Svein Hovland

Model Reduction and Control in Computational Fluid Dynamics

Svein Hovland

Doctoral Thesis

Theses at NTNU, 2008:274

NTNU
Norwegian University of
Science and Technology
Thesis for the degree of
philosophiae doctor
Faculty of Information Technology, Mathematics and
Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU

Svein Hovland

Model Reduction and Control in Computational Fluid Dynamics

NTNU
Norwegian University of
Science and Technology

Summary

Computational fluid dynamics CFD gives engineers and researchers the opportunity to model accurately complex physical processes involving heat transfer and fluid flow. At the same time, one wishes to be able to design optimal model based controllers for such systems, for which no simple analytical solutions or compact models exist. Typically these models have a number of unknowns that exceeds 10 000, and sometimes even millions. Model based controller design for systems of such high dimensionality is infeasible due to the high computational requirements. Through the use of modern model order reduction techniques, one can bypass the high dimensionality of the computational fluid dynamics models during controller design. This thesis combines the scientific disciplines of computational fluid dynamics, model order reduction and control theory, as important steps towards employing real-time, optimal and model based control for systems described by high-dimensional models.

   The history of computational fluid dynamics is reviewed and the procedure is demonstrated through an example using the finite volume method. It is demonstrated how CFD models can be put in standard state-space form for analysis of system properties, such as stability, and a CFD model of an unstable system is stabilized through reduced-order control. Different model reduction techniques are introduced, focusing on methods that are particularly suited for control design and large-scale systems. A new way of selecting snapshots for snapshot-based model reduction is proposed.

   Some selected topics from control theory are included for completeness, in particular model predictive control and also the explicit solution of the

model predictive control problem based on multiparametric programming. This thesis proposes to use model reduction in order to make explicit model predictive control feasible for a larger number of systems, and it is shown that a significant reduction in online controller complexity can be achieved, without compromising performance and stability. Further, we consider output-feedback controller design based on reduced-order models. When using reduced-order models to design model-based controllers for complex systems, there always arises a question of guaranteed closed-loop stability in presence of the uncertainty introduced. Some important properties of the resulting closed-loop systems, and controller and observer criteria, for stability are established.

Moreover, this thesis presents a novel design procedure for robust model predictive control based on reduced-order models. The procedure gives provable closed-loop stability in the presence of the model approximation error introduced in the model reduction process. To our knowledge, this is the first time stability is proven for model predictive control designed based on reduced-order models.

Many physical systems in for instance mechatronics, micro-electric mechanical systems, rotating machinery, aerodynamics and acoustics are best described by CFD models with a large number of states. At the same time, they are characterized by very fast dynamics, such that the controllers applied are required to be equally fast. We develop fast model based controllers with constrained control input, in combination with state estimators in an output-feedback structure. For the first time, reduced-order models developed using a model constrained optimization-based reduction technique are used for constrained optimal control, demonstrating significantly improved performance over control design based on the standard methods for model reduction, such as proper orthogonal decomposition, that is most frequently used for large-scale systems. This is an important step towards achieving and actually implementing real-time, model based and constrained optimal control for such systems.

Preface

THIS thesis is submitted in partial fulfillment of the requirements for the degree Doctor of Philosophy at the Norwegian University of Science and Technology (NTNU), Department of Engineering Cyberetics, and it is based on research done in the period of July 2004 through March 2008.

There are many people who deserve recognition in a moment such as this. First of all, I want to extend my appreciation to my supervisor, Professor Tommy Gravdahl, who has not only guided me in my work and willingly agreed to send me across the world on conferences and research stays, but has also provided a great working environment through his understanding and flexibility.

Further, I would like to thank Associate Professor Karen Willcox for inviting me to stay for 9 fruitful months at the Aerospace Computational Design Laboratory at Massachusetts Institute of Technology, and for treating me as one of her own students. My stay at MIT has had an invaluable impact on the quality of this work, and has greatly improved my knowledge on model reduction and paper writing. I would also like to thank Tan Bui-Thanh for many interesting discussions on model reduction during my stay at MIT.

Further, I wish to thank my fellow PhD students and the staff at the Department of Engineering Cybernetics, NTNU. Tove, for always handling whatever request and problem that arises, and Eva for guidance in the frustrating world of bureaucracy. Stefano for passionate discussions on soccer, Dagfinn, for the coffee breaks and discussions and Bjørnar for creating a nice office environment and taking the time to tutor a fresh PhD student. Ak-

Publications

This doctoral work has resulted in the following publications:

- S. Hovland, C. Løvaas, J. T. Gravdahl and G. C. Goodwin. Stability of Model Predictive Control Based on Reduced-Order Models. Submitted to Control Engineering Practice, 2008.

- S. Hovland, K. Willcox and J. T. Gravdahl. Explicit Model Predictive Control for Large-Scale Systems via Model Reduction. AIAA Journal of Guidance, Control and Dynamics, **31**(4), 2008.

- S. Hovland and C. Løvaas and J. T. Gravdahl and G. C. Goodwin. Stability of MPC Based on Reduced-Order Models. 47th IEEE Conference on Decision and Control. Cancun, Mexico, December, 2008.

- S. Hovland and J. T. Gravdahl. Complexity Reduction in Explicit MPC through Model Reduction. 17th IFAC World Congress. Seoul, Korea, July 6-11, 2008.

- S. Hovland and J. T. Gravdahl. Stabilizing a CFD model of an unstable system through model reduction and feedback control. Modeling, Identification and Control, **27**(3), 2006.

- S. Hovland, K. Willcox and J. T. Gravdahl. MPC for Large-Scale Systems via Model Reduction and Multiparametric Quadratic Programming. 45th IEEE Conference on Decision and Control. San Diego, California, Dec. 13-15, 2006.

- S. Hovland and J. T. Gravdahl. Order Reduction and Output Feedback Stabilization of an Unstable CFD Model. American Control Conference. Minneapolis, Minnesota, June 14-16, 2006.

- S. Hovland and J. T. Gravdahl. Stabilizing a CFD model of an unstable system through model reduction and feedback control. 46th Scandinavian Conference on Simulation and Modeling. Trondheim, October, 2005.

Contents

Chapter 1

Introduction

"640k should be enough for anybody."
-Bill Gates, 1981

CONTRARY to what was envisioned in the opening quote, the world has
seen a formidable increase in computing power over the last decades.
Because of this, engineers and researchers take on greater and greater com-
puting tasks.

Computational fluid dynamics (CFD) has emerged as a powerful tool
in many areas of industry and academia. CFD is a joint designation for
numerical methods for solving and analyzing problems concerning fluid-,
heat- and mass flow by computer simulation. These methods include grid
generation, spatial and temporal discretization, solution of the resulting
equations, and presenting the results to the user.

The underlying phenomena in most CFD applications are described by
partial differential equations, which implies that the system state is infinite-
dimensional. A lot of effort has been put into designing control laws for
these distributed parameter systems. Most of these solutions are restricted
to problems with relatively simple geometries and flows, for example for in-
compressible channel flows, pipe flows and cylinder flows. Moreover, many
physical problems are multi-disciplinary, with several PDEs describing dif-
ferent effects within the problem domain. While this is very difficult to

1

handle with the theory of distributed parameter systems, it is relatively straightforward to set up such a problem in any commercial CFD software package. This is indeed the "raison d'être" for CFD.

Although CFD is a very useful tool for analyzing flow phenomena, the computational cost of solving CFD problems is high. It is not unusual that a CFD code needs hours, and even days and weeks to solve a difficult problem, for instance if three spatial dimensions are considered for a complex geometry and flow pattern. If optimization is to be performed based on a CFD model, for example to optimize a design, hundreds or even thousands of solutions are needed before an optimal design is found.

Moreover, since CFD analysis often gives accurate solutions that can help us understand the behavior of a given system, it is desirable to design control laws based on CFD models. We then face the following problems: CFD models

- are expensive to use for unsteady simulations,

- do not couple well with other disciplines such as active control and

- are too large for model based-, optimal- and robust control design.

Consequently, as engineers and researchers take on greater challenges with the increasing use of CFD, they are inevitably faced with the "curse of dimensionality".

Generally, controllers are of the same order as the plant. Consequently, it is prohibitively expensive to compute common controller structures for large-scale systems. When the plant has high dimensionality, the controllers also require extensive state information, are extremely difficult to tune and are expensive to implement and maintain.

To overcome these problems, the theory of model order reduction has emerged over the last decades. The motivation is clear from the viewpoint of a control engineer: With a low-order model at hand that approximates the necessary behavior of the CFD model well, we regain the opportunity to apply our large control system design toolbox.

Example 1. *Optimal Control of Reservoirs*
Oil and gas wells and reservoirs are usually described by complex CFD models with 10^3-10^6 dynamic variables, and are controlled by engineers based on complex simulation studies and the engineer's experience. There is a great potential for improving the operation by introducing optimal control strategies for the reservoirs (e.g. for water injection strategies). With proper use of model order reduction techniques, one can envision that approximate, low order models can be used to design model-based optimal controllers of low order.

1.1   Scope of Thesis

For systems with relatively simple flow regimes and geometries, one can aim at designing controllers and stabilizing the underlying system of partial differential equations through the use of controllers designed based on mathematical analysis of the PDEs. For the broader specter of systems, this is not feasible as flow regimes and geometries turn complex. It is, however, the great strength of CFD that one is able to describe such problems on a computer, and obtain very accurate simulation results that cannot be achieved by simplifying models and systems of partial differential equations. Through the use of model order reduction techniques, it is then possible to develop models of low dimension that capture the essential dynamics. This way, one can achieve improved performance and closed-loop stability for problems that would otherwise be impossible to even model with conventional analytical tools.

In this work, we consider models that result from spatial (and temporal) discretization of partial differential equations by using CFD techniques and software. We consider the problem of designing low order model-based optimal control for the high-fidelity CFD models. We focus on constrained control, since meeting constraints is important for systems in which safe operation is critical. In particular, we consider model predictive control, and the explicit solution to the model predictive control problem, and we strive to make these technologies applicable to systems described by CFD-models.

This requires model reduction, state estimation, handling of uncertainties and ensuring robust stability.

Chapter 2

Background Material

T HIS chapter introduces the tools and techniques that will be used in subsequent chapters to develop reduced-order models and low order controllers. Section 2.1 gives a brief introduction to CFD, Section 2.2 describes the system representations that we will consider, Section 2.3 gives an overview of the model reduction methodology that will be used, and Section 2.4 presents some control preliminaries. In Section 2.5 we discuss some issues that emerge when we apply controllers based on reduced-order models on the high-fidelity model, and in Section 2.6 we give a motivating example.

2.1   Computational Fluid Dynamics

This section presents the fundamentals of CFD, provides some motivating examples and reviews the basics of the methodology.

Definition 1. *Computational Fluid Dynamics*
Computational Fluid Dynamics or CFD is the analysis of systems involving fluid flow, heat transfer and associated phenomena by means of computer-based simulation (Versteeg and Malalasekera, 1995).

With the need for a better understanding of flow phenomena, the aerospace

industry became the driving force for the development of CFD techniques in the 1960s. The realization that CFD is cheaper and faster than experiments quickly made CFD an important tool in the design, R&D and manufacturing processes of aircraft and jet engines.

Over the years, the development of CFD codes has been intimately coupled to advances in computer hardware capabilities, since the solution of complicated flow problems requires the manipulation of thousands or even millions of numbers. Along with the exponential growth of processing speed and memory capacity[1], CFD has become a powerful and prominent tool that is subject to massive research, and is used within numerous areas of application, such as

- reservoir evaluation and simulation,

- design optimization,

- flow around vehicles, lift and drag computation,

- marine engineering,

- combustion modeling,

- fuel cell design and analysis,

- flow inside rotating passages etc.,

- chemical process engineering,

- electrical end electronic engineering,

- wind loading and ventilation in buildings,

- weather prediction,

- flow in rivers and oceans,

---

[1]Almost every measure of the capabilities of digital electronic devices is linked to Moore's Law; the number of transistors that can be inexpensively placed on an integrated circuit is increasing exponentially, doubling approximately every two years.

- flow in arteries and veins and

- earthquake modeling.

Still, many CFD applications require huge computing resources, and the size of problems that can be solved on an ordinary computer is quite limited. The following example illustrates the potential and computational requirements of state of the art CFD codes.

Example 2. *Earth Quake Simulation*
In Akcelik et al. (2003), the authors carry out 1 Hz simulations of the 1994 Northridge earthquake in Los Angeles with 100 million grid points. Their simulations are among the largest unstructured mesh computations reported to date, requiring multiple hours on thousands of processors.

Example 2 provides a stark contrast to the prophecy of the IBM chairman in the early days of computers:

"I think there is a world market for maybe five computers."
-Thomas Watson, chairman of IBM, 1943

2.1.1 A Brief Introduction to CFD

From a scientific viewpoint, computational fluid dynamics can be divided into three phases;

1. pre-processing,

2. solving equations, and

3. post-processing.

The main parts of these three elements will be summarized in the next three subsections. Most of the material in this subsection is based on Versteeg and Malalasekera (1995), but the literature on CFD is vast, and a number of excellent books exist (Ferziger and Peric, 2002, Anderson, 1995, Wesseling, 2001).

Pre-Processing

In the pre-processing phase, the problem is transformed into a format suitable for the solver. In this step, the user must define the computational domain, the governing equations, fluid properties and which phenomena that need to be modeled. An important part of specifying CFD problems, as well as when solving partial differential equations in general, is specification of appropriate boundary conditions (BC) and initial conditions (IC).

Then comes *gridding*; the sub-division of the computational domain into a number of small sub-domains. The result of the gridding process is a grid (or mesh), consisting of a (large) number of elements. The solution to the governing equations is defined at nodes inside each grid element. Consequently, the accuracy of the solution depends on the number of grid elements. Usually, the grid is finer in areas where large variations occur in the flow, and coarser in regions where little happens. Figure 2.1 shows a grid example for flow around a cylinder. Several different mesh types exist, such as uniform and non-uniform, regular and unstructured. A handbook of grid generation can be found in Thompson et al. (1998).
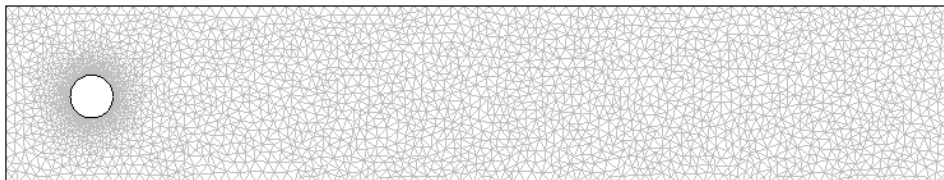
Figure 2.1: Example of a non-uniform, unstructured grid with 5557 elements, used for computing the flow around a cylinder located at the left. The grid is finer close to the cylinder, since this is where we have large gradients. The grid is generated with the commercial software Comsol Multiphysics.

Solving Equations

A common CFD solver performs the following steps:

- Approximation of unknown flow variables by simple functions.

- Discretization by substituting the approximations for the governing equations.

- Solution of the resulting algebraic equations.

There are three different directions when it comes to approximation and discretization; finite difference, finite element and spectral methods. The finite volume method is demonstrated in Section 2.6.

Solvers include familiar algorithms from linear algebra, such as Gauss-Seidel iteration, Krylov subspace methods and the conjugate gradient method. For large problems, the Multigrid method (Briggs and McCormick, 2000) has become very popular in recent years.

Post-Processing

The post-processing stage naturally deals with presenting to the user the results provided by the solver in the previous step. The post-processor usually provides a variety of plotting tools, particle tracking and animations. Figure 2.2 shows a two-dimensional surface plot for the velocity field around the cylinder in Figure 2.1.
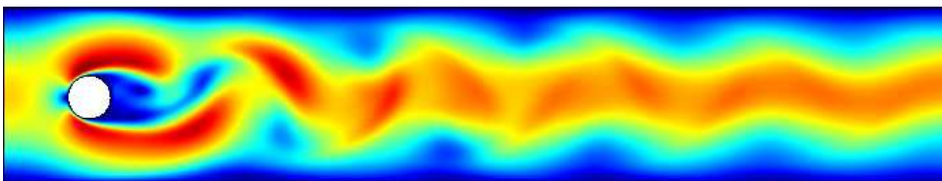
Figure 2.2: Flow around a cylinder. The solution is generated with the commercial software Comsol Multiphysics, using the grid in Figure 2.1.

2.2   System Description

In this section we discuss some properties of the types of systems that we will consider in later chapters.

2.2.1   CFD Models

Models that arise through spatial (and temporal) discretization of PDEs over the computational domain, are subsequently referred to as *CFD models*. The CFD models are assumed to be accurate representations of the underlying PDEs, which can be achieved by selecting a proper grid and numerical algorithm.

When discretizing linear partial differential equations, or when linearizing a nonlinear CFD system, we frequently end up with linear systems in generalized state-space form

$$E\dot{x} = Ax + Bu \tag{2.1a}$$

$$y = Cx, \tag{2.1b}$$

frequently referred to as *descriptor systems*. Here, $x \in \mathbb{R}^n$ represents the descriptor variables, $u \in \mathbb{R}^m$ contains the inputs and $y \in \mathbb{R}^p$ contains the outputs of the system, and $E, A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and $C \in \mathbb{R}^{p \times n}$. In CFD applications, x contains the n unknown flow quantities in the computational grid. Many commercial CFD software packages allow the user to export the CFD descriptions on the format (2.1). For nonlinear CFD codes, the linearization matrices E, A, B, C are evaluated at steady-state flow conditions. The state space matrices are typically sparse matrices of very large dimension, e.g. $n > 10^4$. Although these matrices could be manipulated to obtain a smaller state-space system, such a procedure is often complicated and can destroy the sparsity of the system. The sparsity is useful in numerical methods used in e.g. model reduction. The more general form (2.1) is therefore preferred. However, the state dimension of the system is still prohibitively large for many applications, such as flow control design.

In CFD applications, it is common that the matrix E contains some zero rows, which arise from flow boundary conditions. Consequently, the

matrix E can be singular. In this case, (2.1) consists of a combination of ordinary differential equations and algebraic equations. Such systems are referred to as differential algebraic equations (DAEs). With a slight abuse of notation, we shall subsequently refer to x as the system *state*, also in the case of singular E.

Assumption 1. *It is assumed in the following that the matrix pencil $(A - \lambda E)$ is regular, i.e. $(A - \lambda E)$ is singular only for a finite number of λ.*

Assumption 1 is not restrictive, and guarantees the existence and uniqueness of the solution of (2.1) for any specified initial condition.

In the following we shall use the notation $\mathcal{G}(E, A, B, C)$ to refer to systems of the form (2.1). If $E = I_n$, we use the notation $\mathcal{G}(A, B, C)$. We will also denote by $\mathcal{G}(s)$ and $\mathcal{G}_r(s)$ the transfer functions of the high-fidelity and reduced-order models, respectively.

2.2.2 Stability Properties of Descriptor Systems

The following theorem establishes stability of descriptor models.

Theorem 1. *A descriptor model $E\dot{x} = Ax$ is stable if all finite eigenvalues λ of $(A - \lambda E)$ are in the open left-half complex plane.*

The generalized eigenvalues λ can be obtained by solving the equation

$$\det(A - \lambda E) = 0. \tag{2.2}$$

In the discrete-time case, the system is stable if the generalized eigenvalues lie strictly inside the unit circle.

Remark 1. *Note that if A is negative definite while E is positive definite, the system $E\dot{x} = Ax$ is stable. This is, however, a conservative criterion, since a system may well be stable although this does not hold.*

2.3   Model-Order Reduction

This section defines the problem of model-order reduction, gives a short literature overview and presents some fundamentals and algorithms that are used in subsequent chapters.

2.3.1   Introduction and Problem Statement

Model-order reduction has emerged over the last couple of decades as an important tool to analyze and design controllers for complex systems.

The literature on model reduction is vast, particularly for linear systems. A survey can be found in Antoulas et al. (2001), and the books Antoulas (2005a) and Benner et al. (2005) describe many of these algorithms in detail. The monograph by Obinata and Anderson (2001) treats the application of model reduction techniques *for control* of linear systems, although large-scale systems are not covered specifically. For nonlinear systems, on the other hand, model reduction is still very much an open problem.

The model reduction problem can be stated as follows. For a system modeled by the nonlinear differential equation

$$\dot{x} = f\left(x, u\right) \tag{2.3a}$$

$$y = g\left(x, u\right), \tag{2.3b}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ contains the m inputs to the system and $y \in \mathbb{R}^p$ contains the p outputs; find a new dynamical system

$$\dot{x}_r = \hat{f}\left(x_r, u\right) \tag{2.4a}$$

$$y_r = \hat{g}\left(x_r, u\right), \tag{2.4b}$$

where $x_r \in \mathbb{R}^r$, $u \in \mathbb{R}^m$, and $y_r \in \mathbb{R}^p$ such that $r \ll n$ and the following criteria should be satisfied:

1. The approximation error is "small", preferably with a *global* error bound.

2. System properties, such as stability and passivity, are preserved.

3. The procedure is automatic, numerically stable and efficient.

If the system is modeled by a linear time invariant model of the general form (2.1), we seek an rth order approximation

$$\dot{x}_r = A_r x_r + B_r u \tag{2.5a}$$

$$y_r = C_r x_r, \tag{2.5b}$$

where $r \ll n$, $x_r \in \mathbb{R}^r$, $y_r \in \mathbb{R}^p$, $A_r \in \mathbb{R}^{r \times r}$, $B_r \in \mathbb{R}^{r \times m}$, $C_r \in \mathbb{R}^{p \times r}$, and subject to the same criteria as above.

Comment 1. *An alternative to model-order reduction as described above, is to develop a low-dimensional model by identifying the major characteristics and most important physical phenomena of an initially complex model of the system at hand. Such characteristics could be time scales and spatial variations, for example. Based on this, one can then tailor the low-dimensional model so as to incorporate these characteristics. This procedure is not automatic, and it requires great knowledge about the system in question. On the other hand, one can ensure that specific physical properties and relations are handled properly in the simplification process. Successful use of such an approach is demonstrated by Storkaas, Skogestad, and Godhavn (2003).*

Preservation of system properties such as stability and passivity gives advantages when it comes to controller design. For example, given a passive system described by a passive high-order model[2], a passivity preserving model reduction procedure can be used to find a passive model of low order. Then, a (strictly) passive model based controller of low order can be designed. The closed loop consisting of the plant and the low order controller is then provably stable, using arguments from the theory of interconnections of passive systems. Preservation of passivity is particularly important in applications such as circuit design, where large circuits consisting of passive

---

[2]Although a given plant or system of partial differential equations is passive, the high-fidelity CFD model designed to describe the plant is not necessarily passive. In order to ensure this, a discretization scheme that preserves the passivity property should be used (Kristiansen and Egeland, 2000).

circuit elements are to be replaced by smaller circuits using a smaller number of passive elements. Several researchers have studied this problem, among others Antoulas (2005b), Bai and Freund (2001), and Sorensen (2004). In fluid flow applications, however, the issue of passivity preservation is less important, since the systems encountered are rarely passive.

