal analysis, such as the modal amplitude coherence and modal phase collinearity, and applied the algorithm to test data from vibration excitation of the Galileo spacecraft. [147] proposes an alternative version of Kung's algorithm involving a Hankel matrix twice as large as the Hankel matrix in Kung's work. With this approach, all available data is used to construct the gramians, however the computational burden to compute the singular value decomposition increases further. In [134, 101] recursive versions of ERA are presented which extend ERA to incorporate initial response data. In [72] the authors showed that once a 'maximal' rank ERA model is computed, lower order models can be obtained at almost no additional cost, which resulted from the outer product approximations of system matrices via singular value decomposition. Rowley and co-authors showed in [136] that ERA is the data-driven equivalent to balanced truncation. Where applicable (i.e., when system matrices and their adjoints are available), balanced POD was found to provide superior reduced order models. The authors in [191] propose a randomized POD technique to reduce the computational cost of extracting the dominant modes of the Hankel matrix. A randomized preselection of inputs/outputs/time steps before simulating the primal and dual systems allows for such computational speedup. Singler [171] generalized the ERA in a continuous setting to infinite dimensional systems.

Mechanical systems with multiple sensors and actuators are modeled as as multi-input multi-output (MIMO) dynamical systems. Such systems impose additional computational challenges for system identification, and ERA in particular. For instance, ERA requires a full singular value decomposition of a structured Hankel matrix, whose size scales linearly with the input and output dimension. Moreover, large Hankel matrices can arise if the dynamics of the system decay slowly.

We propose a system identification and model reduction algorithm for MIMO systems which reduces the computational effort and storage compared to standard ERA, see §4.3. The new algorithm projects the Hankel matrix with a carefully chosen left and right projection onto smaller input and output subspaces. Consequently, we do not neglect valuable impulse response data. Computing the SVD of the projected Hankel matrix has then become either feasible or can be executed in shorter time with fewer storage. Moreover, we show that reduced models obtained via the modified ERA retain stability. Numerical results in §4.4 demonstrate the accuracy and computational savings of the modified ERA with projected data.

Remark 4.1.1. A wide range of excellent model reduction techniques for LTI systems exist in the literature, see [8] for an overview. In particular, we shall mention balanced truncation [142, 143] and balanced proper orthogonal decomposition [189, 155], the Iterative Rational Krylov Algorithm [91] and Hankel norm approximations [88]. We do not propose to use ERA as a model reduction technique when state space matrices are available. We rather suggest to use ERA for the combined task of system identification and model reduction where only black-box code or experimental measurements are available. In this case, the aforementioned model reduction techniques are not applicable.

Remark 4.1.2. In this paper, our data will be restricted to time-domain samples of the impulse response of the underlying dynamical systems. In the frequency domain, this corresponds to sampling the transfer function and its derivates around infinity. For the cases where one has the flexibility in choosing the frequency samples, a variety of techniques become available such as the Loewner framework [138], Vector Fitting [95, 76] and various rational least-squares fitting methodologies [161, 36, 77]. However, as stated earlier, our focus here is ERA and to make it computationally more efficient for MIMO systems with large input and output dimensions.

4.2 Partial Realization and Kung's Algorithm

We motivate the problem of partial realization, state Kung's eigensystem realization algorithm and demonstrate the challenges that come with finding a partial realization of complex MIMO systems. In practice, experimental measurements and outputs of black box simulations are sampled at discrete time instances. Therefore, consider the discrete time LTI system

$$x_{k+1} = Ax_k + Bu_k, \tag{4.1}$$
$$y_k = Cx_k + Du_k, \tag{4.2}$$

where $x_k := x(t_k)$ and $t_k := k\Delta t,\ k \in \mathbb{N}_0^+$ is a discrete time instance and $\Delta t \in \mathbb{R}$ is a sampling time. The initial condition is $x(0) = x_0$ and assumed to be zero in the remainder - the system shall be excited through external disturbances. Here, $A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$ are the system, input and output matrices, respectively. The inputs are $u_k \in \mathbb{R}^m$ and the outputs are $y_k \in \mathbb{R}^p$. The system is completely determined by the matrices (A, B, C, D). It is common to define the *Markov parameters*

$$h_k := \begin{Bmatrix} D, & k = 0 \\ CA^{k-1}B, & k = 1, 2, \ldots \end{Bmatrix} \in \mathbb{R}^{p\times m}, \tag{4.3}$$

so the output response equation for system (4.1) - (4.2) becomes

$$y_k = CA^k x_0 + \sum_{i=0}^{k} h_i u_{k-i}.$$

The first term represents the initial response of the system and the second term is the response to past inputs. Assuming zero initial conditions, the external description of the discrete time dynamical system is

$$y_k = \sum_{i=0}^{k} h_i u_{k-i}, \tag{4.4}$$

which is known as the external description of the system; it is fully determined by the Markov parameters.

Definition 4.2.1. [8, Definition 4.1] The external description of a time-invariant, causal, smooth discrete time system with m inputs and p outputs is given by an infinite number of $p \times m$ matrices

$$\Sigma = (h_0, h_1, \ldots, h_k, \ldots).$$

From the previous definition one can see that the infinite series of Markov parameters uniquely defines a dynamical system for all $t > 0$. From a system identification point of view, the definition is impractical since in reality, an infinite amount of data is never available to describe a system. Thus we have to restrict ourselves to a finite data set.

Unfortunately, in some cases, the matrices (A, B, C, D) are not available in a practical setup and rather is the sequence of Markov parameters, describing the reaction of the system to external inputs. If only the Markov parameters (and therefore the external description (4.4)) are available through measurements, how can one reconstruct the internal description (4.1) - (4.2) of a LTI system? This is the classical problem of partial realization.

Definition 4.2.2. [8, Definition 4.46] Given the finite set of $p \times m$ matrices $h_i, i = 1, 2, \ldots, 2s-1$, the partial realization problem consists of finding a positive integer n and constant matrices $A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m},\ C \in \mathbb{R}^{p\times n}$, and $D \in \mathbb{R}^{p\times m}$ such that (4.3) holds.

A finite sequence of Markov parameters is always realizable and there always exists a minimal realization of order $n = \operatorname{rank}(\mathcal{H})$. The matrix $\mathcal{H}$ is called *Hankel matrix* and is defined by the $2s-1$ sampled Markov parameters as

$$\mathcal{H} := \begin{bmatrix} h_1 & h_2 & \ldots & h_s \\ h_2 & h_3 & \ldots & h_{s+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_s & h_{s+1} & \ldots & h_{2s-1} \end{bmatrix} \in \mathbb{R}^{ps\times ms}. \tag{4.5}$$

The size of the Hankel matrix grows linearly with m, p. In this work, we propose to construct a projected Hankel matrix that is independent of the input and output dimensions and therefore does not exhibit such growth. For a better understanding of the algorithms to follow, assume for a moment that the system matrices are known, so that the Hankel matrix reads as

$$\mathcal{H} = \begin{bmatrix} CB & CAB & \ldots & CA^{s-1}B \\ CAB & CA^2B & \ldots & CA^sB \\ \vdots & \vdots & \ddots & \vdots \\ CA^{s-1}B & CA^sB & \ldots & CA^{2s-1}B \end{bmatrix}.$$

It is well known (e.g., [8, Lemma 4.39]) that for a realizable impulse response sequence, the Hankel matrix can be factored into the product of the *observability* *gramian* $\mathcal{O}$ and the *controllability gramian* $\mathcal{C}$:

$$\mathcal{H} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{s-1} \end{bmatrix} [B \; AB \; \dots \; A^{s-1}B] := \mathcal{O} \, \mathcal{C}. \tag{4.6}$$

The shifted observability gramian satisfies

$$\mathcal{O}^{(f)} A = \mathcal{O}^{(l)}, \tag{4.7}$$

where $\mathcal{O}^{(f)}$ and $\mathcal{O}^{(l)}$ denote the first and last $s-1$ block rows of $\mathcal{O}$. A similar result holds for the controllability gramian: $A\mathcal{C}^{(f)} = \mathcal{C}^{(l)}$.

Silverman [164] proposed an algorithm to construct a minimal realization, which requires finding a rank n submatrix of the partially defined Hankel matrix. The algorithm determines the nth order minimal realization directly, and does not involve a model reduction step. Also, the algorithm does not guarantee to retain stability in the process. Kung's Eigensystem Realization Algorithm (ERA), on the other hand, can be divided into two steps, which are briefly reviewed below. To guarantee stability, Kung made the following assumption.

Assumption 4.2.3. *Assume that $2s-1$ Markov parameters are given and that the given impulse response sequence is convergent in the sense that*

$$h_i \to 0 \qquad \textit{for} \quad i > s.$$

We shall refer to this assumption as the convergence-to-zero *property of the Markov parameters.*

The above assumption can be interpreted as follows: The Markov transient dynamics of the Markov parameters are fully captured, i.e., for non-normal systems, the initial growth in the outputs is recorded, and after some $\hat{t} = t_i$, the system decays to zero.

Step 1 of ERA: Low rank approximation of Hankel matrix. Construct the Hankel matrix (4.5) from the given impulse response sequence $(h_1, h_2, \dots, h_{2s-1})$ and compute its singular value decomposition

$$\mathcal{H} = U\Sigma V^T \in \mathbb{R}^{ps \times ms}.$$

Here, $U \in \mathbb{R}^{ps \times ps}$ and $V \in \mathbb{R}^{ms \times ms}$ are orthogonal matrices, respectively, and $\Sigma \in \mathbb{R}^{ps \times ms}$ is a rectangular matrix containing singular values, $\Sigma_{ii} = \sigma_i$, $i =$

$1, \ldots, \min\{ms, ps\}$ (called Hankel singular values) [1] which are ordered as $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > \sigma_{n+1} = 0$. The rank of the Hankel matrix is n, the minimal realization order with $n \leq \min(ms, ps)$. Letting $r \leq n$, the decomposition can also be written as

$$\mathcal{H} = [U_r \ U_2] \begin{bmatrix} \Sigma_r & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_r^T \\ V_2^T \end{bmatrix}, \tag{4.8}$$

where $U_r \in \mathbb{R}^{ps \times r}$ contains the leading r columns of U, the square matrix $\Sigma_r = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$ and $V_r^T \in \mathbb{R}^{r \times ps}$. The matrices U_2, Σ_2, V_2^T and 0 have appropriate dimensions. Consequently, $U_r^T U_r = I_r$ and $V_r^T V_r = I_r$. From the Schmidt-Eckart-Young-Mirsky theorem [8, page 37], it follows that

$$\mathcal{H}_r = U_r \Sigma_r V_r^T$$

is the best rank r approximation of the Hankel matrix. The approximation error is given by $\|\mathcal{H} - \mathcal{H}_r\|_2 = \sigma_{r+1}$ and $\|\mathcal{H} - \mathcal{H}_r\|_F = \sqrt{\sigma_{r+1}^2 + \ldots + \sigma_n^2}$.

Step 2 of ERA: Approximate Realization of LTI System. It is the goal of this step to find a realization (A_r, B_r, C_r) of the *approximate* Hankel matrix $\mathcal{H}_r$ which is the best rank r approximation to $\mathcal{H}$. Kung suggested that $\mathcal{H}_r$ should have "Hankel structure" as well, so that it can be factored into a product of an *approximate* observability and controllability gramian as

$$\mathcal{H}_r = \mathcal{O}_r \mathcal{C}_r, \qquad \text{where} \quad \mathcal{O}_r = U_r \Sigma_r^{1/2}, \quad \mathcal{C}_r = \Sigma_r^{1/2} V_r^T.$$

In light of equation (4.6), if $\mathcal{O}_r$ is the approximation to the observability gramian, then its first block row can be used to estimate C_r, therefore

$$C_r = [I_p \ 0] \ U_r \Sigma_r^{1/2}, \tag{4.9}$$

where I_p is the $p \times p$ identity matrix. Similarly, the first block column of $\mathcal{C}_r$ yields an approximation of the control input matrix B_r:

$$B_r = \Sigma_r^{1/2} V_r^T \ [I_m \ 0]^T. \tag{4.10}$$

To estimate the system matrix A_r, the shift invariance property (4.7) is imposed on the approximate gramians as

$$\mathcal{O}_r^{(f)} A_r = \mathcal{O}_r^{(l)}, \qquad A_r \mathcal{C}_r^{(f)} = \mathcal{C}_r^{(l)}.$$

[1] We use this term to refer to the singular values of the Hankel matrix. Per definition, the Hankel singular values are the singular values of the underlying Hankel *operator*; see, e.g., [8].

Either equality can be used to solve the least squares problem for A_r. Without loss of generality, we focus on the first equality involving the observability gramian. The reader should observe that $\mathcal{O}_r^{(f)}$ is a $p(s-1) \times r$ matrix, so a least squares problem to minimize $\|\mathcal{O}_r^{(f)} A_r - \mathcal{O}_r^{(l)}\|$ has to be solved. The minimizing solution is given by the Moore-Penrose pseudo inverse [89, Ch. 5] as

$$A_r = [\mathcal{O}_r^{(f)}]^{\dagger} \mathcal{O}_r^{(l)}.$$

Note that $\mathcal{O}_r^{(f)} = U_r^{(f)} \Sigma_r^{1/2}$ so that A_r is computed as

$$A_r = \Sigma_r^{-1/2} [U_r^{(f)}]^T U_r^{(l)} \Sigma_r^{1/2}. \tag{4.11}$$

Theorem 4.2.4. *[124] If the Markov parameters satisfy Assumption 4.2.3, then the realization given by (A_r, B_r, C_r) from (4.9), (4.10), (4.11) provides a stable discrete time dynamical system. Then,*

$$\sum_{i=1}^{2s-1} \|C_r A_r^{i-1} B_r - h_i\|_F^2 \leq \sigma_{r+1} \sqrt{r+m+p}.$$

where p is the number of outputs, m is the number of inputs, r is the order of the reduced model and σ_{r+1} denotes the first neglected Hankel singular value.

Theorem 4.2.4 reveals that if the original model is stable, then reduced order models of any order r obtained through ERA are stable, too with an a priori error bound in the impulse response reconstruction. The rank n of the Hankel matrix is the order of the minimal realization. However, n can be very large and the resulting model too big for design and control purposes. Instead, one would like to obtain reduced order models of order $r \ll n$. The choice of r depends on many factors, such as accuracy of the reduced order model, performance criteria, limitations on implementable model orders, etc.

Example 4.2.5. This work has been motivated by the need to generate reduced order models for the indoor-air behavior in buildings, see [38, §IV.B]. The original model of interest has a large number of inputs and outputs, in particular, we are given $m = 26$ control inputs and $p = 42$ measured outputs. The impulse response data is sampled over $3600[s]$ with a Markov parameter measured every $2[s]$. With standard ERA, this requires computing a *full* SVD of size $37,800 \times 23,400$, which is challenging on a standard desktop machine.

4.3 Tangential Interpolation of the Markov Parameters

The main bottleneck of the original eigensystem realization algorithm is the computation of the singular value decomposition of the Hankel matrix. In many cases it might not be feasible to compute the SVD for the large matrix $\mathcal{H}$. Even if one can do so, it might take a significant amount of computation time and memory. Thus, we propose a way to circumvent this problem by interpolating the Markov parameters before assembly of the Hankel matrix. The proposed algorithm, denoted by TERA henceforth, has three stages:

- Compute tangential directions and project the impulse response data, see §4.3.2.
- Use the ERA on the reduced size Hankel matrix $\hat{\mathcal{H}}$ and obtain realizations $(\hat{A}_r, \hat{B}_r, \hat{C}_r)$, see §4.3.3.
- Convert back to the full input and output dimensions and get approximations (A_r, B_r, C_r) of the original impulse response sequence, see §4.3.3.

Our approach is motivated by rational approximation by tangential interpolation, as illustrated in the next section.

4.3.1 Tangential Interpolation from Data

A thorough treatment of rational interpolation of a given dataset along tangential directions can be found in [9, §7]. To illustrate the idea, assume for a moment that a continuous time dynamical system

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned}$$

is given. By applying a Laplace transform, it can be seen that the transfer function $G(s) = C(sI - A)^{-1}B$ maps the inputs to the outputs in the frequency domain through $\hat{y}(s) = G(s)\hat{u}(s)$. Note that the transfer function in the frequency domain is the equivalent to the input-ouput description (4.4) by Markov parameters in the discrete time domain. Model reduction through rational interpolation seeks a reduced order transfer function $G_r(\omega) = C_r(\omega I - A_r)^{-1}B_r$, with $A \in \mathbb{R}^{r\times r}$, $B \in \mathbb{R}^{r\times m}$ and $C \in \mathbb{R}^{p\times r}$, such that $G(s_i) = G_r(s_i)$ for a set of interpolation points $\{s_i : \ i = 1, 2, \ldots, k\}$. However, for MIMO systems, this is too restrictive since it imposes $p{\cdot}m$ conditions for every interpolation point leading to unnecessarily high reduced orders. The concept of tangential interpolation eases those conditions by only enforcing interpolation along certain directions. Assume that transfer function $G(s)$ is sampled at r points $\{\theta_i : \ i = 1, 2, \ldots, r\}$ along *the right tangential directions* $u_i \in \mathbb{C}^m$ and r points $\{\mu_i : \ i = 1, 2, \ldots, r\}$ along *the left tangential directions* $v_i \in \mathbb{C}^p$; i.e., $G(\theta_i)u_i$ and $v_i^T G(\mu_i)$ are measured. Then, the Loewner framework [138] produces a reduced model $G_r(s)$ that tangentially interpolates the given data, i.e.,

$$v_i^T G(\mu_i) = v_i^T G_r(\mu_i) \qquad \text{and} \qquad G(\theta_i)u_i = G_r(\theta_i)u_i,$$

The details of how the interpolant $G_r(s) = C_r(sE_r - A_r)^{-1}B_r$ is constructed can be found in [138, 9]; here we only show how E_r is constructed:

$$E_r(i,j) = -\frac{v_i^T(G(\mu_i) - G(\theta_j))u_j}{\mu_i - \theta_j}, \quad \text{for} \quad i,j = 1,\ldots,r \tag{4.12}$$

E_r is a divided different matrix (called the Loewner matrix) corresponding to $G(s)$. However, in filling the entries of E_r, not the full-matrix data $G(\mu_i) \in \mathbb{C}^{m\times p}$ or $G(\theta_i) \in \mathbb{C}^{m\times p}$, but rather the tangential data $v_i^T G(\mu_i)u_j \in \mathbb{C}$ and $v_i^T G(\theta_j)u_j \in \mathbb{C}$ are used; thus the dependence on the input and output dimensions are avoided. Note that without this modification, the reduced matrix E_r would be of dimension $(r \cdot m) \times (r \cdot p)$ as opposed to $r \times r$. This is the motivation for our modification to ERA.

Remark 4.3.1. The choice of interpolation points and tangential directions is of fundamental importance in model reduction by interpolation. Iterative Rational Krylov Algorithm of [91] provides a locally optimal strategy in the $\mathcal{H}_2$ norm. In [23], IRKA has been recently coupled with the Loewner approach to find optimal reduced models in a data-driven setting. However; this approach cannot be applied here since in the ERA setting, the available frequencies are fixed and one can only sample the Markov parameters, which corresponds to sampling the transfer function at infinity.

4.3.2 Projection of Markov parameters

Inspired by tangential interpolation in the Loewner framework, for systems with high dimensional input and output spaces we will project the impulse response samples

h_i onto low dimensional subspaces via multiplications by tangential directions. However, achieving this goal in the ERA set-up comes with major additional difficulties that do not appear in the Loewner framework. Therein, the elegant construction of the reduced-model quantities B_r and C_r guarantee that the number of rows and columns still match the original input and output dimensions even when the tangential interpolation is employed. In other words, only the system dimension is reduced without changing the input/output dimensions. However, in ERA, once the Markov parameters $h_i \in \mathbb{R}^{p\times m}$ are replaced by the (tangentially) projected quantities $\hat{h}_i \in \mathbb{R}^{\ell_1\times\ell_2}$ where $\ell_1 < p$ and $l_2 < m$, the reduced model via ERA will have l_2 inputs and ℓ_1 outputs; thus the original input and output dimensions will be lost. Therefore, one will need to carefully lift this reduced model back to the original m-inputs and p-outputs spaces. The second difficulty arises from the fact that sampling Markov parameters mean sampling $G(s)$ only around infinity and thus we need to choose the same tangential directions for every sample. Since selecting a single direction for all the Markov parameters will be extremely restrictive, we will pick multiple dominant tangential directions to project all the Markov parameters.

To deal with large input and output spaces, [191] uses a randomized selection of inputs and outputs and subsequently collect primal and dual simulation data reducing computational time and storage requirements for the SVD of the Hankel matrix However, the method assumes that primal and dual simulations can be performed separately, which is not possible in several situations and which we will not assume. In [136] and [155], the authors consider fluid dynamical applications, where the output of interest is often the entire state, which is enormous. Hence, standard ERA is not feasible, especially since the complex dynamical behavior of fluid systems makes it necessary to sample many Markov parameters. The authors suggest to project the output space onto a low dimensional manifold, and use ERA subsequently. However, this is mentioned as a rather short remark without any details and an algorithm to recover the original output dimension is not given, a crucial difficulty arising in the ERA setup as mentioned above. Moreover, not only the inputs, but also the number of outputs, can cause computational challenges. Recall Example 4.2.5, where both input and output dimensions are large ($m = 26$ and $p = 42$), which leads to a challenging computation of the SVD. Therefore, we propose a modified ERA method that works with a two-sided projected version of the Markov parameters while guaranteeing stability of the reduced model endowed with an error bound.

The minimization problem behind the proposed method is to find two projectors P_1

and P_2 that solves

$$\min_{\substack{\mathrm{rank}(\mathrm{P}_1)=\ell_1 \\ \mathrm{rank}(\mathrm{P}_2)=\ell_2}} \sum_{i=1}^{2s-1} ||P_1 h_i P_2 - h_i||_F^2. \tag{4.13}$$

Remark 4.3.2. Ideally, one would like to pick individual projectors $P_1^{(i)}$ and $P_2^{(i)}$ for every Markov parameter to produce the minimal error $\sum_{i=1}^{2s-1}\sum_{j=\ell_1+1}^{p}\sigma_j^2(h_i)$, where $\ell_1 = \ell_2$. However this appears to be impractical since, in an analogy to tangential interpolation, this would correspond to choosing different tangential directions for $G(s)$ and $G'(s)$ for example. Therefore we restrict ourselves to finding two orthogonal projectors, which are used for the entire dataset of Markov parameters. Henceforth, we shall see that this preserves the structure of the Hankel matrix, at the cost of a suboptimal error.