Model reduction *for control* is somewhat different from model reduction for simulation purposes, and it is treated among others by Obinata and Anderson (2001) and Zhou et al. (1996). A reduced-order model that gives good approximation in open loop may not necessarily be a good approximation in closed loop, since the system dynamics change once the feedback loop is closed. If the ultimate objective is the low-order controller (rather than the low order model), then it is essential that the closed-loop performance objective be incorporated in the reduction technique. A common approach is to use frequency weighting in order to emphasize the importance of approximation quality in the bandwidth of the closed-loop system. Another approach is to use iterative plant- and controller reduction in a closed-loop configuration (see e.g. Wortelboer et al., 1999).

Next, we will briefly introduce some model reduction techniques that will be used in later chapters.

2.3.2   Balanced Truncation

Balanced truncation is a standard technique for model reduction of stable, linear systems, and can be found in many standard references on control (see e.g. Zhou et al., 1996). It was originally introduced to the control community by Moore (1981). Although the method is computationally demanding when the system order is large, recent and ongoing research address the extension of these algorithms to large-scale settings (Sorensen and Antoulas, 2002, Gugercin and Antoulas, 2004, Li and White, 2002, Benner et al., 2000). Modern numerical linear algebra techniques has allowed balanced truncation techniques to be applied efficiently to systems of order up to $n = 10^6$ (Benner, 2007).

Loosely speaking, balanced truncation is done by truncating states that give the least contribution to the input-output behavior. This motivates

considering the controllable and observable subspaces of the state space. The controllable subspace contains the set of states that can be reached with zero initial state and a given input $u(t)$, while the observable subspace comprises those states that, as initial conditions, can produce a non-zero output $y(t)$ without external input. The controllability and observability grammians $\mathcal{P}$ and $\mathcal{Q}$ are $n \times n$ matrices whose eigenvectors span the controllable and observable subspaces, respectively. If the system is minimal, the Gramians are positive definite. The following fundamental theorem gives conditions for the existence of the Gramians.

Theorem 2. *If $\mathcal{G}(A, B, C)$ is exponentially stable, then the controllability and observability Gramians $\mathcal{P}$ and $\mathcal{Q}$ exist, and are the unique positive definite solutions to the Lyapunov equations*

$$A\mathcal{P} + \mathcal{P}A^T + BB^T = 0, \tag{2.6}$$

$$A^T\mathcal{Q} + \mathcal{Q}A + C^TC = 0. \tag{2.7}$$

A system is said to be *balanced* when the states that are excited most by input are at the same time the states that produce the most output energy. In such a realization, the grammians are both equal to a diagonal matrix, say Σ, with the elements σ_i on the diagonal in descending order,

$$\mathcal{P} = \mathcal{Q} = \Sigma. \tag{2.8}$$

The diagonal elements σ_i are called the system's Hankel singular values. Model reduction by balanced truncation proceeds by first obtaining the balanced system realization, and then truncating the states with small Hankel singular values.

The error introduced by balanced truncation is upper bounded by

$$\|\mathcal{G}(s) - \mathcal{G}_r(s)\|_\infty \leq 2 \sum_{k=r+1}^{n} \sigma_k. \tag{2.9}$$

This means that the error is equal to twice the sum of the truncated Hankel singular values. The error can also be represented in terms of a time-domain

output error,

$$\|y\left(t\right) - y_r\left(t\right)\|_2 \leq 2 \sum_{k=r+1}^{n} \sigma_k \|u\left(t\right)\|_2. \tag{2.10}$$

Remark 2. *From Theorem 2 it is easily understood that balanced truncation is restricted to* stable *systems.*

Several extensions to balanced truncation exist. It is especially worth mentioning LQG balanced truncation (Jonckheere and Silverman, 1983), that is specifically targeted at control applications by considering a closed-loop balanced realization, and is applicable to unstable systems, contrary to the standard implementation. Some nonlinear extensions also exist, see for example Scherpen (1993) and Lall et al. (2002), and the references therein.

2.3.3 Model Reduction by Projection

Model reduction by projection is a general framework that can be used to describe many reduction algorithms for large-scale systems. For a general system, described as in equation (2.3), model reduction by projection works as follows. It is assumed that the state x can be approximated by a linear combination of r basis vectors

$$x \approx \Phi_r x_r, \tag{2.11}$$

where $x_r \in \mathbb{R}^r$ is the reduced state and $\Phi_r \in \mathbb{R}^{n \times r}$ is a projection matrix containing as columns the r basis vectors $\phi_1, \phi_2, \ldots, \phi_r$. Substituting (2.11) into (2.3), and requiring the resulting residual to be orthogonal to the space spanned by Φ_r gives the reduced model

$$\dot{x}_r\left(t\right) = \Phi_r^T f\left(\Phi_r x_r\left(t\right), u\left(t\right)\right) \tag{2.12a}$$

$$y_r\left(t\right) = g\left(\Phi_r x_r\left(t\right), u\left(t\right)\right), \tag{2.12b}$$

where $x_r \in \mathbb{R}^r$ is the reduced state and $y_r \in \mathbb{R}^p$ is the output of the reduced model.

For linear systems, the reduced state-space model is given by

$$E_r \dot{x}_r = A_r x_r + B_r u \tag{2.13a}$$
$$y_r = C_r x_r, \tag{2.13b}$$

where

$$E_r = \Phi_r^T E \Phi_r, \tag{2.14}$$
$$A_r = \Phi_r^T A \Phi_r, \tag{2.15}$$
$$B_r = \Phi_r^T B, \tag{2.16}$$

and

$$C_r = C \Phi_r. \tag{2.17}$$

Several model reduction algorithms use the general projection framework just described; however, they differ in the computation of the projection matrix Φ_r.

2.3.4   Proper Orthogonal Decomposition

First introduced independently by Karhunen (1946) and Loève (1946), proper orthogonal decomposition (POD) is sometimes called the Karhunen-Loève expansion. The method is also known as principal component analysis. When first applied in the context of fluid mechanics in Lumley (1967), it was used to study turbulent flows. Applicable even for very high-order systems and non-linear problems, POD has become the most popular method within the field of model reduction and control for CFD applications. This approach has been considered for active control purposes by numerous authors (Kunisch and Volkwein, 1999, Astrid et al., 2002, Ravindran, 2000, Benner and Saak, 2005, Atwell et al., 2001, Afanasiev and Hinze, 2001). However, there are several limitations associated with using the POD; in particular, POD-based reduced models lack the quality guarantees of those derived using more rigorous methods, such as balanced truncation. Even in the case of stable LTI systems, reduction via POD can lead to undesirable and unpredictable results, such as unstable reduced models.

POD can be described in view of the projection framework described in Section 2.3.3. In the search of the basis vectors Φ_r, the POD procedure proceeds as follows. Collect a finite number of M samples $x(t_i)$ from (2.1) or (2.3), for $t = t_1, \ldots t_M$, in a matrix of snapshots

$$\mathcal{X} = \left[x^1, x^2, \ldots, x^M \right] = \left[x(t_1), x(t_2), \ldots, x(t_M) \right], \qquad (2.18)$$

where the columns $\{\mathcal{X}_{\cdot,\, j}\}_{j=1}^M$ can be thought of as the spatial coordinate vectors of the system at time step t_j. The rows $\{\mathcal{X}_{i,\, \cdot}\}_{i=1}^n$ describe the time trajectories of the system evaluated at different locations in the spatial domain (Kunisch and Volkwein, 1999). The snapshots may be taken from physical experiments or from computer (CFD) simulations.

For a given number of basis vectors r, the POD basis is found by minimizing the error Δ between the original snapshots and their representation in the reduced space, defined by

$$\Delta = \sum_{i=1}^M \left[x(t_i) - \tilde{x}(t_i) \right]^T \left[x(t_i) - \tilde{x}(t_i) \right], \qquad (2.19)$$

where $\tilde{x}(t_i) = \Phi_r \Phi_r^T x(t_i)$.

The minimizing solution Φ_r can be found via the set of left singular vectors of the snapshot matrix $\mathcal{X}$, which is conveniently computed using the singular value decomposition of $\mathcal{X}$,

$$\mathcal{X} = \Phi \Sigma \Psi^T, \qquad (2.20)$$

where the columns of $\Phi = [\phi_1, \ldots, \phi_M]$ form the optimal orthogonal basis for the space spanned by $\mathcal{X}$. Φ and Ψ are unitary matrices (i.e. $\Phi^{-1} = \Phi^T, \Psi^{-1} = \Psi^T$) and Σ is a diagonal matrix with the singular values σ_i of $\mathcal{X}$ on the diagonal. The r most significant basis functions are associated with the r largest singular values σ_i, $i = 1, \ldots r$, of $\mathcal{X}$. If the singular values σ_i fall of rapidly in magnitude, a reduced-order model may be constructed by projection using Φ_r consisting of the r first columns of Φ. These basis functions are the ones that capture the most salient characteristics of the snapshot data $\mathcal{X}$.

The reduced-order model will capture only the dynamics present in the snapshot data, and so the choice of snapshots is critical. Suitable inputs should therefore be used to excite the system, so that the desired characteristics are present in the data. Frequently, snapshots are taken from the impulse- or step responses of the CFD model. Moreover, some methods exist for adaptively deciding how many snapshots to include, and where to take them, see for example Meyer and Matthies (2003) or Hinze and Volkwein (2005).

Proper orthogonal decomposition is summarized in Algorithm 1.

Algorithm 1. *Proper Orthogonal Decomposition*

1. *Simulate the state equations and record snapshots $\mathcal{X}$ of the system state.*

2. *Perform singular value decomposition of the snapshot data, as in (2.20).*

3. *Extract the r most significant basis vectors Φ_r based on the singular values σ_i of the snapshot matrix $\mathcal{X}$.*

4. *Project the governing equations onto the reduced basis as in (2.12) or (2.14)-(2.17) to find the reduced model.*

2.3.5   Goal-Oriented Model-Constrained Reduction

Goal-oriented model-constrained reduction is a reduction algorithm proposed in Bui-Thanh et al. (2007), that also uses the general projection framework in Section 2.3.3. In this procedure, a cost similar to (2.19) is used as an objective function in an optimization formulation. The optimization problem seeks to find the rth-order basis $\Phi_r = [\phi_1, \ldots, \phi_r] \in \mathbb{R}^{n \times r}$ and the corresponding reduced-order state solution $x_r(t) \in \mathbb{R}^r$ so that the $\mathcal{L}_2$-norm of the error between the full-order and reduced-order output is

minimized[3]. For the linear model (2.1), this can be formulated as

$$\min_{\Phi_r, x} \frac{1}{2} \sum_{l=1}^{\mathcal{S}} \int_0^T \left(y^l - y_r^l \right)^T \left(y^l - y_r^l \right) dt \tag{2.21a}$$

$$+ \frac{\beta}{2} \left[\sum_{j=1}^r \left(1 - \phi_j^T \phi_j \right)^2 + \sum_{i,j=1, i \neq j}^r \left(\phi_i^T \phi_j \right)^2 \right]$$

subject to:

$$\Phi_r^T E_r^l \Phi_r \dot{x}_r^l = \Phi_r^T A_r^l \Phi_r x_r^l + \Phi_r^T B_r^l u^l, \ l = 1, \dots, \mathcal{S} \tag{2.21b}$$

$$\Phi x_r^l (0) = x^l (0), \ l = 1, \dots, \mathcal{S} \tag{2.21c}$$

$$y_r^l = C^l \Phi x_r^l, \ l = 1, \dots, \mathcal{S}. \tag{2.21d}$$

The summation over l allows one to consider a finite set of $\mathcal{S}$ instantiations of the governing equations (2.1) that could arise from variations in the coefficient matrices E, A, B and C, the input u, or the initial state x_0. The superscript l thus denotes the lth instance of the system, which has corresponding state $x^l (t)$, input $u^l (t)$, and output $y^l (t)$. For example, where (2.1) represents a spatially discretized PDE, these variations stem from changes in the domain shape, boundary conditions, coefficients, initial conditions or sources of the underlying PDEs.

The two key differences between the formulation (2.21) and the POD are that the model-constrained optimization approach

1. enforces the reduced-order governing equations as constraints, and

2. minimizes the *output* error, while the POD minimizes the error of state prediction over the entire domain.

The former issue ensures that the error $(y - y_r)^T (y - y_r)$ in (2.21a) is evaluated for y_r that are achieved by simulating the reduced-order model, and

---

[3]If y and y_r are taken to be the impulse response of $\mathcal{G}(s)$ and $\mathcal{G}_r(s)$, respectively, then $\|y - y_r\|_{\mathcal{L}_2}$ is equal to the difference $\|\mathcal{G}(s) - \mathcal{G}_r(s)\|_{\mathcal{H}_2}$ in $\mathcal{H}_2$-norm between the two system transfer functions.

not just from projection, as for $\tilde{x}$ in (2.19). By emphasizing the importance of an accurate approximation of *input-to-output* behavior instead of attempting to minimize the error over the entire state domain, it is hoped that reduced models are obtained that are e.g. more suitable for use in an output-feedback implementation.

The full-order output $y^l(t)$ is obtained from simulating the high-fidelity model over a selected set of inputs and the interval $t \in [0,\ T\rangle$. The second term in the cost function (2.21a) is a regularization term to yield orthonormal basis vectors, with β as a regularization parameter.

This approach retains applicability to nonlinear systems, but addresses some of the limitations of the POD by targeting the projection basis to output functionals of interest, and by bringing additional knowledge of the reduced-order governing equations into the construction of the basis. Formulation of the problem of determining the basis as an optimal control problem has also been considered for distributed parameter systems by Borggaard (2006).

Determining the basis via the optimization procedure will in general be more computationally demanding than using POD. However, this additional offline cost is a tradeoff that can be made, if necessary, to achieve low order models of acceptable quality.

2.4   Control Preliminaries

In this section, some preliminaries about the control theory used subsequently will be described.

2.4.1   The Linear-Quadratic Regulator

The linear-quadratic regulator (LQR) is a model-based optimal control scheme. For a discrete-time linear system given by $x_k = Ax_k + Bu_k$, the feedback control law is found by minimizing the cost functional defined by

$$J = \sum_{k=0}^{\infty} \left(x_k^T Q x_k + u_k^T R u_k \right) \qquad (2.22)$$

where Q and R are design weighting matrices that penalize deviation from zero of the states, and use of control energy, respectively. The feedback control law that minimizes this cost is given by

$$u = -K_{lq}x, \qquad (2.23)$$

where K_{lq} is found as (Kwakernaak and Sivan, 1972)

$$K_{lq} = (R + B^T P B)^{-1} B^T P A, \qquad (2.24)$$

and P is found by solving the discrete-time algebraic Riccati equation

$$P = Q + A^T \left(P - PB \left(R + B^T P B \right)^{-1} B^T P \right) A. \qquad (2.25)$$

2.4.2 Model Predictive Control

Model predictive control (MPC) policies are optimization based control policies that calculate the current control input by solving a constrained optimization problem, with a cost similar to (2.22), parameterized by the current system state. This strategy has been widely adopted in the industrial process control community and implemented successfully in many applications. The greatest strength of MPC is the intuitive way in which constraints can be incorporated in a multivariable control problem formulation. Here we will give a brief introduction to a standard MPC formulation. For further reading on MPC, there exists a number of books (Maciejowski, 2001), (Allgöwer and Zheng, 2000) and tutorials (Rawlings, 2000).

A Standard MPC Formulation

Model predictive control is formulated for a discrete-time state-space model

$$x_{k+1} = Ax_k + Bu_k, \qquad (2.26a)$$
$$y_k = Cx_k, \qquad (2.26b)$$

where $k \in \mathbb{Z}$, and $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$ and $y_k \in \mathbb{R}^p$ denote the state, inputs and outputs, respectively, at time step k. The constant matrices A, B and

C are of appropriate dimensions, and (A, B) is a controllable pair. For the regulator problem (regulating the system states to zero), the model predictive controller solves at time step k the optimization problem

$$\min_{U_k} \left\{ x_{k+N|k}^T P x_{k+N|k} \right. \tag{2.27a}$$

$$\left. + \sum_{i=0}^{N-1} \left(x_{k+i|k}^T Q x_{k+i|k} + u_{k+i}^T R u_{k+i} \right) \right\}$$

subject to:

$$u_{\min} \leq u_{k+i} \leq u_{\max}, \ i = 0, \ldots, N - 1 \tag{2.27b}$$

$$y_{\min} \leq y_{k+i} \leq y_{\max}, \ i = 1, \ldots, N \tag{2.27c}$$

$$u_{k+1} = K x_{k+i|k} \ , \ N_u \leq i \leq N - 1 \tag{2.27d}$$

$$x_{k|k} = x_k \tag{2.27e}$$

$$x_{k+i+1|k} = A x_{k+i|k} + B u_{k+i}, \ i \geq 0 \tag{2.27f}$$

$$y_{k+i|k} = C x_{k+i|k}, \ k \geq 0, \tag{2.27g}$$

where P and Q are design weighting matrices of appropriate dimensions that penalize deviation from zero of the states x_{k+i} at the end of the prediction horizon N and over the entire horizon, respectively. In this work, the final cost matrix P and gain K are calculated from the algebraic Riccati equation, under the assumption that the constraints are not active for $k \geq N$. The weight R penalizes use of control action u. The notation $(\cdot)_{k+i|k}$ is used to emphasize that the predictions $(\cdot)_{k+i}$ are made based on the value at step k. N_u defines the control horizon, which is the number of future control moves to be optimized. In this work, we set $N_u = N$, for convenience. The sequence $U_k = \begin{bmatrix} u_0^T & u_1^T & \ldots & u_{N_u-1}^T \end{bmatrix}^T$ contains the future control inputs that yield the best predicted output with respect to the performance criterion on the prediction horizon. Once this set has been found, the first control input u_0 is applied to the process, before the whole optimization problem is re-solved at the next sample. The optimization problem is then slightly different, having been updated by a new process

measurement, a new starting point and an additional time slice at the end of the time horizon.

It is well established that implementing a linear model predictive controller requires solving a quadratic program (QP) in U_k at each time step (Maciejowski, 2001). With some manipulations, the problem in (2.27) can be written

$$\min_{U_k} \left\{ \frac{1}{2} U_k^T H U_k + x_k^T F U_k \right\} \tag{2.28a}$$

$$\text{subject to: } G U_k \leq W + E x_k, \tag{2.28b}$$

where the matrices H, F, G, W and E are functions of the weighting matrices P, Q, R and the bounds $u_{\min}$, $u_{\max}$, $y_{\min}$ and $y_{\max}$. If the weighting matrices in (2.27a) satisfy $P \succeq 0$, $R \succ 0$ and $Q \succeq 0$, then $H \succ 0$ and the problem is strictly convex. The Karush-Kuhn-Tucker conditions (KKT) are then sufficient conditions for optimality (Nocedal and Wright, 1999, page 333), and the solution U_k can be shown to be unique (Bemporad et al., 2002). The assumptions on Q and R are usually met by choosing Q and R to be diagonal matrices that appropriately penalize the relative importance of state or input values.

This traditional MPC strategy requires significant online computation, limiting the use of this kind of controller to processes with small system state dimension or relatively slow dynamics, since the optimization problem that is solved at each sampling time can otherwise become large.

2.4.3 Soft Constraints

When MPC is applied, a process can operate near, or even *at* specified process constraints. In many cases this leads to the most cost effective operation for a given plant, since constraints are often directly associated with cost. But system constraints sometimes cause problems with respect to the feasibility of the optimization problem to be solved by the model predictive controller. Unexpectedly large disturbances may occur, forcing the system to a state from which there is no way of keeping it within the specified limits without breaking some set of constraints. Feasibility problems may

also occur due to modeling errors, especially for linearized systems, or when initializing the system, potentially outside the intended region of operation.

Preferably, infeasibility of the MPC optimization problem should be avoided at all costs. In Kerrigan and Maciejowski (2001, 2000a) methods are presented that allows one to determine a priori whether or not an MPC controller has this desirable property, when the effects of the disturbances have been neglected in the design of the controller. The authors apply invariant set theory to establish which initial states guarantee feasibility of the MPC controller for all time. Nevertheless, mechanisms should be implemented that ensure that the control system has a way of dealing with feasibility problems. Several possible solutions for handling such problems have been proposed, ranging from simple, but sub-optimal approaches like using the same control signal as in the previous time step, to more refined approaches like that of Vada et al. (2001), where the constraints are relaxed in an optimal manner subject to a user-defined prioritization. The approach that will be considered in this thesis is constraint softening by means of slack variables. One advantage with this approach is that the optimization to be performed by the MPC controller at each step remains a quadratic program.

Constraints are normally divided into two different classes. Input constraints, such as actuator and valve limitations are typical examples of physical limitations that will lead to *hard* constraints. A hard constraint is absolute, in that it can under no circumstances be violated. A valve can only be opened to a certain limit, and this limit cannot be exceeded. Output or state constraints, however, are not necessarily absolute. For example, it may be desirable for a given process to operate within a specific temperature range. But one might consider allowing for the system temperature to exceed the desired range, if this is the only way of keeping the system within some level of control. A constraint that may be violated if required, is called a *soft* constraint.

By introducing *slack variables* to the problem formulation the desired constraints can be softened effectively. The slack variables are zero if no constraints are violated. By penalizing the non-zero values of the slack variables in the cost function, the constraint violations are kept to a minimum.

Penalty functions that lead to constraint violation and use of slack only if

the original problem is otherwise left infeasible are called *exact* penalty functions. Consequently, the constraints will not be violated unnecessarily if the penalty function is exact. In order to achieve an exact penalty function, the 1-norm or the ∞-norm must be used to penalize constraint violations, and the penalty weight must be sufficiently large (Kerrigan and Maciejowski, 2000b, Hovland, 2004).

2.4.4   Explicit MPC via Quadratic Programming

It has recently been shown that a great deal of the computational effort in traditional MPC can be done offline. In Bemporad et al. (2002), the authors proposed solving multiparametric quadratic programs (mpQPs) that are used to obtain explicit solutions to the MPC problem, such that the control input can be computed by evaluating a piecewise affine function of the current system state. Thus, the explicit model predictive controller (eMPC) accomplishes online MPC functionality without solving an optimization problem at each time step.

In parametric programming, the solution to a mathematical program is found explicitly for a range of parameter values. Mathematical programs that contain more than a single parameter are commonly referred to as multiparametric programs (Tøndel, 2003, page 1-2). The problem (2.28) can be viewed as an mpQP in U_k, where x_k is a vector of parameters.

Following Bemporad et al. (2002), consider (2.28), and define

$$z \triangleq U_k + H^{-1}F^T x_k. \qquad (2.29)$$

Then, the problem in (2.28) can be transformed into

$$\min_z \left\{ \frac{1}{2} z^T H z \right\} \qquad (2.30a)$$

$$\text{subject to: } Gz \leq W + Sx_k, \qquad (2.30b)$$

which is an mpQP in z, parameterized by x_k. The matrix S is found as $S = E + GH^{-1}F^T$. By considering the KKT conditions of this quadratic program in z, the solution z^* is seen to remain optimal in a neighborhood

of x_k where the active set remains optimal. The region in which this active set remains optimal can be shown to be a polyhedron in the parameter space (that is, the state space) (Bemporad et al., 2002). The mpQP in z can be solved offline for the state space area of interest. Computing the control input at a time step k then becomes a straightforward task: Given the system state x_k, the optimal control inputs U_k are obtained through an affine mapping,

$$U_k = K_i x_k + k_i, \ i = 1, \ldots, N_p \tag{2.31}$$

where N_p is the number of polyhedra and the subscript i denotes the ith affine function. K_i and k_i are constant within each polyhedron in the parameter space. The online effort is thus reduced from solving a potentially large optimization problem at each time step to evaluating a piecewise affine function of the current state, by determining the region i in which the current state x_k resides.