The projectors are given via the products of orthonormal matrices, i.e.,

$$\begin{aligned} P_1 &= W_1 W_1^T, \quad \mathrm{rank}(\mathrm{P}_1) = \ell_1, \\ P_2 &= W_2 W_2^T, \quad \mathrm{rank}(\mathrm{P}_2) = \ell_2, \end{aligned}$$

where $W_1^T W_1 = I_{\ell_1}$ and $W_2^T W_2 = I_{\ell_2}$. The goal is to find the P_1 and P_2 by considering data-streams of Markov parameters. To compute the projection matrices, arrange the impulse response sequence in a matrix

$$\Theta_L := [h_1 \ h_2 \ \cdots h_{2s-1}] \in \mathbb{R}^{p\times m(2s-1)}$$

and solve the optimization problem

$$P_1 = \arg \min_{\mathrm{rank}(\widetilde{\mathrm{P}}_1)=\ell_1} ||\widetilde{P}_1 \Theta_L - \Theta_L||_F^2. \tag{4.14}$$

The optimal solution of the optimization problem is given by the singular value decomposition (SVD) of $\Theta_L = U\Sigma V^T$, and hence $W_1 = U(:, 1 : \ell_1)$, the leading ℓ_1 columns of U. This minimum error is then given by $||W_1 W_1^T \Theta_L - \Theta_L|| = \sigma_{\ell_1+1}(\Theta_L)$, the $(\ell_1+1)^{\mathrm{th}}$ singular value of Θ_L. To compute the right projector P_2, we again define

$$\Theta_R := \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{2s-1} \end{bmatrix} \in \mathbb{R}^{p(2s-1)\times m}$$

and the corresponding optimization problem

$$P_2 = \arg \min_{\mathrm{rank}(\widetilde{\mathrm{P}}_2)=\ell_2} ||\Theta_R \widetilde{P}_2 - \Theta_R||_F^2. \tag{4.15}$$

Similarly, compute the SVD of $\Theta_R = U\Sigma V^T$, and the optimal solution is $P_2 = W_2 W_2^T$, where $W_2 = V(:, 1 : \ell_2)$. The error is simply $||W_2 W_2^T \Theta_R - \Theta_R|| = \sigma_{\ell_2+1}(\Theta_R)$. The goal is to reduce the size of the Markov parameters, to lessen the cost of the singular value decomposition of the Hankel matrix. The factors W_1 and W_2 are employed to project the Markov parameters as

$$\hat{h}_i = W_1^T h_i W_2 \quad \in \mathbb{R}^{\ell_1 \times \ell_2} \tag{4.16}$$

Equation (4.16) can be considered analogous to tangential interpolation where the transfer function $G(s_i)$ ($s_i = \infty$ in this case) and its derivatives are sampled along different tangential directions; the columns of W_1 and W_2. The projected values $\hat{h}_i$ are subsequently used to construct a reduced size Hankel matrix $\hat{\mathcal{H}}$. For this, define the block diagonal matrices $\mathcal{W}_1 := \mathrm{diag}(\mathrm{W}_1, \mathrm{W}_1, \ldots, \mathrm{W}_1)$ and $\mathcal{W}_2 := \mathrm{diag}(\mathrm{W}_2, \ldots, \mathrm{W}_2)$. Then the projected Hankel matrix becomes

$$\hat{\mathcal{H}} = \mathcal{W}_1^T \mathcal{H} \mathcal{W}_2 \in \mathbb{R}^{s\ell_1 \times s\ell_2}. \tag{4.17}$$

4.3.3 ERA for Projected Hankel Matrix and Recovering Original Dimensions

Once the projected Hankel matrix (4.17) is computed, ERA can be applied. However due to the projected input and output dimensions, control and observation matrices are identified in the reduced output/reduced input space. Thus, the goal of TERA is to lift these spaces optimally back to the original dimension recover the full input and output dimensions. Using the definitions of $\mathcal{W}_1$ and $\mathcal{W}_2$, we can rewrite (4.17) as

$$\begin{aligned}
\hat{\mathcal{H}} = [\hat{h}]_{ij} &= \begin{bmatrix} W_1^T & & \\ & \ddots & \\ & & W_1^T \end{bmatrix} \begin{bmatrix} h_1 & h_2 & \dots & h_s \\ h_2 & h_3 & \dots & h_{s+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_s & h_{s+1} & \dots & h_{2s-1} \end{bmatrix} \begin{bmatrix} W_2 & & \\ & \ddots & \\ & & W_2 \end{bmatrix} \\
&= \begin{bmatrix} W_1^T & & \\ & \ddots & \\ & & W_1^T \end{bmatrix} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{s-1} \end{bmatrix} [B\ AB \dots A^{s-1}B] \begin{bmatrix} W_2 & & \\ & \ddots & \\ & & W_2 \end{bmatrix} \\
&= \begin{bmatrix} \hat{C} \\ \hat{C}A \\ \vdots \\ \hat{C}A^{s-1} \end{bmatrix} [\hat{B}\ A\hat{B} \dots A^{s-1}\hat{B}].
\end{aligned}$$

where we have defined $\hat{C} = W_1^T C$ and $\hat{B} = BW_2$. This illustrates how to identify $(\hat{A}, \hat{B}, \hat{C})$ from the interpolated Hankel matrix. The best rank r approximation of the projected Hankel matrix is given by the truncated singular value decomposition

$$\hat{\mathcal{H}}_r = \hat{U}_r \hat{\Sigma}_r \hat{V}_r^T = \hat{\mathcal{O}}_r \hat{\mathcal{C}}_r,$$

where $\hat{\mathcal{O}}_r = \hat{U}_r \hat{\Sigma}_r^{1/2}$ and $\hat{\mathcal{C}}_r = \hat{\Sigma}_r^{1/2} \hat{V}_r^T$ represent the observability and controllability matrices, respectively. As before, the first block row of $\hat{\mathcal{O}}_r$ gives an approximation for $\hat{C}_r$, the observation matrix matching the interpolated impulse response, so

$$\hat{C}_r = [I_\ell\ 0]\ \hat{U}_r \hat{\Sigma}_r^{1/2} \in \mathbb{R}^{\ell_1 \times r}.$$

Analogously, the first block column of $\hat{\mathcal{C}}_r$ yields an approximation for $\hat{B}_r$, the control input matrix for the interpolated impulse response sequence, which reads as

$$\hat{B}_r = \hat{\Sigma}_r^{1/2} \hat{V}_r^T [I_\ell\ 0]^T \in \mathbb{R}^{r \times \ell_2}.$$

To solve the least squares problem for the system matrix $\hat{A}_r$, one proceeds as in the previous subsection, so that

$$\hat{A}_r = [\hat{\mathcal{O}}_r^{(f)}]^\dagger \hat{\mathcal{O}}_r^{(l)} = \hat{\Sigma}_r^{-1/2} [\hat{U}_r^{(f)}]^T \hat{U}_r^{(l)} \hat{\Sigma}_r^{1/2}, \tag{4.18}$$

which is computed as in (4.11) with appropriate matrices. To illuminate the connection between the A_r in (4.11) obtained from standard ERA and the $\hat{A}_r$ in (4.18) obtained from the projected sequence, let $\widetilde{\mathcal{W}}_1^T$ denote the matrix obtained from deleting the last block row and column from $\mathcal{W}_1^T$, and similarly for $\widetilde{\mathcal{W}}_2^T$. Note that $\hat{\mathcal{O}}_r^{(f)} = \widetilde{\mathcal{W}}_1^T \mathcal{O}_r^{(f)}$. Then, it readily follows that

$$\hat{A}_r = [\widetilde{\mathcal{W}}_1^T \mathcal{O}_r^{(f)}]^\dagger \widetilde{\mathcal{W}}_1^T \mathcal{O}_r^{(l)} = [\mathcal{O}_r^{(f)}]^\dagger \widetilde{\mathcal{W}}_1 \widetilde{\mathcal{W}}_1^T \mathcal{O}_r^{(l)},$$

using the fact that $\widetilde{\mathcal{W}}_1$ is orthogonal. Recall from (4.11) that $A_r = [\mathcal{O}_r^{(f)}]^\dagger \mathcal{O}_r^{(l)}$. Thus, $\hat{A}_r$ works on the $\mathcal{O}_r^{(l)}$ projected onto the range of $\widetilde{\mathcal{W}}_1$. Note that $\widetilde{\mathcal{W}}_1 \widetilde{\mathcal{W}}_1^T \neq I$ unless $\tilde{\mathcal{W}}_1$ is square (i.e., when there is no reduction in input and output dimension in which case one recovers the standard ERA.) The identified system matrices $\hat{A}_r$,$\hat{B}_r$ and $\hat{C}_r$ match the *projected* Markov parameters

$$\hat{h}_i \approx \hat{C}_r \hat{A}_r^{i-1} \hat{B}_r, \quad i = 1, \ldots, 2s-1$$

in the least squares sense.

Note that while $\hat{A}_r$ is an $r \times r$ matrix (matching the original ERA construction), $\hat{B}_r$ has ℓ_2 columns (as opposed to m) and $\hat{C}_r$ has ℓ_1 rows (as opposed to p). Therefore, we need to lift $\hat{B}_r$ and $\hat{C}_r$ to the original input/output dimensions. By virtue of the minimization problem (4.13), the original input-output dimension of the system can be recovered through injection of $\hat{h}_i$ to the $\mathbb{R}^{p \times m}$. Recall that $\hat{h}_i = W_1^T h_i W_2$. Therefore $\{\hat{C}_r \hat{A}_r^{i-1} \hat{B}_r\}$ approximates $\{W_1^T h_i W_2\}$ in the least-squares sense. To approximate the original sequence $\{h_i\}$, we, then, replace $\hat{C}_r$ with $W_1 \hat{C}_r$ and $\hat{B}_r$ with $\hat{B}_r W_2^T$. In other words, the original impulse response sequence is approximated via

$$h_i \approx \underbrace{W_1 \hat{C}_r}_{:=C_r} \hat{A}_r \underbrace{\hat{B}_r W_2^T}_{:=B_r} \tag{4.19}$$

yielding the final reduced-model quantities

$$\begin{aligned} A_r &= \hat{\Sigma}_r^{-1/2} [\hat{U}_r^{(f)}]^T \hat{U}_r^{(l)} \hat{\Sigma}_r^{1/2} \\ B_r &= \hat{\Sigma}_r^{1/2} \hat{V}_r^T [I_{\ell_2}\ 0]^T W_2^T \\ C_r &= W_1 [I_{\ell_1}\ 0]\ \hat{U}_r \hat{\Sigma}_r^{1/2} \end{aligned} \tag{4.20}$$

The modified eigensystem realization algorithm for tangentially interpolated data (TERA) is given in Algorithm 5.

Algorithm 5 : TERA

```
Input: Markov parameters $h_1, h_2, \ldots, h_{2s-1}$;
       Reduced model order $r$;
       Number of tangential directions $\ell_1, \ell_2$ .
Output: State space realization $(A_r, B_r, C_r)$.
Compute svd: $[h_1 \ h_2 \ \cdots h_{2s-1}] = U\Sigma V^T$.
$W_1 = U(:, 1 : \ell_1)$.
Compute svd: $[h_1^T \ h_2^T \ \cdots h_{2s-1}^T]^T = U\Sigma V^T$.
$W_2 = V(:, 1 : \ell_2)$.
for $i = 1 : 2s-1$ do
    $\hat{h}_i = W_1^T h_i W_2$.
end for
Assemble Hankel matrix $\hat{\mathcal{H}}$ as in (4.5).
Compute svd: $\hat{\mathcal{H}} = \hat{U}\hat{\Sigma}\hat{V}^T$.
$\hat{U}_r = \hat{U}(1 : r, :)$, $\hat{\Sigma}_r = \hat{\Sigma}(1 : r, 1 : r)$ and $\hat{V}_r = \hat{V}(1 : r, :)$.
$\hat{U}_r^{(f)} = \hat{U}(1 : r, 1 : p(s-1)($ and $\hat{U}_r^{(l)} = \hat{U}(1 : r, (s+1) : ps)$.
$A_r = \hat{\Sigma}_r^{-1/2}[\hat{U}_r^{(f)}]^T \hat{U}_r^{(l)} \hat{\Sigma}_r^{1/2}$.
$B_r = \hat{\Sigma}_r^{1/2} \hat{V}_r^T [I_l \ 0]^T W_2^T$.
$C_r = W_1 [I_l \ 0] \ \hat{U}_r \ \hat{\Sigma}_r^{1/2}$.
```

4.3.4 Error Analysis and Stability

Since its original publication by Kung, ERA has enjoyed great popularity, which is in part because the obtained reduced order models are stable. Hence, we would like to retain this important feature, and we therefore show first, that stable models yield stable ROM's when TERA is applied.

Proposition 4.3.3. *If Assumption 4.2.3 holds, i.e., if the underlying dynamical systems is stable, then the reduced model given by the matrices A_r, B_r, C_r in (4.20) obtained via TERA from the projected data is stable.*

Proof. The projected Markov parameters are $\hat{h}_i = W_1^T h_i W_2$, where $W_1 \in \mathbb{R}^{p\times \ell_1}$ and $W_2 \in \mathbb{R}^{m\times \ell_2}$ have orthogonal columns. It follows from Assumption 4.2.3, that $\|h_i\|_F \to 0$ when $i > s$. Therefore,

$$\|\hat{h}_i\|_F = \|W_1^T h_i W_2\|_F \leq \|W_1\|_F \|h_i\|_F \|W_2\|_F \to 0 \quad \text{for} \quad i > s.$$

It thus follows that $\hat{h}_i \to 0$ as $i > s$. Since the projected impulse response satisfies the convergence to zero property, Theorem 4.2.4 can be applied for TERA to obtain that $\|A_r\| \leq 1$, which completes the proof. □

From Theorem 4.2.4 we can directly obtain an error bound for the interpolated Markov parameters.

Corollary 4.3.4. *In the reduced input and output dimensions ℓ, the error in the Markov parameter sequence is given by*

$$\sum_{i=1}^{2s-1} \|\hat{C}_r \hat{A}_r^{i-1} \hat{B}_r - \hat{h}_i\|_F^2 \leq \sqrt{r + \ell_1 + \ell_2} \cdot \sigma_{r+1}(\hat{\mathcal{H}}).$$

Proof. Note that when ERA is applied to $\hat{h}_i$, it yields a stable reduced order model as shown in Proposition 4.3.3. Using $m = \ell_1$ and $p = \ell_2$ in Theorem 4.2.4, the result follows directly, by replacing all quantities by the 'hat' quantities. □

Corollary 4.3.4 gives a bound for the error in the interpolated (projected) Markov parameters. However, the real quantity of interest is the least-squares in the reconstruction of the original full Markov parameter sequence $\{h_i\}$. The next results answers this question.

Theorem 4.3.5. *Let h_i be the original sequence of Markov parameters, and let $\{\hat{C}_r\hat{A}_r^{i-1}\hat{B}_r\}$ be the identified sequence via TERA. The approximation error is given by*

$$\sum_{i=1}^{2s-1} ||h_i - W_1\hat{C}_r\hat{A}_r^{i-1}\hat{B}_rW_2^T||_F^2 \leq 4\left(\sum_{i=\ell_1+1}^{p}\sigma_i^2(\Theta_L) + \sum_{i=\ell_2+1}^{m}\sigma_i^2(\Theta_R)\right) + 2\sqrt{r+\ell_1+\ell_2}\cdot\sigma_{r+1}(\hat{\mathcal{H}}) \tag{4.21}$$

Proof. We begin with splitting the error into two parts

$$\begin{aligned}
\sum_{i=1}^{2s-1} ||h_i - W_1\hat{C}_r\hat{A}_r^{i-1}\hat{B}_rW_2^T||_F^2 &= \sum_{i=1}^{2s-1} ||\underbrace{h_i - P_1h_iP_2}_{=:T_i} + \underbrace{P_1h_iP_2 - W_1\hat{C}_r\hat{A}_r^{i-1}\hat{B}_rW_2^T}_{=:Z_i}||_F^2 \\
&= \sum_{i=1}^{2s-1} ||T_i + Z_i||_F^2 \\
&\leq \sum_{i=1}^{2s-1} (||T_i||_F + ||Z_i||_F)^2 \\
&\leq 2\left(\underbrace{\sum_{i=1}^{2s-1}||T_i||_F^2}_{\varepsilon_1} + \underbrace{\sum_{i=1}^{2s-1}||Z_i||_F^2}_{\varepsilon_2}\right),
\end{aligned}$$

where in the last line we used $2||T_i||_F||Z_i||_F \leq ||T_i||_F^2 + ||Z_i||_F^2$. Subsequently, we give estimates for the two error terms ε_1, and ε_2. We begin with ε_1:

$$\begin{aligned}
\varepsilon_1 &= \sum_{i=1}^{2s-1} ||h_i - P_1h_iP_2||_F^2 \\
&= \sum_{i=1}^{2s-1} ||h_i - P_1h_i + P_1(h_i - h_iP_2)||_F^2 \\
&= ||\Theta_L - P_1\Theta_L + P_1(\Theta_L - \Theta_L\mathcal{P}_2)||_F^2 \\
&\leq ||\Theta_L - P_1\Theta_L||_F^2 + ||\Theta_L - \Theta_L\mathcal{P}_2||_F^2 + 2||\Theta_L - P_1\Theta_L||_F||\Theta_L - \Theta_L\mathcal{P}_2||_F \\
&\leq 2\left(||\Theta_L - P_1\Theta_L||_F^2 + ||\Theta_L - \Theta_L\mathcal{P}_2||_F^2\right)
\end{aligned}$$

where $\mathcal{P}_2 = \mathrm{diag}(\mathrm{P}_2, \dots, \mathrm{P}_2)$ is block-diagonal, and we used in the last equality that the Frobenius norm is invariant under orthogonal transformations. For the first term in the sum, it follows from the definition of P_1 in (4.14) (and by the singular value decomposition) that

$$||\Theta_L - P_1\Theta_L||_F^2 = \sum_{i=\ell_1+1}^{p} \sigma_i^2(\Theta_L).$$

The second term in the sum can be rewritten as

$$\begin{aligned} ||\Theta_L - \Theta_L\mathcal{P}_2||_F^2 &= ||[h_1,\ h_2,\ \dots, h_{2s-1}] - [h_1P_2,\ h_2P_2,\ \dots,\ h_{2s-1}P_2]||_F^2 \\ &= ||\Theta_R P_2 - \Theta_R||_F^2 \\ &= \sum_{i=\ell_2+1}^{m} \sigma_i^2(\Theta_R), \end{aligned}$$

where the last equality follows from the definition of P_2 in (4.15). Collecting the terms yields

$$\varepsilon_1 \leq 2\left(\sum_{i=\ell_1+1}^{p} \sigma_i^2(\Theta_L) + \sum_{i=\ell_2+1}^{m} \sigma_i^2(\Theta_R)\right).$$

The term ε_2 can be simplified using the orthogonality of W_1 and W_2 and by using Corollary 4.3.4; namely, we obtain

$$\begin{aligned} \varepsilon_2 &= \sum_{i=1}^{2s-1} ||P_1h_iP_2 - W_1\hat{C}_r\hat{A}_r^{i-1}\hat{B}_rW_2^T||_F^2 \\ &= \sum_{i=1}^{2s-1} ||W_1W_1^Th_iW_2W_2^T - W_1\hat{C}_r\hat{A}_r^{i-1}\hat{B}_rW_2^T||_F^2 \\ &= \sum_{i=1}^{2s-1} ||W_1^Th_iW_2 - \hat{C}_r\hat{A}_r^{i-1}\hat{B}_r||_F^2 \\ &= \sqrt{r+\ell_1+\ell_2}\cdot\sigma_{r+1}(\hat{\mathcal{H}}). \end{aligned}$$

Collecting the terms, we obtain

$$\begin{aligned} \sum_{i=1}^{2s-1} ||h_i - W_1\hat{C}_r\hat{A}_r^{i-1}\hat{B}_rW_2^T||_F^2 &\leq 2(\varepsilon_1+\varepsilon_2) \\ &\leq 4\left(\sum_{i=\ell_1+1}^{p} \sigma_i^2(\Theta_L) + \sum_{i=\ell_2+1}^{m} \sigma_i^2(\Theta_R)\right) + 2\sqrt{r+\ell_1+\ell_2}\cdot\sigma_{r+1}(\hat{\mathcal{H}}), \end{aligned}$$

which completes the proof. □

4.4 Numerical Results

In this section, we present numerical results for TERA (Algorithm 5) and Kung's standard ERA. To test these algorithms, a mass spring damper model (MSD) and a cooling model for steel profiles (Rail) are considered. The main computational difference between the approaches is the size of the full-SVD that need to computed. As we will illustrate, TERA offers significant computational savings by working with the SVD of a reduced Hankel matrix, see Table 4.1.

Example	SVD size for ERA	CPU	SVD size for TERA	CPU
4.4.1 (MSD)	$15,000 \times 15,000$	1216.8s	$3,500 \times 3,500$	18.0s
4.4.2 (Rail)	6000×7000	110.0s	4000×4000	25.2s

Table 4.1: Specifications, CPU times to execute, and time savings for the numerical examples. Solved on cluster with a 6-core Intel Xeon X5680 CPU at 3.33GHz and 48GB RAM, with `Matlab2013b`.

Note that ERA assumes a discrete-time model. The examples we consider are continuous-time dynamical systems, i.e., they have the form

$$\dot{x}(t) = A_c x(t) + B_c u(t) \tag{4.22}$$

$$y(t) = C_c x(t) \tag{4.23}$$

where the subscripts are used to emphasize the continuous time parameters. *We emphasize that the matrices A_c, B_c, C_c are never used in the algorithm. Both ERA and TERA have only access to impulse response data.* Once the reduced-models are computed via ERA and TERA, we use the original system dynamics only for illustration purposes to present a more detailed comparison both in the time-domain by comparing time-domain simulations and in the frequency domain by comparing Bode plots. Since we have access to continuous time matrices, the original transfer function can be compared to $G_r(\cdot)$ obtained from the reduced order models. In continuous time, the output error between the full and reduced order model is bound as

$$\|y - y_r\|_2 \leq \|G - G_r\|_{\mathcal{H}_\infty} \|u\|_2,$$

where the norm of the transfer function is defined as

$$\|G - G_r\|_{\mathcal{H}_\infty} := \sup_{\omega \in \mathbb{R}} \|G(i\omega)\|_2.$$

Note, that ERA assumes a discrete time model, therefore we convert the continuous models to discrete time via a bilinear transformation (e.g., [3]), mapping the complex open left half plane into the unit circle in $\mathbb{C}$. Let T_s be the sampling period for the discrete time system. The bilinear transformation is

$$z = e^{t\Delta t} = \frac{1 + t\frac{\Delta t}{2}}{1 - t\frac{\Delta t}{2}},$$

where z is a discrete time variable, $z \in \mathbb{C}, |z| \leq 1$ and $t \in (-\infty, 0]$ is the continuous time variable. The system matrices have to be converted to

$$\begin{aligned}
A &= \frac{\Delta t}{2}\left(\frac{2}{\Delta t}I + A_c\right)\left(\frac{2}{\Delta t}I - A_c\right)^{-1}, \\
B &= \sqrt{2}\left(\frac{2}{\Delta t}I - A_c\right)^{-1} B_c, \\
C &= \sqrt{2}C_c\left(\frac{2}{\Delta t}I - A_c\right)^{-1}, \\
D &= C_c\left(\frac{2}{\Delta t}I - A_c\right)^{-1} B_c.
\end{aligned}$$

The above conversion is also known as Tustin transformation and is implemented in `Matlab` when using the `c2d` command with the option `tustin`. The SLICOT AB04MD routine can alternatively be used for the bilinear transformation.

4.4.1 Mass Spring Damper System

This model is taken from [92] and describes a mass spring damper system with masses m_i, spring constants k_i and damping coefficients $c_i \geq 0$ for $i = 1, 2, \ldots, n/2$. The state variables are the displacement and momentum of the masses, and the outputs are the velocities of some selected masses. We refer to [92, §6] for more details about the model.

The model dimension is $n = 1000$, which is equivalent to 500 mass spring damper elements. All masses are $m_i = 4$, the spring constants are $k_i = 4$ and the damping coefficients are $c_i = 0.1$ for $i = 1, 2, \ldots, 500$. The number of inputs is equal to the number of outputs, namely $m = p = 30$. The largest eigenvalue of A_c is $\lambda_{max} = 3.9 \cdot 10^{-5}$, and the system is simulated up to $T = 2 \cdot 10^4 s$. We collect $2s = 1000$ Markov parameters, which corresponds to a sampling period of $\Delta t = 20s$ for the conversion of continuous to discrete time systems. In Figure 4.1, the left plot, decay of the normalized Markov parameters is plotted, $\|h_i\|/\|h_1\|$. Note the steep initial decay and then a slower decay after about one hour. The right-plot of Figure 4.1 shows the singular values of Θ^L and Θ^R.

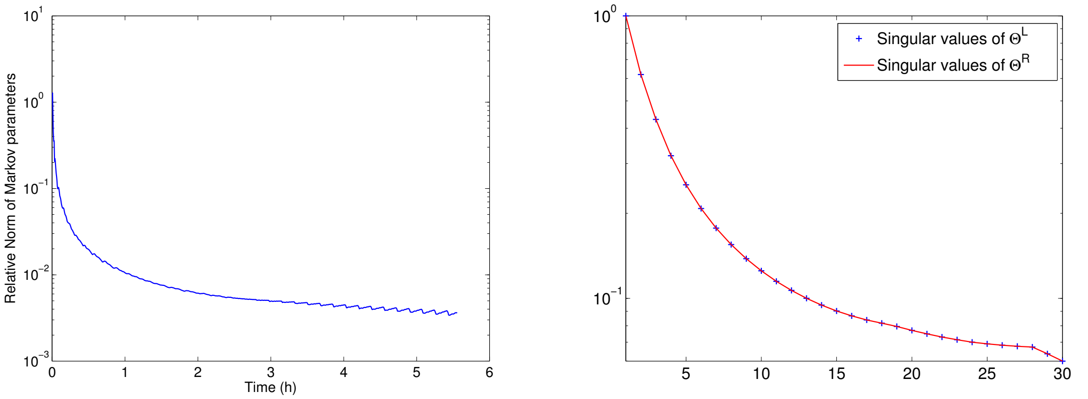

Figure 4.1: MSD model. (left): Norm of the Markov parameters. (right): Singular values of Θ^L and Θ^R.

Application of the standard ERA requires computing an SVD of size $15,000 \times 15,000$. In TERA, we pick $\ell_1 = \ell_2 = 7$ and with projection through $\ell_1 = \ell_2 = 7$ directions, an SVD of size 3500×3500 has to be computed. Figure 4.2, left, shows the transfer function of the full model, and the two reduced order models. Both reduced models have problems matching the resonance at $\omega \approx 20Hz$. The standard eigensystem realization algorithm shows high oscillations and does not match the low frequency behavior of the transfer function. The projected ERA performs better at low frequencies, but also shows obvious mismatches around the resonance frequency.

The leading hundred normalized singular values of both the full Hankel matrix $\mathcal{H}$ and several projected Hankel matrices $\hat{\mathcal{H}}$ are shown in Figure 4.2, right. Note the drastic difference in the decay of the singular values. At the truncation order $r = 30$, the singular values of $\hat{\mathcal{H}}$ have already dropped significantly. In contrast, the singular values of the full Hankel matrix start a rapid decay only after $r \approx 60$. We choose the reduced model order as $r = 30$, and apply standard ERA and TERA.

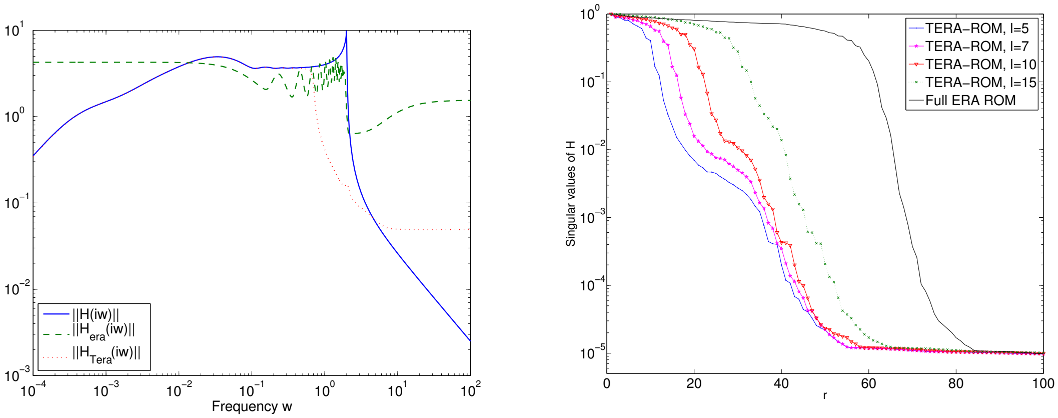

Figure 4.2: MSD model. (left): Transfer function of the full and two reduced models of order $r = 30$ (right): Normed singular values of $\mathcal{H}$ and $\hat{\mathcal{H}}$. Convergence with respect to the number of interpolation directions $\ell_1 = \ell_2$.