This has several advantages: Firstly, the online computational time can be reduced to the microsecond-millisecond range, and secondly, MPC functionality is achieved with low complexity, easily verifiable real-time code. Further, execution is deterministic, and there is no need for floating point arithmetics (no recursive numerical computations). All these advantages justify the employment of eMPC in embedded and safety-critical systems. Hegrenæs et al. (2005) consider using eMPC for spacecraft attitude control. In Johansen et al. (2006) the authors consider hardware implementation of eMPC, where memory requirements, computational speeds and hardware architecture design is studied using field programmable gate arrays (FPGA) and an application specific integrated circuit (ASIC).

2.5 Low-Order Controllers for Large-Scale Systems

In this section, we discuss some issues relevant to the task of developing model-based or optimal controllers of low order to a high-fidelity model.

2.5.1   Different Paths to a Low Order Controller

Simple controllers are normally preferred over complex controllers, since the computational requirements are smaller, hardware design and implementation is less complex and error-prone, and they are more transparent to the user. For this reason, low order controllers are preferred over high order controllers. Also, the need for real-time control of many physical systems necessitates controllers that are of low order. In general, model-based or optimal controllers, such as LQG and $\mathcal{H}_\infty$ controllers, designed for a given plant have roughly the same dimension as the plant. The need for complexity reduction is therefore evident whenever the plant model is large. There are several fundamentally different approaches to designing controllers of low order, as illustrated in Figure 2.3.
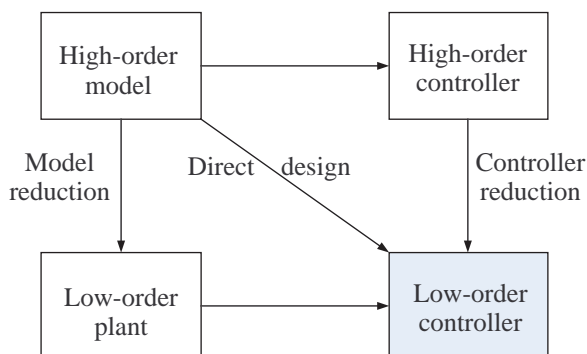

Figure 2.3: Different avenues for low order controller design.

The different procedures can be summarized as follows:

1. Perform direct design of a low-order controller based on a high-order model.

2. Design an initial controller for the plant/high-order model, and then reduce the order of the controller.

3. Perform plant model reduction and design a controller based on the reduced-order plant model.

Procedure 1 usually depends heavily on some properties of the plant, and requires great computations if the state dimension of the plant is large. The approach is outside the scope of this work and interested readers are referred to the literature (Hsu et al., 1994, Bernstein and Haddad, 1989, Iwasaki and Skelton, 1993, Gu et al., 1993, Gu. et al., 1993). Procedure 2 is very common for systems of medium size, for instance in the robust control community, where tools such as $\mathcal{H}_\infty$ design are frequently used to design an initial controller, followed by controller reduction. This procedure has been studied for CFD models by, among others, Atwell et al. (2001), Atwell and King (2005). The main drawback of this approach is that it requires the design of an appropriate initial controller, which is not feasible in many applications where the state dimension is large. This leaves us with the third approach, albeit this procedure is often criticized for introducing approximation (and consequently errors) at an earlier stage in the design process, which may propagate errors into the controller design. This can, however, be compensated for by designing controllers robust to uncertainties and modeling errors. Also, with a plant model with small state dimension available, we may use our large toolbox for control system design. Model reduction for control of large-scale systems has been considered in a number of settings (Kunisch and Volkwein, 1999, Ravindran, 2000, Atwell et al., 2001, Afanasiev and Hinze, 2001, Ahuja et al., 2007, Cohen et al., 2006, Kunisch and Volkwein, 2006, Willcox and Megretski, 2005, Evans, 2003). One recently proposed approach that seems promising, is the Optimality System POD method (Kunisch and Volkwein, 2006), which generates reduced models for control by iteratively computing a POD basis that targets the closed-loop optimality system.

An alternative to the approaches sketched in Figure 2.3, is to obtain a low-order model directly by closed-loop identification, where the identification criterion takes the control performance objective into account, and to use this model for controller design. According to Codrons et al. (1999), the question whether to use model reduction or identification is of secondary im-

portance, whereas the critical issue is to include closed-loop considerations in the process. In our opinion, however, if a high-fidelity model is available, one should make use of this knowledge when constructing a low-order model. We therefore prefer to use model reduction rather that closed-loop identification, although both approaches are viable.

2.5.2   Output-Feedback Control with Reduced-Order Model

When a controller is designed, we need to connect the controller to the plant or high-fidelity model. When we are using controllers designed based on a reduced-order model, we need to compute an estimate of the reduced-order state variable x_r, based on the output of the CFD model, using some sort of state estimator. The structure of the closed loop is illustrated in Figure 2.4.
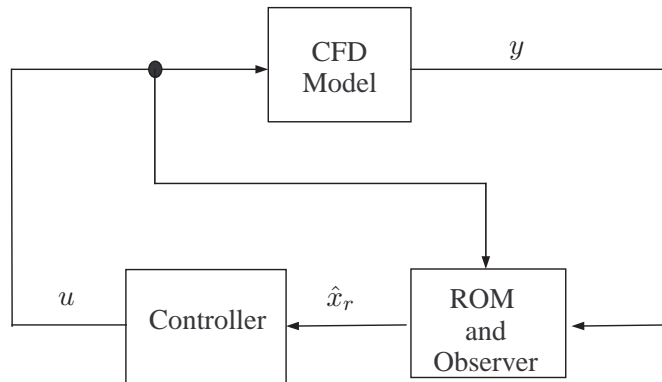
Figure 2.4: Block diagram of the reduced-order output-feedback setup. $\hat{x}_r$ is an estimate of the reduced state based on an observer, using the reduced model (ROM) and measurements y from the CFD model.

Output feedback control combined with model uncertainty may lead to system instability, although the original model is stable and the controller

stabilizes the reduced model. The mere existence of controllers stabilizing the reduced-order model and not the plant (Linnemann, 1988) necessitates stability analysis of the closed-loop system.

2.5.3 Closed-Loop Stability of Linear CFD models

In this section we will present an example of closed-loop stability analysis for a simple control structure.

Consider a linear high-fidelity model of the form (2.1), for which we have derived a reduced-order model of the form (2.5). Based on this model, we can design a controller using any model-based controller synthesis tool, such as LQG, LQR, or a robust controller using tools such as $\mathcal{H}_\infty$ design. The controller, which can also contain a state observer, is given by the general controller state-space model

$$\dot{x}_c = A_c x_c + B_c u_c \tag{2.32}$$

$$y_c = C_c x_c, \tag{2.33}$$

where $x_c \in \mathbb{R}^r$ is the controller state, u_c contains the inputs to the controller, such as the plant output, and the output of the controller is the input to the plant, i.e. $y_c = u$.

The closed-loop system is given by

$$E\dot{x} = Ax + Bu = Ax + BC_c x_c \tag{2.34}$$

$$\dot{x}_c = A_c x_c + B_c u_c = A_c x_c + B_c Cx, \tag{2.35}$$

or

$$\bar{E}\dot{\bar{x}} = \bar{A}\bar{x}, \tag{2.36}$$

where $\bar{x} = \begin{bmatrix} x^T & x_c^T \end{bmatrix}^T$,

$$\bar{E} = \begin{bmatrix} E & 0 \\ 0 & I_r \end{bmatrix} \tag{2.37}$$

and

$$\bar{A} = \begin{bmatrix} A & BC_c \\ B_c C & A_c \end{bmatrix} \tag{2.38}$$

We then have the following result:

Theorem 3. *The closed loop system consisting of the full model* (2.1) *and the output-feedback controller* (2.32) *is stable, provided that the generalized eigenvalues of* $(\bar{A} - \lambda \bar{E})$ *are stable, i.e.* $\lambda(\bar{A}, \bar{E}) \subset \mathbb{C}^- \cup \{\infty\}$, *where*

$$\bar{A} = \begin{bmatrix} A & BC_c \\ B_cC & A_c \end{bmatrix}$$

and

$$\bar{E} = \begin{bmatrix} E & 0 \\ 0 & I_r \end{bmatrix}.$$

Proof. The result follows directly from Theorem 1. ■

In the case where $E = I_n$, it suffices to check the eigenvalues of $\bar{A}$.

The following example illustrates the design process for a particular output-feedback design.

Example 3. *Based on the ROM, we design the continuous-time LQR counterpart of Section 2.4.1,*

$$u = -K_r x_r. \tag{2.39}$$

We design an observer

$$\dot{\hat{x}}_r = A_r \hat{x}_r + B_r u + L_r (y - C_r \hat{x}_r) \tag{2.40}$$

$$\hat{y}_r = C_r \hat{x}_r, \tag{2.41}$$

such that $(A_r - L_r C_r)$ *is Hurwitz, and we use feedback from the estimated reduced state, i.e.*

$$u = -K_r \hat{x}_r. \tag{2.42}$$

Our control structure takes the form of Figure 2.4. Now, the closed-loop system is stable provided that the generalized eigenvalues of $(\bar{A} - \lambda \bar{E})$ *are stable, where* $\bar{A}$ *and* $\bar{E}$ *are given by*

$$\bar{A} = \begin{bmatrix} A & -BK_r \\ L_rC & (A_r - B_rK_r - L_rC_r) \end{bmatrix}, \tag{2.43}$$

and

$$\bar{E} = \begin{bmatrix} E & 0 \\ 0 & I_r \end{bmatrix}. \tag{2.44}$$

2.6 Order Reduction and Stabilization of an Unstable CFD Model

This section serves as a motivating example, in which we consider stabilization of a computational fluid dynamics model of an *unstable* process model. We illustrate how to set ut a simple CFD model based on partial differential equations and discretization via the finite volume method. It is further shown how the CFD model can be put in a standard state-space form. A stabilizing controller is found based on optimal control design for the reduced-order model and then applied to the full model, where it is shown to stabilize the system. This section is based on Hovland and Gravdahl (2006a,b,c).

2.6.1 Introduction

While most of the CFD models in the reduced-order control literature using POD are nominally stable, we here extend the focus to *unstable* models in this chapter. This contribution demonstrates the possibility of designing stabilizing controllers to a class of systems that would otherwise be very computationally demanding or maybe even infeasible, due to the large state-dimension of such CFD models.

2.6.2 Case Study: Heated Plate

CFD Model

To demonstrate how an unstable system can be stabilized using POD and feedback control, we study heat conduction in a plate. The plate is $1\,\text{m} \times 1\,\text{m}$, defining the two-dimensional computational domain $\Omega = [0, 1] \times [0, 1]$ depicted in figure 2.5. The plate is insulated along the boundaries, apart from

the center of each boundary, where four flux actuators are located. This defines Neumann boundary conditions on all boundaries.
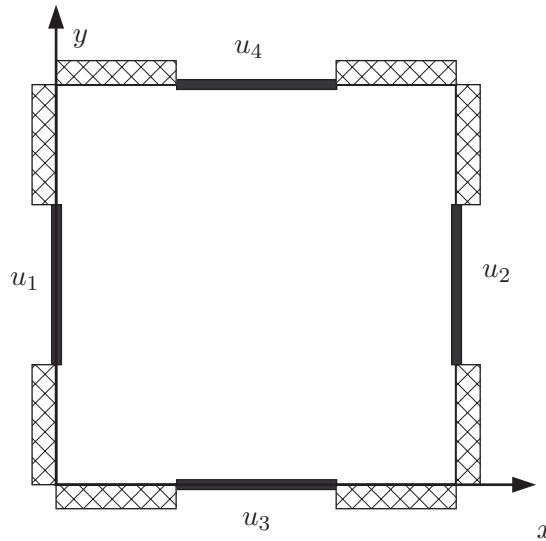
Figure 2.5: Sketch of plate with actuators on boundaries (bold lines).

The temperature $T(t, x, y)$ of the plate is governed by the unsteady linear two-dimensional heat equation

$$\rho c_p \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2} + k \frac{\partial^2 T}{\partial y^2} + S, \qquad (2.45)$$

where ρ and c_p are the density and specific heat capacity of the plate, respectively, and k is the thermal conductivity, that is assumed to be uniform over the computational domain and independent of temperature. Note that x now and in the following denotes a spatial coordinate and no longer the state variable. The source term $S \triangleq S_c + S_T$ is a term containing heat sinks and sources. In the present problem, convective heat transfer to the surroundings gives rise to a sink term

$$S_c = hA(T - T_\infty) \ \ [\text{W}], \qquad (2.46)$$

where h is the convective heat transfer coefficient, A is the heat transfer area of the surface and T_∞ is the ambient temperature. Due to electric current, the plate is subject to an internal temperature-dependent heat source

$$S_T = k_1 T \; \left[\mathrm{W} \, / \, \mathrm{m}^3 \right],$$ (2.47)

where $k_1 > 0$, at all points except from the boundary. Intuitively, this positive feedback from the temperature to the source may lead to a physically unstable system if the convective heat loss to the surroundings is not large enough. An increase in temperature will then lead to a stronger source, which again increases the temperature, and so on.

Discretizing the governing equation by the finite volume method, (2.45) is integrated over each control volume (CV) and over the time interval from t to $t + \Delta t$, to obtain (Versteeg and Malalasekera, 1995)

$$\int_{CV} \left(\int_t^{t+\Delta t} \rho c_p \frac{\partial T}{\partial t} dt \right) dV = $$
$$\int_t^{t+\Delta t} \int_{CV} \left(k \frac{\partial^2 T}{\partial x^2} \right) dV \, dt$$
$$+ \int_t^{t+\Delta t} \int_{CV} \left(k \frac{\partial^2 T}{\partial y^2} \right) dV \, dt + \int_t^{t+\Delta t} \int_{CV} S \; dV \, dt,$$

where the order of integration has been changed for the first term. Using the numerically unconditionally stable backward Euler (fully implicit) temporal discretization and n grid points over the spatial domain Ω, the system (2.45) can be written as a system of n equations of the form

$$a_P T_P = a_W T_W + a_E T_E + a_S T_S + a_N T_N + a_P^0 T_P^0 + S_u,$$ (2.48)

where the a's are coefficients and T_P is the temperature at the grid point (point P) under consideration at time step $k + 1$. S_u and S_P arise from discretizing the source term S as

$$\Delta V \cdot S = S_u + S_P T_P,$$ (2.49)

ρ	c_p	k	h	T_∞	k_1
1000	1000	1000	100	293	1000

Table 2.1: Numerical values of parameters.

where S_P is included in a_P. Using the convenient compass notation, T_W, T_E, T_S and T_N are the temperatures at the west, east, south and north adjacent grid points, respectively, at time step $k + 1$.

T_P^0 is the temperature at grid point P at time step k. Collecting the temperature at all grid points in a row vector $T(k) \in \mathbb{R}^n$ leads to a discrete linear system of the form

$$
\begin{array}{rcl}
ET(k+1) & = & \bar{A}T(k) + \bar{B}u(k) + \bar{V}, \\
y(k) & = & \bar{C}T(k),
\end{array}
$$
(2.50)

where $E \in \mathbb{R}^{n \times n}$ is a penta-diagonal matrix containing the coefficients a_p, a_W, a_E, a_S and a_N and $\bar{A} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with a_P^0 on the main diagonal.

$\bar{B} \in \mathbb{R}^{n \times m}$ contains the contributions from the inputs, while the constant source terms give rise to a constant term $\bar{V} \in \mathbb{R}^n$.

To validate that the plate model is unstable, we compute the generalized eigenvalues λ of $(A - \lambda E)$, using the numerical parameter values in Table 2.1, which confirms that the system has a pole outside the unit circle, at $\lambda = 1.0001$.

When the system matrices are of very high order, designing a model-based stabilizing controller is a computationally demanding task. This motivates the search for a reduced-order model.

Reduced-Order Model

The PDE (2.45) is discretized using 50 grid points in both the x- and y-direction. This gives in total 2500 states in the CFD model. To construct a model of reduced order, we use proper orthogonal decomposition, as outlined in Section 2.3.4, Algorithm 1. The system (2.50) is simulated for

$M = 600$ time steps, thus forming the matrix of snapshots $\mathcal{X}$. During this simulation the inputs are varied randomly taking moderate step changes over a suitable range to excite as much of the system dynamics as possible. SVD of the snapshot matrix is performed, and the singular values are considered in order to form the POD basis Φ_r, as depicted in figure 2.6.
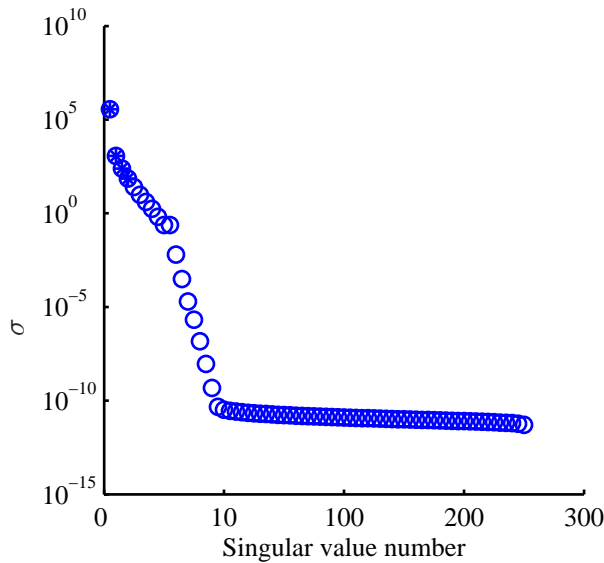
Figure 2.6: Singular values σ of the snapshot matrix. The $*$'s indicate singular values corresponding to the extracted basis functions. Note that the ordinate axis is logarithmic.

As can be seen from the figure the singular values fall off quite rapidly, and many of the singular values are close to zero, indicating that the basis functions corresponding to those singular values can be omitted without loss of information. There is no systematic approach to establish how many basis functions that should be included in Φ_r. The heuristic criterion

$$P = \frac{\sum_{i=1}^{r} \sigma_i^2}{\sum_{i=1}^{M} \sigma_i^2}, \tag{2.51}$$

gives an indication on how much of the energy that is conserved in the reduced-order model. If $P \approx 1$ most of the energy is captured in the first r basis functions, indicating a fairly accurate reduced-order model (Astrid et al., 2002). If we choose $r = 4$ basis functions, $P = 99.99\%$. Moreover, if the reduced-order model has four states the number of states in the reduced-order model is equal to the number of inputs. Consequently, the reduced-order model is fully actuated, which might be favorable when tracking a reference profile for the complete state. The reduced-order model is seen to be controllable and hence also stabilizable.

Using the projection framework outlined in Section 2.3.3 we get the reduced-order model

$$\Phi_r^T E \Phi_r T_r (k+1) = \Phi_r^T A \Phi_r T_r (k) + \Phi_r^T B u (k) + \Phi_r^T V. \tag{2.52}$$

Defining $E_r \triangleq \Phi_r^T E \Phi_r$ allows us to write

$$T_r (k+1) = E_r^{-1} \Phi_r^T A \Phi_r T_r (k) + E_r^{-1} \Phi_r^T B u (k)$$
$$+ E_r^{-1} \Phi_r^T V, \tag{2.53}$$

where E_r is invertible since E, Φ_r^T and Φ_r are all nonsingular. This yields the reduced-order model on discrete state-space form

$$T_r (k+1) = A_r T_r (k) + B_r u (k) + V_r \tag{2.54a}$$
$$y_r (k) = C_r T_r (k) , \tag{2.54b}$$

where $T_r \in \mathbb{R}^r$, $u \in \mathbb{R}^m$, $y_r \in \mathbb{R}^p$, $A_r = E_r^{-1} \Phi_r^T A \Phi_r \in \mathbb{R}^{r \times r}$, $B_r = E_r^{-1} \Phi_r^T B \in \mathbb{R}^{r \times m}$, $V_r = E_r^{-1} \Phi_r^T V \in \mathbb{R}^r$ and $C_r \in \mathbb{R}^{p \times r}$. In this example, $r = m = 4$. To ensure tracking for the plate temperature, we set C to be the $n \times n$ identity matrix. Consequently, $C_r \in \mathbb{R}^{n \times r}$.

The reduced-order model (2.54) is unstable since

$$\rho (A_r) = 1.0001. \tag{2.55}$$

Remark 3. *Note that the general POD procedure does not automatically preserve stability properties during the reduction process. Nominally stable*

models may result in unstable reduced-order models, and vice versa. One criterion for preserving stability properties in POD is presented in Prajna (2003). The result is however not applicable to models of very high order.

The reduced-order state $T_r(k)$ is estimated online through a linear observer of the form

$$\hat{T}_r(k+1) = (A_r - LC_r)T_r(k) + B_r u(k) + V_r + Ly(k), \qquad (2.56)$$

where $y(k)$ is the output from the high-order CFD model and L is chosen such that $\rho(A_r - LC_r) < 1$.

2.6.3 Controller Design

Feedback control is performed by use of heat flux actuators on parts of the boundary of the domain, shown as the bold lines in figure 2.5. The control objective is to reach a constant temperature reference T^d while at the same time rejecting disturbances. The reference temperature T^d is set to be a uniform temperature of $300°$ K.

Since the full model is too large for controller design the reduced-order model is analyzed instead. The reduced-order reference T_r^d is found as $T_r^d = \Phi_r^T T^d$. Given the unstable reduced-order model (2.54), the control objective is to stabilize the system around the reference temperature. Defining the tracking error as

$$e(k) \triangleq T_r^d - T_r(k), \qquad (2.57)$$

the control input is chosen as

$$u = Ke = K\left(T_r^d - T_r(k)\right), \qquad (2.58)$$

where K is chosen such that $\rho(A_r - B_r K) < 1$. The controller gain K is taken to be the solution to the linear quadratic regulator problem as defined in Section 2.4.1.

Using feedback from the estimated temperature $\hat{T}_r$, we can construct the closed-loop matrices $\bar{A}$ and $\bar{E}$ as in (2.43) and (2.44). By computing the generalized eigenvalues of the closed-loop system, we can then conclude

that the closed-loop system is stable, since the poles of the closed-loop systems lie strictly inside the unit disc. The largest closed-loop eigenvalue lies at $z = 0.9973$. The will, however, be a steady state error, due to the disturbance V.