First, we note that Theorem 4.3.5 via the upper bound in (4.21) can give valuable insight into the success of TERA. Choosing $r = 30$, and $\ell_1 = \ell_2 = 7$, we obtain

$$\frac{\sum_{i=1}^{2s-1} ||h_i - W_1 \hat{C}_r \hat{A}_r^{i-1} \hat{B}_r W_2^T||_F^2}{\sum_{i=1}^{2s-1} ||h_i||_F^2} = 0.1127$$

and the upper bound in (4.21) yields

$$\frac{4\left(\sum_{i=\ell_1+1}^{p} \sigma_i^2(\Theta_L) + \sum_{i=\ell_2+1}^{m} \sigma_i^2(\Theta_R)\right) + 2\sqrt{r+\ell_1+\ell_2} \cdot \sigma_{r+1}(\hat{\mathcal{H}})}{\sum_{i=1}^{2s-1} ||h_i||_F^2} = 0.8688,$$

thus the bound is in the same order of magnitude. The main contribution to the upper bound results from the truncation of Θ^L and Θ^R.

To compare ERA and TERA, both reduced models are converted back to continuous time, yielding

$$\dot{x}_r(t) = A_{r,c} x_r(t) + B_{r,c} u(t), \qquad y_r(t) \qquad = C_{r,c} x_r(t) + D_{r,c} u(t) \tag{4.24}$$

The system (4.24) was solved from zero to $60s$ with zero initial conditions. The input function was chosen as in [92, Ex.6.3], $u_i(t) = e^{-0.05t} \sin(5t)$ and the input vector consists of 30 copies of the input function.

Figure 4.3 shows outputs six and eleven of time domain simulations. Both ERA and TERA eventually converge to the outputs of the full model, i.e. to the zero steady state. The model predictions of the ERA reduced order model are far from the actual output and hence produce erroneous results in the short term.

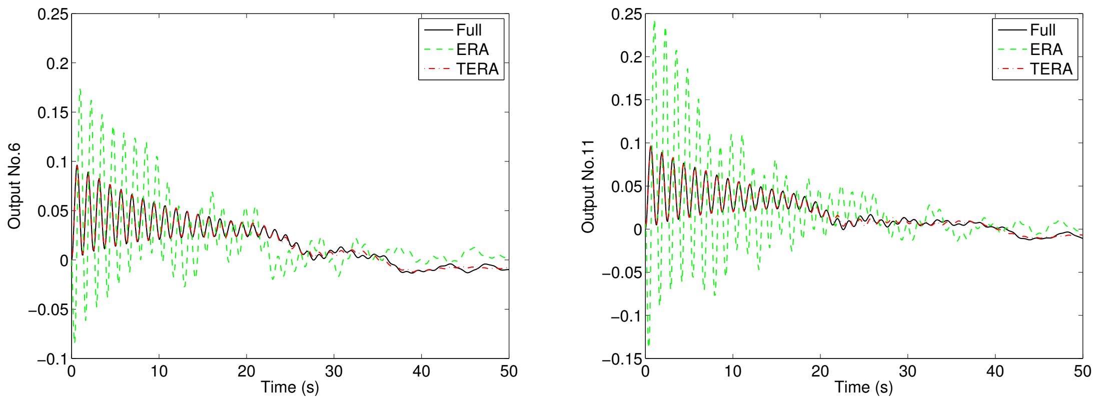

Figure 4.3: MSD model: Outputs of continuous time simulations of the full model, and reduced models with $r = 30$. (left): Output No.6. (right): Output No.1.

Figure 4.3 illustrates that the reduced model order $r = 30$ seems too low for ERA to produce satisfactory results. Based on the plot of the Hankel singular values, Figure 4.2, the singular values of the full Hankel matrix start decaying at order $r \approx 60$. Figure 4.4, compares the continuous time simulations of the full order model, and both reduced order models with $r = 60$ (the left and right interpolation directions for TERA are kept at $\ell_1 = \ell_2 = 7$). The outputs of the ERA model have improved in accuracy compared to the full model. From several numerical experiments, we found that we had to go up to $r = 60$ to have a good match of ERA and the full system.

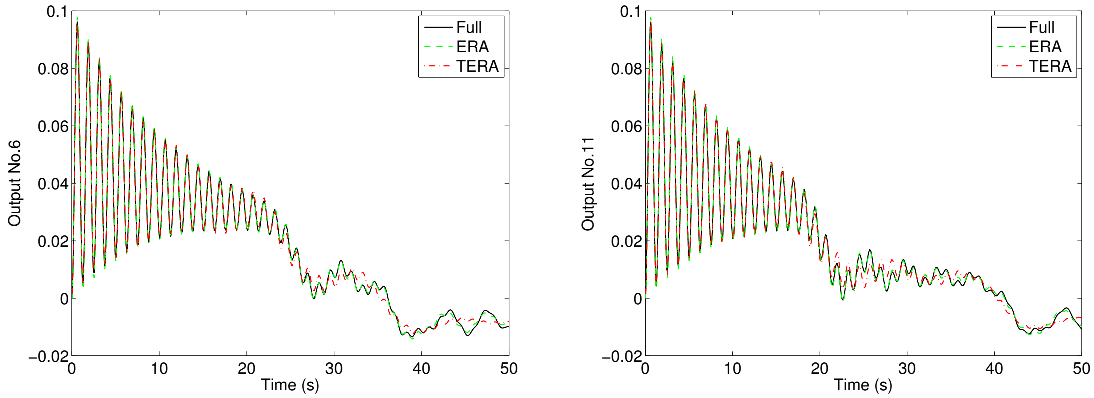

Figure 4.4: MSD model: Outputs of continuous time simulations of the full model, and reduced models with $r = 80$. (left): Output No.6. (right): Output No.1.

In conclusion, TERA produced a better reduced order model than ERA at model order $r = 30$ while reducing the effort for the SVD from a $15,000 \times 15,000$ matrix to a $3,500 \times 3,500$ matrix. At reduced model order $r = 60$, ERA provides a slightly better match of the in terms of the output of time domain simulations, yet it still remains expensive to compute. The advantage of computational effort for TERA is still persistent. Moreover, the reader should note that a careful balance of the number of interpolation directions ℓ_1, ℓ_2, and the reduced order model size r, led to a satisfactory accuracy in the ROM, while saving computational time.

4.4.2 Cooling of Steel Profiles

The model is taken from the Oberwolfach benchmark collection for model reduction [104] and is further described in [180]. The model describes the cooling process of steel profiles in a rolling mill. Different steps in the production process require different temperatures of the raw material. In order to achieve a high throughput and therefore high profitability of the mill, it is necessary to reduce the temperature of the profile as fast as possible to the required level needed for entering the next production phase. However, there are limits to the cooling procedure, so that material properties such as durability and porosity stay within specified quality standards. The cooling is achieved by spraying fluids on the surface of the material. The process is modeled by a two dimensional heat equation with boundary control input. A finite element discretization results in a model (E, A, B, C) with $n = 1357$ states,

$m = 7$ outputs and $p = 6$ outputs. The maximal eigenvalue of the system matrix is $\lambda_{max} = -1.767 \cdot 10^{-5}$, which implies that the Markov parameters will decay slowly. It is therefore necessary to sample many Markov parameters to capture enough of the system dynamics.

The physical problem resides at temperatures of approximately 1000 degrees centigrade down to about 500-700 degrees depending on calculation time.[2] The state values are scaled to 1000 being equivalent to 1.0. This results in a scaling of the time line with factor 100, meaning that computer times have to be divided by 100 to get the real (physical) time in seconds. All plots are given in real time.

Initially, a Cholesky factorization of the mass matrix was performed, $E = LL^T$ and a change of variables yields $L^{-1}AL^{-T} \mapsto A, L^{-1}B \mapsto L$ and $CL^{-T} \mapsto C$. The model is converted to a discrete time model through the bilinear transformation, as described above. The system was simulated for $200,000s$ in simulation time and a Markov parameter sampled every $100s$. This is equivalent to $2000s = 33.3$m in real time and collecting snapshots every second.

Figure 4.5, left, shows the normalized decay of the Markov parameters over time, so $\|h_i\|/\|h_1\|$ for $i = 1 : 20 : 2000$. The plot can guide the choice of when to stop collecting data. Figure 4.5, right, shows the normalized singular values of the matrices Θ^L and Θ^R, respectively. In addition to computational cost limitations, the decay of the singular values gives valuable insight into choosing the tangential truncation orders ℓ_1 and ℓ_2.

The dimension of the reduced order model was chosen a priori to be $r = 20$. First, standard ERA was applied to the sequence of $2s = 2000$ Markov parameters in $\mathbb{R}^{6\times 7}$ requiring an SVD of size 6000×7000. The reduced order matrices (A_r, B_r, C_r, D_r) were obtained in discrete time and converted back to continuous time, where the transfer function $G_r(s) = C_{r,c}(sI - A_{r,c})^{-1}B_{r,c}$ is evaluated. In a second experiment, the Markov parameters are projected with $\ell_1 = \ell_2 = 4$ tangential directions, so that $\hat{h}_i \in \mathbb{R}^{4\times 4}$. Therefore, only a singular value decomposition of size 4000×4000 has to be computed. Note, that the singular value decomposition scales cubic with the matrix dimension n. We applied the TERA and obtained a reduced order model $(\hat{A}_r, \hat{B}_r, \hat{C}_r, \hat{D}_r)$ and again computed the transfer function for the continuous representation. Figure 4.6, left, shows the transfer functions of the full model, and both reduced models, producing indistinguishable results. Figure 4.6, right, shows the convergence of the singular values of the tangentially interpolated Hankel matrix for various values of ℓ_1, ℓ_2, as the dimension of the reduced order model r increases.

[2] Information about the model from IMTEK-simulation [104].

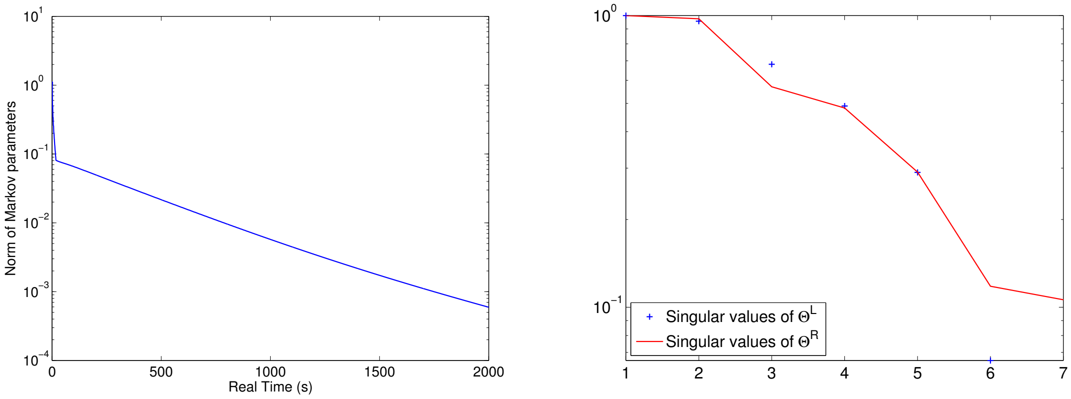

Figure 4.5: Rail model. (left): Norm of the Markov parameters. (right): Singular values of Θ^L and Θ^R.

The continuous time reduced order models (4.24) are simulated with an input vector $u(t) \in \mathbb{R}^m$ with $u_i(t) = 0.2e^{-.005t}$, for $i = 1, \ldots, m = 7$. For time stepping, we used `ode45` in `Matlab` with standard error tolerances. The outputs are compared to the outputs of simulations of the full model, equations (4.22)–(4.23). Figure 4.7 shows outputs one, two and five computed from the full model, as well as the reduced models obtained through both standard ERA and TERA. In addition to reducing the computational time and memory requirements of standard ERA, the TERA framework performs well in time domain simulations.

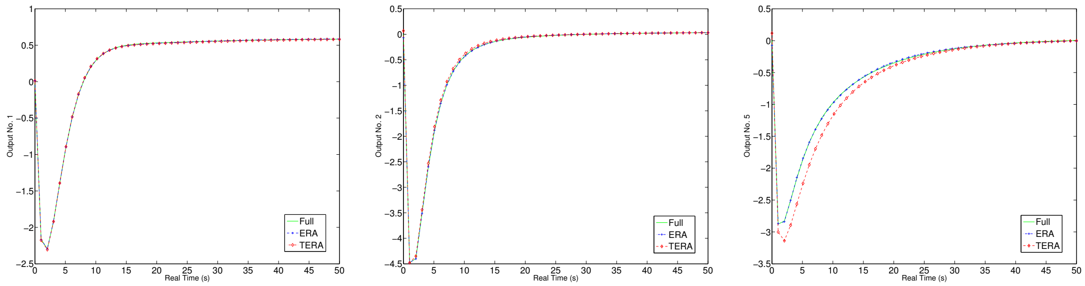

Figure 4.7: Rail model. Time domain simulations of the full and reduced order models. (left): Output No.1. (middle): Output No.2. (right): Output No.5.

The error introduced by the standard eigensystem realization algorithm for the $r =$

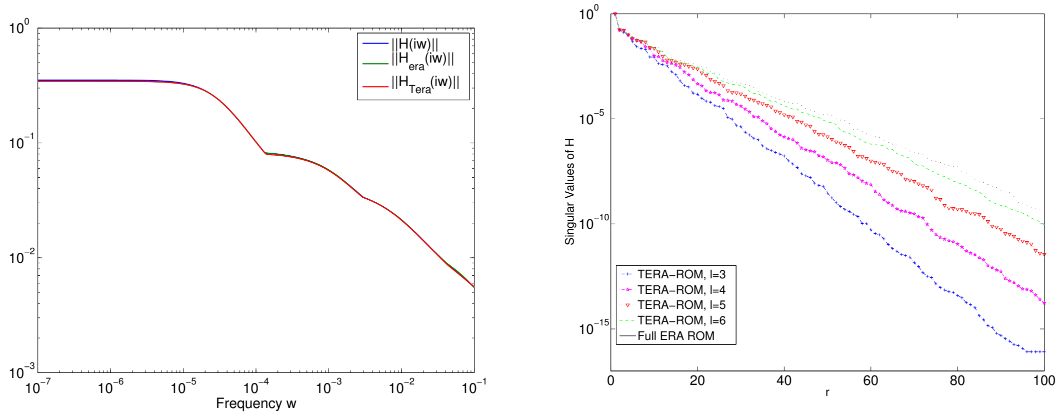

Figure 4.6: Rail model. (left): Transfer function for full model, ERA reduced model and TERA reduced model. (right): Convergence of the (normed) singular values of $\mathcal{H}$ and $\hat{\mathcal{H}}$ for various interpolated models, as r increases.

20 reduced order model (relative to the summed norm of the Markov parameters) is

$$\frac{\sum_{i=1}^{2s} \|C_r A_r^{i-1} B_r - h_i\|_F^2}{\sum_{i=1}^{2s} \|h_i\|_F^2} = 3.5 \cdot 10^{-5},$$

and the error resulting from applying TERA with four tangential directions on both sides is

$$\frac{\sum_{i=1}^{2s-1} ||h_i - W_1 \hat{C}_r \hat{A}_r^{i-1} \hat{B}_r W_2^T||_F^2}{\sum_{i=1}^{2s-1} ||h_i||_F^2} = 3.0 \cdot 10^{-3}$$

Moreover, the upper bound from Theorem 4.3.5 is

$$\frac{4\left(\sum_{i=\ell_1+1}^{p} \sigma_i^2(\Theta_L) + \sum_{i=\ell_2+1}^{m} \sigma_i^2(\Theta_R)\right) + 2\sqrt{r+\ell_1+\ell_2} \cdot \sigma_{r+1}(\hat{\mathcal{H}})}{\sum_{i=1}^{2s-1} ||h_i||_F^2} = 8.5 \cdot 10^{-2}.$$

As we can see from this, and the previous example, the upper bound is not sharp. In fact, Kung's original upper bound is not sharp as well, as we have seen in computations. The actual error in this example is one order of magnitude smaller than the upper bound suggest. We shall also mention, that the error of the interpolated method converges to the original ERA error as the number of tangential directions is increased.

In both numerical experiments, we also generated tangential directions from a random normal distribution. This approach gave unsatisfactory results in several test runs and we therefore safely exclude it as a choice for tangential directions. For illustration purposes, we plotted the results of the continuous time simulations for output two and six in Figure 4.8, where random interpolation directions were used for $\hat{\mathcal{H}}$.

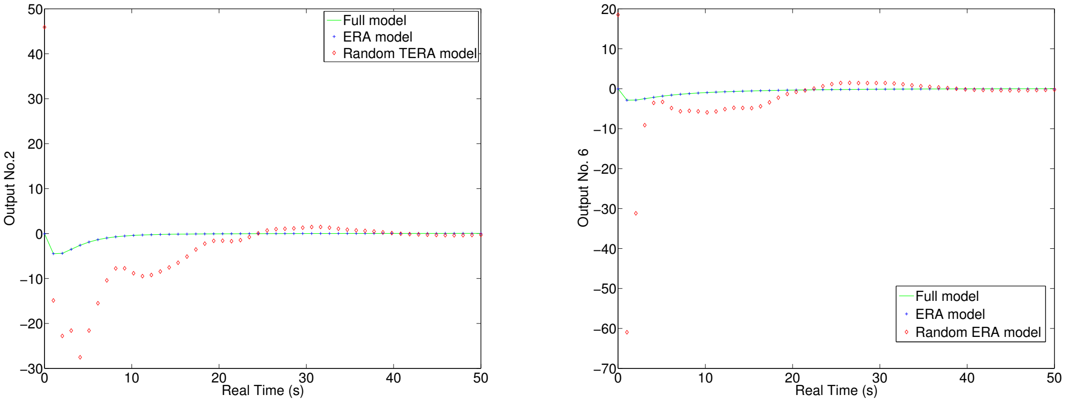

Figure 4.8: Rail model. Time domain simulations of the full and reduced order models. The TERA model was obtained by interpolation with random directions. (left): Output No.2. (right) Output No.6.

4.4.3 Indoor-Air Model for Thermal-Fluid Dynamics

Here, we give a brief heuristic of the limitations of the above approach. As mentioned in Example 4.2.5, an application where the tangential interpolation approach would be beneficial is given by the model for indoor-air behavior [38]. Therein, system matrices are not at hand (which is usually the case for non-academic settings), and therefore we cannot provide the same level of detailed comparison. Nonetheless, one would like to know a priori, whether a tangentially interpolated eigensystem realization would be successful. The tools developed earlier can help us decide whether TERA could be applied here. As mentioned in Example 4.2.5, there are 1437 Markov parameters available from simulations of the complex model, which are generated by $FLUENT$ simulations with an underlying grid of $n \approx 200,000$ finite volume elements used in a three dimensional domain. The version of the model we consider here has $m = 26$ inputs and $p = 19$ outputs. Figure 4.9, left, shows the decay of the singular values of Θ^L and Θ^R to determine the number of necessary interpolation directions. The reader should compare this to Figures 4.5 and 4.1, where a steeper decay in the singular values can be observed. Moreover, since the error bound in Theorem 4.3.5 contains the summed tail of the neglected singular values of Θ^R and Θ^L, the upper bound is large. The second ingredient to the error bound in Theorem 4.3.5 is the first neglected Hankel singular value in the reduced order model. The singular values of the (interpolated) Hankel matrix are shown in Figure 4.9, right. As in the previous examples, the Hankel singular values converge to the true values as we increase the interpolation directions $\ell_1 = \ell_2 = \ell$. However, the convergence is noticeably slower than in the previous two examples. Taken together, one would expect TERA to not yield satisfactory results for low ℓ_1, ℓ_2, which could hint at the fact that all inputs and outputs are highly relevant for this particular model.

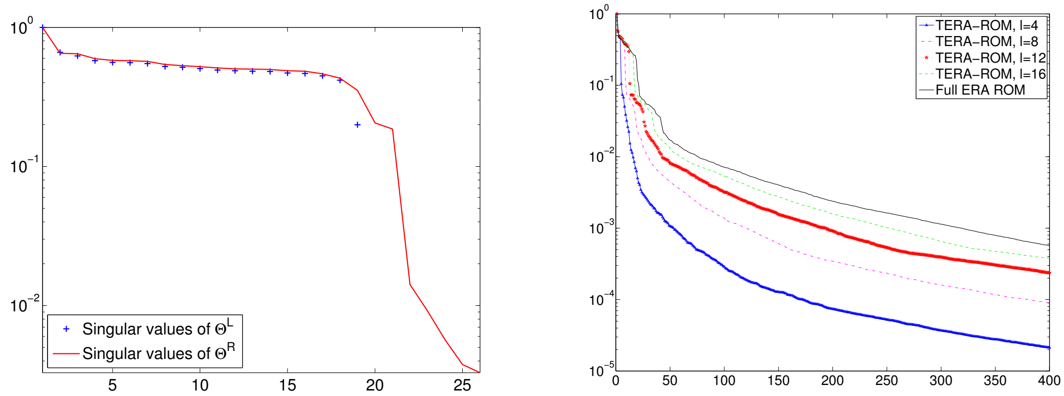

Figure 4.9: Indoor-air model: (left) Singular values of Θ^L and Θ^R. (right): Convergence of the (normed) singular values of $\mathcal{H}$ and $\hat{\mathcal{H}}$ for various interpolated models, as r increases.

4.5 Conclusions

We modified the standard Eigensystem Realization Algorithm to handle MIMO systems efficiently. The input and output dimensions were reduced by tangential interpolation of the impulse response data. The standard ERA was then used on the low dimensional input and output space. The observation and control matrices were injected back to the original input and output dimensions. The computational savings for the necessary singular value decomposition were significant, as demonstrated in the numerical examples, in particular since the complexity of the SVD grows cubic with the size of the Hankel matrix. Moreover, we provided an error bound for the tangentially interpolated version of ERA (called TERA), which clearly showed how both the truncation of the Hankel matrix, as well as of the input and output dimension (via Θ^L and Θ^R) affect the reduced order models.

We would like to note, that there is a significant difference between reducing the input and output dimension and the state space dimension itself. Model reduction techniques for state space systems often reduce systems from 10^{5-8} to only a few hundred modes while retaining a tremendous amount of accuracy. This is partly because the dynamics of such systems are found to be less complex than the model order suggests. In contrast, sensors and actuators (often less than 100 each) are critically important to the design, effectiveness and performance of the plant. In extreme cases, by removing sensors or actuators, stabilizability and detectability of the system can be lost, essentially creating subspaces that cannot be controlled or observed. Thus, input and output projection is a sensitive task and we do not expect an equally impressive order reduction. In this light, the reduction of the inputs and outputs in the above examples is completely satisfactory to us, and allowed to achieve accurate continuous time models. Moreover, we provided an example, where TERA fails, because the singular values of Θ^R and Θ^L rarely converge. This is helpful for practitioners, since it gives a priori information whether TERA should be applied or not.

The numerical findings demonstrate that TERA identifies accurate reduced order models from data only. Thus, the algorithm can run with inputs from experiments or black-box code and accurately identify reduced order dynamics. We shall note that it may be possible to obtain a error bound in line with Kung's result. It would be interesting to see the performance of TERA on experimental data.

Chapter 5

Compressed Sensing and Regime Selection

5.1 Introduction

In this chapter, a new, data-driven algorithm for sensing and classification of complex, coupled, thermally driven airflows is presented [33, 34]. The goal is to use online sensor data, to classify a "dynamic regime" in which the system currently resides. With the availability of large amounts of data from physical systems, we address the challenge of using the valuable information in this (online or offline) data for controller and observer design. Using modern compressed sensing methods, we reduce the amount of online sensor data required for the classification algorithm. Moreover, we propose a new sensing basis using dynamic mode decomposition, which enables us to increase the robustness of the sensing method to sensor noise.

The design of a new generation of passive heating, ventilation and air conditioning (HVAC) systems can tremendously benefit from incorporating airflow dynamics into the control and sensing mechanisms. Utilizing natural convection in this process can save hardware, such as fans, and provides proper air circulation. Fortunately, convective air flows are well studied theoretically (in simple domains), computationally, and experimentally. The Boussinesq approximations are a commonly accepted mathematical model for the dynamics of buoyancy driven flows, when temperature differences are small. Moreover, Computational Fluid Dynamics (CFD) and the availability of vast computational resources allow for accurate simulation of the governing equations, even in complex spatial domains. Due to the lack of an experimental setup, we use CFD simulations to generate data for convective airflows.

Controlling the dynamics of such flows poses numerous challenges. First and foremost, computational resources for control in industrial devices are limited due to space, cost and other feasibility requirements. Thus, in practice, controllers have to be cheaply computable, which we address by using low dimensional models. Second, in the indoor environments considered herein, the geometry, boundary conditions and external disturbances are time-varying. Therefore, controllers and sensing techniques have to be robust to those parametric changes during runtime. Third, real sensors provide noisy measurements, which has to be addressed through noise-robust sensing methods. Our goal is to design efficient, low order controllers that are robust to both parametric changes and noise in the sensing mechanism.

Here, we present a method that takes into account those challenges, and provide numerical results, based on synthetic data, that demonstrate robustness and accuracy of the proposed sensing method. The interplay of compressed sensing (allows for cheap hardware, in that it only needs few sensors) and reduced order modeling (enables computationally cheap software implementation) outlines a promising avenue to this challenging problem. A key to the success of the proposed sensing method is that the considered flow dynamics settles on low dimensional attractors, thus fulfilling the sparsity assumption in compressed sensing. Another important aspect is that one only has to sense what is deemed “necessary” for an effective control synthesis, and not focus on smallest scale phenomena.

The proposed method is summarized as follows: In a first step, we compute the dominant structures in the flow by using Proper Orthogonal Decomposition (POD) [13, 98, 135], or Dynamic Mode Decomposition (DMD) [162, 65]. Those structures represent high energy (POD) or dynamically important (DMD) modes, that are used to sparsely represent the data. In a second step, we design a flow-regime classification algorithm based on compressed sensing [57, 58, 73] and sparse representation to classify and reconstruct flow regimes from few measurements. In particular, we numerically demonstrate the effectiveness of practical boundary sensing techniques and compare the results to using a distributed sensor array.

For decades, considerable attention has been devoted to the field of airflow sensing, observer design and reconstruction from (sparse) measurements. The authors in [1] used few heuristically located sensors in the wake of a cylinder to reconstruct the von Karman vortex street flow pattern. This was possible in large part due to the special structure of this well-known flow field and its dependence on only a few characteristic quantities. A practical sensing technique is considered in [40], where the pressure field of a flow past a cylinder is reconstructed from sparse cylinder-surface measurements. Moreover, the authors in [159, 86] optimize sparse, distributed measurement locations and employ POD to predict the temperature profiles in data storage centers. Taylor and Glauser [176] derive a framework with POD and Linear Stochastic Estimation techniques for remote sensing of flow around a pitched airfoil. The stochastic framework estimates POD coefficients from pressure data on the boundary of the airfoil, and is demonstrated to estimate the true flow field fairly accurately. The authors in [15] reconstruct the flow data from sparse measurements via compressed sensing. Moreover, Willcox [188] introduces "gappy POD" for efficient flow reconstruction, and proposes a sensor selection methodology based on a condition number criterion. In the references [47, 82], sensor placement strategies based on optimization of observability gramians and related system theoretic measures are considered. The resulting locations are distributed in the spatial domain. The theory for those algorithms was developed in the 1970's, see [145], and the references therein. Recently, machine learning approaches for optimal placement of sensors in flow applications have also emerged [118].

5.2 Dynamic Model and Feature Extraction

The model under consideration is given by the Boussinesq equations (1.3)–(1.5) for thermally driven airflow. We consider two different examples, one model of forced airflow, and a model of purely boundary-driven natural convection in a differentially heated cavity. Specifications and details for both models follow at the beginning of §5.4, respectively. A common feature of both models is their parametric dependence, motivating the definition of "dynamic regimes", i.e. states of the flow, where solutions are "similar", in a specified sense. Fortunately, the field of Computational Fluid Dynamics (CFD) offers a wide range of theory and software (`FLUENT,COMSOL,OpenFOAM,NEK5000,`...) for those types of models. Nonetheless, a high fidelity simulation of the Boussinesq equations (for a few minutes of a flow solution) can easily take multiple days to compute. To extract the dominant features of the flow solutions, we use Dynamic Mode Decomposition (DMD) and Proper Orthogonal Decomposition (POD), as introduced in §1.6.

5.3 Regime Classification

As a first step to use reduced order models for controllers and observers of parameter dependent industrial systems, we focus on the problem of classifying operating conditions. Therefore, the flow patterns arising from different geometric configurations, boundary conditions, and parameters, are used to define *regimes*.[1] A library of anticipated flow regimes is defined, and compressed sensing is employed to match present flow conditions to any of those predefined library elements; this step is referred to as *classification*.

Definition 5.3.1. Let $\hat{y} \in \mathbb{R}^p$ be a (possibly noise corrupted) measurement and $\Phi := [\Phi^1 \ \dots \ \Phi^d] \in \mathbb{R}^{n \times R}$ be a library of sparsity basis elements. The *classification problem* is to find

$$\hat{k} = \arg \min_{k=1,\dots,d} ||\hat{y} - C\Phi^k a^k||_2, \tag{5.1}$$

where a^k denotes the unknown coefficients in the sparsity basis Φ^k.

In other words, one wants to find the regime in the library that best represents the data. The problem statement is of course valid for $p = n$, i.e. for full state information. In this work, however, we consider partial (sensed) information only, hence the focus on measurements in the above definition.

5.3.1 Library Generation

The generation of a sparsity basis for the dynamic regimes is accomplished through POD and DMD. Let $\mathcal{Q} = \{q_1, q_2, \dots, q_d\}$ denote a set of d different configurations (parameters, boundary conditions, geometry). For each configuration, simulation data is generated from the Boussinesq (or Navier-Stokes) equations as

$$X(q_i) \in \mathbb{R}^{n \times s} \qquad q_i \in \mathcal{Q},$$

where each column of $X(q_1)$ is a snapshots of the solution from parameter q_i. The solutions are generated on an equidistant time grid. Through DMD or POD, we compute r_i basis functions for every regime, $i = 1, \dots, d$, which yields the sparse library (or dictionary)

$$\Phi := [\Phi^1 \ \dots \ \Phi^d] \quad \in \mathbb{R}^{n \times R},$$

where $R = \sum_{i=1}^d r_i$ and for notational convenience $\Phi(q_k) := \Phi^k$.

[1] A precise definition of dynamic regimes is not available in the current literature and we shall use the word with a certain vagueness.

5.3.2 Sensing Matrix

Sensors typically collect local information, therefore motivating the use of point measurements in a mathematical setting. In particular, for sensing indoor-air flows, we restrict ourselves to the boundaries of the domain, e.g. walls or ceilings. A point measurement matrix C is defined as

$$C_{i,j} := \{0, 1\}, \qquad \sum_{j=1}^{n} C_{i,j} = 1, \qquad \sum_{i=1}^{p} C_{i,j} = 1.$$

In compressed sensing, it is customary to use Gaussian or Bernoulli matrices C, since they satisfy the restricted isometry property, see §1.7. Unfortunately, such sensor arrays are impractical for flow sensing applications. At best, averaged velocities over a small spatial region are sensed, which would lead to a sequence of entries in the measuring matrix. This is left for further study.

5.3.3 Classification

The problem of identifying the active library block that represents the current data is discussed below. Assume that a sample of the state x^k of the k^{th}-regime is given. The measurements available for classification are

$$\hat{y} = Cx^k + \eta, \tag{5.2}$$

which are corrupted with white sensor noise η with zero mean. The goal then becomes to identify the regime best matching the data $\hat{y}$, which ideally would be the k^{th} regime. If for some reason the algorithm chooses another regime, we say that it *confused* the regime. To construct an algorithm to achieve minimal confusion, one has to address the following questions:

- How many measurements (sensors) p are necessary for successful classification?
- How can one make the algorithm robust to sensor noise?
- Which a priori conditions on the dictionary Φ and sensing mechanism C guarantee, or give high confidence, that the classification works?

The above questions are addressed in the remainder of this section. First, we begin with stating the decision algorithm for classification. By using the sparse basis (DMD and POD), we can express the unknown state as

$$x^k = \Phi^k a^k, \quad a^k \in \mathbb{C}^{r_k}, \quad r_k \ll n,$$

where again k labels the regime membership. In a next step, define

$$\Theta^k := C\Phi^k \quad \in \mathbb{C}^{p\times r_k},$$

which contains the rows of the extracted (low order) features available to the sensing mechanism. With this definition, the library of sensed regimes becomes

$$\Theta = \left[\Theta^1 \; \Theta^2 \ldots \Theta^d\right]. \tag{5.3}$$

Note that $p > r_k$ is a necessary requirement for any sensing algorithm, since all r_k coefficients a_k are unknown, and we do not assume additional structure amongst the coefficients. Consequently, multiplying (5.2) with $[\Theta^k]^*$ yields

$$(\Theta^k)^*\hat{y} = (\Theta^k)^*\Theta^k a^k + (\Theta^k)^*\eta,$$

and $(\Theta^k)^*\Theta^k$ is then invertible, which can be ensured by selecting appropriate sensor locations. In other words, if linear independent measurements are taken, the matrix $(\Theta^k)^*\Theta^k$ is invertible. This can be guaranteed with high probability with a random array of sensors. Thus, the above equation is rewritten as

$$[(\Theta^k)^*\Theta^k]^{-1}(\Theta^k)^*(\hat{y} - \eta) = a^k.$$

The classifier should be defined on the measurement space $\mathbb{R}^p$. The injection into $\mathbb{R}^p$, reads as

$$\Theta^k[(\Theta^k)^*\Theta^k]^{-1}(\Theta^k)^*(\hat{y} - \eta) = y^k,$$

and $y^k = Cx^k$ is the uncorrupted measurement. This procedure defines an orthonormal projection onto the space of measurements for the k^{th} regime as

$$P_k := \Theta^k[(\Theta^k)^*\Theta^k]^{-1}(\Theta^k)^*. \tag{5.4}$$

To classify a signal to a subspace, consider the norm of the projection to each subspace $||P_k\hat{y}||_2$ and pick the maximum of all the candidates. Then, the estimated subspace is given as

$$\hat{k} = \arg\max_{k=1,\ldots,d} ||P_k\hat{y}||_2, \tag{5.5}$$

which then yields the regime that aligns most with the current, noise corrupted data. Once the best matching library regime is found, the coefficients a^k can be recovered via a least squares solution

$$a^k = (\Phi^k)^\dagger\hat{y},$$

where $(\Phi^k)^\dagger$ denotes the Moore-Penrose pseudoinverse of Φ^k. The above algorithm produces excellent results in practice and is robust to noise, as we see in §5.4. Moreover, there is a simple sufficient condition, under which this holds true. While this might give a conservative bound, it provides valuable insight into techniques and success of the above method. Consider d subspaces $\{\mathcal{W}_k, k = 1, \dots, d\}$ with bases $\{\Phi^1,\ \Phi^2, \dots, \Phi^d\}$ and the corresponding projection matrices $\{P_1, P_2, \dots, P_k\}$, as computed in 5.4. A signal approximately lies in a single subspace, $\hat{k}$, under the following model:

$$x = x_{in} + x_{out},\ \ x_{in} \in \mathcal{W}_{\hat{k}},\ \ x_{out} \perp \mathcal{W}_{\hat{k}}, \tag{5.6}$$

where x_{in} and x_{out} denote the in-subspace and the out-of-subspace components, respectively. For classification, we solve the optimization problem (5.5). For the following proposition, we define

$$\eta_{ij} := \max_{i \neq j} ||P_i P_j||_2 = \max_{i \neq j} \frac{||P_i P_j x||_2}{||x||_2}, \quad i, j \in 1, \dots, d, \tag{5.7}$$

as a measure for the alignment of two subspaces.

Proposition 5.3.2. *Let d subspaces $\mathcal{W}_k,\ k = 1, \dots, d$ be given, and let the signal $x \in \mathcal{W}_{\hat{k}}$ for some $\hat{k} \in 1, \dots, d$ according to (5.6). Moreover, assume that $\|x_{out}\|_2^2 \leq \epsilon \|x\|_2^2$, with $\epsilon < 1/2$ and let η_{ij} be defined as above. Then, if*

$$\eta_{\hat{k}k} < \sqrt{1 - \frac{\epsilon}{1-\epsilon}}, \quad \forall\ k \neq \hat{k}, \tag{5.8}$$

the classification in (5.5) is successful.

Before proceeding to the proof of the proposition, we would like to comment on this result. The constant ϵ is an upper bound for the relative energy of the given signal x in the subspace $\mathcal{W}_{\hat{k}}$. The reader should note, that for a given set of d subspaces, the η_{ij} are computable quantities, and one can compute $\eta_{max} = \max_{i \neq j} \eta_{ij}$. Therefore, one can only expect to be able to classify signals that satisfy (5.8) for η_{max}. This in turn gives a priori guidance, whether a dictionary is suitable for classification, or if fewer or more appropriate regimes should be included to account for signals that might be far off any of the present subspaces, in a vector alignment sense. In the example below, we show that the requirement of which signals to classify influences the choice of the dictionary.

Example 5.3.3. Let two subspaces, $\mathcal{W}_k$ and $\mathcal{W}_{\hat{k}}$ be given, and consider a signal with energy of 90% in the subspace $\mathcal{W}_{\hat{k}}$. Then $\epsilon = .1$ and $\frac{\epsilon}{1-\epsilon} = \frac{1}{9} \Rightarrow 1 - \frac{\epsilon}{1-\epsilon} = \frac{8}{9}$.

Consequently, if $\eta_{\hat{k}k} < \sqrt{\frac{8}{9}} \approx .94$, we guarantee correct classification of the signal x to the subspace $\mathcal{W}_{\hat{k}}$.

Proof. Let $x = x_{in} + x_{out},\ x_{in} \in \mathcal{W}_{\hat{k}},\ x_{out} \perp \mathcal{W}_{\hat{k}}$, as above. Since $x_{out} \perp \mathcal{W}_{\hat{k}}$, the projection on the correct subspace has norm

$$\|P_{\hat{k}}x\|_2^2 = \|x_{in}\|_2^2.$$

The projection to the other subspaces k is bounded through the obvious

$$||P_k x_{out}||_2 \leq ||P_k||_2||x_{out}||_2 = ||x_{out}||_2.$$

Additionally, we have that $x_{in} + x_{out} = P_{\hat{k}}x_{in} + x_{out}$, which yields the following estimate:

$$\|P_k x\|_2^2 \leq \|P_k P_{\hat{k}} x_{in}\|_2^2 + \|P_k x_{out}\|_2^2 \leq \eta^2\|x_{in}\|_2^2 + \|x_{out}\|_2^2.$$

The classification is considered accurate (sufficient condition), if the projection onto regime $\hat{k}$ retains the most information of the signal, in other words, if

$$||P_k x||_2 \leq ||P_{\hat{k}}x||_2, \quad \forall \hat{k} \in \{1, \ldots, d\}, \hat{k} \neq k,$$

which holds true whenever

$$||P_k x||_2 \leq \eta^2\|x_{in}\|_2^2 + \|x_{out}\|_2^2 \overset{!}{<} ||P_{\hat{k}}x||_2 = \|x_{in}\|_2^2,$$

and consequently the condition can be rewritten as

$$\|x_{out}\|_2^2 < (1 - \eta^2)\|x_{in}\|_2^2.$$

Assume that at most a portion ϵ of the signal lies out of the correct subspace $\|x_{out}\|_2^2 \leq \epsilon\|x\|_2^2$, which then yields

$$\|x_{in}\|_2^2 = \|x - x_{out}\|_2 \geq \|x\|_2 - \|x_{out}\|_2 \geq (1 - \epsilon)\|x\|_2^2.$$

Next, note that $(1 - \eta^2)\|x_{in}\|_2 \geq (1 - \eta^2)(1 - \epsilon)\|x\|_2$, and if we assume that

$$(1 - \eta^2)(1 - \epsilon)\|x\|_2^2 > \epsilon\|x\|_2^2$$

the sufficient condition holds true and we obtain correct classification. For $\epsilon < 1/2$, this is equivalent to

$$\eta < \sqrt{1 - \frac{\epsilon}{1 - \epsilon}}.$$

Note, that we can define $SOR = \frac{\epsilon}{1-\epsilon}$, where SOR is the subspace-to-outside ratio, so $\eta < \sqrt{1 - SOR}$ is the sufficient condition. □

5.3.4 Theory of Block Sparse Recovery

The concepts of block-coherence (between different regimes) and sub-coherence (within each regime) of the sparse library (5.3) can help to understand the classification performance and requirements on the library, as follows. In [80], a theory of block sparse recovery is considered, and we shall briefly state the relevant results therein. Casting the recovery problem into the block-sparsity framework, requires the following definition.

Definition 5.3.4. Let $R = \sum_{i=1}^{d} r_i$ and $a(q_i) := a^i \in \mathbb{C}^{r_i}$ be a vector. The blockwise vector $a = [a^*(q_1)\ a^*(q_2)\ \dots\ a^*(q_d)]^* \in \mathbb{C}^R$ is called *block k-sparse*, if exactly k of its blocks $a(q_i)$ are nonzero.

By the above definition, we have that $y = \Theta a$, with only few coefficients in a being nonzero. We are interested in conditions on the sensed library Θ, such that a block-sparse recovery of the vector a from k-measurements is possible. However, the results below apply to the general case of $x = \Phi a$ of full state reconstruction as well.

Definition 5.3.5. [80] The *block-coherence* of the library Θ is defined as

$$\mu_B := \max_{i,j \text{ s.t. } i \neq j} \left[\frac{1}{r} \sigma([\Theta^i]^* [\Theta^j]) \right], \tag{5.9}$$

where it is assumed that $r_i = r$ for $i = 1, \dots, d$. Moreover, $\sigma(A)$ denotes the spectral norm, i.e. the largest singular value of A.

Definition 5.3.6. [80] The *sub-coherence* is a property of the individual library blocks, and defined as

$$\nu := \max_{l} \max_{i,j \text{ s.t. } i \neq j} ||\theta_i^* \theta_j||_2, \qquad \text{s.t. } \theta_i = \Theta^l(:, i).$$

In other words, the sub-coherence is an expression of the orthogonality of the dictionary elements (blocks). Note, that $\nu = 0$ if the bases within each regime are orthogonal, although this is not required. In particular, when using DMD as a feature extraction method, the basis functions are not orthogonal, and therefore $\nu \neq 0$. Given the previous definitions, the main result is quoted below.

Theorem 5.3.7. *[80, Thm.3] A sufficient condition*[2] *to recover the k-sparse vector a from $y \in \mathbb{R}^p$ measurements via the library $\Theta \in \mathbb{C}^{p \times R}$ is*

$$k \cdot r < \frac{1}{2} \left(\frac{1}{\mu_B} + r - (r-1) \frac{\nu}{\mu_B} \right),$$

[2] For certain recovery algorithms, such as Block-OMP

where it is assumed that all library blocks have the same number of elements, namely r. Since we are interested in $k = 1$ sparse solutions (classification of one regime), the above inequality simplifies to

$$r < \frac{1+\nu}{\mu_B + \nu}.$$

Note, that the above result provides a *sufficient* condition, and might be conservative.

5.3.5 Augmented DMD - Opportunities for Robust Sensing

The dynamic mode decomposition is a model reduction/feature extraction technique as introduced in §1.6.3. To present the main idea, we first consider a single regime, say k, and drop the subscripts. Later, we extend the approach to the classification problem for multiple regimes. Recall from equation (1.47), that $\Phi = \Phi^k$ approximates the eigenvectors of the linear advance operator A, such that

$$A\Phi = \Phi\Lambda, \qquad \Phi \in \mathbb{R}^{n\times r},$$

where $\Lambda \in \mathbb{C}^{r\times r}$ denotes the diagonal matrix of the first r DMD eigenvalues. Here, we develop a method to incorporate this property into the sensing mechanism. With the concept of an *augmented basis*, we use *batches* (time-trajectories) of data (consecutive measurements) to classify a single regime. In several numerical experiments, this approach increases both the classification performance, as well as the robustness of the process to sensor noise.

Consider a state vector $x_t = x(t)$, sampled from the underlying discrete dynamical system. The state is expressed in the sparse DMD basis as

$$x_t = \Phi\beta,$$

where $\beta = \beta(t)$ is the unknown vector of coefficients.[3] It follows from the previous two equations, that

$$x_{t+1} = Ax_t = A\Phi\beta = \Phi\Lambda\beta,$$

and recursively

$$x_{t+2} = Ax_{t+1} = A\Phi\Lambda\beta = \Phi\Lambda^2\beta.$$

[3] $\beta \neq a$, since the time is picked arbitrary. Moreover, the basis Φ depends on the data and time sampling frequency Δt. Therefore, in practical sensing, this sampling should be kept the same as the one used for generation of the basis.

By iterating this process one can easily see that $x_{t+j} = \Phi\Lambda^j\beta$, which we incorporate into the sensing mechanism. Therefore, subsequent snapshots can be expressed via the *same* β. In the classification setting below, this implies that in particular, β has the same block sparsity pattern over a few time steps, which allows us to use more data to make a confident classification decision. The above information can be written in batch-form as

$$\begin{bmatrix} x_t \\ x_{t+1} \\ \vdots \\ x_{t+j} \end{bmatrix} = \begin{bmatrix} \Phi\beta \\ \Phi\Lambda\beta \\ \cdots \\ \Phi\Lambda^j\beta \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 & \dots & \phi_r \\ \lambda_1\phi_1 & \lambda_2\phi_2 & \dots & \lambda_r\phi_r \\ \vdots & \vdots & & \vdots \\ \lambda_1^j\phi_1 & \lambda_2^j\phi_2 & \dots & \lambda_r^j\phi_r \end{bmatrix} \cdot \beta.$$

Next, define the *augmented DMD basis vector* as

$$\hat{\phi}_i^j := \begin{bmatrix} \phi_i \\ \lambda_i\phi_i \\ \vdots \\ \lambda_i^j\phi_i \end{bmatrix} \in \mathbb{C}^{jn}, \quad \text{with} \quad \hat{\phi}_i^0 := \phi_i,$$

so that the previous equation can be rewritten as

$$\underbrace{\begin{bmatrix} x_t \\ x_{t+1} \\ \vdots \\ x_{t+j} \end{bmatrix}}_{x_t^{t+j}} = \underbrace{\begin{bmatrix} \hat{\phi}_i & \hat{\phi}_i^1 & \dots & \hat{\phi}_i^j \end{bmatrix}}_{\hat{\Phi}} \cdot \beta.$$

By considering the outputs of the dynamical system, $y_t = Cx_t$, the recursion remains unchanged. To this end, let $C \in \mathbb{R}^{p\times r}$ be a given sensing matrix, and define $\mathcal{C} :=$ blkdiag$(C, \dots, C)$ having k copies of C on its diagonal. Similarly, we define $y_t^{t+j} = \mathcal{C}x_t^{t+j} \in \mathbb{R}^{pj}$. Recall, that the derivation is for a single dynamic regime say i, so $=$ $\hat{\Phi} = \hat{\Phi}^i$, yielding

$$y_t^{t+j} = \mathcal{C}\hat{\Phi}^i\beta.$$

Extension to the classification problem. The previous approach can be straightforwardly extended to the classification problem. Assume that r_i DMD modes $\Phi^i = \Phi_{r_i}(q_i)$ and eigenvalues $\Lambda_i = \Lambda_{r_i}(q_i)$ of each dynamic regime i are computed, so that the augmented DMD library reads as

$$\hat{\Phi} := \begin{bmatrix} \Phi^1 & \Phi^2 & \cdots & \Phi^d \\ \Phi^1\Lambda_1 & \Phi^2\Lambda_2 & \cdots & \Phi^d\Lambda_d \\ \vdots & \vdots & \vdots & \vdots \\ \Phi^1\Lambda_1^j & \Phi^2\Lambda_2^j & \cdots & \Phi^d\Lambda_d^j \end{bmatrix}. \tag{5.10}$$

The problem of regime classification can then be recast as

$$\hat{i} = \arg \min_{i=1,\dots,d} ||y_t^{t+j} - \mathcal{C}\hat{\Phi}^i\beta||_2, \tag{5.11}$$

and the reader should note, that $\beta \in \mathbb{C}^R$ is 1-block-sparse and $y_t^{t+j} \in \mathbb{R}^{pj}$ is the available data. Therefore, the augmented classification problem incorporates more system data over time, i.e. dynamics, with the same number of sensors (the states are sensed at the same spatial locations, and the time evolution of the sensor data is recorded). The subsequent numerical part demonstrates the effectiveness of this approach for various airflow sensing test problems.

5.4 Numerical Results

Two test cases are considered herein - A data set from a `FLUENT` simulation of an indoor-air environment with forced-convection and a natural convection dataset obtained from spectral element solver `NEK5000`. Only few results are available in the literature for classification of dynamic regimes via compressed sensing, see [40] and [42, Table 6]. Our results compare to those publications, and often show even improved classification. However, the algorithms were not compared on the same examples.

5.4.1 Two dimensional Navier-Stokes

We consider velocity data $\boldsymbol{u}(t_i)$ from a Navier-Stokes simulation in an enclosed domain. The physical domain is a room of height $H = 16$ft, length $L = 8$ft, and one inlet on the top left corner and two outlets on the center right (window) and bottom left, all of height $w_i = w_o = 1$ft. A schematic of the room is shown in Figure 5.1 below. We consider a two-dimensional problem, and therefore assume that the dimension normal to the plane (z-direction) is much larger than H, and that boundary effects from the wall in z-direction are negligible. In other words, the cross section can be thought of as located in the center of a long z-dimension. The Reynolds number is defined as $Re := \boldsymbol{u}_{in} \cdot (2w_i)/\nu$ with $\nu = \mu/\rho$ is the kinematic viscosity. Here, $\mu = 1.7894 \cdot 10^{-5} \frac{Ns}{m^2}$ and $\rho = 1.225 \frac{kg}{m^3}$, which are typical values for air, and w_i, the height of the inlet was set as $1ft = 0.3048m$. The term $2w_i$ then denotes the hydraulic diameter. The Reynolds number is the ratio of inertia and viscous forces so for larger values of Reynolds the impact of viscosity is less pronounced and the flow is governed by the convective terms in Navier-Stokes equations. The boundary conditions on the wall are assumed no slip, i.e. $\boldsymbol{u} = \boldsymbol{v} = 0$. Here, $\boldsymbol{u}$ and $\boldsymbol{v}$ are the horizontal and vertical components of velocity. On the inlet, the velocities are prescribed, and the resulting Reynolds numbers are given in Table 5.1, for which the flow is laminar. Note, that the for regimes two and four, the inflow velocity is ten times the value of regimes one and three. Those four scenarios define the flow regimes to be distinguished in the sensing experiment. Initially, there is no flow and the temperature resides uniformly at $300K$. The flow field and contour plots of a snapshot of the simulations at $600s$ of the four flow regimes is shown in Figures 5.2–5.3.