Taking into consideration the disturbance V, the controller should include integral action in order to minimize the steady-state tracking error. To do this in a straightforward way, we define the augmented state

$$\tilde{T}(k) \triangleq \begin{bmatrix} T_r(k) \\ u(k-1) \end{bmatrix} \in \mathbb{R}^{r+m}, \tag{2.59}$$

giving an augmented state-space model

$$\begin{aligned} \tilde{T}(k+1) &= \tilde{A}\tilde{T}(k) + \tilde{B}\Delta u(k) + \tilde{V}, \\ \tilde{y}(k) &= \tilde{C}\tilde{T}(k), \end{aligned} \tag{2.60}$$

where

$$\begin{aligned} \tilde{A} &\triangleq \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}, \quad \tilde{C} \triangleq \begin{bmatrix} C & 0 \end{bmatrix}, \\ \tilde{B} &\triangleq \begin{bmatrix} B \\ I \end{bmatrix}, \qquad \tilde{V} \triangleq \begin{bmatrix} V \\ 0 \end{bmatrix}, \end{aligned} \tag{2.61}$$

and $\Delta u(k) = u(k) - u(k-1)$. In this augmented state-space model, integral action is built-in, and the input increment $\Delta u(k)$ is found as

$$\Delta u(k) = K\left(T_r^d - T_r(k)\right),$$

where K is the feedback gain matrix found above.

2.6.4 Numerical Simulation

Initially, the plate temperature is at rest, and equal to the ambient temperature at $293\,\mathrm{K}$. At $t = 0$ the inner source is switched on. Without control the temperature of the plate is strictly increasing. The plate temperature is shown for four different time instants in figure 2.7.
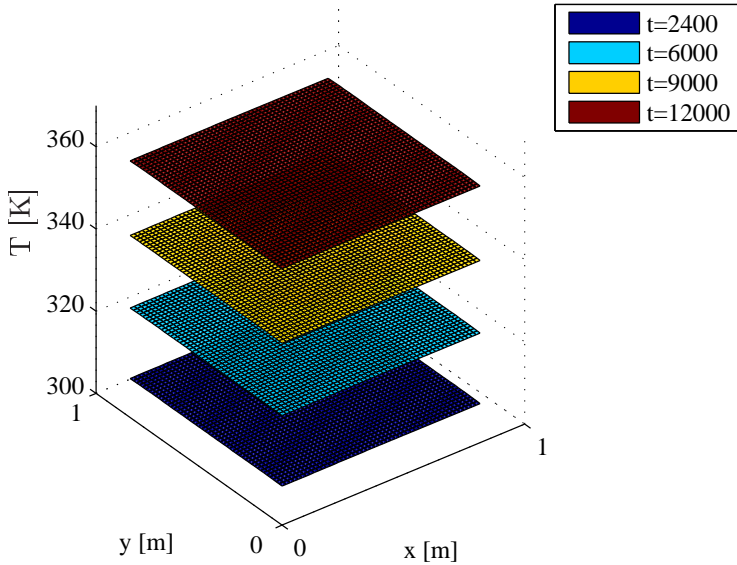
Figure 2.7: Plate temperature without control, shown for $t = 2400$, 6000, 9000 and 12000 s. The temperature is strictly increasing with time.

If the simulation is run for a longer period of time the temperature continues to increase, illustrating the instability of the system.

Now, the full CFD-model is simulated with the controller designed for the reduced-order model in section 2.6.3. The weighting matrices Q and R are set to $Q = 50 \cdot I_r$ and $R = 10^{-4} \cdot I_m$. The system is stabilized, and it is simulated until steady-state is reached, after approximately $t = 100$ minutes. The largest steady-state error is close to $3\,\mathrm{K}$, as shown in figure 2.8.

It is seen that although the original CFD model is symmetric, the controller based on the reduced-order model does not manage to exploit this
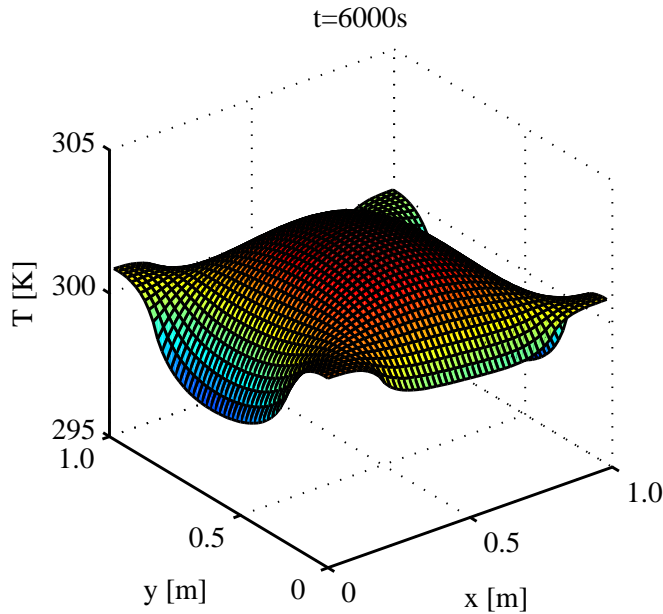
Figure 2.8: Steady state temperature, shown here for $t = 6000\,\mathrm{s}$.

symmetry, since the symmetry is not preserved in the model-order reduction scheme.

2.6.5   Concluding Remarks

In this section we have demonstrated, using a case study, that a CFD-model of an unstable system can be stabilized through model-order reduction and a controller designed for the reduced-order model. This makes it possible to design stabilizing controllers for systems that would otherwise be very computationally demanding.

It should be noted that expansion into orthonormal basis function is only applicable for square-integrable signals. Unstable systems generally have responses which are not square-integrable, and consequently the theory

of POD does not apply. In this work, however, the instability is slow, and so the responses do not blow up, and we are able to collect meaningful snapshots and the subsequent POD expansion works well. However, one should take care when using POD on unstable systems, as these responses may blow up and make approximation by an orthonormal basis impossible.

Chapter 3

Complexity Reduction in Explicit MPC

I$^{\text{N}}$ this chapter we propose to use model reduction techniques to make explicit model predictive control possible for a larger number of applications and for longer control horizons. The material deviates slightly from the rest of the thesis, since we mainly consider models with a relatively low number of states and, at this point, the results are not applicable to most CFD-models. However, we present a design procedure that can prove essential for achieving this goal eventually, as the field of explicit MPC and multiparametric programming is further developed. The chapter is based on Hovland and Gravdahl (2008).

3.1   Introduction

The traditional MPC strategy presented in Section 2.4.2 demands a significant amount of online computation, limiting the use of this kind of controller to processes with relatively slow dynamics, since an optimization problem is solved at each sampling time. The *explicit* solution of the model predictive control problem, presented in Section 2.4.4, leads to online constrained optimal control without having to solve an optimization problem at each

time step.

The main drawback of eMPC is the large increase in both offline and online complexity as the state dimension of the system model grows larger and the control horizon and the number of constraints are increased. For this reason, the procedure is limited to models of relatively low order, typically with less than 10 states. This has motivated the use of complexity reduction techniques, such as input parametrization, as discussed in Tøndel and Johansen (2002).

The main contribution of this chapter is the combination of eMPC and rigorous model reduction techniques with upper bounds on the approximation error, thereby reducing the complexity of eMPC. This makes the control scheme attractive for a number of systems that would otherwise be excluded due to the high complexity of the resulting controllers. The proposed use of model reduction techniques is demonstrated for several applications, among others for control of fuel cell breathing. In all applications, a significant reduction in controller complexity is achieved.

For clarity, we use the basic balanced truncation algorithm presented in Section 2.3.2 to compute reduced-order models in this chapter, albeit techniques focusing on closed-loop approximation quality, such as LQG balanced truncation or frequency-weighted balanced truncation, are assumed to further improve performance in our results.

3.2   Reduced-Order MPC

Reduced-order models will be used to design output-feedback eMPC controllers for the systems. The eMPC control input is computed based on the reduced state vector $x_r(k)$ at every time step k, and x_r must therefore be estimated by an observer, based on measurements from the plant (or the output of the original model). When we are dealing with output constraints, it is particularly important that the output of the reduced-order model is a good estimate of the plant output, in order to satisfy the output constraints for the plant. The observer(s) should therefore account for the approximation error in the reduced model.

A basic linear observer such as the Luenberger observer, does not account explicitly for uncertainties, that are amplified by the observer gain matrices. Consequently, the state estimate may not be accurate enough in the presence of model perturbation. We therefore follow common practice in the MPC literature (Astrid et al., 2002, Muske and Rawlings, 1993), and use a Kalman filter, which is known to have desirable properties for systems with noise in outputs and state equations. The Kalman filter is here defined in terms of the discretized reduced model with added noise,

$$\hat{x}_r(k+1) = A_r \hat{x}_r(k) + B_r u(k) + \Gamma w(k) \tag{3.1a}$$

$$y_r(k) = C_r \hat{x}_r(k) + v(k), \tag{3.1b}$$

where $v(k)$ and $w(k)$ are assumed to be zero mean white noise processes with covariance matrices $R_k = R_k^T \succ 0$ and $Q_k = Q_k^T \succ 0$, respectively, and where Γ defines the mapping between w and the different states. In this setup, the noise processes are expected to account for uncertainty in the state equations through $\Gamma w(k)$, and the uncertainty in the output through $v(k)$. The closed-loop system with Kalman filter and explicit model predictive controller takes the general form of Figure 2.4.

A number of questions regarding robust stability, feasibility and robust constraint fulfillment arises when the reduced model is used to control the high-order model. Since the explicit MPC solution is equivalent to the standard MPC solution, many methods for robust stability analysis developed for standard MPC (see e.g. Bemporad and Morari, 1999) can be used to conclude stability for the reduced-order eMPC in the presence of the uncertainty introduced through the model reduction process. Some recent results on MPC stability in the presence of model uncertainty have been developed (Heath et al., 2005b, Heath and Wills, 2005, Heath et al., 2005a). Also, tests for robust MPC stability of input-constrained systems with unstructured uncertainty have recently been established by Løvaas et al. (2007b).

In Chapter 4 we develop criteria for guaranteeing stability of MPC based on reduced-order models. In this chapter, however, we use the nominal model (the reduced model) for controller design, and address certain robustness issues during the design stage. While we do not explicitly analyze the

robustness of the reduced model predictive controller in this chapter, good
performance is achieved by ad hoc tuning based on exhaustive simulations
for ranges of operating conditions. In many cases this approach leads to bet-
ter performance than using robust MPC techniques (Bemporad and Morari,
1999). Choosing the right robust MPC technique is an art, and much ex-
perience is necessary to make it work.

Given the uncertainty introduced through the model reduction process,
one cannot guarantee that feasibility of the underlying optimization problem
is maintained and that the constraints on the states/outputs are fulfilled.
This problem can be handled through the use of soft constraints. Con-
straints on the states/outputs often represent desirable operational limits
rather than fundamental operational constraints. In addition, from a prac-
tical point of view it does not make sense to use tight state constraints
because of the presence of noise, disturbances and numerical errors. Relax-
ing the state constraints in effect removes the feasibility problem, at least
for stable systems (Bemporad and Morari, 1999). *Exact* penalty functions
can be used to allow constraint violation only when absolutely necessary
(Kerrigan and Maciejowski, 2000b).

3.3   Case Studies

The proposed control structure will be demonstrated using 6 different ran-
dom systems to illustrate the potential for complexity reduction, and two
specific examples to show performance when using reduced-order eMPC.

By implementing the piecewise affine function as a binary search tree,
the online computational time is logarithmic in the number of polyhedra
in the state space partition (Tøndel et al., 2003). The online memory and
processing requirements increase with the number of regions in the partition.
This number is therefore used in the following as a measure of complexity
of the explicit model predictive controller, and a reduction in the number
of regions is considered to be a reduction in controller complexity.

3.3.1   Example 1

Without considering approximation quality and closed-loop performance, 6 different random systems of order $n = 6$, with two inputs and two outputs have been considered. For all six systems, the inputs and outputs are constrained such that

$$|u_i| \leq 1, \ i = 1, 2 \tag{3.2}$$
$$|y_i| \leq 1, \ i = 1, 2 \tag{3.3}$$

and the control horizon is fixed at $N_u = 4$. The resulting controller complexity is tabulated in Table 3.1. The table shows that eMPC for the original system is very demanding, with $O\left(10^5\right)$ polyhedra in the state space partition. But by truncating only one state, the controller complexity is reduced to a manageable level, as the number of regions is reduced by two orders of magnitude.

System/r	3	4	5	6
1	603	1447	1487	117573
2	625	1549	1589	122675
3	519	1095	1145	109656
4	539	1125	1136	95896
5	537	1033	1755	116438
6	513	1461	2145	109711

Table 3.1: Example 1: Controller complexity (in terms of number of regions in the state space) for 6 random systems with two inputs and two outputs, with upper and lower bounds on inputs and outputs.

3.3.2   Example 2

For a random stable LTI system of order $n = 15$, the input is constrained such that $|u| \leq 5$ and the output is constrained such that $|y| \leq 1$. Figure 3.1 compares the complexity of the eMPC solution for different model orders

r and different control horizons N_u for this example. For all r and N_u, we set $Q = 10^3 \cdot C_r^T C_r$ and $R = 10^{-3}$. The figure illustrates that the controller
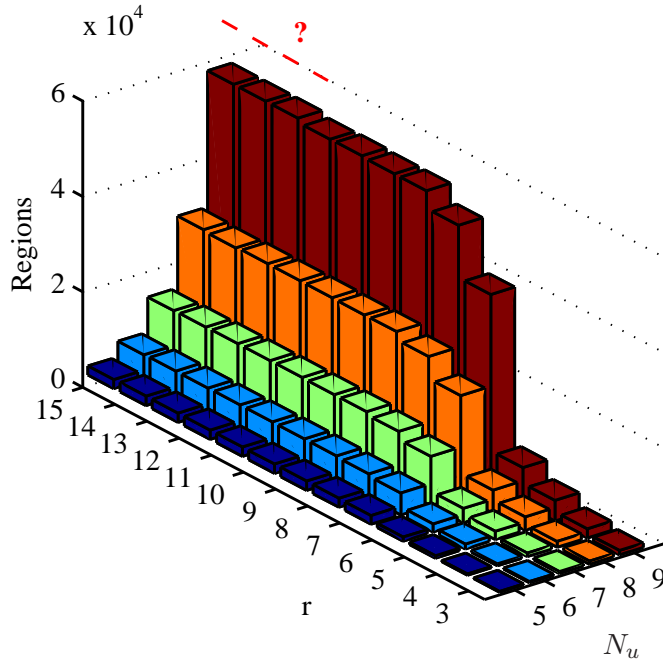
Figure 3.1: Example 2: Complexity in terms of number of regions in the eMPC solution, for different model orders r and different control horizons N_u. For $r = 13$, 14 and 15, no solutions have been found with control horizon $N_u = 9$, indicated by the dotted line and the question mark. The system order should be reduced to $r = 7$ or even $r = 6$ to obtain a significant reduction in complexity.

complexity increases by over an order of magnitude as we include more states in the reduced model and increase the control horizon N_u. For $r = 3$, the number of regions ranges from 155 for $N_u = 5$ to 1287 for $N_u = 10$. For the original 15th order model, we are unable to compute the state space

r	Error bound
3	1.4×10^{-1}
4	7.4×10^{-2}
5	3.3×10^{-2}
6	6.7×10^{-3}
7	3.1×10^{-3}
8	1.5×10^{-4}
9	2.0×10^{-6}
10	3.5×10^{-7}
11	2.7×10^{-8}
12	4.5×10^{-10}
13	5.5×10^{-14}
14	4.3×10^{-17}

Table 3.2: Bound on model reduction error for Example 2.

partition for $N_u > 8$, due to the formidable computational requirement. The state space partition comprises 27442 regions for $N_u = 8$. For $r = 12$, the number of regions in the state space partition is 55139 for $N_u = 9$.

The model reduction error bound (2.9) is shown in Table 3.2, and illustrates the trade-off that must be made between controller complexity and quality of the reduced model, and consequently the quality of the resulting controller.

From Figure 3.1 it can be seen that by reducing the number of states down to 6, the controller complexity remains relatively low for the control horizons considered. We therefore generate our explicit model predictive controller using 6 states in the reduced model. For $r = 6$, the error bound is $\|\mathcal{G}(s) - \mathcal{G}_r(s)\|_\infty \leq 6.7 \times 10^{-3}$. Still, the eMPC controller based on the 6th order reduced model is sufficient for control, as illustrated in Figure 3.2, where it can be seen that both the input and the output are kept within their bounds, when the plant is initialized with a representative non-zero state vector. The horizon length is $N_u = 9$ and the explicit MPC solution based on the reduced-order model consists of 7625 polyhedra. The figure

shows the performance with eMPC based on the full-order model, with a control horizon $N_u = 8$, for which the controller consists of 27442 regions.

Although the error bound merely establishes a bound on the error between the two transfer functions in open loop, it does not guarantee performance, degree of sub-optimality and constraint satisfaction for the closed loop system. It is nevertheless an indication that a great reduction in complexity might be achieved without compromising the performance.
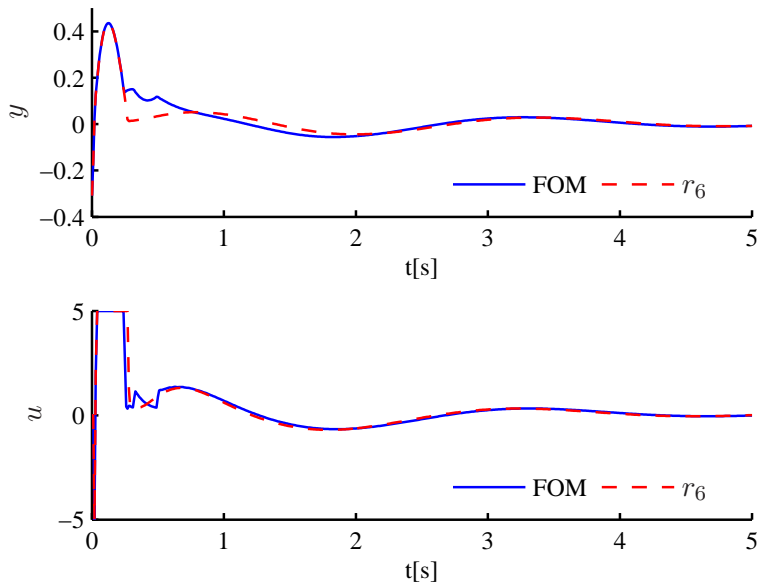
Figure 3.2: Top: Output y of Example 2 with eMPC based on full order model (FOM) with $N_u = 8$ and reduced-order model with $r = 6$ and $N_u = 9$. The output is constrained between ± 1. Bottom: Control input, constrained between ± 5.

3.3.3   Example 3

This example is a scaled, linearized model for control of fuel cell breathing, as described in Pukrushpan et al. (2004). The model is a stable LTI system with one input (compressor voltage), two performance variables z (system net power and oxygen excess ratio) and 8 states. Focusing on the methodology presented above, we use a slightly simplified version of the model in Pukrushpan et al. (2004). In our simplified model, we ignore disturbances (stack current), and assume that the performance variables z are measured, which amount to setting the output $y = z$. We discretize the model with sampling time $T_s = 1\,\mathrm{ms}$, and derive reduced-order models with $r = 3$ to $r = 7$ states. For these reduced models, we solve the eMPC offline problem for eMPC horizons 1-5, with bounds on the input and outputs:

$$|u| \le 5, \quad |y_1| \le 0.03, \quad |y_2| \le 0.2. \tag{3.4}$$

We set the weight matrices to be $Q = 1000 \times C_r^T C_r$ and $R = 1$. The complexity of different eMPC controllers for this example is shown in Table 3.3, while the model reduction error bound (2.9) is shown in Table 3.4. It can be seen from Table 3.3 that the complexity of the controller increases rapidly for the original model ($r = 8$), while the increase is less pronounced for $r = 3$ and $r = 4$. The tables also show that by truncating 4 states, the controller complexity is reduced by an order of magnitude for $N_u = 5$, at the cost of introducing an approximation error $\|\mathcal{G}(s) - \mathcal{G}_r(s)\|_\infty \le 1.3 \times 10^{-4}$. If we reduce the number of states down to $r = 3$, the number of regions in the state space partition is reduced by over two orders of magnitude compared to the original model, for $N_u = 5$. By truncating only one state, the number of regions is reduced by 34% for $N_u = 5$.

The simulation in Figure 3.3 shows the difference in closed loop behavior when using the full-order model with 8 states, and reduced-order models with 3 and 7 states.

In this simulation, the eMPC horizon is $N_u = N = 5$, which gives 105 regions in the controller for $r = 3$, 9964 regions for $r = 7$ and 14999 regions for the full-order model with 8 states. Moreover, it can be seen that both

r/N_u	1	2	3	4	5
3	7	19	41	69	105
4	7	51	237	740	1813
5	7	55	333	1472	5020
6	7	55	331	1575	6068
7	7	57	393	2186	9964
8	7	61	445	2695	14999

Table 3.3: Controller complexity for Example 3. $r = 8$ corresponds to no model truncation ($r = n$).

r	Error bound
3	1.6×10^{-3}
4	1.3×10^{-4}
5	4.9×10^{-5}
6	4.4×10^{-6}
7	2.6×10^{-7}

Table 3.4: Bound on model reduction error for Example 3.

outputs remain within their bounds. The sub-optimality of the reduced-order controllers is clearly illustrated in the plot.

3.4 Concluding Remarks

It has been demonstrated that the performance of eMPC based on reduced-order models is of comparable quality to that of eMPC based on the original systems. It is possible to use longer control horizons, while at the same time keeping the controller complexity low, at the cost of some controller sub-optimality. The degree of complexity reduction depends on the application, but is shown to be significant in all our examples. For input-constrained and soft-constrained systems, the approach is especially attractive, since the requirements to satisfy the output constraints need not be met. However,
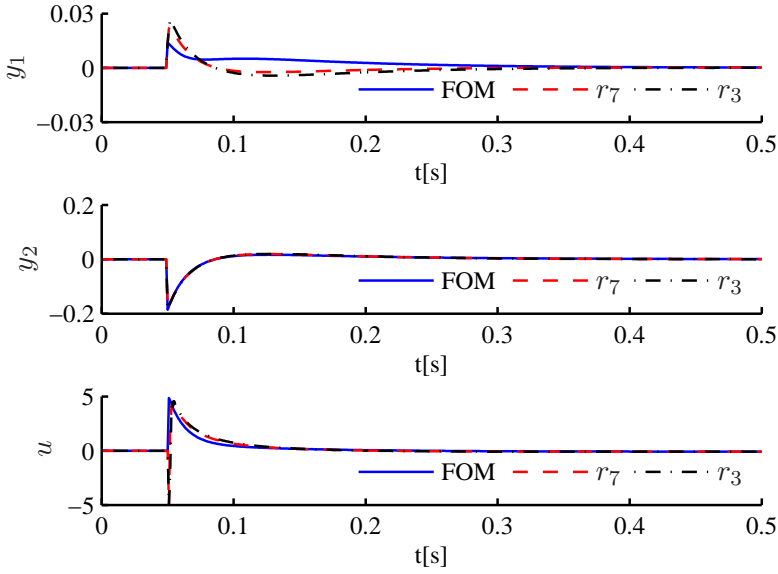
Figure 3.3: Example 3: Closed-loop response to a disturbance at $t = 0.05\,\text{s}$. The figure compares the performance for the full-order model (FOM), and reduced models with $r = 3$ (r_3) and $r = 7$ (r_7), all with $N_u = 5$.

further work should focus on developing guarantees for satisfaction of output constraints.