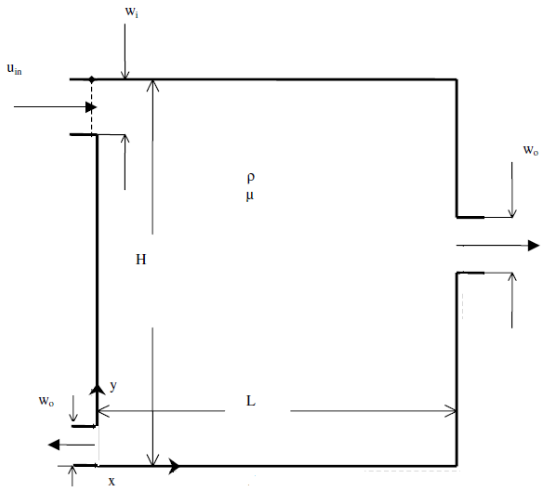

Figure 5.1: Geometry for indoor room, inlets, outlets and window.

The reader should note, that despite an open window in case four, the flow does not leave the room. Due to the large inlet velocity, the velocities in the room are large, and therefore cause low pressures around the window area, which causes the flow to continue downstream. However, the formation and arrangement of vortices are non-trivially different compared to case 2 (window closed).

`FLUENT` uses an optimized, unstructured spatial grid so solve the Navier-Stokes equa-

Regime	Re	Bottom outlet	Window
R1	104.3	open	closed
R2	1043	open	closed
R3	104.3	open	open
R4	1043	open	open

Table 5.1: 2D Navier-Stokes: Flow regimes are defined via the Reynolds number Re, where the inflow velocity is varied by a factor of 10.

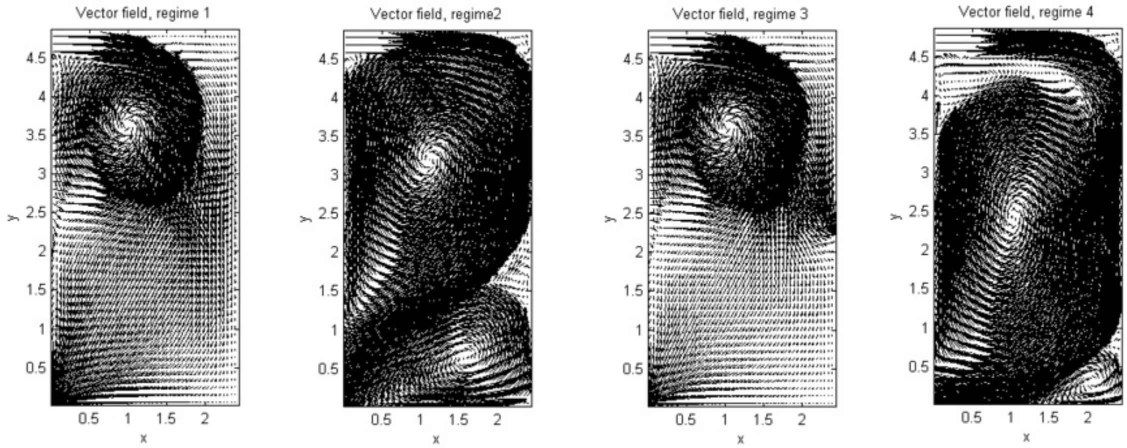

Figure 5.2: Velocity field to solution of Navier-Stokes equations for regimes R1–R4 (cases), as given in Table 5.1 at 600s.

tions, which can vary across regimes. However, the sensor locations should not be affected by this, and hence we interpolated the solution data on a regular 49 × 49 grid, whose nodes are possible sensor locations (in case of distributed sensing). As a technical note, the boundaries of the interpolated grid are chosen slightly away from the physical domain, in order to sense the flow there (The flow in x, y direction on the walls would be zero, due to the no-slip boundary condition). DMD and POD modes were computed from the interpolated data.

Classification of the aforementioned flow scenarios is performed through 100 experiments for every regime. At each experiment per regime, a snapshot of the simulation data for this regime is picked, and $p = 30$ measurement locations are chosen at random from the set of boundary nodes. At the sensor locations, u_x, u_y are measured. Consequently, $2p$ measured values are available. Moreover, white noise is added with a signal-to-noise-ratio of $SNR = 10$. We compare standard DMD and POD for gen-

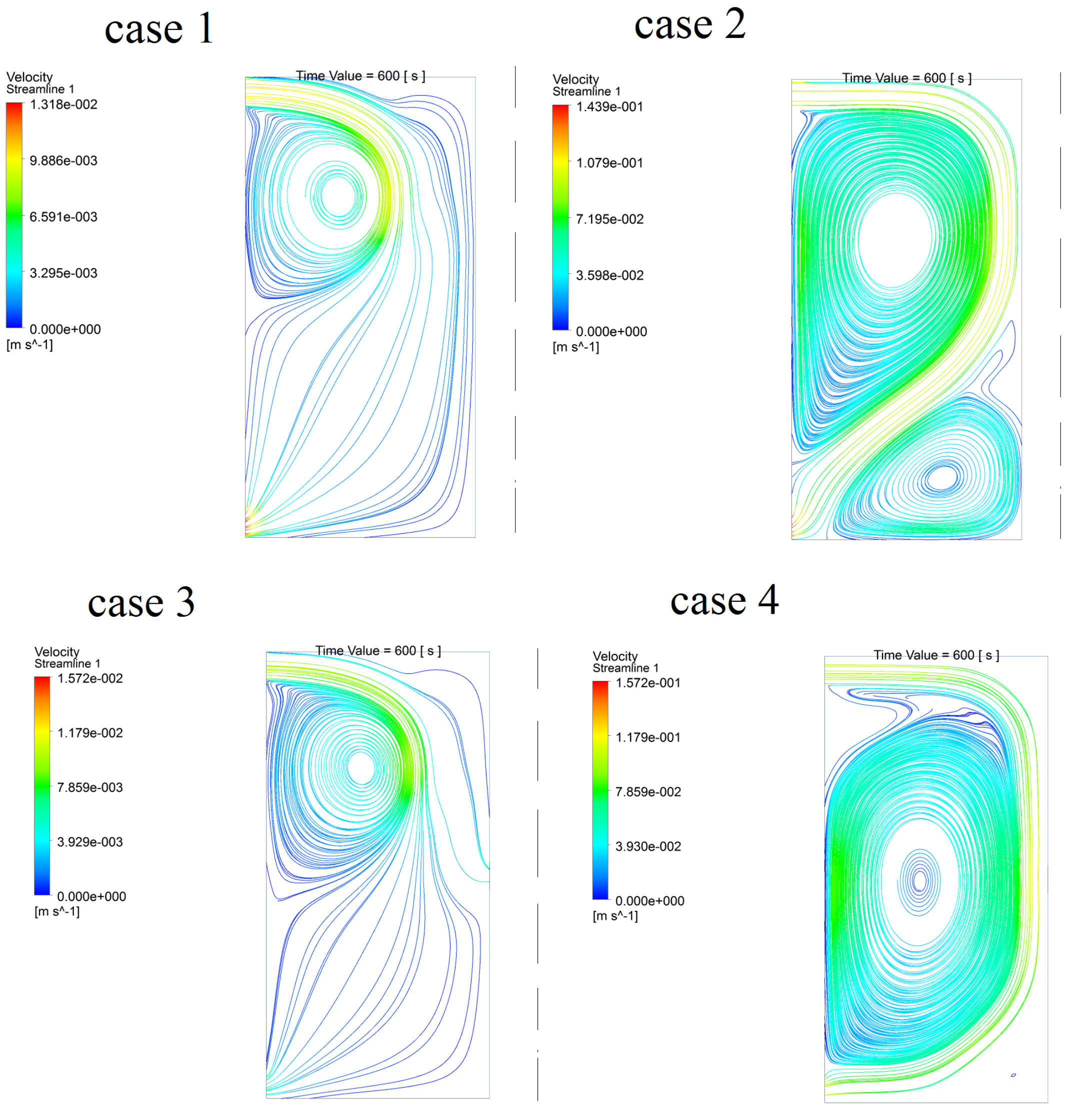

Figure 5.3: Contour plot of solution to Navier-Stokes equations for regimes R1–R4 (cases), as given in Table 5.1 at 600s.

	R1	R2	R3	R4
R1	98.0%	2.0%	0%	0%
R2	1.0%	97.5%	0%	1.5%
R3	0.5%	0%	99.5%	0%
R4	0%	0.5%	4.0%	95.5%

Table 5.2: 2D Navier-Stokes: Confusion matrix for classification with $p = 30$ boundary sensors using DMD basis.

	R1	R2	R3	R4
R1	88.5%	0%	11.5%	0%
R2	0%	99.5%	0%	0.5%
R3	9.0%	0.5%	90.5%	0%
R4	0%	0.5%	0%	99.5%

Table 5.3: 2D Navier-Stokes: Confusion matrix for classification with $p = 30$ boundary sensors using POD basis.

eration of the library. For DMD, $r_i = 6$ basis functions are computed, and for POD, the energy level is set to $\mathcal{E}_{min} = 99.9\%$, resulting in $r_1 = 3,\ r_2 = 7, r_3 = 3, r_4 = 6$ POD basis functions in the respective regimes. Tables 5.2 - 5.3 show the confusion matrices for boundary sensing with POD, and DMD, respectively. The reader should observe, that the classification performance increases when using the DMD modes, which we have seen in many experiments.

Effect of Augmented DMD. We investigate the robustness to noise of the basis functions, where the effect of the augmentation of the DMD basis is particularly noticeable. Therefore, we increase the sensor noise to $SNR = 5$, and reduce the number of flow measurements to $p = 10$. The confusion matrix in this case is shown in Table 5.4, and we can see that the confusion between regimes increased significantly. Below, we demonstrate that augmentation of the DMD basis improves classification performance in the presence of significant noise.

In Table 5.5 the confusion matrix is shown for the case of $j = 4$, hence the DMD basis is augmented as in equation (5.10) by four additional block rows. Moreover, Table 5.6 shows the confusion matrix for the case of $j = 7$, i.e. the DMD basis is augmented by seven more blocks. In other words, seven consecutive time samples from the dynamical system are taken to classify the current regime, which clearly

	R1	R2	R3	R4
R1	47.5%	27.5%	13.5%	11.5%
R2	10.5%	61.5%	6.0%	22.0%
R3	17.5%	11.5%	52.5%	18.5%
R4	5.0%	13.0%	12.5%	69.5%

Table 5.4: 2D Navier-Stokes: Confusion matrix for classification with $p = 10$ boundary sensors and high noise using DMD basis.

	R1	R2	R3	R4
R1	89.5%	3.5%	6.5%	0.5%
R2	2.5%	88.0%	1.0%	8.5%
R3	7.0%	1.5%	90.0%	1.5%
R4	1.5%	2.5%	1.0%	95.0%

Table 5.5: 2D Navier-Stokes: Confusion matrix for classification with $p = 10$ boundary sensors, high noise and augmented DMD basis with $j = 4$ consecutive time measurements.

contributes to the robustness of the sensing method.

Subsequently, we investigate the convergence behavior of the identification as the number of measurements p increases, as well as convergence when using a DMD basis augmented by j block-rows. The parameters for this test problem are set as follows: The overall state dimension is $n = 4801$, every library block in DMD contains $r_i = 6$ basis functions. The number of sensors was increased as $p = 5 : 1 : 30$ and the basis was augmented by $j = 0 : 1 : 9$ blocks. One hundred experiments were performed, and the results averaged over those. In every element, a random selection of the boundary nodes for flow sensing was chosen, and the white noise generated

	R1	R2	R3	R4
R1	93.5%	2.0%	4.5%	0%
R2	1.5%	96.5%	0%	2.0%
R3	6.5%	0%	93.0%	0.5%
R4	0.5%	3.0%	0.5%	96.0%

Table 5.6: 2D Navier-Stokes: Confusion matrix for classification with $p = 10$ boundary sensors and high noise using augmented DMD basis with $j = 7$ consecutive time measurements.

with a signal-to-noise-ratio of $SNR = 20$.

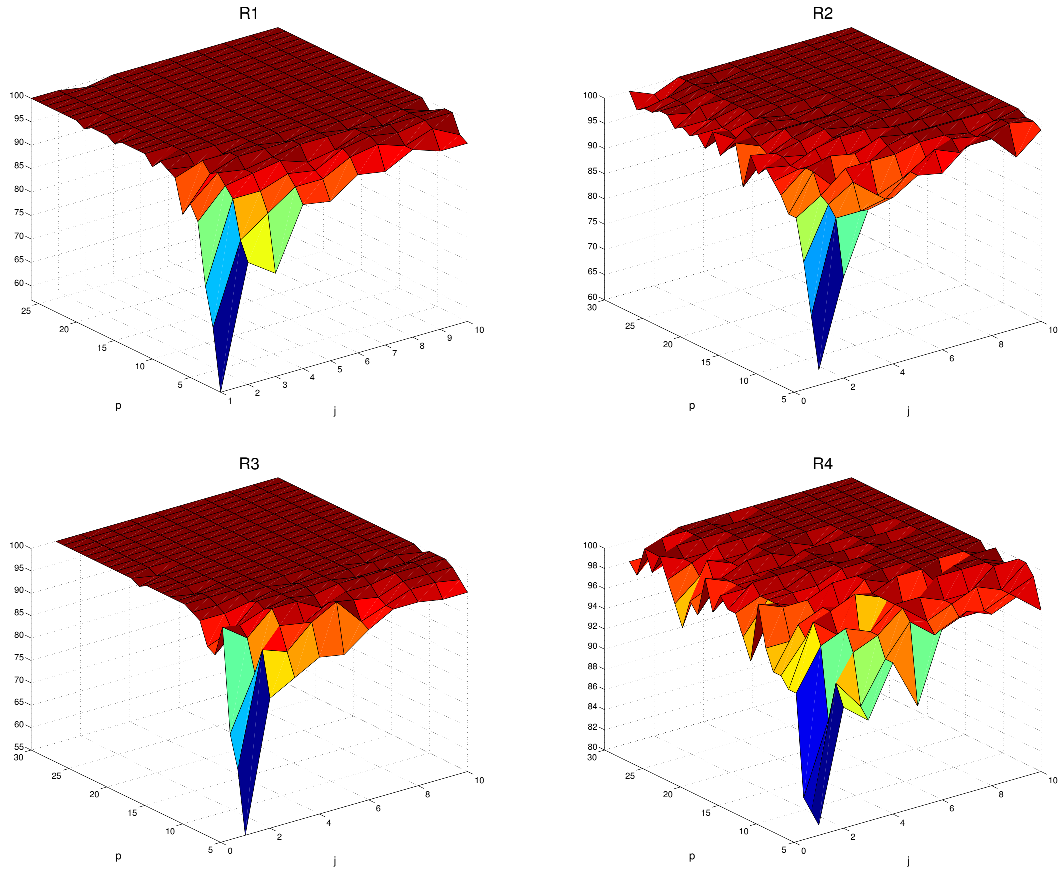

Figure 5.4: 2D Navier-Stokes: Confusion plot for the four regimes R1–R4 as the number of boundary sensors p increases, and the DMD basis is augmented with j additional blocks.

As we can see from Figure 5.4, the identification quality converges rapidly with the number of sensors as well as the augmentation of the basis function. One should note, that especially for a small number of measurements, the augmented DMD approach improves results significantly. For reconstruction (if uniformly and randomly sampled in domain) the number of required sensors is $p \approx k \log_{10}(n/k) \approx 17$ by the theory of compressed sensing.

5.4.2 Boussinesq Equations

We consider a coupled thermal-fluid dynamics application, where the focus is on the interaction of the forced incoming flow (displacement ventilation type) with the natural convection induced by the buoyancy force. Velocity and temperature distributions of the flow are determined with a finite volume scheme using `FLUENT`. The geometry is chosen as in Figure 5.1, with the corresponding values from the previous example. The bottom floor is heated at $303K$ and the top floor resides at $297K$, the velocity inlet and outlets are unchanged. The parametric study should evolve around the Archimedes number $Ar := Gr/Re^2$, as the measure of natural to forced convection. Here, Gr denotes the Grashoff number, and the simulations are performed for the same Reynolds numbers as in Table 5.1, with $Gr = 1.6 \cdot 10^{11}$.

Due to viscous effects a vortex starts to form in the upper part of the room. The instability of the hot air adjacent to the warm floor at this location will cause another large circulation to develop which is purely formed due to buoyancy forces. Such a structure is common in Rayleigh-Benard problems and in some literature referenced as “mean wind”. For the current parameters, it can be seen that the mean wind is strong enough to override the upper vortex and merge with it. As a result, after some time there is a governing vortex at the center of the cavity. In case of mixed convection examined here, the dynamics is even more complicated because the mean wind also interacts with incoming flow (interaction of thermal/natural convection and displacement/forced convection). Therefore, even for regime one with small inlet velocity the central vortex collapses into two (or even more) vortices at seemingly random times, which later merge again. By increasing the inlet velocity, i.e. regime 2, the inertial forces increase and thereby the Archimedes number decreases. Fundamentally, the formation of flow structures is highly dependent on Ar. Thus, the ratio of inlet velocity (Re number) and temperature difference between top and bottom floor (Gr number) result in a variety of regimes, as listed in Table 5.7.

Regime	Ar	Bottom outlet	Window
R1	17.4	open	closed
R2	173.8	open	closed
R3	17.4	open	open
R4	173.8	open	open

Table 5.7: 2D Boussinesq: Flow regimes are defined via the Archimedes number Ar.

The coupling of dependent variables can be used for practical flow sensing. Airflow

	R1	R2	R3	R4
R1	1.00	0.45	0.60	0.46
R2	0.45	1.00	0.39	0.37
R3	0.60	0.39	1.00	0.54
R4	0.46	0.37	0.54	1.00

Table 5.8: 2D Boussinesq: Subspace alignment matrix $\eta_{ij} = ||P_i P_j||_2$, with $P_i = \Phi_i(\Phi_i^* \Phi_i)^{-1}\Phi_i$ for the library of four regimes R1–R4.

has been sensed successfully through pressure measurements, and reconstructing the velocity map from limited pressure sensor information is possible [40, 15]. In this example, both the thermal as well as the flow states can be sensed. However, temperature measurements are easiest to acquire with the current sensor technology, so the goal is to use a minimal number of velocity sensors. At this point, it is not obvious if the full velocity field can be classified by using temperature measurements only, and we shed some light on this through our numerical experiments.

Proper Orthogonal and Dynamic Modes are computed from an interpolated dataset, and classification with various sensor locations and quantities are performed therewith. A convergence study of the interpolation grid size with respect to the heat flux resulted in a 49 × 49 grid (Nusselt number convergence). In each of the 100 experiments, a white noise was added to the signal, to corrupt the measurement and investigate robustness of the classification algorithm. The POD modes were computed from the stacked data of temperature and velocity snapshots. The goal here is to retain as much of the coupling in the data as possible, hence we choose not to compute the modes separately, as done for the structure preserving model reduction in Chapter 2. The energy contained in the POD basis is $\mathcal{E}_{min} = 0.995\%$, leading to $r = 2, 7, 2, 2$ POD modes in the four regimes, respectively. Table 5.8 shows the alignment of the subspaces spanned by the full modes, as defined by η in equation (5.7). In contrast, Table 5.9 gives the alignment of the subspaces spanned by the modes restricted to the boundary. The reader should note, that subspace alignments are more pronounced for the full modes Φ in Table 5.8. In other words, that subspaces are less distinguishable from the boundary content only.

Table 5.10 shows the confusion matrix for the case of boundary sensors with the DMD basis. The signal-to-noise-ratio is set to $SNR = 10$, $p_V = 5$ velocity sensor locations are chosen, and the temperature is sensed at $p_T = 180$ boundary nodes, which is more than 90% of the available temperature information. As can be seen from the results, boundary sensing can be highly effective, and classifies regimes

	R1	R2	R3	R4
R1	1.00	0.62	0.60	0.53
R2	0.62	1.00	0.55	0.46
R3	0.60	0.55	1.00	0.48
R4	0.53	0.46	0.48	1.00

Table 5.9: 2D Boussinesq: Subspace alignment matrix for the library of four regimes R1–R4 after sensing: $\eta_{ij} = ||P_i P_j||_2$, with $P_i = \Theta_i(\Theta_i^* \Theta_i)^{-1}\Theta_i$, and $\Theta = C\Phi$.

	R1	R2	R3	R4
R1	91.0%	4.0%	5.0%	0%
R2	0%	100.0%	0%	0%
R3	5.0%	0%	95.0%	0%
R4	0%	1.0%	0%	99.0%

Table 5.10: 2D Boussinesq: Confusion matrix with boundary sensors using DMD for the four regimes R1–R4.

correctly and robustly with more than 90% success.

In §5.3.5, we introduced a method to use subsequent time sensor measurements to increase robustness to noise and confidence of the classification, and called it augmented DMD. Here, we show how the classification improves both this the number of sensors, but also with the amount of subsequent data used in the sensing basis. Figure 5.5 shows amount of correct classification of each regime, given the number of velocity and temperature measurements, for various augmented basis. The results were obtained for a moderate signal to noise ratio of $SNR = 10$. The identification improves significantly as the basis is augmented. Note, that for this approach, no additional sensors are needed required.

5.4.3 Differentially Heated Square Cavity - Direct Numerical Simulation Data

A model of a differentially heated square cavity is considered, where again the Boussinesq approximation for the Navier-Stokes equations are used. This model is of particular interest, since a characteristic parameter, the Rayleigh number, defines the avenue to turbulence, from laminar solutions, to periodic, quasi-periodic and finally turbulent steady-state solutions. As such, those flow patterns arising from various

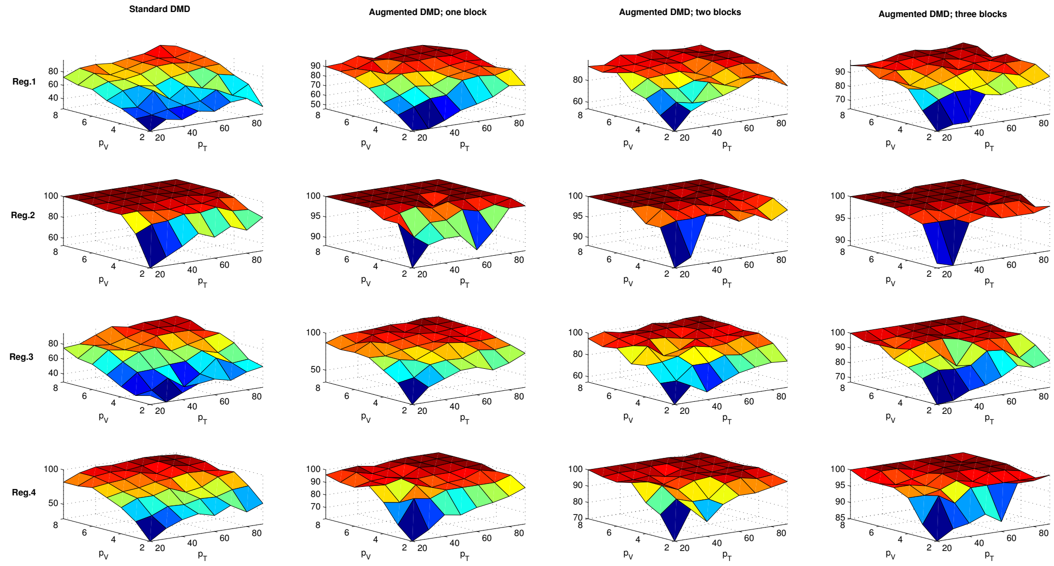

Figure 5.5: 2D Boussinesq: Confusion values for regimes R1–R4 as the velocity sensors p_V, the temperature sensors p_T, and the augmentation j increases.

Raleigh numbers can provide a natural way of defining “regimes” for classification, see Tables 5.11–5.12. The differentially heated square cavity model has been well studied for laminar regimes in [100, 137] and for unsteady behavior in [129]. Recently, Borggaard and San [160] presented low-order models for the differentially heated square cavity, and proposed Reynolds-number based closure models for more complex flow scenarios.

The spatial domain is the unit square, so $L = H = 1$, and the variables are non-dimensionalized. A schematic of the domain is shown in Figure 5.6. The cold (left) and hot (right) walls are isothermal and set as $T(x = 0) = 0$, and $T(x = 1) = 1$. The top and bottom walls are free temperature boundaries. A no-slip boundary condition on all four walls (i.e. $u = v = 0$) is imposed.

A natural convection is initiated, as the heated fluid is rising along the hot wall, while cooled fluid is falling along the cold wall. The Prandtl number is 0.71 (typical value for air), and the temperature difference and other fluid properties are chosen such that the Rayleigh number is $Ra = \frac{\rho^2 g\beta\Delta T H^3}{\mu^2}Pr$ varies between 10 and 10^9. The simulation data is obtained with the spectral element software NEK5000. Direct numerical simulation, as opposed to simulations with turbulence models as in FLUENT

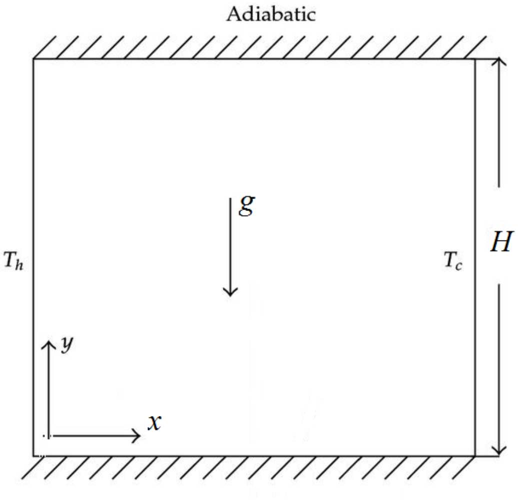

Figure 5.6: Differentially Heated Square: Schematic of the computational domain for the DNS example and NEK5000 simulation.

can give deeper insight into the mechanisms causing turbulence and unsteady behavior in flows, as finer scales are resolved in the solution. The mean Nusselt number, defined as $Nu = \frac{1}{L}\int_0^L \frac{\partial T}{\partial x} dx$, matches results in the literature for various values of Rayleigh number. The spectral element grid for the simulations is finer for higher Rayleigh numbers, see Tables 5.11–5.12.

	R1	R2	R3	R4	R5	R6	R7	R8	R9
Ra	10	10^2	10^3	10^4	10^5	10^6	10^7	10^8	$1.82 \cdot 10^8$
$3n$	1728	1728	1728	1728	1728	1728	1728	248,832	110,592

Table 5.11: Differentially Heated Square: Flow regimes R1–R9 with corresponding Raleigh numbers and spectral grid size.

	R10	R11	R12	R13	R14	R15	R16
Ra	$1.83 \cdot 10^8$	$1.85 \cdot 10^8$	$2 \cdot 10^8$	$4 \cdot 10^8$	$6 \cdot 10^8$	$8 \cdot 10^8$	10^9
$3n$	110,592	110,592	110,592	110,592	110,592	110,592	248,832

Table 5.12: Differentially Heated Square: Flow regimes R10–R16 with corresponding Raleigh numbers and spectral grid size.

The velocity and temperature data is stacked into the combined state $x = [u\ v\ T]^* \in$

$\mathbb{R}^{3n}$, and the matrix X contains the snapshots (in time) as columns. Moreover, the velocities are scaled by a factor of 5000, to have both temperature and velocity in the same order of magnitude. This way, the reduced basis computation (via svd) is not biased towards larger magnitude entries. To circumvent dimensionality issues, the full simulation data is subsequently interpolated on a 40 × 40 equidistant grid. Afterwards, the solutions are defined on vector spaces with identical dimension. A convergence study with respect to the interpolation grid size is performed, to assure that the important information in the flow solutions is retained. We compute the DMD eigenvalues from the full and interpolated data, and found that those indeed converge, and a good agreement and trade-off was given for the 40 × 40 size of the spectral grid, so $3n = 4800$. For regimes R1–R7, the solutions are in fact extrapolated onto this grid, yet this does not change the eigenvalues considerably. In Figure 5.7, a plot of the DMD spectrum of the first twenty eigenvalues computed from standard DMD is given. Importantly, the eigenvalues close to the unit circle (mainly oscillatory behavior) converge noticeably quick.

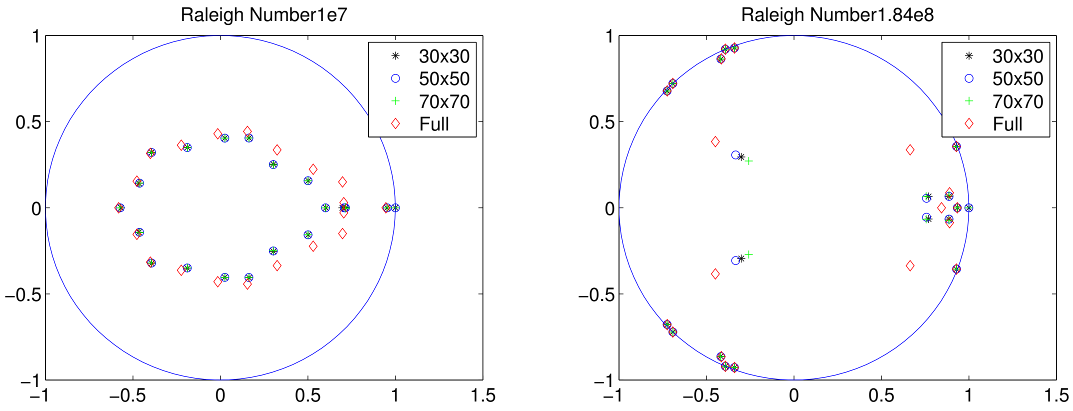

Figure 5.7: Differentially Heated Square: Convergence of the first twenty DMD eigenvalues computed from the full data, extrapolated (left: R7), and interpolated (right: R11) data.

For every regime, we compute a DMD basis Φ_i of size $r_i = 8$, and subsequently assemble the library of regimes, $\Phi \in \mathbb{R}^{3n\times R}$, where $R = \sum_{i=1}^{16} r_i$. Moreover, when a time series of measurements is taken for better classification, we augment Φ to $\hat{\Phi}$ by j additional blocks. First, we investigate the alignment and coherence of the regime library. The measure

$$\gamma_{ij} = \frac{||P_i P_j||_F}{||P_i||_F}, \tag{5.12}$$

where $P_i = \Phi_i(\Phi_i^* \Phi_i)^{-1}\Phi_i^*$ gives insight into the alignment of the projections. To have a second look at the effectiveness of the DMD basis to represent the data, the scalar

$$\kappa_{ij} = \frac{||P_i X_j||_F}{||X_j||_F}, \quad i, j = 1, \ldots, d, \tag{5.13}$$

indicates how much of the information of the data is retained in the regimes. When proper orthogonal decomposition is used as a feature extraction technique, κ_{ij} attains its maximum, as POD is the optimal basis to represent the data, see equation (1.41). In contrast, DMD has its merits in representing the dynamics of the system, and therefore one should have a closer look at its ability to represent the data. Figure 5.8 shows the measure γ_{ij} as defined in equation (5.12), both for the full projection onto Φ_i (left), and the projection onto the boundary modes $C\Phi_i$ only (right). This measure indicates, how much information is retained, by projecting onto a subspace first, and subsequently onto another subspace. The diagonal contains ones, and the off diagonal entries are generally decreasing with the off-diagonal index, indicating that only neighboring regimes (in terms of Raleigh number) share similar features. By definition, the matrices are symmetric. Moreover, two blocks of regimes appear, the first one from $Ra = 10$ to $Ra = 10^7$, and the second block from $Ra = 10^8$ to $Ra = 8 \cdot 10^8$. Note, that the last regime for Raleigh number $Ra = 10^9$, resulting in a “chaotic” flow solution, is considerably different from the other regimes. Interestingly, the blue colored fields get darker from full to boundary projection, which gives better distinction of regimes with large differences in Raleigh number. In contrast, within the two blocks, the alignment increased. Based on this information, we conjecture that through boundary sensing, misidentification slightly gets worse within the blocks; we also expect to see a confusion matrix similar in structure to Figure 5.8.

Figure 5.9 sheds a different light onto the alignment of the subspaces. The measure κ_{ij}, as defined in equation (5.13), indicates how much information is preserved by projecting on the basis Φ_i, through the projection P_i. As a first observation, the projection onto the (non-optimal) DMD modes still retains a high energy content of the data, in that the diagonal entries are mostly above 98%. Additionally, neighboring regimes share similar features; for instance, the projection of the data from regime 3 onto the basis of regime 1 retains a high amount of energy (measured in the Frobenius norm). As before, two groups of regimes appear. In analogy to the considerations above for the measure γ_{ij}, the similarity within the two blocks increases (darker red in the right plot with boundary), and the distinction between other the blocks increases. Consequently, we expect some confusion between the two clusters. From a physical point of view, this is not surprising, since a bifurcation

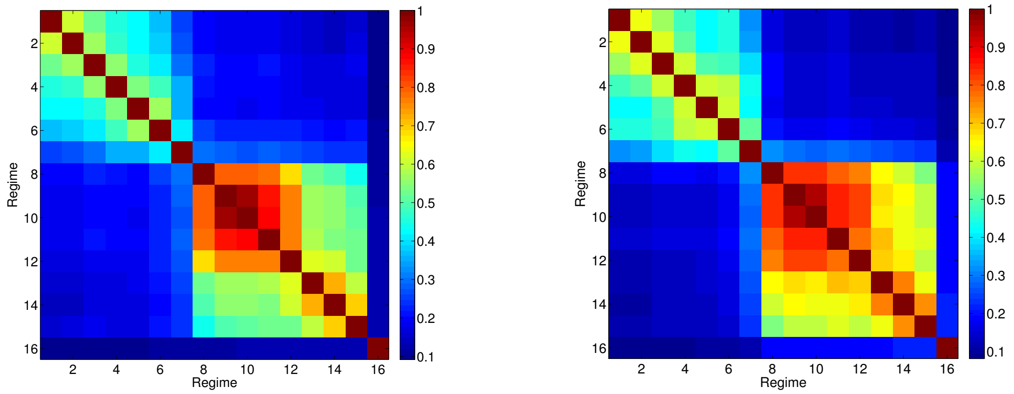

Figure 5.8: Differentially Heated Square: The subspace alignment measure γ_{ij} from definition (5.12).

towards a periodic steady state has been observed in the literature [129] around $Ra \approx 1.83 \cdot 10^8$. Therefore, flows with Raleigh number close to the bifurcation will show similar features.

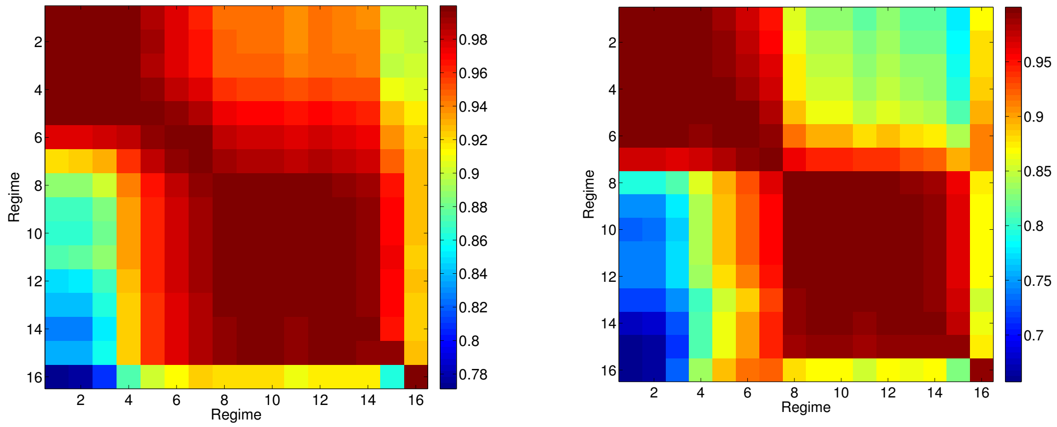

Figure 5.9: Differentially Heated Square: The subspace alignment measure κ_{ij} from definition (5.13).

Next, the results for the classification of the flow regimes are shown. For this task, $p_V = 10$ velocity, and p_T=50 temperature sensors are used, which are placed on the boundary. For technical reasons (no-slip boundary condition), the velocity is sensed on the nearest grid point to the walls. The signal to noise ratio is set to $SNR = 20$. The DMD basis is augmented by two block rows. For each regime, one hundred experiments are performed, where at each test, a snapshot from a given regime is picked and the best match to one of the 16 regimes is found by projection (5.5). In Figure 5.10, the confusion matrix for the 16 regimes is plotted. The block with the highest confusion (worst identification) is between regimes R9–R12, so Raleigh numbers $1.82 \cdot 10^8 - 2 \cdot 10^8$, which is intuitive from a physical perspective. Moreover, there appears to be confusion between the regimes one and two, and subsequently misidentification of the high Raleigh number scenarios. However, this is a local phenomenon, and typically only confuses neighboring regimes.

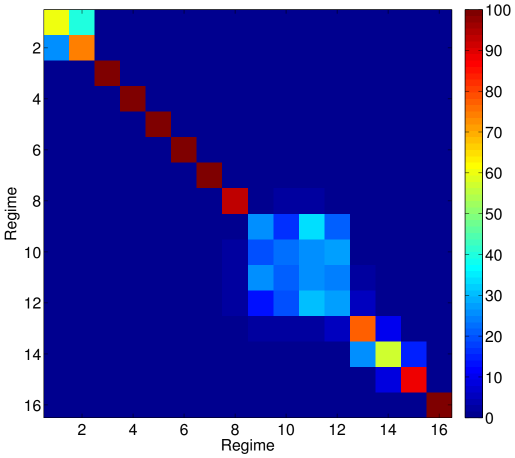

Figure 5.10: Differentially Heated Square: Confusion matrix for flow regimes R1–R16.

Below, the results for the last nine regimes (second cluster) are presented in more detail. Tables 5.13–5.14 show the data fit measure κ_{ij}, as defined in equation (5.13). Moreover, Tables 5.15–5.16 show the closeness of projections, computed from the measure γ_{ij} in equation (5.12). In all cases, the values computed from the full, spatially distributed DMD modes in Table 5.15 are smaller than the value computed from the boundary values of the DMD modes $C\Phi$, in Table 5.16. This indicates again, that one has to expect a slight deterioration of classification when using boundary information only. Lastly, Table 5.17 shows the confusion values for the nine regimes.

	R8	R9	R10	R11	R12	R13	R14	R15	R16
R8	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.97	0.93
R9	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.97	0.93
R10	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.97	0.92
R11	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.97	0.92
R12	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.97	0.93
R13	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.92
R14	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.92
R15	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.93
R16	0.92	0.92	0.92	0.91	0.92	0.92	0.92	0.86	1.00

Table 5.13: Differentially Heated Square: Data fit measure κ_{ij} from equation (5.13), where the basis was augmented by two blocks ($j = 2$), and the projection is computed from the full modes Φ^i for every regime. Refer to Tables 5.11–5.12 for the definition of the regimes.

	R8	R9	R10	R11	R12	R13	R14	R15	R16
R8	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.96	0.88
R9	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.97	0.87
R10	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.97	0.87
R11	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.97	0.87
R12	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.97	0.87
R13	0.99	1.00	1.00	1.00	1.00	1.00	0.99	0.97	0.85
R14	0.99	0.99	0.99	0.99	1.00	1.00	1.00	0.97	0.87
R15	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.87
R16	0.88	0.87	0.87	0.85	0.87	0.86	0.87	0.83	0.99

Table 5.14: Differentially Heated Square: Data fit measure κ_{ij} from equation (5.13), where the basis was augmented by two blocks ($j = 2$), and the projection is computed from the boundary modes $C\Phi^i$. Refer to Tables 5.11–5.12 for the definition of the regimes.

	R8	R9	R10	R11	R12	R13	R14	R15	R16
R8	1.00	0.79	0.80	0.78	0.69	0.52	0.49	0.44	0.12
R9	0.79	1.00	0.97	0.86	0.76	0.57	0.55	0.50	0.12
R10	0.80	0.97	1.00	0.87	0.76	0.57	0.55	0.51	0.12
R11	0.78	0.86	0.87	1.00	0.76	0.59	0.55	0.53	0.12
R12	0.69	0.76	0.76	0.76	1.00	0.63	0.61	0.52	0.12
R13	0.52	0.57	0.57	0.59	0.63	1.00	0.72	0.60	0.13
R14	0.49	0.55	0.55	0.55	0.61	0.72	1.00	0.69	0.13
R15	0.44	0.50	0.51	0.53	0.52	0.60	0.69	1.00	0.13
R16	0.12	0.12	0.12	0.12	0.12	0.13	0.13	0.13	1.00

Table 5.15: Differentially Heated Square: Measure for shared projection information γ_{ij} from definition (5.12). The projection is computed from the full modes Φ^i.

	R8	R9	R10	R11	R12	R13	R14	R15	R16
R8	1.00	0.84	0.84	0.79	0.76	0.64	0.60	0.53	0.19
R9	0.84	1.00	0.96	0.84	0.82	0.67	0.65	0.60	0.19
R10	0.84	0.96	1.00	0.85	0.83	0.66	0.63	0.59	0.18
R11	0.79	0.84	0.85	1.00	0.77	0.70	0.63	0.61	0.18
R12	0.76	0.82	0.83	0.77	1.00	0.68	0.66	0.60	0.19
R13	0.64	0.67	0.66	0.70	0.68	1.00	0.77	0.64	0.20
R14	0.60	0.65	0.63	0.63	0.66	0.77	1.00	0.74	0.21
R15	0.53	0.60	0.59	0.61	0.60	0.64	0.74	1.00	0.22
R16	0.19	0.19	0.18	0.18	0.19	0.20	0.21	0.22	1.00

Table 5.16: Differentially Heated Square: Measure for shared projection information γ_{ij} from definition (5.12). The projection is computed from the boundary modes $C\Phi^i$.

	R8	R9	R10	R11	R12	R13	R14	R15	R16
R8	93	3	0	0	3	1	0	0	0
R9	2	27	23	27	20	1	0	0	0
R10	1	19	17	30	30	3	0	0	0
R11	1	23	13	35	23	3	0	2	0
R12	1	17	19	28	32	3	0	0	0
R13	2	2	3	4	5	71	11	2	0
R14	2	1	3	0	1	16	65	12	0
R15	1	0	1	0	1	3	12	82	0
R16	0	0	0	0	0	0	0	0	100

Table 5.17: Differentially Heated Square: Confusion matrix over 100 experiments with randomly selected boundary sensors ($p_T = 50$ for temperatures and $p_V = 10$ for velocity). The noise level was around 10% ($SNR = 20$) and the basis was augmented by two blocks ($j = 2$). Refer to Tables 5.11–5.12 for the definition of the regimes. It is obvious, that the dynamic regimes 9–12 show high confusion, as they dynamics defined by those Raleigh numbers is similar.

In the previous study, all available data is used for generation of the sparse library, and subsequently, the same data used for classification. Here, training and testing data is separated, to investigate the performance of the algorithm to unknown data. In particular, the library of sparse DMD basis is generated for the regimes given in Table 5.18.

Reg.	R8	R12	R13	R14	R15	R16
Ra	10^8	$2 \cdot 10^8$	$4 \cdot 10^8$	$6 \cdot 10^8$	$8 \cdot 10^8$	10^9
$3n$	248,832	110,592	110,592	110,592	110,592	248,832

Table 5.18: Differentially Heated Square: Flow regimes for sparse library with corresponding Raleigh numbers and spectral grid size. The regime numbering is kept consistent with the prior convention in Tables 5.11–5.12.

The corresponding subspace measures (κ_{ij} and γ_{ij}) can be read out from Tables 5.13–5.16 above. The testing data is taken from regimes R9, R10, and R11, which correspond to Raleigh numbers between $1.82 \cdot 10^8$ and $1.85 \cdot 1.85$. As noted earlier, there is a bifurcation [129] of the flow at $Ra \approx 1.83 \cdot 10^8$, which has been observed both experimentally, as well as numerically. From a physical point of view, one expects the classification of the testing data to match to regimes R8 or R12. In

Regime	R8	R12	R13	R14	R15	R16
Classification	6%	87%	6%	0%	0%	0%

Table 5.19: Differentially Heated Square: Correct identification of unknown data (taken from regimes R9–R12) for six regimes at high Raleigh numbers. As expected from the physics, the data is classified largely as regime R12.

fact, this is very much the case when using the sensing method developed here. We performed 600 independent experiments, where at each experiment a flow snapshot (or subsequent snapshots for the augmented sensing algorithm) is taken from the test data, and classified into the six sparse basis regimes given in Table 5.18. The signal-to-noise ratio is set to $SNR = 20$, and $p_V = 10$ flow sensors are used, together with $p_T = 50$ temperature sensors, all placed on the boundary of the unit square (to be precise, recall that "boundary" for the velocity sensors means slightly inside the domain, since the velocity is set zero at the boundaries). For better classification performance, and more robustness to noise, the DMD basis is augmented by two blocks, i.e. three subsequent time snapshots are taken for (batch-wise) classification.

Table 5.19 contains the classification results for identifying the regimes from the unknown data. Note, that the sparse basis library does not contain any of this data. Nonetheless, the sensing method is able to match the testing data to the (physically) correct flow patterns. In practice, this is important, since one does not expect the data to repeat itself in a given situation. Hence, the sensing mechanism needs to be able to match data to its "closest" subspace in the library collection.

One way to cope with this situation, while not discarding any data, would be to cluster subspaces. In particular, the subspaces corresponding to R8–R12 share a considerable amount of similarities, and therefore a subspace clustering algorithm could be used to combine those into a single basis. This is part of future work.

5.5 Conclusion

The compressive sensing based classification algorithm developed here, effectively used boundary data to classify dynamic regimes. This is important for the development of regime-based controllers and observers, and we believe it to have great potential for industrial applications. Through using successive-in-time data from sensors (batch-data), we were able to improve the robustness to noise, and the classification performance overall. This comes at no additional cost for sensors.

In the last example, where DNS data from a differentially heated cavity was considered, we have seen (as expected from the physics), that clusters of data appear. On the one hand, one could keep a representative regime, and discard the others. However, this would not incorporate all the available data. Instead, we are interested in subspace clustering methods, which would retain the most information possible, and reduce the data to a few relevant clusters. This is particularly helpful, where the solution behavior and bifurcations with respect to predefined parameters, are not obvious from a physics based point of view. Hence, data-based clustering algorithms could work on experimental data, and improve the classification results even further.

Additionally, we presented several measures, such as γ_{ij}, defined in equation (5.12), and κ_{ij} from (5.13), which give a good a priori indication, as to how well the classification will work. In essence, those measures quantize the optimality of the projection of the data onto the DMD modes, and the alignment of the computed subspaces. An observability result for compressive sensing for linear systems is available in [80, 68]. Combining observability theory and compressive sensing, we plan to obtain results for optimal sensor placement and classification performance. In particular, these probabilistic results should be linked to the subcoherence and block-coherence of the assumed library of sparse basis functions.

Chapter 6

Conclusions and Future Work

The goal of this thesis was to leverage the advantages of physics and data-based reduced order models for design, optimization, and control of physical systems. New data and model reduction algorithms were developed, which are either an improvement of available methods, or a new way to solve a challenging problem with a particular emphasis on practical applications. Specifically, the following results were obtained in this dissertation:

- An algorithm to solve Algebraic Riccati equations, which produces higher accuracy at lower solution rank compared to a state of the art Krylov method, and can use available sparse linear solvers and software. The algorithm was designed to be easily implementable for practitioners of proper orthogonal decomposition. Moreover, this provides a first step towards a completely matrix free approach to solve ARE. This was presented in Chapter 3.

- A method for system identification and model reduction that addresses the challenges of a large number of inputs and outputs. This circumvents the computation burden of current algorithms, and makes system identification for MIMO systems computationally more efficient. A stability result and an error bound for this algorithm were also developed. This was presented in Chapter 4.

- A method for classification and sensing of parameter dependent data, based on a library of dynamic regimes. This addresses the challenge of utilizing a large amount of offline and online data, in the design of sensors and observers. As an application, we considered thermal-flow data, which depends on parameters such as the Reynolds and Rayleigh number. We developed a new, augmented sensing basis, which improves robustness to noise of the classification method, and uses only boundary sensor data as inputs to the algorithm. Through the method of compressed sensing, we were able to reduce the amount of required online data to classify flow scenarios. This algorithm has great promise for the use in online-control mechanisms in industrial systems. This was presented in Chapter 5.

The focus and motivation was to use model reduction and compressive sensing techniques to solve challenging problems in control, estimation and identification of large-scale systems. In particular, we attempt to combine the advantages of physics-based models with data-based methods to use "the best of both worlds". The algorithms in this dissertation were not designed for implementation on high performance computers, but rather on typical clusters and workstations, which are widely used in academia and industry.

The results in this dissertation certainly do not "close the chapter" on these problems. They raise several interesting questions, which remain to be addressed, and where the author sees potential for further research. The topics of particular interest include:

- Developing a rigorous convergence theory for the Riccati solver from Chapter 3. As we pointed out, there are parallels between the POD method and rational Krylov subspace techniques (as discussed in §3.4.1), and this could help to obtain a thorough analysis of the method.
- Developing subspace clustering algorithms to automatically combine nearby (in some metric) subspaces. So far, for a simple geometry, a clustering is possible due to physical insight into the dynamical behavior (bifurcations, transitions) of the system. For more general geometries and increasingly realistic (hence complex) boundary conditions, this approach might not be feasible.
- Relating the block- and subcoherence of the sparse dictionary to the success of classification. Finding sharp necessary and sufficient conditions for the algorithm to classify a given regime correctly.

Appendix A

Further Technical Results

The appendix adds some further, known technical results, which are meant for the interested reader. Those are important, but would have distracted from the main ideas in the above sections.

A.1 Approximation Theory for Control of PDEs

Theorem 1.5.3 guarantees the existence of the optimal controller for the dynamical system generated by Burgers equation with a bounded control operator. However, to compute such a control, a discretization scheme is needed. Thus, one seeks a finite dimensional approximation K_N to the gain operator $\mathcal{K}$. This poses the natural question of convergence. Does the optimal gain K_N computed from the discretized system converge to the infinite dimensional gain operator $\mathcal{K}$? This question is addressed by K.Ito in [105]. To keep this thesis rather self-contained, we recall the key results from this paper. With regard to notation, it is customary to use subscripts to denote finite dimensional approximations, hence let $x \in \mathcal{X}$ be the infinite dimensional state, and let x_n be the approximation in finite dimensions.

$$\pi_n : \mathcal{X} \mapsto \mathcal{X}_n \subseteq \mathbb{R}^n \tag{A.1}$$

be the orthogonal projection of $\mathcal{X}$ onto $\mathcal{X}_N$. The projection should satisfy the convergence property

$$||\pi_n x - x||_{\mathcal{X}} \leq M n^{-s} ||x||_{\mathcal{D}(\mathcal{A})},$$

where $M \geq 0$ is a generic constant independent of the state space dimension n and s is the speed of convergence. Through this projection π_n one can define a sequence of approximating problems $(\mathcal{X}_n, A_n, B_n, C_n)$. The linear, finite rank operators can then be represented via matrices A_n, B_n, C_n are matrix representations of appropriate size of the finite dimensional operators.