Chapter 4

Stability of MPC Based on Reduced-Order Models

I<small>N</small> this chapter, we present a novel, systematic procedure for obtaining closed-loop stable output-feedback model predictive control based on reduced-order models. The design uses linear state estimators, and applies to open-loop stable systems with hard input- and soft state constraints. Robustness against the model reduction error is obtained by choosing the cost function parameters so as to satisfy a linear matrix inequality condition. We also show by means of an example, that performance is maintained even when the model reduction error is relatively large. This chapter is based on Hovland et al. (2008a,b).

4.1 Introduction

The use of model reduction techniques along with MPC is desirable in many applications, in order to reduce the online complexity in implementations that would otherwise run too slowly. In Section 3 we demonstrated how a significant reduction in complexity could be achieved by truncating only a few number of states, in particular when the MPC horizons are large. The online complexity reduction came at the cost of introducing an ap-

proximation error in the closed-loop system. With the introduction of the approximation error, questions concerning closed-loop stability and feasibility arise. These are very important issues to address, since controllers designed based on reduced-order models might stabilize the reduced-order model and not the plant (Linnemann, 1988).

The results in this chapter are based on the previous work Løvaas et al. (2007a, 2008a,b) on robust output-feedback MPC for systems with uncertainties. Here, we specialize these results to the case of reduced-order models. We ensure stability by choosing the cost function parameters so as to satisfy a linear matrix inequality (LMI) condition, which guarantees the existence of a suitable Lyapunov function. To the best of our knowledge, this is the first result that deals systematically with the model reduction error in model predictive control. The results make MPC more attractive for a number of systems that would otherwise be excluded due to the high complexity of the associated full-order controllers.

In order to guarantee feasibility of the MPC problem, we adopt the soft constraints formulation of Løvaas et al. (2008b), in which an additional horizon is introduced into the well-known approach of Scokaert and Rawlings (1999) to reduce the number of slack variables. Consequently, the size of the optimization problem we consider may be significantly smaller than what can be achieved using the approach of Scokaert and Rawlings (1999) in combination with the reduced-order model. This extra feature fits nicely into our design, since our goal is to to make our MPC procedure more efficient by introducing reduced-order models.

The traditional MPC strategy requires significant online computation, limiting the use of this kind of controller to processes with small system state dimension or relatively slow dynamics, since the optimization problem that is solved at each sampling time can otherwise become too large. Remedies such as "input blocking", short horizons etc. are commonly used to reduce the complexity and online computational times. Fast implementation of model predictive control in real-time systems has been considered, among others, by Bleris and Kothare (2005) and Pannocchia et al. (2007). Also, it was proposed in Bemporad et al. (2002) to solve multiparametric quadratic programs (mpQPs) that can be used to obtain explicit solutions to the MPC

problem, such that the control input can be efficiently computed by evaluating a piecewise affine function of the system state. Nevertheless, even a moderate increase in the model dimension may lead to a prohibitive increase in the complexity of both the offline- and the online problem. Hence, to address the complexity issue, we consider the use reduced-order models in this work.

The chapter outline is as follows: In Section 4.2 we describe the system formulations that we will consider. The nominal state-feedback design presented in Section 4.3 lays the foundation for the reduced-order MPC described in Section 4.4, of which we prove stability in Section 4.5. In Section 4.6 we propose a procedure for synthesis of a robust MPC design, and we demonstrate performance through a numerical example in Section 4.7. Concluding remarks can be found in Section 4.8.

Throughout we use the following notation: $\|x\|_P^2$ denotes $x^T P x$, $[a, \cdots, c]$ denotes $\begin{bmatrix} a^T & \cdots & c^T \end{bmatrix}^T$ and I_n denotes the $n \times n$ identity matrix.

4.2 System Description

We consider a stable, linear, discrete-time plant, described by the known model

$$x_{k+1} = A x_k + B u_k \tag{4.1a}$$

$$y_k = C x_k, \tag{4.1b}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^p$ denote the state, input and output, respectively, and the matrices A, B and C are of appropriate dimensions. It has not been considered whether the following theory can be extended to descriptor models of the form (2.1). For descriptor models with non-singular mass matrix E, one can of course apply the theory by inverting E and multiplying throughout the state equation. The system is subject to the following constraints

$$V u_k \leq v, \ \forall k \geq 0 \tag{4.2a}$$

$$H x_k \leq h, \ \forall k \geq 0, \tag{4.2b}$$

where $V \in \mathbb{R}^{n_v \times m}$, $v \geq 0$, and $H \in \mathbb{R}^{n_h \times n}$.

The input constraints (4.2a) are hard constraints, that must be respected at all time, whereas the state constraints (4.2b) are soft constraints, and will be treated by penalizing constraint violation in the MPC cost function. This is a natural assumption, since input constraints, such as actuator- and valve limitations are physical limitations that cannot be exceeded. State- and output constraints, on the other hand, often represent desirable, rather than absolute limitations.

4.2.1   Reduced-Order Nominal Model

The plant model (4.1) is assumed to be of such a dimension that the online computational requirements conflict with the time available to compute the control input. For the purpose of MPC design, we therefore generate a reduced-order model (ROM), by reducing the order of (4.1) using an appropriate model reduction technique, such as any of the methods presented in the previous chapters.

The nominal model obtained by model reduction is denoted by

$$x_{r_{k+1}} = A_r x_{r_k} + B_r u_k \qquad (4.3a)$$
$$y_{r_k} = C_r x_{r_k}, \qquad (4.3b)$$

where $x_r \in \mathbb{R}^r$ such that $r < n$, $y_r \in \mathbb{R}^p$, $A_r \in \mathbb{R}^{r \times r}$, $B_r \in \mathbb{R}^{r \times m}$ and $C_r \in \mathbb{R}^{p \times r}$. The nominal model must respect the constraints (4.2). To enable this, we make the following assumption:

Assumption 2. *It is assumed that the constraints* (4.2b) *apply to the outputs of* (4.1)*, and consequently apply naturally to the outputs of* (4.3)*. This can easily be achieved by choosing any states that should be constrained as outputs of the plant.*

Remark 4. *Associated with the reduced-order model is an approximation error that can be quantified in general terms as follows: When substituting* (4.3) *for* (4.1)*, the* minimum *achievable Hankel norm of the error system is equal to the* $(r + 1)$*-st Hankel singular value of the original system* (4.1)

(Adamjan et al., 1971, Glover, 1984, Gu, 2005). This error needs to be accounted for in the controller design.

4.3   Nominal Case with State Feedback

In this section we disregard the approximation error and present the soft-constrained state-feedback MPC policy proposed in Løvaas et al. (2008b) for the nominal system (4.3), when disregarding the approximation error. The state-feedback policy will subsequently be used in Section 4.4 to develop an exponentially stable robust output-feedback policy for the system (4.1) based on the reduced-order model (4.3).

The following optimization problem leads to an MPC scheme with guaranteed nominal stability:

$$\left[P^{N,N_\varepsilon} \right] : J^* \left(x_r \right) = \min_{U, \varepsilon, e} J \left(x_r, U, \varepsilon, e \right)$$

$$\text{s.t.} \begin{cases} x_{r_0} & = x_r \\ x_{r_{i+1}} & = A_r x_{r_i} + B_r u_i \\ V u_i & \leq v, \ \forall i \in \{0, \cdots, N_u - 1\} \\ u_i & = 0, \ \forall i \geq N_u \\ H x_{r_i} & \leq h + \epsilon_i, \ \forall i \in \{0, \cdots, N_\epsilon - 1\} \\ H x_{r_i} & \leq h + H A_r^{i - N_\epsilon} e, \ \forall \in i \{N_\epsilon, \cdots, N - 1\} \\ T x_{r_N} & \leq t + T A_r^{N - N_\epsilon} e, \end{cases} \tag{4.4}$$

Here,

$$U = \begin{bmatrix} u_0 \\ \vdots \\ u_{N_u - 1} \end{bmatrix}$$

and

$$\varepsilon = \begin{bmatrix} \epsilon_0 \\ \vdots \\ \epsilon_{N_\epsilon - 1} \end{bmatrix}$$

are the sequences of N_u inputs and N_ϵ slack variables to be optimized over the horizons N_u and N_ϵ, and $e \in \mathbb{R}^r$ is an additional vector of slack variables

that has been introduced to summarize constraint violation beyond the prediction time $i = N_\epsilon - 1$. N is the prediction horizon. Further,

$$
J\left(x_r, U, \varepsilon, e\right) \triangleq \begin{bmatrix} x_r \\ U \\ \varepsilon \\ e \end{bmatrix}^T P \begin{bmatrix} x_r \\ U \\ \varepsilon \\ e \end{bmatrix} \tag{4.5}
$$

is the cost function, for some appropriate matrix P whose selection will be explained below, and the matrix T and the vector t describe a "terminal constraint set". T and t can e.g. be chosen so that the terminal constraint set equals the maximal output admissible set associated with the state constraints (4.2b) (see e.g. Gilbert and Tan, 1991). We let U^*, ε^* and e^* denote the optimal values of U, ε and e, resulting from $\left[P^{N,N_\varepsilon}\right]$. We let the set

$$
\mathbb{S} \triangleq \left\{ \begin{bmatrix} x_r & U & \varepsilon & e \end{bmatrix}^T \mid \begin{bmatrix} x_r & U & \varepsilon & e \end{bmatrix}^T \text{satisfy} \left[P^{N,N_\varepsilon}\right] \right\}, \tag{4.6}
$$

such that we can write the constraints in $\left[P^{N,N_\varepsilon}\right]$ as

$$
\begin{bmatrix} x_r \\ U \\ \varepsilon \\ e \end{bmatrix} \in \mathbb{S}. \tag{4.7}
$$

Remark 5. *Note that by choosing the parameters in $\left[P^{N,N_\varepsilon}\right]$ in an appropriate way (see Løvaas et al., 2008b), our formulation is equivalent to the standard soft-constrained MPC in Scokaert and Rawlings (1999). Some special features of our particular formulation is however crucial in our quest for robustly stable MPC based on reduced-order models.*

To help describe various conditions on $\left[P^{N,N_\varepsilon}\right]$ and on the cost function matrix P, consider the following autonomous prediction system:

$$
\begin{bmatrix} x_{r_{n+1}} \\ U_{n+1} \\ \varepsilon_{n+1} \\ e_{n+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_r & [B_r\ 0\ \cdots\ 0] & 0 & 0 \\ 0 & \Gamma\left(N_u, n_u\right) & 0 & 0 \\ 0 & 0 & \Gamma\left(N_\epsilon, n_h\right) & \bar{H} \\ 0 & 0 & 0 & A_r \end{bmatrix}}_{\bar{A}_0} \begin{bmatrix} x_{r_n} \\ U_n \\ \varepsilon_n \\ e_n \end{bmatrix}, \tag{4.8}
$$

where

$$\bar{H} = \begin{bmatrix} 0 \\ \vdots \\ H \end{bmatrix},$$

and where $\Gamma\left(\bar{N}, \bar{n}\right)$ is a matrix such that, using

$$\bar{U} = \begin{bmatrix} \bar{u}_0 \\ \vdots \\ \bar{u}_{\bar{N}-1} \end{bmatrix},$$

we have

$$\Gamma\left(\bar{N}, \bar{n}\right)\bar{U} = \begin{bmatrix} \bar{u}_1 \\ \vdots \\ \bar{u}_{\bar{N}-1} \\ 0 \end{bmatrix},$$

that is

$$\Gamma\left(\bar{N}, \bar{n}\right) = \begin{bmatrix} 0 & I_{\bar{n}} & 0 & \cdots & 0 \\ \vdots & 0 & I_{\bar{n}} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \vdots & 0 & \cdots & 0 & I_{\bar{n}} \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{\bar{N}\bar{n} \times \bar{N}\bar{n}}. \tag{4.9}$$

Remark 6. *Note that if $N_\epsilon = N$ and P satisfies*

$$\bar{A}_0^T P \bar{A}_0 - P + \bar{C}_0^T \, diag\,[Q, R, S]\,\bar{C}_0 = 0, \tag{4.10}$$

where $\bar{A}_0$ is defined in (4.8), $Q \in \mathbb{R}^{r \times r}$, $Q \geq 0$, $R \in \mathbb{R}^{m \times m}$, $R > 0$, $S \in \mathbb{R}^{n_h \times n_h}$, $S > 0$, and where the matrix $\bar{C}_0$ is such that

$$\bar{C}_0 \begin{bmatrix} x_r \\ U \\ \varepsilon \\ e \end{bmatrix} = \begin{bmatrix} x_r \\ u_0 \\ \epsilon_0 \end{bmatrix},$$

then the cost function (4.5) satisfies

$$J\left(x_r, U, \varepsilon, e\right) = \left\|x_{r_{N_u}}\right\|_{P_F}^2 + \sum_{i=0}^{N_u-1} \left(\|x_{r_i}\|_Q^2 + \|u_i\|_R^2\right)$$

$$+ \|e\|_\Pi^2 + \sum_{i=0}^{N-1} \|\epsilon_i\|_S^2, \tag{4.11}$$

where

$$A_r^T P_F A_r - P_F = -Q$$

and

$$A_r^T \Pi A_r - \Pi = -H^T S H,$$

and where x_{r_i} is given by $\left[P^{N,N_\varepsilon}\right]$ (Løvaas et al., 2008b).

Remark 7. *Note that the set $\mathbb{S}$ is invariant for the system (4.8), namely*

$$\bar{A}_0 \begin{bmatrix} x_r & U & \varepsilon & e \end{bmatrix}^T \in \mathbb{S}, \quad \forall \begin{bmatrix} x_r & U & \varepsilon & e \end{bmatrix}^T \in \mathbb{S}. \tag{4.12}$$

The state-feedback MPC design proposed in Løvaas et al. (2008b) is based on $\left[P^{N,N_\epsilon}\right]$ as follows:

Algorithm 2. *Nominal State-Feedback MPC*
Offline:

1. *Choose any integers N, N_u and N_ϵ satisfying $N \geq N_u \geq 1$, $N \geq N_\epsilon \geq 1$.*

2. *Choose any matrices $Q \geq 0$, $R > 0$ and $S > 0$.*

3. *Choose P that satisfies (4.10).*

4. *Choose any T and t such that the set $X_F \triangleq \{x_r | T x_r \leq t\}$ satisfies*

$$A_r x_r \in X_F, \forall x_r \in X_F, \qquad X_F \subseteq \{x_r | H x_r \leq h\}. \tag{4.13}$$

Online: *At each time step $k \geq 0$, solve $\left[P^{N, N_\varepsilon} \right]$, using $x_r = x_{r_k}$, then apply $u_k = \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} U^* (x_r)$ to (4.3).*

Remark 8. *Note that $\left[P^{N, N_\varepsilon} \right]$ is always feasible, since a particular feasible solution is given by*

$$
\begin{bmatrix} x_r \\ U \\ \varepsilon \\ e \end{bmatrix} = K_F x_r,
$$

where

$$
K_F = \begin{bmatrix} 0 \\ H \\ H A_r \\ \vdots \\ H A_r^{N_\epsilon - 1} \\ A_r^{N_\epsilon} \end{bmatrix}. \tag{4.14}
$$

The following theorem establishes closed-loop stability when applying Algorithm 2 to the nominal system (4.3), disregarding the plant (4.1) altogether.

Theorem 4. *The closed-loop system under Algorithm 2 is globally exponentially stable. Moreover, the closed-loop trajectories satisfy*

$$
\sum_{k=0}^{\infty} \|x_{r_k}\|_Q^2 + \|u_k\|_R^2 + \|\epsilon_k^*\|_S^2 \leq J^* (x_{r_0}), \tag{4.15}
$$

where ϵ_k^* denotes the first block component of $\varepsilon^* (x_{r_k})$.

Proof. This is theorem 3 in Løvaas et al. (2008b), where the proof can be found. ∎

We have now established stability of the MPC design of Algorithm 2, when applied to (4.3) only. In other words, we have shown that the closed-loop system consisting of only the reduced-order model and the model pre-

dictive controller is stable. In the next section, we take the model approximation errors into account.

4.4   Reduced-Order MPC with Output Feedback

In this section, we propose an output-feedback MPC procedure based on the reduced-order model (4.3), in which we take into account the error introduced through the model reduction process. We also prove closed-loop stability when applying this controller to the plant (4.1).

The MPC control input is computed based on the reduced-order state vector x_k at each time step, and x_k should therefore be estimated by an observer, using measurements y_k from the plant. We consider a linear estimator of the form

$$\hat{x}_{r_{k+1}} = A_r \hat{x}_{r_k} + B_r u_k + L \left(y_k - C_r \hat{x}_{r_k} \right), \qquad (4.16)$$

where $\hat{x}_{r_k}$ denotes the estimated reduced state at time step k, and we choose L such that $(A_r - LC_r)$ is Schur (i.e. the eigenvalues lie strictly inside the unit disc). (It seems possible to allow for other observer structures, however this possibility has not been explored in detail.)

When uncertainties are taken into account, we will make use of the following matrix function:

$$\Sigma_{\{Q,R,S\}}(P) \triangleq \bar{A}_0^T P \bar{A}_0 - P + \bar{C}_0^T \text{diag}\left[Q, R, S\right] \bar{C}_0. \qquad (4.17)$$

The nominal cost function matrix, denoted by P_0, is retrieved by solving the "Lyapunov equality" $\Sigma_{\{Q,R,S\}}(P) = 0$, that is, we have

$$\Sigma_{\{Q,R,S\}}(P_0) = 0. \qquad (4.18)$$

Requiring $\Sigma_{\{Q,R,S\}}(P) \leq 0$ implies $P \geq P_0$. We will use the constraint $\Sigma_{\{Q,R,S\}}(P) \leq 0$ at a later stage to search for a P that gives a cost function for the robust case that is an upper bound on the nominal cost.

The proposed output-feedback policy for the system, considering the uncertainties, can now be described as follows:

Algorithm 3. *Output-Feedback MPC with Reduced Model*
Offline:

1. *Generate a reduced-order model* (4.3).

2. *Design a state estimator* (4.16) *based on the reduced-order model.*

3. *Choose any integers N, N_u and N_ϵ satisfying $N \geq N_u \geq 1$, $N \geq N_\epsilon \geq 1$.*

4. *Choose any matrices $Q \geq 0$, $R > 0$ and $S > 0$.*

5. *Choose any matrix P satisfying $\Sigma_{\{Q,R,S\}}(P) \leq 0$.*

6. *Choose any T and t such that the set $X_F = \{x_r | Tx_r \leq t\}$ satisfies* (4.13).

Online: *At each time step $k \geq 0$, solve $\left[P^{N,N_\varepsilon}\right]$ using $x_r = \hat{x}_{r_k}$, then apply $u_k = \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} U^*(\hat{x}_{r_k})$ to* (4.1).

Remark 9. *Note that we can always find P such that*

$$\Sigma_{\{Q,R,S\}}(P) \leq 0.$$

This follows trivially from stability of (4.8), *and by recognizing that*

$$\bar{A}_0^T P \bar{A}_0 - P + \bar{C}_0^T \, diag\,[Q, R, S]\, \bar{C}_0$$

is nothing more than a particular discrete-time Lyapunov equation for system (4.8). *Hence, since*

$$\bar{C}_0^T \, diag\,[Q, R, S]\, \bar{C}_0 \geq 0,$$

there always exists a P such that $\Sigma_{\{Q,R,S\}}(P) = 0$.

In the following section, we will prove stability of the proposed output-feedback policy.

4.5   Robust Stability Test

Now, we propose an LMI condition on the cost function matrix P which is sufficient for closed-loop stability. To this end, we define the augmented state

$$\bar{x} \triangleq [x, \hat{x}_r], \tag{4.19}$$

where x is the plant state and $\hat{x}_r$ is the estimated ROM state. The dynamics of $\bar{x}$ in closed-loop are described by

$$\bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{B}\mu_k, \qquad \bar{x}_0 = [x_0, \hat{x}_{r_0}] \tag{4.20}$$

$$\hat{x}_{r_k} = \bar{C}\bar{x}_{r_k}, \tag{4.21}$$

where

$$\bar{A} = \begin{bmatrix} A & 0 \\ LC & A_r - LC_r \end{bmatrix}, \tag{4.22}$$

$$\bar{B} = \begin{bmatrix} BD_1 \\ B_r D_1 \end{bmatrix}, \tag{4.23}$$

$$\bar{C} = \begin{bmatrix} 0 & I \end{bmatrix}, \tag{4.24}$$

and

$$D_1 = \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix}$$

is such that

$$u_k = D_1 \mu_k,$$

where

$$\mu_k = \begin{bmatrix} U_k^* \\ \varepsilon_k^* \\ e_k^* \end{bmatrix} \tag{4.25}$$

contains the minimizers of $\left[P^{N,N_\varepsilon}\right]$ at time step k. The matrix L is the gain of the state estimator (4.16).

For the purpose of stability analysis, we need to establish a feasible solution μ_{k+1}^F to $\left[P^{N,N_\varepsilon}\right]$ at time step $k+1$, based on the optimal solution μ_k at the previous time step k. The following lemma establishes the existence such a solution.

Lemma 1. *Let $\bar{A}$ and $\bar{B}$ be defined as in (4.22) and (4.23). Then*

$$F_1 = K_F \begin{bmatrix} LC & -LC_r \end{bmatrix} \tag{4.26}$$

and

$$F_2 = \begin{bmatrix} \Gamma\left(N_u, n_u\right) & 0 & 0 \\ 0 & \Gamma\left(N_\epsilon, n_h\right) & \bar{H} \\ 0 & 0 & A_r \end{bmatrix}, \tag{4.27}$$

are such that

$$\mu_{k+1}^F = F_1 \bar{x}_k + F_2 \mu_k \tag{4.28}$$

is a feasible solution to $\left[P^{N,N_\varepsilon}\right]$ at time step $k+1$, where, K_F is as in (4.14).

Proof. The closed-loop dynamics are given by (4.20) and (4.28), which we can write

$$\begin{bmatrix} \bar{x}_{k+1} \\ \mu_{k+1} \end{bmatrix} = \begin{bmatrix} \bar{A} & \bar{B} \\ F_1 & F_2 \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ \mu_k \end{bmatrix}. \tag{4.29}$$

We need to verify that

$$\begin{bmatrix} \bar{A} & \bar{B} \\ F_1 & F_2 \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ \mu_k \end{bmatrix} \in \mathbb{R}^n \times \mathbb{S}, \quad \forall \begin{bmatrix} \bar{x}_k \\ \mu_k \end{bmatrix} \in \mathbb{R}^n \times \mathbb{S}, \tag{4.30}$$

where $\mathbb{S}$ is as in (4.6). Expanding (4.29) allows us to write

$$\begin{bmatrix} x_{k+1} \\ \hat{x}_{r_{k+1}} \\ \mu_{k+1} \end{bmatrix} = \begin{bmatrix} Ax_k + BD_1\mu_k \\ A_r\hat{x}_{r_k} + BD_1\mu_k + \begin{bmatrix} LC & -LC_r \end{bmatrix} \bar{x}_k \\ K_F \begin{bmatrix} LC & -LC_r \end{bmatrix} \bar{x}_k + F_2\mu_k \end{bmatrix}. \tag{4.31a}$$

Now, it is straightforward to find a matrix G such that the set $\mathbb{S}$ in (4.6) can be written as

$$\mathbb{S} = \left\{ \begin{bmatrix} x_r & \mu \end{bmatrix}^T | G\mu - GK_F x_r \le g \right\}, \tag{4.32}$$

where $g \triangleq \begin{bmatrix} v & v & \dots & h & \dots & h & t \end{bmatrix}^T \ge 0$. To verify (4.30) we therefore need to show that

$$G\mu_{k+1} - GK_F\hat{x}_{r_{k+1}} \tag{4.33a}$$

$$= GF_2\mu_k - GK_F\left(A_r\hat{x}_{r_k} + BD_1\mu_k\right) \le g. \tag{4.33b}$$

Now, to see that inequality (4.33b) indeed holds, we note from (4.8) that

$$
\begin{bmatrix} A_r \hat{x}_{r_k} + B D_1 \mu_k \\ F_2 \mu_k \end{bmatrix} = \bar{A}_0 \begin{bmatrix} \hat{x}_{r_k} \\ \mu_k \end{bmatrix}. \tag{4.34}
$$

Consequently, the result follows from (4.12). ∎

As the final step towards our stability result, we need to find a suitable cost function matrix P. To this end we introduce the following definitions:

$$
\Omega\left(\Omega_0, P\right) \triangleq \begin{bmatrix} \Omega_0 & 0 \\ 0 & 0 \end{bmatrix} + D_P^T P D_p, \tag{4.35}
$$

with

$$
D_P = \begin{bmatrix} \bar{C} & 0 \\ 0 & I_{n_\mu} \end{bmatrix}, \tag{4.36}
$$

and $\Omega_0 \in \mathbb{R}^{(n+r) \times (n+r)}$.

$$
\Phi\left(\Omega_0, P\right) \triangleq \begin{bmatrix} \bar{A} & \bar{B} \\ F_1 & F_2 \end{bmatrix} \Omega\left(\Omega_0, P\right) \begin{bmatrix} \bar{A} & \bar{B} \\ F_1 & F_2 \end{bmatrix} - \Omega\left(\Omega_0, P\right). \tag{4.37}
$$

The stability test for Algorithm 3 can now be stated as follows.

Theorem 5. *Assume that, for a given P, there exists a matrix $\Omega_0 \in \mathbb{R}^{(n+r) \times (n+r)}$ such that,*

$$
\Omega\left(\Omega_0, P\right) > 0 \tag{4.38a}
$$
$$
\Phi\left(\Omega_0, P\right) < 0, \tag{4.38b}
$$

where $\Omega\left(\Omega_0, P\right)$ is as defined in (4.35) and $\Phi\left(\Omega_0, P\right)$ is as defined in (4.37). Then the closed-loop system under Algorithm 3 is exponentially stable.

Proof. Proving stability follows the well-known path (Mayne et al., 2000) of first showing recursive feasibility, and then showing that there exists a Lyapunov function for the closed-loop system that decreases at each time

step. Feasibility at each time step has been established in Lemma 1. Now, consider the Lyapunov function candidate

$$V\left(\bar{x}, \mu\right) \triangleq \left\|\begin{bmatrix} \bar{x} \\ \mu \end{bmatrix}\right\|^2_{\Omega(\Omega_0, P)}, \tag{4.39}$$

which is positive definite in view of (4.38a), and where μ denotes the minimizers of $\left[P^{N,N_\varepsilon}\right]$, as in (4.25). At time step k, we have

$$V_k^* \triangleq V\left(\bar{x}_k, \mu_k\right) = \left\|\begin{bmatrix} \bar{x}_k \\ \mu_k \end{bmatrix}\right\|^2_{\Omega(\Omega_0, P)} \tag{4.40}$$

$$= \|\bar{x}_k\|^2_{\Omega_0} + \left\|\begin{bmatrix} \bar{C}\bar{x}_k \\ \mu_k \end{bmatrix}\right\|^2_P \tag{4.41}$$

$$= \|\bar{x}_k\|^2_{\Omega_0} + \left\|\begin{bmatrix} \hat{x}_{r_k} \\ \mu_k \end{bmatrix}\right\|^2_P \tag{4.42}$$

$$= \|[\bar{x}_k]\|^2_{\Omega_0} + J_k^*, \tag{4.43}$$

where $\hat{x}_r$ takes the place of the nominal state. Similarly, at the next time step $k+1$, the Lyapunov function candidate is given by

$$V_{k+1}^* \triangleq V\left(\bar{x}_{k+1}, \mu_{k+1}\right) = \left\|\begin{bmatrix} \bar{x}_{k+1} \\ \mu_{k+1} \end{bmatrix}\right\|^2_{\Omega(\Omega_0, P)} \tag{4.44}$$

$$= \|[\bar{x}_{k+1}]\|^2_{\Omega_0} + J_{k+1}^*. \tag{4.45}$$

Now μ_{k+1}^F, as in (4.28), can be used to derive a bound for V_{k+1}^*. Since

$$V_{k+1}^F \triangleq V\left(\bar{x}_{k+1}, \mu^F\right) = \left\|\begin{bmatrix} \bar{x}_{k+1} \\ F_1\bar{x}_k + F_2\mu_k \end{bmatrix}\right\|^2_{\Omega(\Omega_0, P)} \tag{4.46}$$

$$= \|\bar{x}_{k+1}\|^2_{\Omega_0} + \left\|\left[\hat{x}_{r_{k+1}}, U_{k+1}^F, \epsilon_{k+1}^F, e_{k+1}^F\right]\right\|^2_P \tag{4.47}$$

and

$$V_{k+1}^* = \left\|\left[x_{k+1}, \hat{x}_{r_{k+1}}\right]\right\|^2_{\Omega_0} + J_{k+1}^*, \tag{4.48}$$

we have that

$$(\delta V)_{k+1} \triangleq V\left(\bar{x}_{k+1}, \mu_{k+1}\right) - V\left(\bar{x}_{k+1}, \mu_{k+1}^F\right) \tag{4.49}$$

$$= \|\bar{x}_{k+1}\|_{\Omega_0}^2 + J_{k+1}^* - \|\bar{x}_{k+1}\|_{\Omega_0}^2 \tag{4.50}$$

$$- \left\| \left[\hat{x}_{r_{k+1}}, U_{k+1}^F, \epsilon_{k+1}^F, e_{k+1}^F \right] \right\|_P^2$$

$$= J_{k+1}^* - \left\| \left[\hat{x}_{r_{k+1}}, U_{k+1}^F, \epsilon_{k+1}^F, e_{k+1}^F \right] \right\|_P^2, \tag{4.51}$$

and it follows that

$$(\delta V)_{k+1} \leq 0, \tag{4.52}$$

since μ_{k+1}^F is feasible and

$$J_{k+1}\left(\mu_{k+1}^F\right) \geq J_{k+1}^*.$$

Obviously, this implies

$$V_{k+1}^* \leq V_{k+1}^F. \tag{4.53}$$

Now, it remains to show that

$$V_{k+1}^F - V_k^* \leq \alpha \|\bar{x}_k\|^2, \tag{4.54}$$

for (some arbitrarily small) scalar $\alpha > 0$. For that purpose, we use the property (4.38b). At time step k, we have

$$\begin{bmatrix} \bar{x}_k \\ \mu_k \end{bmatrix}^T \Phi\left(\Omega_0, P\right) \begin{bmatrix} \bar{x}_k \\ \mu_k \end{bmatrix} \tag{4.55}$$

$$= \begin{bmatrix} \bar{x}_k \\ \mu_k \end{bmatrix}^T \begin{bmatrix} \bar{A} & \bar{B} \\ F_1 & F_2 \end{bmatrix}^T \Omega\left(\Omega_0, P\right) \begin{bmatrix} \bar{A} & \bar{B} \\ F_1 & F_2 \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ \mu_k \end{bmatrix} - V_k^*. \tag{4.56}$$

Now, note that

$$\begin{bmatrix} \bar{A} & \bar{B} \\ F_1 & F_2 \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ \mu_k \end{bmatrix} = \begin{bmatrix} \bar{A}\bar{x}_k + \bar{B}\mu_k \\ F_1\bar{x}_k + F_2\mu_k \end{bmatrix} \tag{4.57}$$

$$= \begin{bmatrix} \bar{x}_{k+1} \\ \mu_{k+1}^F \end{bmatrix}, \tag{4.58}$$

where μ_{k+1}^F is the feasible solution, as defined in equation (4.28). By inserting (4.58) into (4.56), we have that

$$\begin{bmatrix} \bar{x}_k \\ \mu_k \end{bmatrix}^T \Phi\left(\Omega_0, P\right) \begin{bmatrix} \bar{x}_k \\ \mu_k \end{bmatrix} \tag{4.59}$$

$$= \begin{bmatrix} \bar{x}_{k+1} \\ \mu_{k+1}^F \end{bmatrix}^T \Omega\left(\Omega_0, P\right) \begin{bmatrix} \bar{x}_{k+1} \\ \mu_{k+1}^F \end{bmatrix} - V_k^* \tag{4.60}$$

$$= \left\| \begin{bmatrix} \bar{x}_{k+1} \\ \mu_{k+1}^F \end{bmatrix} \right\|_{\Omega(\Omega_0, P)} - V_k^* \tag{4.61}$$

$$= V_{k+1}^F - V_k^* \tag{4.62}$$

Since the inequality (4.38b) is strict it then follows that (4.54) holds for some $\alpha > 0$. ∎

Remark 10. *We note that the stability test of the above theorem may be generalized along the lines of Løvaas et al. (2007a), Løvaas (2008) to include sum quadratic constraints (and associated scalar multipliers) that describe the static nonlinearity associated with the on-line optimization. However, whilst such an approach leads to a less conservative stability test, it produces a bilinear matrix inequality (BMI) in the cost function matrix P. The resulting non-convex BMI condition is useful for checking stability of a particular P (since the relevant condition then becomes an LMI), but problematic (and therefore not exploited) when we next consider the general synthesis problem of choosing P subject to stability.*

4.6 Robust Design

Note that, given P, it is a standard LMI feasibility problem to search for Ω_0 that satisfies (4.38), thereby checking robust stability of a particular design. Such a P is, however, likely to give a conservative design. In this section, we propose a semi-definite program (SDP) that may be used to compute a matrix $P \geq P_0$ that satisfies the stability criterion (28) and is as "close" as possible to the nominal cost function matrix P_0. The SDP is similar to

those proposed in Løvaas et al. (2008a,b) and as follows:

$$\inf_{P_1, P_2, \Omega_0} \text{trace}\,(P_1) + q\text{trace}\,(P_2) \tag{4.63a}$$

$$\text{s.t.} \begin{cases} P & = \text{diag}\{P_1, P_2\} \\ \Sigma_{\{Q,R,S\}}\,(P) & \leq 0 \\ \Phi\,(\Omega_0, P) & < 0 \\ \Omega\,(\Omega_0, P) & > 0 \end{cases} \tag{4.63b}$$

where $q > 0$ is a scalar, and where we have also added the structural constraint $P = \text{diag}\{P_1, P_2\}$, such that the cost (4.5) takes the form $J\,(x, U, \varepsilon, e) = \|[x, U]\|_{P_1}^2 + \|[\varepsilon, e]\|_{P_2}^2$. Regarding the feasibility of the above SDP, we have the following strong result:

Theorem 6. *If the matrices, A and $A_r - LC_r$, are both stable, then the problem* (4.63) *is feasible.*

Proof. To construct a feasible solution and thereby prove Theorem 6, we will adapt the arguments used to prove Theorem 4.5 in Løvaas (2008). To this end, let $\hat{\Omega}$ be a "Lyapunov matrix" satisfying

$$\bar{A}^{\text{T}}\hat{\Omega}\bar{A} - \hat{\Omega} < 0.$$

Note that, using $\hat{\Omega}$, any scalar $\epsilon_1 > 0$ and some sufficiently large scalar $\alpha_1 > 0$, the following inequality holds:

$$\begin{bmatrix} \bar{A} & \bar{B} \end{bmatrix}^{\text{T}} \hat{\Omega} \begin{bmatrix} \bar{A} & \bar{B} \end{bmatrix} - \begin{bmatrix} I & 0 \end{bmatrix}^{\text{T}} \hat{\Omega} \begin{bmatrix} I & 0 \end{bmatrix}$$
$$- \text{diag}\{0, \alpha_1 D_1^{\text{T}} D_1 + \epsilon_1 I\} < 0. \tag{4.64}$$

Also, define matrices, H_1 and H_2, satisfying the following two Lyapunov inequalities:

$$\Gamma^{\text{T}}(N_u, n_u) H_1 \Gamma(N_u, n_u) - H_1 < -\alpha_1 D_1^{\text{T}} D_1,$$

$$\begin{bmatrix} \Gamma(N_\epsilon, n_h) & \bar{H} \\ 0 & A_r \end{bmatrix}^{\text{T}} H_2 \begin{bmatrix} \Gamma(N_\epsilon, n_h) & \bar{H} \\ 0 & A_r \end{bmatrix} - H_2 < 0.$$

Here, the various matrices are as in the definition of F_2 in (4.27), and the scalar $\alpha_1 > 0$ is as in (4.64). From the strict inequalities above and from the structure of the matrix F_2, we note that the following inequality holds

$$
\begin{aligned}
&\begin{bmatrix} F_1 & F_2 \end{bmatrix}^{\mathrm{T}} \operatorname{diag}\{H_1, \epsilon_2 H_2\} \begin{bmatrix} F_1 & F_2 \end{bmatrix} \\
&- \begin{bmatrix} 0 & I \end{bmatrix}^{\mathrm{T}} \operatorname{diag}\{H_1, \epsilon_2 H_2\} \begin{bmatrix} 0 & I \end{bmatrix} \\
&+ \operatorname{diag}\{-\epsilon_2 \alpha_2 I, \alpha_1 D_1^{\mathrm{T}} D_1 + \epsilon_2 I\} \leq 0,
\end{aligned}
\tag{4.65}
$$

using some sufficiently large scalar $\alpha_2 > 0$ and any sufficiently small scalar $\epsilon_2 > 0$. By choosing $\epsilon_2 = \epsilon_1 > 0$ small enough and adding (4.64) to (4.65) we obtain

$$
\Phi\left(\hat{\Omega}, \hat{P}\right) < 0, \quad \hat{P} \triangleq \operatorname{diag}\{0, H_1, \epsilon_2 H_2\}.
\tag{4.66}
$$

It can then be verified that the following is a feasible solution to (4.63):

$$
P = \operatorname{diag}\{P_1, P_2\} = P_0 + c\hat{P}, \quad \Omega_0 = c\hat{\Omega},
\tag{4.67}
$$

where $c > 0$ is some sufficiently large scalar. To see this, note from (4.18) and the definitions of H_1, H_2, that $\Sigma_{\{Q,R,S\}}(P_0 + c\hat{P}) \leq 0$, for any $c > 0$. Furthermore, using (4.67), we have

$$
\Phi(\Omega_0, P) = c\Phi(\hat{\Omega}, \hat{P}) + \Phi(0, P_0).
$$

∎

In the sequel, we denote by P^* a feasible and (near) optimal solution to (4.63).

Remark 11. *Since $\Sigma_{\{Q,R,S\}}(P^*) \leq 0$, we have that $P^* \geq P_0$, where P_0 is as in* (4.18).

By use of $P = P^*$ we obtain the following robust design.

Algorithm 4. *Robust Output-Feedback Reduced-Order MPC Offline:*

1. Choose any integers N, N_u and N_ϵ satisfying $N \geq N_u \geq 1$, $N \geq N_\epsilon \geq 1$.

2. Generate a reduced-order model.

3. Choose any T and t such that the set $X_F = \{x_r | T x_r \leq t\}$ satisfies (4.13).

4. Choose any observer gain such that $A_r - LC_r$ is stable.

5. Choose any matrices $Q \geq 0$, $R > 0$ and $S > 0$ and determine $P = P^*$ by solving (4.63).

Online: At each time step $k \geq 0$, solve $\left[P^{N,N_\varepsilon}\right]$ using $x_r = \hat{x}_{r_k}$, then apply $u_k = \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} U^*(\hat{x}_{r_k})$ to (4.1).

We next address the important question of conservatism of the above robust reduced-order design. Specifically, we show that, under a reasonable assumption, the proposed design is non-conservative in the sense that $P^* \approx P_0$ provided that the neglected dynamics $\Delta(z) \triangleq C_p(zI - A_p)^{-1}B_p - C(zI - A)^{-1}B$ are sufficiently small.

Consider the following assumption which relates the plant model to the reduced order model:

Assumption 1. We have

$$A = \begin{bmatrix} A_r & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_r \\ B_2 \end{bmatrix}, \quad C = \begin{bmatrix} C_r & C_2 \end{bmatrix}.$$

Furthermore, the matrix A_{22} is stable.

Remark 12. Note that A_r can be placed in the upper left corner of A by using a balanced realization of the plant model. Furthermore, the requirement that A_{22} is stable, is always satisfied when the reduced order model is obtained using, for example, balanced truncation.

Under the above assumption, we will show that Algorithm 4 converges to the associated nominal design obtained using $P = P_0$ as the neglected dynamics goes to zero $\Delta(z) \triangleq C(zI - A)^{-1}B - C_r(zI - A_r)^{-1}B_r$. To this end, note that, replacing the matrices A_{21}, B_2 in Assumption 1 by δA_{21}, δB_2 using some scalar δ (and thereby changing the plant model) amounts to shrinking the neglected dynamics by a factor to obtain $\Delta(z) \leftarrow \delta \Delta(z)$. Thus we shall be concerned with establishing the following theorem, which shows that: if the matrices A_{21}, B_2 are "small", then $P \approx P_0$.

Theorem 7. *Suppose Assumption 1 holds. For any given $\epsilon > 0$, there exists a $\delta > 0$, such that, if we make the assignments $A_{21} \leftarrow \delta A_{21}$, $B_2 \leftarrow \delta B_2$, then*

$$\text{Trace}\,(P^* - P_0) \leq \epsilon.$$

Proof. Let P_I be the solution to $\Sigma_{\{0,0,0\}}(P_I) + I = 0$. For any given $\epsilon > 0$, consider

$$P = P_0 + \alpha P_I, \quad \alpha = \frac{\epsilon \tilde{q}}{\theta(P_I)} > 0, \tag{4.68}$$

where $\tilde{q} \triangleq \min\{1, q\} > 0$ and where $\theta(\text{diag}\{P_1, P_2\}) \triangleq \text{Trace}\,(P_1) + q\text{Trace}\,(P_2)$ [see (4.63)]. In view of (near) optimality of P^*, it suffices to show that there exists a $\delta > 0$ such that P in (4.68) is feasible provided we make the assignments $A_{21} \leftarrow \delta A_{21}$, $B_2 \leftarrow \delta B_2$. Moreover, since the inequality $\Phi(\Omega_0, P) < 0$ in (4.63b) is strict, it suffices, by continuity arguments, to show that P is feasible when $A_{21} = 0$, $B_2 = 0$ (i.e., using $\delta = 0$). To this end, let $A_{21} \leftarrow 0$, $B_2 \leftarrow 0$ and consider the following matrix which is similar to

$$\begin{bmatrix} \bar{A} & \bar{B} \\ F_1 & F_2 \end{bmatrix}$$

(when $A_{21} = 0$, $B_2 = 0$):

$$\Upsilon \triangleq [\text{diag}\{T, I\}] \begin{bmatrix} \bar{A} & \bar{B} \\ F_1 & F_2 \end{bmatrix} [\text{diag}\{T, I\}]^{-1}$$

$$= \begin{bmatrix} A_r - LC_r & A_{12} - LC_2 & 0 & 0 \\ 0 & A_{22} & 0 & 0 \\ LC_r & LC_2 & A_r & B_r D_1 \\ K_F LC_r & K_F LC_2 & 0 & F_2 \end{bmatrix}, \tag{4.69}$$

where

$$T \triangleq \begin{bmatrix} I_{n_x} & 0 & -I_{n_x} \\ 0 & I_{(n-n_x)} & 0 \\ 0 & 0 & I_{n_x} \end{bmatrix}, \tag{4.70}$$

and where have made use Assumption 1. Since the matrices, $A_r - LC_r$, A_{22}, are stable and $\Sigma_{\{Q,R,S\}}(P) < 0$, it follows by the structure of the matrix Υ that there exist some positive definite symmetric matrix $X \in \mathbb{R}^n$ such that

$$\Upsilon^\text{T} \text{diag}\{X, P\} \Upsilon - \text{diag}\{X, P\} < 0. \tag{4.71}$$

Since the above inequality is equivalent to $\Phi(\Omega_0, P) < 0$ with $\Omega_0 = T^\text{T} XT$, the result follows. That is, choosing $\Omega_0 = T^\text{T} XT$ and $\text{diag}\{P_1, P_2\} = P = P_0 + \alpha P_I$ yields a feasible solution provided we make the assignments $A_{21} \leftarrow \delta A_{21}$, $B_2 \leftarrow \delta B_2$, using some sufficiently small, but positive, scalar δ. ∎

Theorem 7 shows that Algorithm 4 converges to a certainty equivalence implementation of the design of Scokaert and Rawlings (1999) as the model uncertainty tends to zero, provided that we make suitable choices for T, t, N and N_ϵ. A numerical example illustrating this convergence is presented in the following section.

4.7   Numerical Examples

In this section we will consider two different systems. The first is a random non-minimum phase 6th order plant with oscillatory dynamics that we will

use to illustrate the procedure. Since this system is non-minimum phase, which leads to a challenging control task, the example suggests that our procedure can be used on systems that contain complex dynamics. The second example is a CFD model describing the motion in a building, which will demonstrate the usefulness of the procedure in real-world problems.

4.7.1   Random 6th-Order System

We consider a 6th order plant given by

$$A = \begin{bmatrix} 0.2809 & 0.2505 & -0.1990 & -0.2232 & 0.0321 & -0.5003 \\ 0.2505 & -0.4756 & 0.3022 & 0.1714 & -0.1126 & -0.1190 \\ -0.1990 & 0.3022 & 0.4621 & 0.0965 & -0.0284 & -0.0891 \\ -0.2232 & 0.1714 & 0.0965 & 0.6050 & -0.0633 & 0.1457 \\ 0.0321 & -0.1126 & -0.0284 & -0.0633 & 0.4647 & -0.1332 \\ -0.5003 & -0.1190 & -0.0891 & 0.1457 & -0.1332 & -0.2399 \end{bmatrix},$$

$$B = \begin{bmatrix} 1.0159 & 0 & 0.5988 & 1.8641 & 0 & -1.2155 \end{bmatrix}^T,$$

and

$$C = \begin{bmatrix} 1.2920 & 0 & 0 & 0.2361 & 0.8428 & 0 \end{bmatrix}.$$

The system has a zero at $z = 6.83$, outside the unit circle, and is consequently non-minimum phase. The output y_k is subject to soft unit bound constraints, and the input u_k is subject to hard unit bound constraints. We choose $N_u = N = 10$, $N_\epsilon = 2$, $Q = I$, $R = 0.1$ and $S = 1000I$.