Now that the finite dimensional optimal control problem is set up, the issue of convergence of solutions of the finite dimensional Riccati equation (1.36) to the infinite dimensional solutions of (1.24) shall be addressed. Therefore, suitable assumptions need to be made. In particular, one needs to assume convergence and dual convergence of the system. Moreover, assumptions on the preservation of exponential stabilizability and detectability are necessary. In particular, the following assumptions are made in [105]:

Assumption A.1.1. *(convergence): For each $x \in \mathcal{X}$ and $u \in U = \mathbb{R}$ there holds:*
(a) $e^{A_n t}\pi_n x \to S(t)x$,
(b) $B_n u \to \mathcal{B}u$,
(c) $C_n x \to \mathcal{C}x$.

Assumption A.1.2. *(dual convergence): For each $x \in \mathcal{X}$ and $y \in Y$ there holds:*
(a) $e^{A_n^* t}\pi_n x \to S^*(t)x$,
(b) $B_n^* \pi_n x \to \mathcal{B}^* x$,
(c) $C_n^* y \to \mathcal{C}^* y$.

Assumption A.1.3. *(preservation of exponential stabilizability/detectability): Assume that there is a n_0 such that for all $n \geq n_0$ the following hold:*
(a) The family of pairs (A_n, B_n) is uniformly stabilizable, i.e. there exists a sequence of operators K_n and positive constants $M_1 \geq 1, \omega_1 > 0$ such that $\sup ||K_n|| < \infty$ and the semigroups generated by the closed loop operators $A_n - B_n K_n$ are exponentially bounded as

$$||e^{(A_n - B_n K_n)t}\pi_n|| \leq M_1 e^{-\omega_1 t}, \quad t \geq 0. \tag{A.2}$$

(b) The family of pairs (A_n, C_n) is uniformly detectable, i.e. there exists a sequence of operators G_n and positive constants $M_2 \geq 1, \omega_2 > 0$ such that $\sup ||G_n|| < \infty$ and the semigroups generated by the closed loop operators $A_n - G_n C_n$ are exponentially bounded as

$$||e^{(A_n - G_n C_n)t}\pi_n|| \leq M_2 e^{-\omega_2 t}, \quad t \geq 0. \tag{A.3}$$

Theorem A.1.4. *([105]) Let Assumptions A.1.1 - A.1.3 be satisfied. Then there exists an integer n_0 such that for all $n \geq n_0$ the Riccati equation has a unique* *nonnegative solution P_n with bounded operator norm,* $\sup ||P_n|| < \infty$*, and there exist constants* $\omega > 0$ *(independent of n) and* $M \geq 1$ *such that*

$$||e^{(A_n - B_n B_n^* P_n)t} \pi_n|| \leq M e^{-\omega t}, \quad t \geq 0. \tag{A.4}$$

Corollary A.1.5. *([105]) Let Assumptions A.1.1 - A.1.3 be satisfied and* (A_n, B_n) *be stabilizable and* (A_n, C_n) *detectable. Then the solution to the finite dimensional Riccati equation,* P_n*, converges strongly to* Π*, the solution to the operator Riccati equation.*

Theorem A.1.6. *([105]) Suppose that* $\mathcal{B}$ *is compact and* $B_n = \pi_n \mathcal{B}$ *and that Assumptions A.1.1(a) and A.1.3(a) are satisfied. Then* $(\mathcal{A}, \mathcal{B})$ *is stabilizable.*

Theorem A.1.7. *([112],p.116) Let* $\mathcal{A} : \mathcal{D}(\mathcal{A}) \mapsto \mathcal{X}$ *be the generator of an analytic semigroup and* $\mathcal{B} \in \mathcal{L}(U, Z)$ *a bounded control operator. Moreover, let* π_n *denote the projection operator in (A.1) associated with the FEM discretization. Under certain natural assumptions on the approximating scheme, the finite dimensional control law is*

$$u^*(t) = -K_n \pi_n z(t). \tag{A.5}$$

This linear feedback exponentially stabilizes the dynamical system

$$\dot{x}(t) = [\mathcal{A} - \mathcal{B} K_n \pi_n] x(t), \qquad x(0) = x_0. \tag{A.6}$$

In other words, under certain assumptions on the FEM, the finite dimensional gain operator indeed stabilizes the infinite dimensional dynamical system.

A.2 Non-Dimensionalization and Vectorization of Data

It is customary in the fluid dynamics community to non-dimensionalize the data. In the first example (FLUENT simulations) of Section 5.4, the flow variables and temperature are simulated in standard units. The flow is forced through an inflow velocity (at the upper left corner of the room), which we denote by $(u_{in})_i$ for regime i. The top wall temperature resides at $T_c = 297K$ and the bottom wall is heated at

$T_h = 303K$. The number of grid points in x and y direction is denoted by n_x and n_y, respectively, and A denotes the area. The spatially averaged temperature is given by

$$\bar{T}(t_i) = \frac{\int \hat{T}(t_i)dA}{|A|} = \frac{\sum_{k=0}^{n_x \cdot n_y}(\hat{T}(t_i))_k \Delta x \Delta y}{\sum_{k=0}^{n_x \cdot n_y} \Delta x \Delta y} = \frac{\sum_{k=0}^{n}(\hat{T}(t_i))_k}{n},$$

where $n_x \cdot n_y = n$. The value of the spatially averaged temperature is approximately $300K$ recorded over $1000s$ simulation time. The non-dimensionalized variables are then defined via

$$u_x := \frac{\hat{u}_x}{[u_{in}]_i}, \qquad u_y := \frac{\hat{u}_y}{[u_{in}]_i}, \qquad T(t_i) := \frac{\hat{T}(t_i) - \bar{T}(t_i)}{|T_h - T_c|}.$$

For ease of presentation and computation, the data is vectorized in the form

$$X = \begin{bmatrix} u_x(t_i) \\ u_y(t_i) \\ T(t_i) \end{bmatrix}_{i=1,..,n_t},$$

which is the data format we use throughout for feature extraction and sensing throughout.

A.3 POD - Mean flow averaging

For computation of the POD modes in Section 5.4, the mean of the dataset is removed before the feature extraction is applied [135, 108]. Note, that the solution to the dynamical system can be decomposed as

$$\mathbf{u}(t, \mathbf{x}) = \mathbf{u}_0(\mathbf{x}) + \mathbf{u}'(t, \mathbf{x}),$$

where $\mathbf{u}_0(\mathbf{x}) = \frac{1}{s}\sum_{i=1}^{s} \mathbf{u}(t_i, \mathbf{x})$ represents the time average of the data. To this end, let the data be stored in the snapshot matrix

$$X = \begin{bmatrix} | & | & & | \\ \mathbf{u}(t_0, \mathbf{x}) & \mathbf{u}(t_1, \mathbf{x}) & \cdots & \mathbf{u}(t_s, \mathbf{x}) \\ | & | & & | \end{bmatrix}$$

After subtracting the mean of the dataset,

$$\tilde{X} = \begin{bmatrix} | & | & & | \\ \mathbf{u}(t_0, \mathbf{x}) & \mathbf{u}(t_1, \mathbf{x}) & \cdots & \mathbf{u}(t_s, \mathbf{x}) \\ | & | & & | \end{bmatrix} - \begin{bmatrix} | & | & & | \\ \mathbf{u}_0 & \mathbf{u}_0 & \cdots & \mathbf{u}_0 \\ | & | & & | \end{bmatrix},$$

we apply the POD technique to the mean-subtracted data $\tilde{X}$.

Remark A.3.1. Note, that when using Dynamic Mode Decomposition (DMD), mean flow subtraction is not advisable. If done so, the dynamic modes coincide with the discrete Fourier modes, implying that the dependence on the specific dataset is lost, see [65, Sec.4].

A.4 Sparse DMD

Unlike POD, the DMD method does not provide a natural way of ordering modes, therefore modal selection becomes important. It is customary to use 'standard' DMD, as described in Section 1.6.3. However, there are alternative approaches, that do not involve a rank reduction at the first step, and rather use a ℓ_1 optimization to achieve sparsity and data-fit. The concept of *sparse DMD* was introduced in [113]. Since we pursue this in the future, we include the method here for the interested reader. The first steps follow the standard steps of DMD, as introduced earlier, except that the singular value decomposition of Ψ_0 is not significantly truncated, and modes are selected differently. To illustrate the idea, let

$$\Psi_0 = [x_0 \; x_1 \; \ldots \; x_{s-1}] \qquad \Psi_1 = [x_1 \; x_2 \; \ldots \; x_s]$$

be the snapshot matrices of the simulation data, so that

$$\Psi_1 = \mathbf{A}\Psi_0.$$

The singular value decomposition of the snapshot matrix yields

$$X_0 = U\Sigma V^T,$$

and the advance operator A is projected onto $r \approx s$ (a slight truncation for numerical rank deficiency can be performed) dimensions as

$$U_r^T \mathbf{A} U_r =: S_r,$$

where one should again observe that $U_r \in \mathbb{R}^{n\times r}$ has the POD basis functions as columns. We shall emphasize again, that in standard DMD, $r \ll s$, and here $r \approx s$. One can decompose S_r via the eigendecomposition into

$$S_r = Y\Lambda Y^{-1}.$$

We can then see, that by the state in r dimensions, denoted by $\hat{x}$ evolves as

$$\hat{x}_t = Y\Lambda^t Y^{-1}\hat{x}_0 = \sum_{i=1}^{r} y_i \lambda_i^t [y_i]^{-1}\hat{x}_0 = \sum_{i=1}^{r} y_i \lambda_i^t \alpha_i,$$

and consequently, since $\Phi := UY$ are the DMD modes in the original n dimensional space, one has

$$x_t = \sum_{i=1}^{r} \phi_i \lambda_i^t \alpha_i, \quad \text{for} \quad t \in \{0, 1, \ldots, s-1\}.$$

One can interpret the coefficients α_i as the contribution of the initial condition to the dynamics. The authors in [113] propose to use a selection based on the α_i to determine the modes kept for the DMD. Rewriting the above yields

$$\begin{aligned}
\Psi_0 &= [x_0 \; x_1 \; \ldots \; x_{n-1}] \\
&\approx [\phi_1 \; \phi_2 \; \ldots \; \phi_n] \begin{bmatrix} \alpha_1 & & & \\ & \alpha_2 & & \\ & & \ddots & \\ & & & \alpha_r \end{bmatrix} \begin{bmatrix} 1 & \lambda_1 & \cdots & \lambda_1^{n-1} \\ 1 & \lambda_2 & \cdots & \lambda_2^{n-1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & \lambda_r & \cdots & \lambda_r^{n-1} \end{bmatrix} \\
&:= \Phi D_{\boldsymbol{\alpha}} V_{and}(\lambda),
\end{aligned}$$

where $V_{and}(\lambda)$ is a Vandermonde matrix. The idea of sparse DMD is to optimize the coefficients $\boldsymbol{\alpha}$, so that the approximation best fits the data and the vector α is sparse, i.e.

$$\min_{\boldsymbol{\alpha}} ||\Psi_0 - \Phi D_{\boldsymbol{\alpha}} V_{and}(\lambda)||_F^2 + \gamma ||\boldsymbol{\alpha}||_1.$$

However, since $W_r^T W_r = I_r$, and $X_0 \approx W_r \Sigma_r V^T$ and $\Phi_r = W_r Y$, we get the smaller problem

$$\min_{\boldsymbol{\alpha}} ||\Sigma_r V_r^T - Y D_{\boldsymbol{\alpha}} V_{and}(\lambda)||_F^2 + \gamma ||\boldsymbol{\alpha}||_1.$$

The above problem can be solved with standard convex optimization routines.

Remark A.4.1. During some preliminary studies, we have used CVX to solve the optimization, which we observed to be slow. Moreover, the parameter γ, which imposes sparsity of the solution has to be tuned. This is done by brute-force, i.e. running a loop over decreasing values of γ. The iteration terminates when a maximally acceptable error on the data-fit is achieved. Alternatively, one could rewrite $\Psi_0 - \Phi D_{\boldsymbol{\alpha}} V_{and}(\lambda)$, which is a linear term in λ as $Y - Z\lambda$, and subsequently use faster ℓ_1 optimization tools.

Bibliography

[1] Akanyeti, O., Venturelli, R., Visentin, F., Chambers, L., Megill, W. M., and Fiorini, P. What information do Kármán streets offer to flow sensing? *Bioinspiration & biomimetics 6*, 3 (2011), 036001. 5.1

[2] Akhtar, I., Borggaard, J. T., Stoyanov, M., and Zietsman, L. On commutation of reduction and control: linear feedback control of a von Kármán street. In *Proceedings of the 5th Flow Control Conference, American Institute of Aeronautics and Astronautics* (2010), pp. 1–14. 3.1

[3] Al-Saggaf, U. M., and Franklin, G. F. Model reduction via balanced realizations: an extension and frequency weighting techniques. *IEEE Transactions on Automatic Control 33*, 7 (1988), 687–692. 4.4

[4] Allen, E. J., Burns, J. A., and Gilliam, D. S. Numerical approximations of the dynamical system generated by Burgers equation with Neumann–Dirichlet boundary conditions. *ESAIM: Mathematical Modelling and Numerical Analysis 47*, 05 (2013), 1465–1492. 2.1

[5] Allen, E. J., Burns, J. A., Gilliam, D. S., Hill, J., and Shubov, V. The impact of finite precision arithmetic and sensitivity on the numerical solution of partial differential equations. *Math. Comput. Modelling 35*, 11-12 (2002), 1165–1195. 2.1

[6] Amodei, L., and Buchot, J.-M. A stabilization algorithm of the Navier–Stokes equations based on algebraic Bernoulli equation. *Numerical Linear Algebra with Applications 19*, 4 (2012), 700–727. 3.2

[7] Amsallem, D., and Farhat, C. Stabilization of projection-based reduced-order models. *International Journal for Numerical Methods in Engineering 91*, 4 (2012), 358–377. 1.6.2, 2

[8] Antoulas, A. C. *Approximation of Large-Scale Dynamical Systems (Advances in Design and Control)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2005. 1.2.5, 1.2, 1.5, 1.5.2, 1.5.4, 1.5.10, 1.6.3, 3.1, 3.2.1, 3.5.1, 4.1.1, 4.2.1, 4.2.2, 4.2, 4.2, 1

[9] Antoulas, A. C., Beattie, C. A., and Gugercin, S. Interpolatory model reduction of large-scale dynamical systems. *Efficient Modeling and Control of Large-Scale Systems* (2010), 3–58. 4.3.1

[10] Arnold, W. F., and Laub, A. J. Generalized eigenproblem algorithms and software for algebraic Riccati equations. *Proceedings of the IEEE 72*, 12 (1984), 1746–1754. 3.5

[11] Atwell, J. A., Borggaard, J. T., and King, B. B. Reduced order controllers for Burgers' equation with a nonlinear observer. *International Journal of Applied Mathematics and Computer Science 11* (2001), 1311–1330. 2, 3.1

[12] Atwell, J. A., and King, B. B. Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations. *Math. Comput. Modelling 33*, 1-3 (2001), 1–19. 1.6.2, 2, 2.3.2, 3.1

[13] Aubry, N., Holmes, P., Lumley, J., and Stone, E. The dynamics of coherent structures in the wall region of a turbulent boundary layer. *Journal of Fluid Mechanics 192* (1988), 115–173. 1.6.2, 5.1

[14] Bai, Z., Demmel, J., and Gu, M. An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems. *Numer. Math 76*, 0 (1997), 279–308. 3.1

[15] Bai, Z., Wimalajeewa, T., Berger, Z., Wang, G., Glauser, M., and Varshney, P. Low-dimensional approach for reconstruction of airfoil data via compressive sensing. *AIAA Journal* (2014), 1–14. 5.1, 5.4.2

[16] Baker, J., Embree, M., and Sabino, J. Fast singular value decay for Lyapunov solutions with nonnormal coefficients. arXiv:1410.8741, 2015. 3.5

[17] Banks, H., and Burns, J. Hereditary control problems: numerical methods based on averaging approximations. *SIAM Journal on Control and Optimization 16*, 2 (1978), 169–208. 1.3.2

[18] Banks, H. T., Beeler, S. C., Kepler, G. M., and Tran, H. T. Reduced order modeling and control of thin film growth in an HPCVD reactor. *SIAM J. Appl. Math. 62*, 4 (2002), 1251–1280. 3.1

[19] Banks, H. T., del Rosario, R. C. H., and Smith, R. C. Reduced-order model feedback control design: numerical implementation in a thin shell model. *IEEE Trans. Automat. Control 45*, 7 (2000), 1312–1324. 3.1

[20] Banks, H. T., and Ito, K. A numerical algorithm for optimal feedback gains in high-dimensional linear quadratic regulator problems. *SIAM J. Control Optim. 29*, 3 (1991), 499–515. 3.2

[21] Baraniuk, R., Cevher, V., Duarte, M., and Hegde, C. Model-based compressive sensing. *Information Theory, IEEE Transactions on 56*, 4 (2010), 1982–2001. 1.7

[22] Barrault, M., Maday, Y., Nguyen, N., and Patera, A. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Math. Acad. Sci. Paris 339*, 9 (2004), 667–672. 1.6.1

[23] Beattie, C., and Gugercin, S. Realization–independent H_2 approximation. In *Proceedings of the 51st IEEE Conference on Decision & Control* (2012), IEEE, pp. 4953–4958. 4.3.1

[24] Beattie, C. A., Borggaard, J. T., Gugercin, S., and Iliescu, T. A domain decomposition approach to POD. In *Proceedings of the 45th IEEE Conference on Decision and Control* (2006). 3.4

[25] Benner, P. Accelerating Newton's method for discrete-time algebraic Riccati equations. In *Proceedings MTNS 98* (1998), pp. 569–572. 3.1

[26] Benner, P. Balancing-related model reduction for parabolic control systems. In *Proceedings of the 1st IFAC Workshop on Control of Systems Governed by Partial Differential Equations* (2014), pp. 257–262. 1.6, 3.1

[27] Benner, P., and Bujanović, Z. On the solution of large-scale algebraic riccati equations by using low-dimensional invariant subspaces. Preprint MPIMD/14-15, Max Planck Institute Magdeburg, Aug. 2014. Available from http://www.mpi-magdeburg.mpg.de/preprints/. 3.1

[28] Benner, P., and Heiland, J. Lqg-balanced truncation low-order controller for stabilization of laminar flows. In *Active Flow and Combustion Control 2014*, R. King, Ed., vol. 127 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*. Springer International Publishing, 2015, pp. 365–379. 3.1

[29] Benner, P., Li, J.-R., and Penzl, T. Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems. *Numer. Linear Algebra Appl. 15*, 9 (2008), 755–777. 3.3.3

[30] Benner, P., Mehrmann, V., and Xu, H. A note on the numerical solution of complex Hamiltonian and skew-Hamiltonian eigenvalue problems. *Electronic Transactions on Numerical Analysis 8* (1998), 115–126. 3.1

[31] Benner, P., and Saak, J. A Galerkin-Newton-ADI method for solving large-scale algebraic Riccati equations. Preprint SPP1253-090, DFG Priority Programme 1253 Optimization with Partial Differential Equations, Available from http://www.am.uni-erlangen.de/home/spp1253/wiki/images/2/28/Preprint-SPP1253-090.pdf, 2010. 3.1

[32] Benner, P., and Saak, J. Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: A state of the art survey. *GAMM-Mitteilungen 36*, 1 (June 2013), 32–52. 3.1, 3.3.2, 3.3.3, 3.5

[33] Benosman, M., Boufounos, P., Kramer, B., and Grover, P. System and method for controlling operations of air-conditioning system. *US Patent Application 14/640,052, filed March 6, 2015* (2015). 5.1

[34] Benosman, M., Boufounos, P., Kramer, B., and Grover, P. System and method for controlling operations of air-conditioning system. *US Patent Application 14/714,428, filed May 18, 2015* (2015). 5.1

[35] Bensoussan, A. *Representation and control of infinite dimensional systems*. Systems & control. Birkhäuser, 2007. 1.5.3

[36] Berljafa, M., and Güttel, S. Generalized rational Krylov decompositions with an application to rational approximation. *SIAM Journal on Matrix Analysis and Applications, to appear* (2015). 4.1.2

[37] Bini, D., Iannazzo, B., and Meini, B. *Numerical Solution of Algebraic Riccati Equations*. Fundamentals of algorithms, 2012. 3.1

[38] Borggaard, J. T., Cliff, E., and Gugercin, S. Model reduction for indoor-air behavior in control design for energy-efficient buildings. In *Proceedings of the American Control Conference* (2012), IEEE, pp. 2283–2288. 4.1, 4.2.5, 4.4.3

[39] Borggaard, J. T., and Stoyanov, M. An efficient long-time integrator for Chandrasekhar equations. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on* (2008), pp. 3983–3988. 3.1

[40] Bright, I., Lin, G., and Kutz, J. N. Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements. *Physics of Fluids (1994-present) 25*, 12 (2013), 127102. 5.1, 5.4, 5.4.2

[41] Brunton, S. L., Proctor, J. L., and Kutz, J. N. Compressive sampling and dynamic mode decomposition. *arXiv preprint arXiv:1312.5186* (2013). 1.6.3, 1.6.3

[42] Brunton, S. L., Tu, J. H., Bright, I., and Kutz, J. N. Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems. *SIAM Journal on Applied Dynamical Systems 13*, 4 (2014), 1716–1732. 5.4

[43] Bryan, K., and Leise, T. Making do with less: An introduction to compressed sensing. *SIAM Review 55*, 3 (2013), 547–566. 1.7, 1.7.2

[44] Bunse-Gerstner, A., and Mehrmann, V. A symplectic QR-like algorithm for the solution of the real algebraic Riccati equation. *EEE Transactions on Automatic Control AC-31* (1986), 1104–1113. 3.1

[45] Burns, J. A. *Introduction to theCalculus of Variations and Control—With Modern Applications.* CRC Press, Boca Raton, FL, 2014. 3.1

[46] Burns, J. A., Balogh, A., Gilliam, D. S., and Shubov, V. I. Numerical stationary solutions for a viscous Burgers' equation. *J. Math. Systems Estim. Control 8*, 2 (1998), 16 pp. (electronic). 2.1

[47] Burns, J. A., Borggaard, J. T., Cliff, E., and Zietsman, L. An optimal control approach to sensor/actuator placement for optimal control of high performance buildings. In *International High Performance Buildings Conference* (2012). 1.5.1, 5.1

[48] Burns, J. A., He, X., and Hu, W. Control of the Boussinesq equations with implications for sensor location in energy efficient buildings. In *American Control Conference* (2012), Institute of Electrical and Electronics Engineers. 1.5.1

[49] Burns, J. A., and Hulsing, K. P. Numerical methods for approximating functional gains in LQR boundary control problems. *Math. Comput. Modelling 33*, 1-3 (2001), 89–100. Computation and control, VI (Bozeman, MT, 1998). 3.1

[50] Burns, J. A., Ito, K., and Powers, R. K. Chandrasekhar equations and computational algorithms for distributed parameter systems. In *Decision and Control, 1984. The 23rd IEEE Conference on* (1984), vol. 23, pp. 262–267. 3.1

[51] Burns, J. A., and Peichl, G. H. Control system radii and robustness under approximation. In *Robust Optimization-Directed Design.* Springer, 2006, pp. 25–61. 1.3.2

[52] Burns, J. A., Sachs, E. W., and Zietsman, L. Mesh independence of Kleinman-Newton iterations for Riccati equations in Hilbert space. *SIAM J. Control Optim. 47*, 5 (Oct. 2008), 2663–2692. 3.1

[53] Byers, R. Solving the algebraic Riccati equation with the matrix sign function. *Linear Algebra and its Applications 85*, 0 (1987), 267 – 279. 3.1

[54] Caicedo, J.-M., Dyke, S. J., and Johnson, E. A. Natural excitation technique and eigensystem realization algorithm for phase I of the IASC-ASCE benchmark problem: Simulated data. *Journal of Engineering Mechanics 130*, 1 (2004), 49–60. 4.1

[55] Candes, E., and Romberg, J. Sparsity and incoherence in compressive sampling. *Inverse problems 23*, 3 (2007), 969. 1.7

[56] Candès, E. J. Compressive sampling. In *Proceedings of the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures* (2006), pp. 1433–1452. 1.1, 1.7, 1.7.3, 1.7.4, 1.7.5

[57] Candès, E. J., Romberg, J., and Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory 52*, 2 (2006), 489–509. 1.7, 5.1

[58] Candès, E. J., and Tao, T. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory 52*, 12 (2006), 5406–5425. 1.7, 1.7, 5.1

[59] Casta-Selga, R., Lohmann, B., and Eid, R. Stability preservation in projection-based model order reduction of large scale systems. *Eur. J. Control 18* (2012), 122–132. 1.6.1

[60] Chambers, D. H., Adrian, R. J., Moin, P., Stewart, D. S., and Sung, H. J. Karhunen-Love expansion of Burgers model of turbulence. *Phys. Fluids 31* (1988), 2573–2582. 2.1

[61] Chaturantabut, S., and Sorensen, D. C. Discrete empirical interpolation for nonlinear model reduction. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on* (2009), IEEE, pp. 4316–4321. 1.6.1

[62] Chen, K. K., and Rowley, C. W. $\mathscr{H}_2$ optimal actuator and sensor placement in the linearised complex Ginzburg-Landau system. *J. Fluid Mech. 681* (2011), 241–260. 3.1

[63] Chen, K. K., and Rowley, C. W. Normalized coprime robust stability and performance guarantees for reduced-order controllers. *IEEE Transactions on Automatic Control 58*, 4 (2013), 1068–1073. 2

[64] Chen, K. K., and Rowley, C. W. Fluid flow control applications of $\mathscr{H}_2$ optimal actuator and sensor placement. In *Proceedings of the American Control Conference* (2014), pp. 4044–4049. 3.1

[65] Chen, K. K., Tu, J. H., and Rowley, C. W. Variants of dynamic mode decomposition: boundary condition, koopman, and fourier analyses. *Journal of nonlinear science 22*, 6 (2012), 887–915. 1.6.3, 5.1, A.3.1

[66] Chojnowska-Michalik, A., Duncan, T. E., and Pasik-Duncan, B. Uniform operator continuity of the stationary riccati equation in hilbert space. *Applied Mathematics and Optimization 25*, 2 (1992), 171–187. 3.2.3

[67] Curtain, R. F., and Zwart, H. *An introduction to infinite-dimensional linear systems theory*, vol. 21. Springer Science & Business Media, 1995. 1.4.1, 1.5.1

[68] Dai, W., and Yuksel, S. Observability of a linear system under sparsity constraints. *IEEE Transactions on Automatic Control 58*, 9 (2013), 2372–2376. 5.5

[69] Darivandi, N., Morris, K., and Khajepour, A. An algorithm for LQ optimal actuator location. *Smart Materials and Structures 22*, 3 (2013), 035001. 3.1

[70] Demmel, J. W. *Applied Numerical Linear Algebra.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, USA, 1997. 1.2, 3.3.2

[71] Dickinson, B. T., and Singler, J. R. Nonlinear model reduction using group proper orthogonal decomposition. *Int. J. Numer. Anal. Model. 7*, 2 (2010), 356–372. 2.1, 2.2

[72] Döhler, M.and Mevel, L. Fast multi-order computation of system matrices in subspace-based system identification. *Control Engineering Practice 20*, 9 (2012), 882–894. 4.1

[73] Donoho, D. L. Compressed sensing. *IEEE Transactions on Information Theory 52*, 4 (2006), 1289–1306. 1.7, 1.7, 5.1

[74] Donoho, D. L. For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution. *Communications on pure and applied mathematics 59*, 6 (2006), 797–829. 1.7

[75] Doyle, J. C. Guaranteed margins for LQG regulators. *IEEE Transactions on Automatic Control 23*, 4 (1978), 756–757. 1.5.3

[76] Drmač, Z., Gugercin, S., and Beattie, C. Quadrature-based vector fitting for discretized $\mathcal{H}_2$ approximation. *SIAM J. Sci. Comp. 37*, 2 (2015), A625–A652. 4.1.2

[77] Drmač, Z., Gugercin, S., and Beattie, C. Vector fitting for matrix-valued rational approximation. Tech. rep., Available as http://arxiv.org/abs/, 2015. 4.1.2

[78] Druskin, V., and Knizhnerman, L. Extended Krylov subspaces: approximation of the matrix square root and related functions. *SIAM Journal on Matrix Analysis and Applications 19*, 3 (1998), 755–771. 3.3.2

[79] Druskin, V., Knizhnerman, L., and Simoncini, V. Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation. *SIAM Journal on Numerical Analysis 49*, 5 (2011), 1875–1898. 3.4.1

[80] Eldar, Y. C., Kuppinger, P., and Bolcskei, H. Block-sparse signals: Uncertainty relations and efficient recovery. *IEEE Transactions on Signal Processing 58*, 6 (2010), 3042–3054. 5.3.4, 5.3.5, 5.3.6, 5.3.7, 5.5

[81] Fahl, M. Computation of POD basis functions for fluid flows with Lanczos methods. *Math. Comput. Modelling 34*, 1-2 (2001), 91–107. 3.4

[82] Fang, H., Sharma, R., and Patil, R. Optimal sensor and actuator deployment for hvac control system design. In *American Control Conference* (2014), IEEE, pp. 2240–2246. 5.1

[83] Feitzinger, F., Hylla, T., and Sachs, E. W. Inexact Kleinman-Newton method for Riccati equations. *SIAM J. Matrix Analysis Applications 31*, 2 (2009), 272–288. 3.1

[84] Fletcher, C. A. J. The group finite element formulation. *Comput. Methods Appl. Mech. Engrg. 37*, 2 (1983), 225–244. 2.2

[85] Foucart, S., and Rauhut, H. *A mathematical introduction to compressive sensing*. Springer, 2013. 1.7

[86] Ghosh, R., and Joshi, Y. Proper orthogonal decomposition-based modeling framework for improving spatial resolution of measured temperature data. *IEEE Transactions on Components, Packaging and Manufacturing Technology 4*, 5 (May 2014), 848–858. 5.1

[87] Giraud, L., Langou, J., Rozložník, M., and van den Eshof, J. Rounding error analysis of the classical Gram-Schmidt orthogonalization process. *Numer. Math. 101*, 1 (2005), 87–100. 3.4

[88] Glover, K. All optimal hankel-norm approximations of linear multivariable systems and their l∞-error bounds. *International Journal of Control 39*, 6 (1984), 1115–1193. 4.1.1

[89] Golub, G. H., and Van Loan, C. F. Matrix computations. 1996. *Johns Hopkins University, Press, Baltimore, MD, USA* (1996), 374–426. 1.2, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 3.3.2, 3.3.3, 4.2

[90] Grasedyck, L. Existence of a low rank or H-matrix approximant to the solution of a Sylvester equation. *Numer. Linear Algebra Appl 11* (2004), 371389. 3.3.2

[91] Gugercin, S., Antoulas, A. C., and Beattie, C. A. H2 model reduction for large-scale linear dynamical systems. *SIAM J. Matrix Analysis Applications 30*, 2 (2008), 609–638. 1.5.2, 1.6, 4.1.1, 4.3.1

[92] Gugercin, S., Polyuga, R. V., Beattie, C., and Van Der Schaft, A. Structure-preserving tangential interpolation for model reduction of port-Hamiltonian systems. *Automatica 48*, 9 (2012), 1963–1974. 1.6, 4.4.1, 4.4.1

[93] Gunzburger, M. D. *Finite element methods for viscous incompressible flows: a guide to theory, practice, and algorithms*. Computer science and scientific computing. Academic Press, 1989. 1.4.3

[94] Guo, C., and Laub, A. J. A Schur method for solving algebraic Riccati equations. *IEEE Trans. Auto. Control* (1979), 913–921. 3.1

[95] Gustavsen, B., and Semlyen, A. Rational approximation of frequency domain responses by vector fitting. *IEEE Transactions on Power Delivery 14*, 3 (1999), 1052–1061. 4.1.2

[96] Hay, A., Borggaard, J. T., and Pelletier, D. Local improvements to reduced-order models using sensitivity analysis of the proper orthogonal decomposition. *J. Fluid Mech. 629* (2009), 41–72. 1.6.2, 2.1

[97] Heyouni, M., and Jbilou, K. An extended block Arnoldi algorithm for large-scale solutions of the continuous-time algebraic Riccati equation. *Electron. Trans. Numer. Anal. 33* (2009), 53–62. 3.1, 3.3, 3.3.2, 1, 3.3.2, 3.3.3, 3.5, 3.6

[98] Holmes, P., Lumley, J. L., and Berkooz, G. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press, 1998. 1.6.2, 1.6.2, 5.1

[99] Holmes, P., Lumley, J. L., Berkooz, G., and Rowley, C. W. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, second ed. Cambridge Monographs on Mechanics. Cambridge University Press, Cambridge, 2012. 1.6.2, 1.7

[100] Hortmann, M., Perić, M., and Scheuerer, G. Finite volume multigrid prediction of laminar natural convection: Benchmark solutions. *International Journal for Numerical Methods in Fluids 11*, 2 (1990), 189–207. 5.4.3

[101] Houtzager, I., van Wingerden, J., and Verhaegen, M. Recursive predictor-based subspace identification with application to the real-time closed-loop tracking of flutter. *IEEE Transactions on Control Systems Technology 20*, 4 (2012), 934–949. 4.1

[102] Huang, S.-C., and Kim, J. Control and system identification of a separated flow. *Physics of Fluids (1994-present) 20*, 10 (2008), 101509. 4.1

[103] Iliescu, T., and Wang, Z. Are the snapshot difference quotients needed in the proper orthogonal decomposition? *SIAM Journal on Scientific Computing 36*, 3 (2014), A1221–A1250. 1.6.2

[104] IMTEK Simulation. *Oberwolfach Benchmark Collection*, 2003. Available at https://simulation.uni-freiburg.de/downloads/benchmark. 4.4.2, 2

[105] Ito, K. *Strong Convergence and Convergence Rates of Approximating Solutions for algebraic Riccati Equations in Hilbert Spaces*. No. v. 178302 in ICASE report. National Aeronautics and Space Administration, Langley Research Center, 1987. 1.3.2, 2.3.1, A.1, A.1, A.1.4, A.1.5, A.1.6

[106] Jaimoukha, I. M., and Kasenally, E. M. Krylov subspace methods for solving large Lyapunov equations. *SIAM J. Numer. Anal. 31*, 1 (Feb. 1994), 227–251. 3.1, 3.3, 3.3.2

[107] Jarvis, C. Reduced order model study of Burgers equation using proper orthogonal decomposition. Master's thesis, Virginia Tech, 2012. 1.6.2, 2.1

[108] Jarvis, C. *Parameter Dependent Model Reduction for Complex Fluid Flows*. PhD thesis, Virginia Tech, March 2014. 1.6.2, A.3

[109] Jbilou, K. Block Krylov subspace methods for large algebraic Riccati equations. *Numerical algorithms 34*, 2-4 (2003), 339–353. 3.1, 3.3

[110] Jbilou, K. An Arnoldi based algorithm for large algebraic Riccati equations. *Applied mathematics letters 19*, 5 (2006), 437–444. 3.1, 3.3

[111] Jbilou, K., and Riquet, A. J. Projection methods for large Lyapunov matrix equations. *Linear Algebra and its Applications 415* (2006), 344 – 358. Special Issue on Order Reduction of Large-Scale Systems. 3.3

[112] Ji, G., and Lasiecka, I. Partially observed analytic systems with fully unbounded actuators and sensors-FEM algorithms. *Computational Optimization and Applications 11* (1998), 111–136. A.1.7

[113] Jovanović, M. R., Schmid, P. J., and Nichols, J. W. Sparsity-promoting dynamic mode decomposition. *Physics of Fluids (1994-present) 26*, 2 (2014), 024103. A.4

[114] Juang, J.-N., and Pappa, R. S. An eigensystem realization algorithm for modal parameter identification and model reduction. *Journal of Guidance, Control, and Dynamics 8*, 5 (1985), 620–627. 4.1

[115] Juillet, F., Schmid, P. J., and Huerre, P. Control of amplifier flows using subspace identification techniques. *Journal of Fluid Mechanics 725* (2013), 522–565. 4.1

[116] Kailath, T. Some Chandrasekhar-type algorithms for quadratic regulators. In *Decision and Control, 1972 and 11th Symposium on Adaptive Processes. Proceedings of the 1972 IEEE Conference on* (1972), vol. 11, pp. 219–223. 3.1

[117] Kasinathan, D., and Morris, K. $\mathscr{H}_\infty$-optimal actuator location. *IEEE Trans. Automat. Control 58*, 10 (2013), 2522–2535. 3.1

[118] Kasper, K., Mathelin, L., and Abou-Kandil, H. A machine learning appraoch for constrained sensor placement. In *American Control Conference* (2015), pp. 4479–4484. 5.1

[119] Kenney, C., Laub, A. J., and Wette, M. Error bounds for Newton refinement of solutions to algebraic Riccati equations. *Mathematics of Control, Signals and Systems 3*, 3 (1990), 211–224. 3.3.3, 3.3.2, 3.3.3

[120] Kleinman, D. L. On the iterative technique for Riccati equation computations. *IEEE Transactions on Automatic Control 13* (1968), 114–115. 3.1

[121] Knezevic, D. J., Nguyen, N.-C., and Patera, A. T. Reduced basis approximation and a posteriori error estimation for the parametrized unsteady Boussinesq equations. *Mathematical Models and Methods in Applied Sciences 21*, 07 (2011), 1415–1442. 1.6

[122] Kramer, B. Model reduction of the coupled Burgers equation in conservation form. Master's thesis, Virginia Tech, 2011. 2, 2.2, 2.2.1

[123] Kramer, B. Solving algebraic Riccati equations via proper orthogonal decomposition. In *Proceedings of the 19th IFAC World Congress* (2014), pp. 7767–7772.

[124] Kung, S.-Y. A new identification and model reduction algorithm via singular value decomposition. In *Proc. 12th Asilomar Conf. Circuits, Syst. Comput., Pacific Grove, CA* (1978), pp. 705–714. 1.1, 1.6, 4.1, 4.2.4

[125] Kunisch, K., and Volkwein, S. Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition. *J. Optim. Theory Appl. 102*, 2 (1999), 345–371. 2.1

[126] Kwakernaak, H., and Sivan, R. *Linear Optimal Control Systems*. Wiley-Interscience, New York, 1972. 1.5, 1.5.5, 1.5.7, 1.5.8, 1.5.2, 1.5.9, 1.5.12, 1.5.13, 2.1, 3.1

[127] Lall, S., Marsden, J. E., and Glavaški, S. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International journal of robust and nonlinear control 12*, 6 (2002), 519–535. 1.6.2

[128] Layton, W. *Introduction to the numerical analysis of incompressible viscous flows*, vol. 6. Siam, 2008. 1.3, 1.3.1, 1.3.1, 1.3.1

[129] Le Quéré, P., and Behnia, M. From onset of unsteadiness to chaos in a differentially heated square cavity. *Journal of fluid mechanics 359* (1998), 81–107. 5.4.3, 5.4.3, 5.4.3

[130] Lee, C. H., and Tran, H. T. Reduced-order-based feedback control of the Kuramoto-Sivashinsky equation. *J. Comput. Appl. Math. 173*, 1 (2005), 1–19. 3.1

[131] LeVeque, R. J. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*, vol. 98. Siam, 2007. 1.4.3, 1.4.4, 1.4.3

[132] Lin, Y., and Simoncini, V. A new subspace iteration method for the algebraic Riccati equation. *Numerical Linear Algebra with Applications 22*, 1 (2015), 26–47. 3.1

[133] Lindquist, A. A new algorithm for optimal filtering of discrete-time stationary processes. *SIAM Journal on Control 12*, 4 (1974), 736–746. 3.1

[134] Longman, R. W., and Juang, J.-N. Recursive form of the eigensystem realization algorithm for system identification. *Journal of Guidance, Control, and Dynamics 12*, 5 (1989), 647–652. 4.1

[135] Luchtenburg, D. M., Noack, B. R., and Schlegel, M. An introduction to the POD Galerkin method for fluid flows with analytical examples and MATLAB source codes. Tech. rep., Institut für Strömungsmechanik und Technische Akustik, TU Berlin, 2009. 1.6.2, 5, 5.1, A.3

[136] Ma, Z., Ahuja, S., and Rowley, C. W. Reduced-order models for control of fluids using the eigensystem realization algorithm. *Theoretical and Computational Fluid Dynamics 25*, 1-4 (2011), 233–247. 4.1, 4.3.2

[137] Massarotti, N., Nithiarasu, P., and Zienkiewicz, O. Characteristic-based-split (cbs) algorithm for incompressible flow problems with heat transfer. *International Journal of Numerical Methods for Heat & Fluid Flow 8*, 8 (1998), 969–990. 5.4.3

[138] Mayo, A. J., and Antoulas, A. C. A framework for the solution of the generalized realization problem. *Linear algebra and its applications 425*, 2 (2007), 634–662. 4.1.2, 4.3.1

[139] Mendel, J. M. Minimum-variance deconvolution. *IEEE Transactions on Geoscience and Remote Sensing GE-19*, 3 (1981), 161–171. 4.1

[140] Mezić, I. Analysis of fluid flows via spectral properties of the koopman operator. *Annual Review of Fluid Mechanics 45* (2013), 357–378. 1.6.3

[141] Miller, R. K., and Michel, A. N. *Ordinary Differential Equations.* Academic Press, Incorporated, 1982. 1.4, 1.4.1, 1.4.2, 1.4.3

[142] Moore, B. Principal component analysis in linear systems: Controllability, observability, and model reduction. *Automatic Control, IEEE Transactions on 26*, 1 (1981), 17–32. 1.5.2, 1.6, 4.1.1

[143] Mullis, C. T., and Roberts, R. A. Synthesis of minimum roundoff noise fixed point digital filters. *IEEE Transactions on Circuits and Systems 23*, 9 (1976), 551–562. 4.1.1

[144] Needell, D., and Tropp, J. A. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis 26*, 3 (2009), 301–321. 1.7

[145] Omatu, S., Koide, S., and Soeda, T. Optimal sensor location problem for a linear distributed parameter system. *Automatic Control, IEEE Transactions on 23*, 4 (1978), 665–673. 5.1

[146] Opmeer, M. Decay of singular values of the Gramians of infinite-dimensional systems. Preprint. Available at http://www.maths.bath.ac.uk/~mo221/publications.htm. 3.1

[147] Pitstick, G., Cruz, J., and Mulholland, R. J. Approximate realization algorithms for truncated impulse response data. *IEEE Transactions on Acoustics, Speech and Signal Processing 34*, 6 (1986), 1583–1588. 4.1

[148] Pralits, J. O., and Luchini, P. Riccati-less optimal control of bluff-body wakes. In *Seventh IUTAM Symposium on Laminar-Turbulent Transition*. Springer Netherlands, 2010, pp. 325–330. 3.1

[149] Proctor, J. L., Brunton, S. L., and Kutz, J. N. Dynamic mode decomposition with control. *arXiv preprint arXiv:1409.6358* (2014). 1.6.3

[150] Qin, S. J. An overview of subspace identification. *Computers & chemical engineering 30*, 10 (2006), 1502–1513. 4.1

[151] Ravindran, S. S. Error analysis for galerkin pod approximation of the nonstationary boussinesq equations. *Numerical Methods for Partial Differential Equations 27*, 6 (2011), 1639–1665. 1.6.2

[152] Rebolho, D. C., Belo, E. M., and Marques, F. D. Aeroelastic parameter identification in wind tunnel testing via the extended eigensystem realization algorithm. *Journal of Vibration and Control 20*, 11 (2014), 1607–1621. 4.1

[153] Reynders, E. System identification methods for (operational) modal analysis: review and comparison. *Archives of Computational Methods in Engineering 19*, 1 (2012), 51–124. 4.1

[154] Roberts, J. D. Linear model reduction and solution of algebraic Riccati equations by use of the sign function. Engineering Report, CUED/B-Control Tr-13, Cambridge Univ, Cambridge, England, 1971. 3.1

[155] Rowley, C. W. Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos 15*, 03 (2005), 997–1013. 1.6, 1.6.2, 2, 3.4, 4.1.1, 4.3.2

[156] Rowley, C. W., MEZIĆ, I., Bagheri, S., Schlatter, P., and Henningson, D. S. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics 641* (2009), 115–127. 1.6, 1.6.3

[157] Rozložník, M., Tůma, M., Smoktunowicz, A., and Kopal, J. Numerical stability of orthogonalization methods with a non-standard inner product. *BIT 52*, 4 (2012), 1035–1058. 3.4

[158] Saad, Y. Numerical solution of large Lyapunov equations. In *Signal processing, scattering and operator theory, and numerical methods*, vol. 5 of *Progr. Systems Control Theory*. Birkhäuser Boston, 1990, pp. 503–511. 3.1, 3.3.2, 3.4

[159] Samadiani, E., Joshi, Y., Hamann, H., Iyengar, M. K., Kamalsy, S., and Lacey, J. Reduced order thermal modeling of data centers via distributed sensor data. *Journal of Heat Transfer 134*, 4 (2012), 041401. 5.1

[160] San, O., and Borggaard, J. T. Basis selection and closure for pod models of convection dominated boussinesq flows. In *21st International Symposium on Mathematical Theory of Networks and Systems* (2014). 5.4.3

[161] Sanathanan, C., and Koerner, J. Transfer function synthesis as a ratio of two complex polynomials. *IEEE Trans. Autom. Control 8*, 1 (1963), 56–58. 4.1.2

[162] Schmid, P. J. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics 656* (2010), 5–28. 1.6, 1.6.3, 1.6.3, 5.1

[163] Semeraro, O., Pralits, J. O., Rowley, C. W., and Henningson, D. S. Riccati-less approach for optimal control and estimation: an application to two-dimensional boundary layers. *Journal of Fluid Mechanics 731* (2013), 394–417. 3.1

[164] Silverman, L. M. Realization of linear dynamical systems. *IEEE Transactions on Automatic Control 16*, 6 (1971), 554–567. 4.2

[165] Silvester, D., Elman, H., and Ramage, A. *Incompressible Flow and Iterative Solver Software (IFISS), version 3.3*, October 2013. 3.6

[166] Simoncini, V. A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM J. Sci. Comput 29* (2007), 1268–1288. 3.3, 3.3.2, 3.5

[167] Simoncini, V. Computational methods for linear matrix equations. Submitted, January 2014. Available at `http://www.dm.unibo.it/$\sim$simoncin/list.html`. 3.4.1

[168] Simoncini, V., Szyld, D. B., and Monsalve, M. On two numerical methods for the solution of large-scale algebraic Riccati equations. *IMA Journal of Numerical Analysis* (2013). 3.1, 3.3, 3.4.1

[169] Singler, J. R. Convergent snapshot algorithms for infinite-dimensional Lyapunov equations. *IMA Journal of Numerical Analysis 31*, 4 (2011), 1468–1496. 3.4, 3.6

[170] Singler, J. R. Balanced POD for model reduction of linear PDE systems: convergence theory. *Numer. Math. 121*, 1 (2012), 127–164. 3.6

[171] Singler, J. R. Model reduction of linear PDE systems: A continuous time eigensystem realization algorithm. In *Proceedings of the American Control Conference* (2012), pp. 1424–1429. 4.1

[172] Singler, J. R. New POD error expressions, error bounds, and asymptotic results for reduced order models of parabolic pdes. *SIAM Journal on Numerical Analysis 52*, 2 (2014), 852–876. 1.6.2

[173] Singler, J. R., and Batten, B. A. Balanced POD for linear PDE robust control computations. *Comp. Opt. and Appl. 53*, 1 (2012), 227–248. 3.1

[174] Sirovich, L. Turbulence and the dynamics of coherent structures. i-coherent structures. ii-symmetries and transformations. iii-dynamics and scaling. *Quarterly of applied mathematics 45* (1987), 561–571. 1.6.2, 1.6.2, 3.4

[175] Sun, J. Residual bounds of approximate solutions of the algebraic Riccati equation. *Numerische Mathematik 76*, 2 (1997), 249–263. 3.3.3, 3.3.3

[176] Taylor, J. A., and Glauser, M. N. Towards practical flow sensing and control via pod and lse based low-dimensional tools. *Journal of fluids engineering 126*, 3 (2004), 337–345. 5.1

[177] Tissot, G., Cordier, L., Benard, N., and Noack, B. R. Model reduction using dynamic mode decomposition. *Comptes Rendus Mécanique 342*, 6 (2014), 410–416. 1.6.3

[178] Trefethen, L. N., and Embree, M. *Spectra and Pseudospectra*. Princeton University Press, Princeton, NJ, 2005. 3.5

[179] Tu, J. H., Rowley, C. W., Luchtenburg, D. M., Brunton, S. L., and Kutz, J. N. On dynamic mode decomposition: Theory and applications. *arXiv preprint arXiv:1312.0041* (2013). 1.6.3

[180] Unger, A., and Tröltzsch, F. Fast solution of optimal control problems in the selective cooling of steel. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik 81*, 7 (2001), 447–456. 4.4.2

[181] Unger, B. Impact of discretization techniques on nonlinear model reduction and analysis of the structure of the pod basis. Master's thesis, Virginia Tech, 2013. 1.6.2

[182] van den Berg, E., and Friedlander, M. P. Sparse optimization with least-squares constraints. *SIAM Journal on Optimization 21*, 4 (2011), 1201–1229. 1.7

[183] Verriest, E. I. Low sensitivity design and optimal order reduction for the LQG-problem. In *Proc. 24th Midwest Symp. Circ. Syst., Albuquerque, NM, USA* (1981), pp. 365–369. 3.1

[184] Viberg, M. Subspace-based methods for the identification of linear time-invariant systems. *Automatica 31*, 12 (1995), 1835–1851. 4.1

[185] Volkwein, S. Proper orthogonal decomposition for linear-quadratic optimal control. Lecture Notes, University of Konstanz, available at http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/scripts.php, 2013. 1.6.2, 1.6.2

[186] Volkwein, S. Proper orthogonal decomposition: Theory and reduced-order modelling. Lecture Notes, University of Konstanz, available at http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/scripts.php, 2013. 1.6.2, 1.6.2, 3.4

[187] Wales, C., Gaitonde, A., and Jones, D. Stabilisation of reduced order models via restarting. *International Journal for Numerical Methods in Fluids 73*, 6 (2013), 578–599. 4.1

[188] Willcox, K. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & fluids 35*, 2 (2006), 208–226. 5.1

[189] Willcox, K., and Peraire, J. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal* (2002), 2323–2330. 1.6, 1.6.2, 3.4, 3.4, 3.6, 4.1.1

[190] Wolf, T., Panzer, H., and Lohmann, B. On the residual of large-scale Lyapunov equations for Krylov-based approximate solutions. In *Proceedings of the American Control Conference* (2013), pp. 2606–2611. 3.5.1, 3.5.1

[191] Yu, D., and Chakravorty, S. A randomized proper orthogonal decomposition technique. *arXiv preprint arXiv:1312.3976* (2013). 4.1, 4.3.2