First, we reduce the system order from $n = 6$ to $r = 5$ and $r = 4$ using balanced reduction (although other model reduction methods could have been used), and we impose the same constraints on the reduced-order models. Reduced-order models with $r = 5$ and $r = 4$ leads to model reduction errors $\|\Delta(z)\|_\infty = 6.9885 \times 10^{-6}$ and $\|\Delta(z)\|_\infty = 0.0221$, respectively. The plant is initialized at

$$x_0 = [-0.9044, \ -9.1380, \ -2.5036, \ 0.6696, \ -0.0821, \ -4.0350]$$

while the observer is initialized at $\hat{x}_{r_0} = C_r^+ y_0$, where C^+ denotes the Moore-Penrose pseudoinverse of C_r, and y_0 is the initial plant output. The

SDP (4.63) is solved using MATLAB with YALMIP (Löfberg, 2004), Se-DuMi (Sturm, 1999) and Matlab Invariant Set Toolbox (Kerrigan, 2005).

Figure 4.1 compares the closed-loop responses of different robust MPC designs computed using Algorithm 4. The figure also shows the response when using the associated nominal design (NMPC), which is algorithm 4 but using $P = P_0$ as in (4.18).
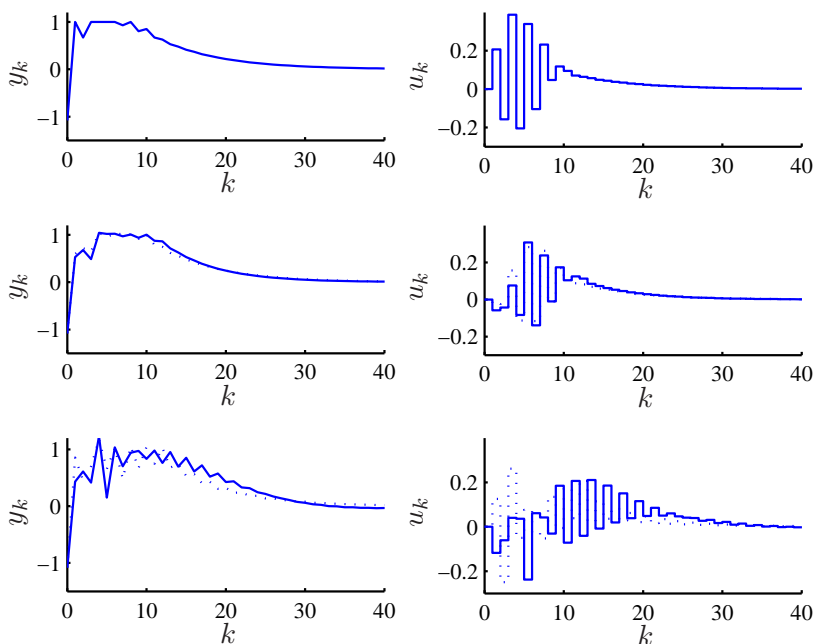

Figure 4.1: Top: NMPC using the plant as the nominal model. Center: NMPC (dotted) and robust MPC (solid) using a ROM with $r = 5$. Bottom: NMPC (dotted) and robust MPC (solid) using a ROM with $r = 4$.

For this initial condition, the open-loop response overshoots the upper

output constraint significantly, and so the robust design is good at keeping its soft constraints. Figure 4.1 suggests that the robust MPC is not overly conservative when the model uncertainty is relatively small.

If we proceed by truncating to $r = 3$, the model reduction error increases by an order of magnitude to $\|\Delta(z)\|_\infty = 0.1373$. In this case, the nominal MPC design fails severely, as illustrated in Figure 4.2. In fact, the output for the nominal design oscillates between its soft constraints. On the other hand, the "robustified" design obtained by applying Algorithm 4 still performs well.
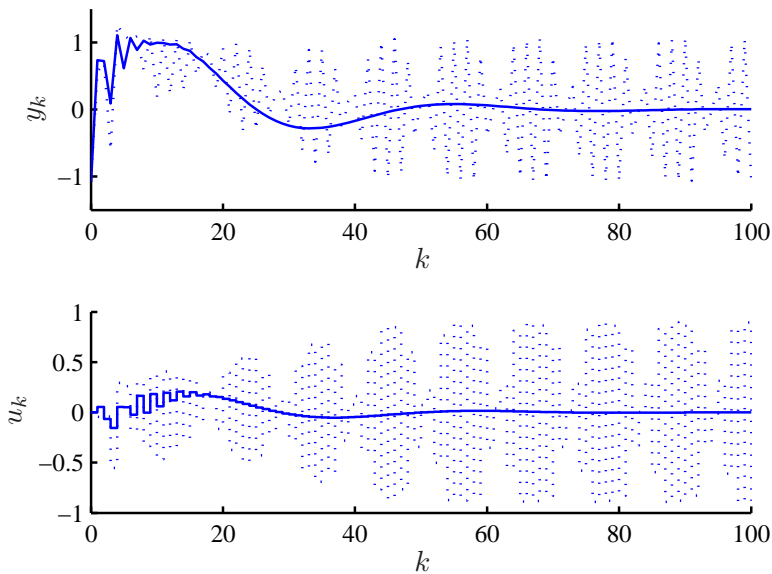
Figure 4.2: NMPC (dotted) and robust MPC (solid) using a ROM with $r = 3$.

4.7.2   Vibration Control of Hospital Building

To investigate the potential of using the design procedure for control of CFD models, we consider a model of the Los Angeles University Hospital building (Chahlaoui and van Dooren, 2002). The building has 8 floors, each with 3 degrees of freedom; vertical and horizontal displacements, and rotation. The CFD model of the building is given as an LTI, which has 48 states, one input and one output. The system is lightly damped, with long lasting oscillations in response to an impulse input (representing the building's response to, for example, an earthquake).

The relatively large number of states in this CFD model, combined with the need for a fast controller in order to effectively counteract the vibrations, would rule out an MPC design based on the full-order model with 48 states. To generate reduced-order models for this problem, we use balanced truncation, and first obtain a model with 8 states, for which the model reduction error $\|\Delta(z)\|_\infty = 0.0755$.

Based on the reduced-order model, model predictive controllers are designed. The controller objective is to reduce the magnitude and the durations of the oscillations. In open loop, the building keeps oscillating for up to 15 seconds, as shown in Figure 4.3. The controller parameters are chosen as $N_u = N = 10$, $N_\epsilon = 4$, $Q = 10^8 C_r^T C_r$, $R = 0.001$ and $S = 1000I$. From Figure 4.3 it can be seen that both the robust MPC and the nominal MPC are able to significantly reduce the oscillations present in the open loop response. The robust MPC is slightly conservative in this simulation.

Now, we proceed by using 6, 5 and 4 states in the reduced-order models. The impulse responses of the CFD model and the reduced-order models in open loop are compared in Figure 4.4.

Figure 4.5 shows the closed-loop performance of the different controllers, where it can be seen that the nominal design fails for $r = 5$ and $r = 4$, while the robust design is still stable, as established in Theorem 5.
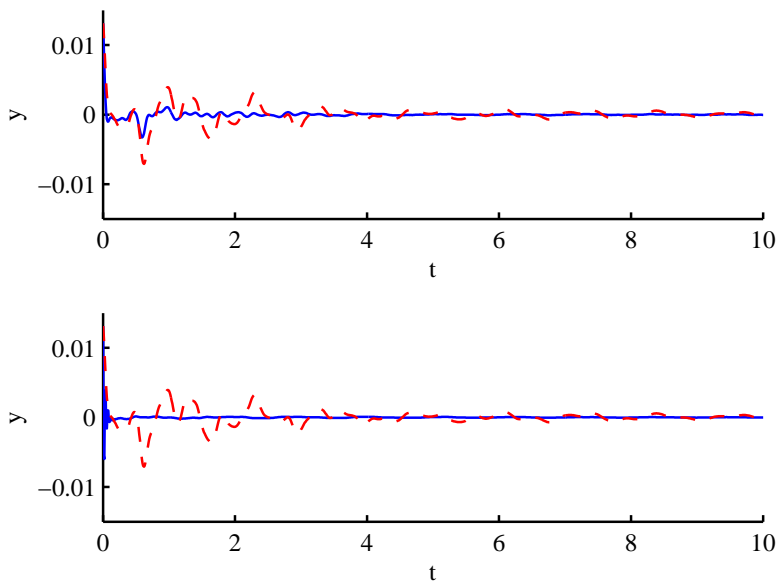
Figure 4.3: Performance with robust MPC (top) and NMPC (bottom) designed based on a model with 8 states. The open loop response is shown as the red, dashed line.
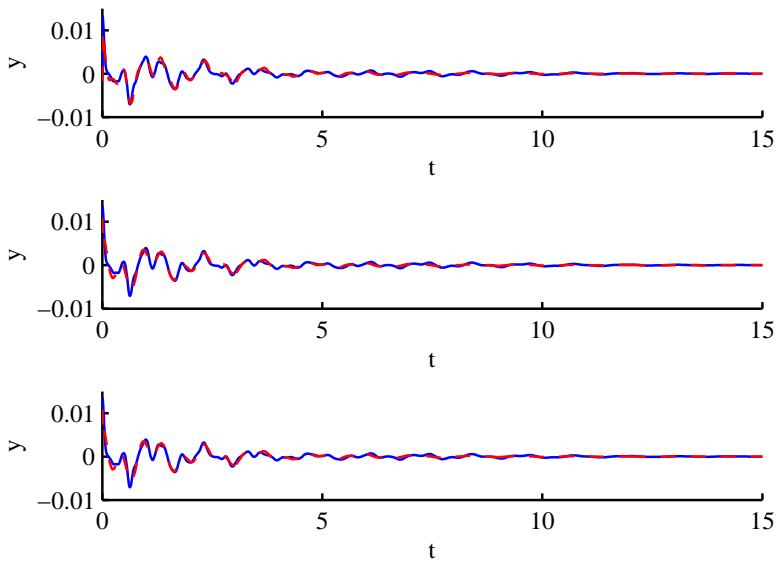
Figure 4.4: Impulse response from reduced-order models with 6 states (top plot), 5 states (center plot) and 4 states (lower plot)..
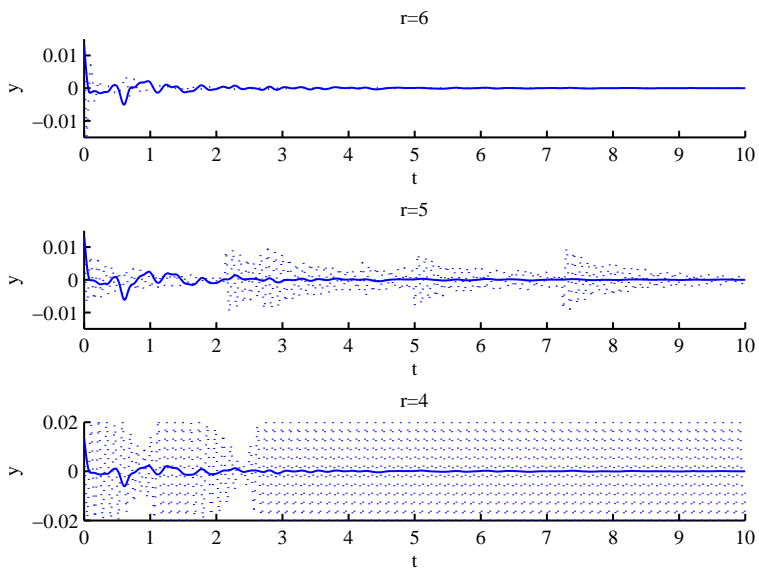
Figure 4.5: Closed-loop impulse response using NMPC (dotted) and robust MPC (solid).

4.8   Concluding Remarks

In this chapter we have developed a procedure for obtaining closed-loop stability of output-feedback MPC based on reduced-order models. The procedure uses the information available in the original plant model in the offline phase of determining the cost function parameters. Since our main objective is to design an efficient online controller, it is reasonable to put extra work into the offline stage.

For large-scale systems, our procedure may be too computationally demanding, since it requires solving LMIs involving the full-order system matrices. However, it seems possible to further develop the procedure described here by treating parts of the dynamics as model uncertainty.

Chapter 5

Explicit MPC for Large-Scale Systems

I~N~ this chapter we present a framework for achieving constrained optimal real-time control for large-scale systems with fast dynamics. The methodology uses the explicit solution of the model predictive control problem combined with model reduction, in an output-feedback implementation. Reduced-order models are derived using the goal-oriented, model-constrained optimization formulation from Section 2.3.5, that yields efficient models tailored to the control application at hand. The approach is illustrated on a simple example for a 1D heat equation, and for a challenging large-scale flow problem that aims to control the shock position in a supersonic diffuser. We compare the results with control based on reduced-order models using POD. This chapter is based on Hovland et al. (2006, 2008c).

5.1 Introduction

With the increasing interest in fluid flow control over the last decade, there arises a need for control methodology that can achieve constrained optimal real-time control of distributed systems with fast dynamics, such as e.g. in mechatronics, MEMS, rotating machinery and acoustics. Model

reduction combined with MPC has been applied in process control systems, such as in Astrid and Weiland (2005), where the authors use POD to generate a reduced-order model that is used to control an industrial glass feeder. eMPC, however, has several advantages for implementation in real-time systems: 1) The online computational time can be reduced to the microsecond–millisecond range, and 2) constrained, optimal control is achieved with low complexity, easily verifiable real-time code, justifying the employment of eMPC in embedded and safety-critical systems. However, the use of eMPC is critically dependent on having a system model of low order, typically with a maximum of ten states. For CFD applications, this motivates use of model order reduction methodology applicable for large-scale systems, that can provide reduced models of very low order, that at the same time are suitable for control. CFD models of systems such as those mentioned above, typically have state dimensions exceeding 10^4, which is prohibitive for model-based controller design. In order to achieve real-time control, the control structure must be capable of computing the control input faster than the sampling rate of the system. Therefore, we need approximate simulation models that are of sufficiently low order for control design, and a framework for coupling the controller with the plant based on the approximate models, while accounting for the error inherent in the approximate model. Such designs were also considered in Chapter 3 and 4, but here we extend the methodology to large-scale systems, for which the model reduction methods from Chapter 3 and 4 are too computationally demanding. We present a new framework for achieving real-time constrained optimal control for large-scale systems with fast dynamics that exploits recent advances in a goal-oriented model reduction methodology and eMPC.

The contribution of this chapter is twofold: 1) We propose an approach for achieving constrained optimal control in applications that are described by models of high order, while being characterized by fast sampling rates, by combining a goal-oriented model reduction method with the explicit solution to the MPC problem. We attach the control structure to the plant with a Kalman filter that accounts for the error introduced in the model approximation process. 2) We demonstrate the performance of reduced

models obtained by goal-oriented optimization in control system design.

Demonstrating the feasibility of achieving real-time constrained optimal control for large-scale systems with fast dynamics is essential if reduced-order modeling methods are to be adopted in applications, such as onboard actual aerospace systems. Even with the considerable recent progress in model reduction to enable flow control, achieving real-time control in a constrained setting has not previously been possible. It is only the application of the recently developed model reduction methodology, which targets the control problem to give models of very low dimension, that makes explicit MPC a feasible approach in this setting. To our knowledge, this is the first time that model reduction has been used in an explicit MPC setting to address the issue of constraints.

5.2 Reduced-Order MPC

We use the control structure of Figure 2.4, and a Kalman filter as in (3.1) to estimate the reduced-order states based on the output of the CFD model, and we denote by $\hat{x}_r$ the resulting estimate of the reduced state x_r.

The framework for guaranteeing robust stability of reduced-order MPC described in Chapter 4 relies on solving LMIs that are of the same dimension as the number of states in the CFD model. For large-scale systems such as those considered in this chapter, this is not feasible with the current setup, due to the large computational requirements involved when solving LMIs. We therefore use the nominal model (the reduced model) for controller design, and address certain robustness issues during the design stage.

Given the uncertainty introduced through the model reduction process, one cannot guarantee that feasibility of the underlying optimization problem is maintained and that the constraints on the states/outputs are fulfilled. This problem is handled through the use of soft constraints. Relaxing the state constraints in effect removes the feasibility problem, at least for stable systems (Bemporad and Morari, 1999).

5.2.1   Implementation of Model-Constrained Reduction

We will use the model-constrained optimization approach described in Section 2.3.5 to derive reduced-order models.

In practice, the optimization problem (2.21) may not be tractable for large-scale problems. In a computationally efficient implementation of the method (Bui-Thanh et al., 2007), the basis functions are assumed to be a linear combination of a finite collection of full-state snapshots $\mathcal{X}$:

$$\Phi_r = \mathcal{X}\Xi, \tag{5.1}$$

where $\Xi \in \mathbb{R}^{M \times r}$, M is the number of snapshots and r is the dimension of the reduced state. Then, the elements of the matrix Ξ become the optimization variables, and the number of optimization variables is reduced from $r \times n$ to $M \times r$. As a consequence, neither the gradient computation nor the optimization step computation (which dominate the cost of an optimization iteration) scale with the full system size n.

If the model reduction procedure is to be implemented on a computer for a particular problem, a discrete formulation is required. Consequently, the integrals in equation (2.21a) are replaced by summation, which leads to the following formulation of the optimization problem:

$$\min_{\Phi_r, x_r} \frac{1}{2} \sum_{\ell=1}^{\mathcal{S}} \sum_{k=1}^{M} \left(y_k^\ell - y_{r_k}^\ell\right)^T \left(y_k^\ell - y_{r_k}^\ell\right)$$
$$+ \frac{\beta}{2} \left[\sum_{j=1}^{r} \left(1 - \phi_j^T \phi_j\right)^2 + \sum_{i,j=1, i \neq j}^{r} \left(\phi_i^T \phi_j\right)^2 \right] \tag{5.2a}$$

subject to:
$$\Phi_r^T E \Phi_r x_{r_{k+1}}^\ell = \Phi_r^T A^\ell \Phi_r x_{r_k}^\ell + \Phi_r^T B^\ell u_k^\ell, \ \ell = 1, \ldots, \mathcal{S}, \ k = 1, \ldots, M \tag{5.2b}$$

$$\Phi_r x_{r_1}^\ell = x_0^\ell, \ \ell = 1, \ldots, \mathcal{S}, \tag{5.2c}$$

$$y_{r_k}^\ell = C^\ell \Phi_r x_{r_k}^\ell, \ \ell = 1, \ldots, \mathcal{S}, \ k = 1, \ldots, M, \tag{5.2d}$$

where the system matrices E, A, B and C correspond to the discrete-time state-space model.

To solve the constrained optimization problem (5.2), we choose to eliminate the state variables x_r and state equations (5.2b)-(5.2d) and solve an equivalent unconstrained optimization problem in the Ξ-variables. The analytic gradient can be found through basic calculus of variations and use of adjoint variables, and an unconstrained optimization algorithm that uses a trust-region-based Newton method (Coleman and Li, 1996) can be used to determine the optimal basis. Since the optimization problem is nonlinear and nonconvex, it is important to generate a good initial guess. One possibility is to pick the POD basis as an initial guess. Alternatively, the initial guess for the case of r basis vectors can be chosen to be the solution of the optimization problem for $r-1$ basis vectors plus an arbitrary rth vector. This iterative procedure can be initialized at any value $r \geq 1$ with the POD basis vectors as an initial guess on the first iteration.

5.2.2 Complexity

The complexity of the proposed control scheme is given by the offline model reduction cost plus the cost of solving the eMPC problem offline for the reduced model. The former is determined by the number of optimization variables in the optimization problem (5.2), which is Mr, as well as the cost of solving the high-fidelity model (to generate the snapshots and to compute the gradient information required by the optimizer). The cost of solving the eMPC problem is problem dependent, but increases rapidly with the number of parameters, the number of input steps to be optimized and the number of constraints in the mpQP. For problems whose solutions consist of a large number of regions, one can easily run into numerical problems. Also, the memory required to store the eMPC solution online increases rapidly as the size of the solution grows. A large number of polyhedra in the online solution requires a large search tree with many nodes, which entails a longer searching process which might compromise real-time requirements. The scheme is therefore limited to cases where the reduced models can be made reasonably small, typically with around ten states.

Further complexity reduction techniques, such as input blocking, can be used to make the eMPC procedure more tractable in cases where the

problem is large.

In the next two sections, we will study in detail both model reduction and closed-loop results for two specific model reduction benchmarks.

5.3   Case Study: Heat Diffusion

To investigate the implementation of the reduced-order control setup described above, we consider a benchmark described by Chahlaoui and van Dooren (2002), describing heat diffusion in a one-dimensional rod. In this benchmark, discretization of the one-dimensional heat diffusion equation leads to a single-input single-output LTI of the form (2.1) with $E = I$. The model has 200 states, which are the temperatures at different locations in the rod. The input u is a heat source located at $1/3$ of the rod length, and the output y is the temperature recorded at $2/3$ of the length.

5.3.1   Model Reduction

We will compare results using both POD and model reduction by model constrained optimization. First, we discuss how to select the snapshots for the model reduction procedure.

Snapshot selection

Deciding how, how many and how often to pick snapshots is non-trivial in snapshot-based model reduction schemes. Collecting a large number of snapshots for the method in Section 2.3.5 leads to a large number of optimization variables, which in turn increases the complexity of the optimization problem.

Instead, we propose to use non-uniform time grids for the snapshots. M snapshots can found in the interval $t \in [0, \ T]$, with the kth snapshot time t_k as

$$t_k = \frac{T(s^{k-1} - 1)}{s^{M-1} - 1}, \ k = 1, 2, \ldots, M \tag{5.3}$$

where $s > 0$ is a constant stretching factor. T can be chosen by the user, for example based on the step response settling time, or the time to reach steady-state, for the high-fidelity model. While an increase in M is expected to increase the quality of the reduced-order models, it also leads to an increase in the size of the optimization problem that must be solved to determine the basis Φ_r. The choice of M must reflect this trade-off between reduced model quality and reduction cost. The effect of s is to ensure that the snapshots are collected more frequently when the response is changing more rapidly, and it can be tuned for the application at hand. The snapshot distribution is more dense in the beginning of the interval $t \in [0, T]$ if s is chosen so that $s > 1$, and more dense at the end of the interval if $s < 1$. By tuning s, the user may pick snapshots to better fit the nature of the response for the application at hand.

If we choose $M = 20$ snapshots distributed uniformly in the interval $t \in \langle 0, 60]$, the steady-state approximation is good, but the transient response is inaccurate, as shown in Figure 5.1.

To further illustrate the difference in approximation quality with different snapshot selections, consider Figure 5.2, where the reduced-order models are derived using the optimization framework.

The figure compares the step responses for two different reduced-order models with the step response of the high-fidelity model. The reduced model in the upper plot was found by solving the problem (5.2) for $M = 20$ snapshots chosen uniformly over the interval $t \in \langle 0, 5]$ for a step input to the large-scale model. The reduced-order model approximates the transient response quite well, but there is evidently a steady-state error.

For this particular example, collecting snapshots the way described above gives better results than uniform time grids while at the same time keeping the number of snapshots low, which is illustrated in the lower plot of Figure 5.2. The figure also visualizes the non-uniform time grid used to generate the reduced-order model. The approximation quality is obviously higher than for the model in the upper plot, using the same number of snapshots.
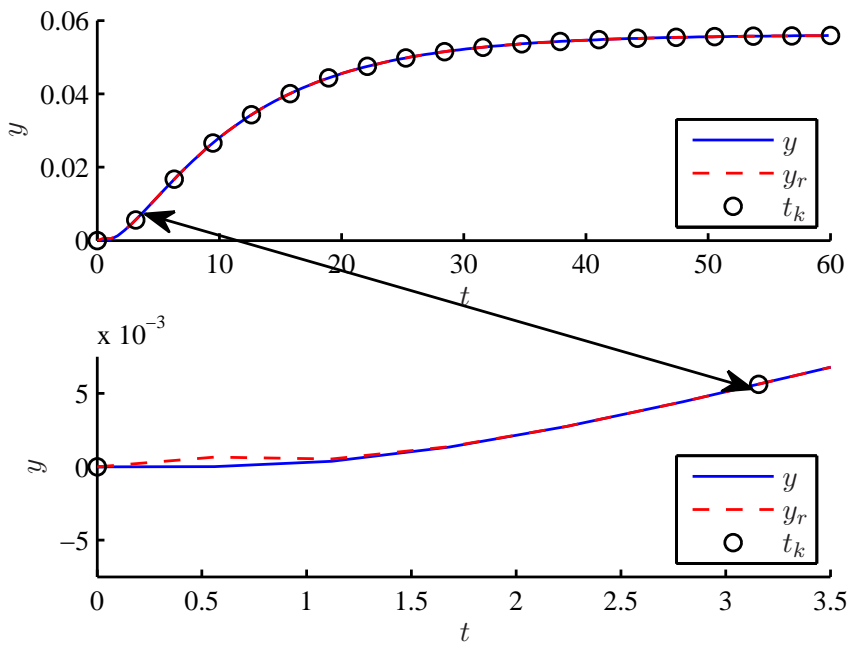
Figure 5.1: Step response of high-fidelity model and reduced-order model of order $r = 4$, generated using 20 snapshots uniformly distributed between 0 and 60 s. The transient error is emphasized by zooming in the bottom plot.
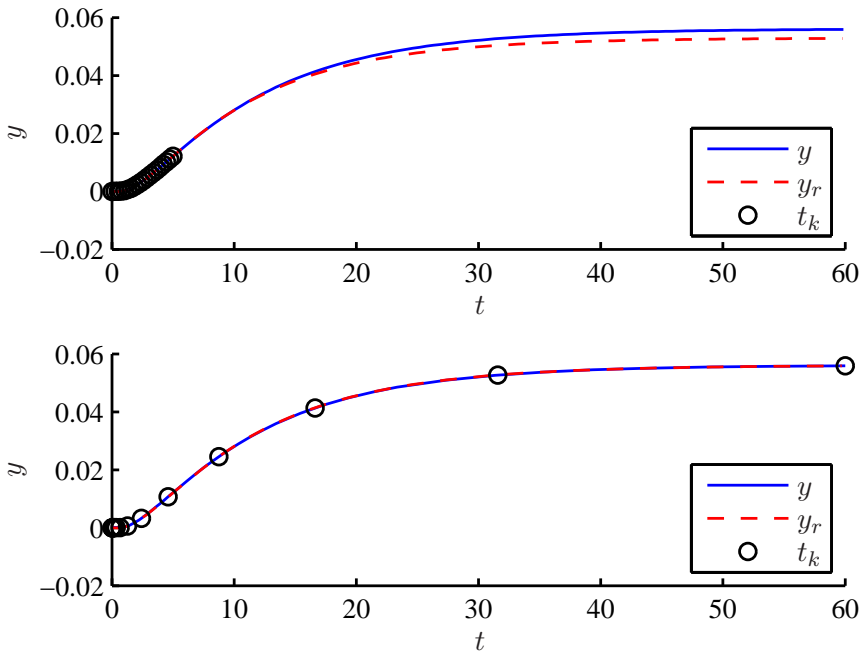
Figure 5.2: Step response of high-fidelity model and reduced-order models of order $r = 4$, generated by solving optimization problem (5.2) by comparing snapshots at time instants indicated by the black circles. Top: 20 snapshots uniformly distributed between 0 and 5 s. The steady-state error is evident. Bottom: Non-uniform time grid. Here, $T = 60$, $M = 20$ and $s = 1.9$.

Model Reduction Results

Reduced-order models of order 1 to 10 are compared in Table 5.1 in terms of the relative $\mathcal{H}_2$ norm of the corresponding error systems, defined as

$$\mathcal{H}_2^e \triangleq \frac{\|\mathcal{G}(s) - \mathcal{G}_r(s)\|_{\mathcal{H}_2}}{\|\mathcal{G}(s)\|_{\mathcal{H}_2}}. \tag{5.4}$$

The reduced-order models are generated by comparing snapshots of the step response of the high-fidelity model at 20 time instants. It is seen that the goal-oriented model based reduction algorithm leads to a significant increase in approximation quality from POD in most cases for this metric, especially for low r.

r	$\mathcal{H}_2^e$ for GOMBR	$\mathcal{H}_2^e$ for POD
1	0.6213	0.7959
2	0.0647	0.5023
3	0.0230	0.0692
4	0.0217	0.0627
5	0.02087	0.0841
6	0.02085	0.0742
7	0.0207	0.0468
8	0.0020	0.0020
9	0.0012	0.0012
10	8.6236×10^{-4}	38×10^{-4}

Table 5.1: Assessment of reduced-order models of order 1 to 10, using the model-constrained reduction algorithm (GOMBR) and the POD. The reduced-order models with the optimized basis give a significant reduction in the relative 2-norm of the error system, especially for low orders.

5.3.2 Closed-Loop Results

To compare the performance of the reduced-order models in closed loop, we first implement an output-feedback infinite horizon LQ-regulator based

on the reduced-order models. We consider the objective of regulating the output of the large-scale system to zero based on the reduced-order models. The controller weights are chosen to reflect this objective, by setting $Q = C^T \tilde{Q} C$, where $\tilde{Q} \in \mathbb{R}^{p \times p}$ is the weight on the output. The input computed by the LQ regulator is given by $u = -K \hat{x}_r$, where K is a constant feedback matrix, and $\hat{x}_r$ is the estimated reduced state. The results are shown in Figure 5.3 and 5.4 for simulation of an optimized and a POD reduced-order model, respectively, with the same weights and $r = 3$. The figures clearly illustrate that the reduced-order model obtained with an optimized basis performs much better in closed loop than the one with a POD basis, and emphasizes the observation from Table 5.1, that the optimized reduced-order models give a better approximation, particularly for small r.

In real-world control problems there will always be some constraints on the state, input and/or output variables. To handle this, eMPC is a better choice than the unconstrained LQ regulator. To illustrate and visualize the setup, we first consider the case where $r = 2$, that is we have only 2 states in the reduced-order model. We set the prediction (and control) horizon $N = 2$. To demonstrate the controller's ability to enforce constraints, we constrain the control input such that $|u| < 1000$. First, the explicit solution to the MPC problem is solved in an offline phase for the relevant area of the reduced-order state space. This solution is used to control the high-fidelity model in an output-feedback setup. The system is initialized with a non-zero output. The resulting response is shown in Figure 5.5 for an optimized basis, where it is seen that the bound constraint on the control input is active during the first half second. It can also be observed that the output from the reduced-order model converges relatively slowly to the output of the high-fidelity model, after about $0.5 \, \text{s}$. The partition of the state space into regions with constant (K_i, k_i) is shown in Figure 5.6, with the phase plane trajectory of the reduced state $\hat{x}_r$ for the simulation in Figure 5.5 indicated by the dotted line.
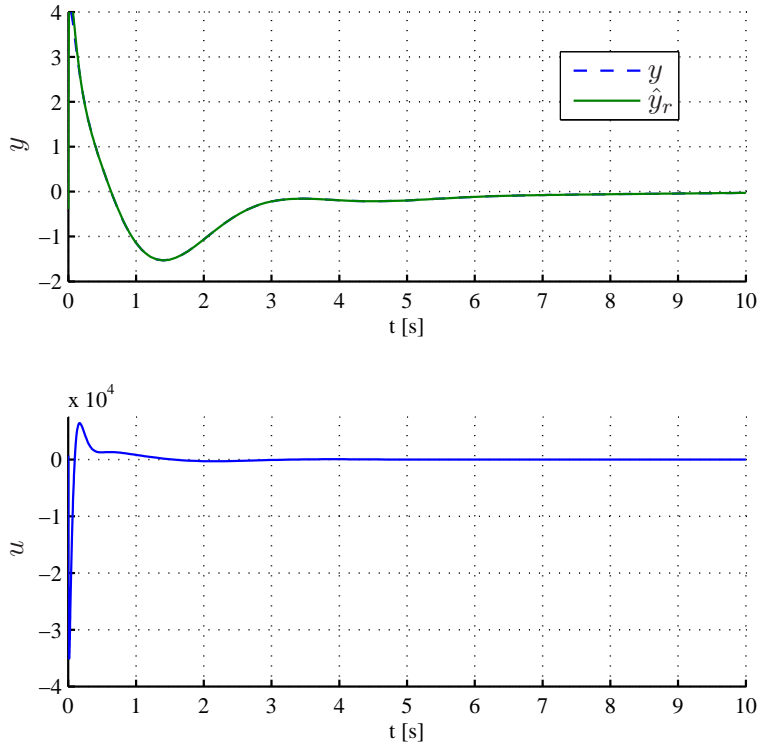
Figure 5.3: Output-feedback LQ regulator for the high-fidelity model based on a reduced-order model with optimized basis for $r = 3$. Top: Estimated output from the reduced-order model $\hat{y}_r$ vs output from the high-fidelity model y. Bottom: Control input.
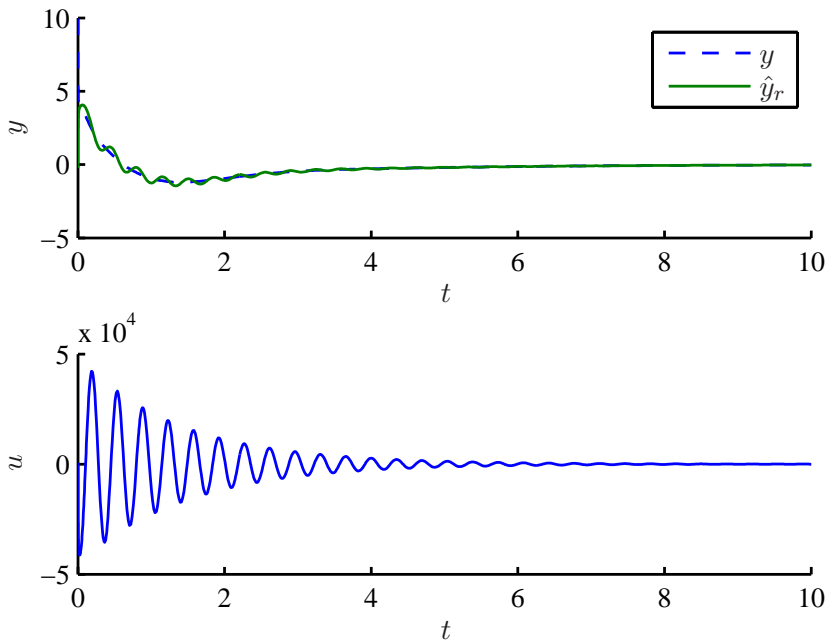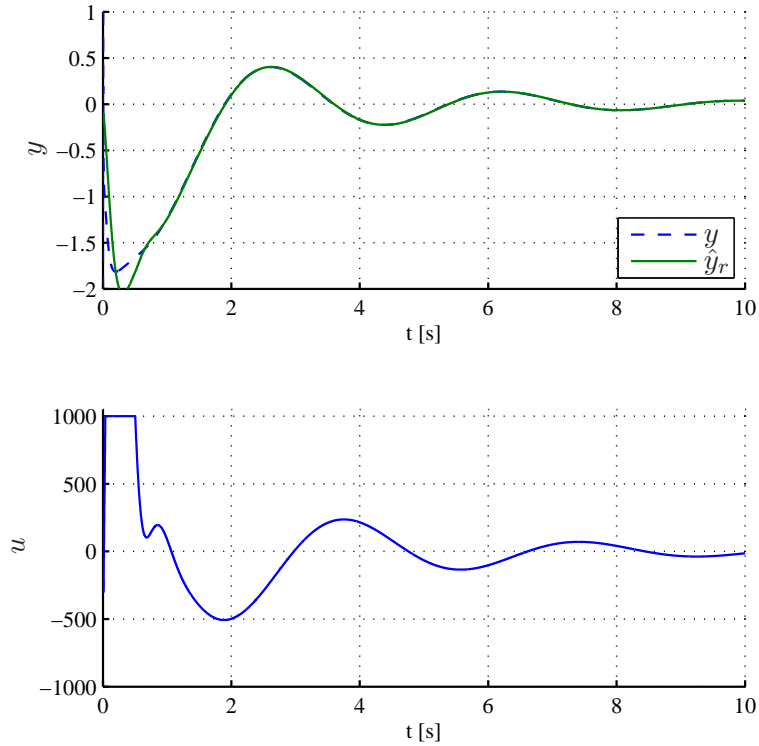
Figure 5.4: Output-feedback LQ regulator for the high-fidelity model based on a reduced-order model with POD basis for $r = 3$. Top: Estimated output from the reduced-order model $\hat{y}_r$ vs output from the high-fidelity model y. Bottom: Control input.

Figure 5.5: Closed-loop performance using eMPC, with $r = 2$ and an optimized basis. Top: High-fidelity y and estimated $\hat{y}_r$ from the reduced-order model. Bottom: eMPC control input, constrained such that $|u| < 1000$. The input constraint is seen to be active during the first half second.
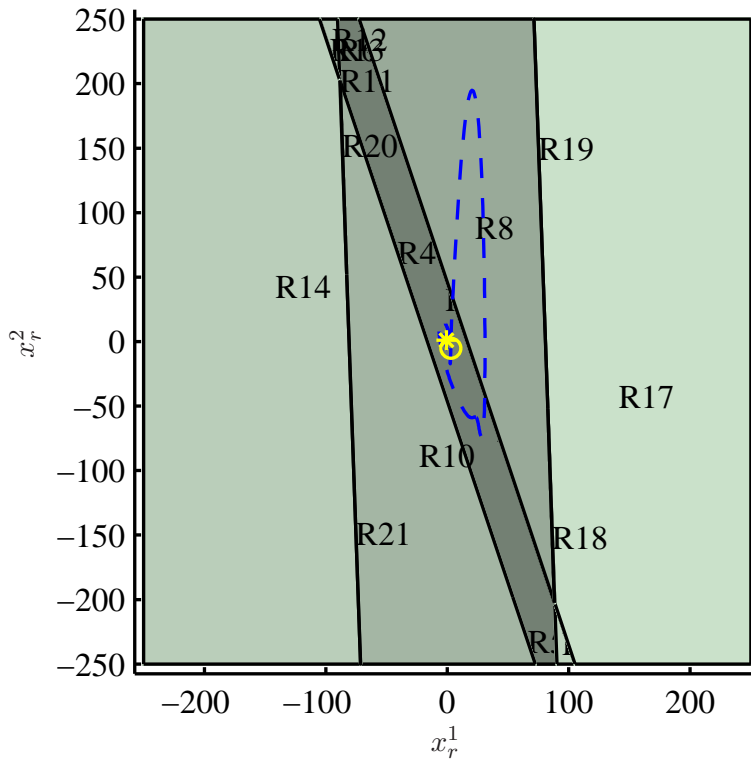
Figure 5.6: Example of state-space partition for a reduced-order model with $r = 2$, with state trajectory starting in o and ending in $*$. The different color shades indicate the 21 regions $\mathbb{R}_i$ in the state space. The controller feedback matrices (K_i, k_i) are constant within each region.

Based on simulations, the reduced-order models generated with the optimized basis perform better in closed loop than the POD models. For this benchmark, they are able to handle higher controller gains, the output is regulated faster to the origin and the control action is smoother. This is illustrated by Figure 5.7. The difference in performance may be attributed to the way in which the goal-oriented models are targeted to give an accurate approximation of the output. For $r = 5$ it is also observed that the output from the reduced-order models converge to the true output an order of magnitude faster than for $r = 2$, resulting in a better closed-loop response. This is what one would expect; adding more states to the reduced-order model leads to better approximations.
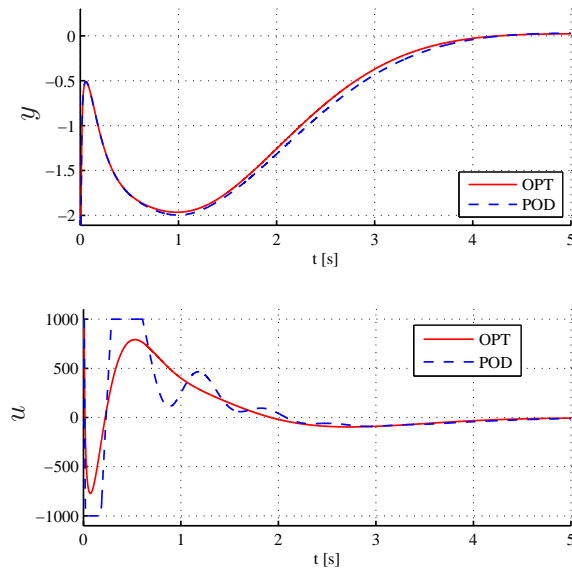

Figure 5.7: Performance comparison for $r = 5$ with eMPC horizon $N = 10$. Top: Output of the full model using reduced-order control based on optimized- and POD basis. Bottom: Control input for the two different cases.

5.4   Case Study: Supersonic Diffuser

This example is a challenging model reduction problem where the objective is to control the position of a shock in a supersonic inlet. The problem is based on an unsteady CFD formulation to simulate subsonic and supersonic flows through a jet engine inlet that is designed to provide a compressor with air at the required conditions (Willcox and Lassaux, 2005). Figure 5.8 shows mach contours in the diffuser at nominal operation.
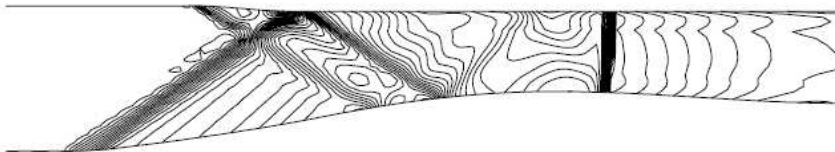

Figure 5.8: Steady-state mach contours in diffuser. A shock sits downstream of the throat. (Willcox and Lassaux, 2005)

The case considered has a steady-state Mach number of 2.2. The flow is assumed inviscid and is modeled by the Euler equations. The underlying CFD code is nonlinear, and the model is linearized about a steady-state solution, giving a stable continuous-time model of the form (2.1), where the continuous-time state $x(t)$ contains the $n = 11,730$ unknown perturbation flow quantities at each point in the computational grid, and the matrices A, B, C and E result from the CFD spatial discretization of the Euler equations[1]. The vector $u \in \mathbb{R}^2$ contains the inputs to the system and $y \in \mathbb{R}$ contains the system output. In this case, the flow state quantities are density, flow velocity components and enthalpy, and the output y is the average Mach number at the throat. There are $3,078$ grid points in the computational domain, giving a total of $n = 11,730$ unknowns. The descriptor matrix E is sparse, and some rows contain only zeros; consequently, E is singular and the inlet model represents a general differential algebraic equa-

---

[1]The system matrices are available in the Oberwolfach Model Reduction Benchmark Collection; http://www.imtek.uni-freiburg.de/simulation/benchmark/.

tion system. The input u contains bleed actuation b (manipulated variable) and an incoming density disturbance d, i.e.

$$u \triangleq \begin{bmatrix} b \\ d \end{bmatrix}.$$

(5.5)

A discrete-time system is obtaining by applying a backward Euler time integration method.

5.4.1   Model Reduction

Reduced-order models of order 1 to 10 are compared in Table 5.2 in terms of the metric (5.4). The reduced-order models are generated by comparing snapshots of the step response of the high-fidelity model at 20 time instants. It is seen that the goal-oriented model based reduction algorithm leads to a significant increase in approximation quality from POD in most cases for this metric, especially for low r. The goal-oriented basis is optimized with the POD basis as the initial guess. In all these cases, the reduced-order model obtained by POD is unstable, while the optimized reduced-order models are not.

   In order to better evaluate the reduced-order models, we compare time-domain and frequency-domain responses for the CFD model with models of reduced order obtained from an optimized basis. We consider a reduced model with 10 states, which was the lowest order that gave satisfactory approximation quality. The optimized basis is found by minimizing the output error for 200 samples in the time interval $t \in (0, 2)\,\mathrm{s}$ in response to a step in each of the two inputs. That is, first we set $b \equiv 1$ and $d \equiv 0$ and collect 200 samples in the time interval, and then we re-initialize the model, set $b \equiv 0$ and $d \equiv 1$ and collect another 200 samples in the same time interval. We use the POD basis vectors generated from the snapshot data as an initial guess for the optimization algorithm.

   The transfer function

$$\mathcal{G}_1 = \frac{y}{b},$$

(5.6)

from bleed b to output y, is shown in Figure 5.9 for the CFD model and the reduced model obtained with an optimized basis. Figure 5.10 illustrates the

r	$\mathcal{H}_2^e$ for GOMBR	$\mathcal{H}_2^e$ for POD
1	0.6213	0.7959
2	0.0647	0.5023
3	0.0230	0.0692
4	0.0217	0.0627
5	0.02087	0.0841
6	0.02085	0.0742
7	0.0207	0.0468
8	0.0020	0.0020
9	0.0012	0.0012
10	8.6236×10^{-4}	38×10^{-4}

Table 5.2: Assessment of reduced-order models of order 1 to 10, using model-constrained optimization (GOMBR) and the POD. The reduced-order models with the optimized basis give a significant reduction in the relative 2-norm of the error system, especially for low orders.

same comparison for the transfer function

$$\mathcal{G}_2 = \frac{y}{d}, \tag{5.7}$$

from the disturbance input d to output y. The transfer function from the disturbance to the output contains a delay, and is consequently more difficult for the reduced-order model to approximate. The reduced-order model is accurate for low frequencies, but does not capture the disturbance response at higher frequencies. However, these higher frequencies are unlikely to occur in typical atmospheric disturbances (Willcox and Megretski, 2005); thus, the reduced model performance shown in Figures 5.9 and 5.10 is deemed acceptable for the purposes of controller design. Figure 5.11 shows the time-domain responses to a step in bleed actuation and a Gaussian density disturbance input. The frequency content of this disturbance input is representative of that expected in practical flight conditions. It can be seen that the reduced model obtained by optimization accurately predicts the time-domain response, confirming its suitability for conditions of practical
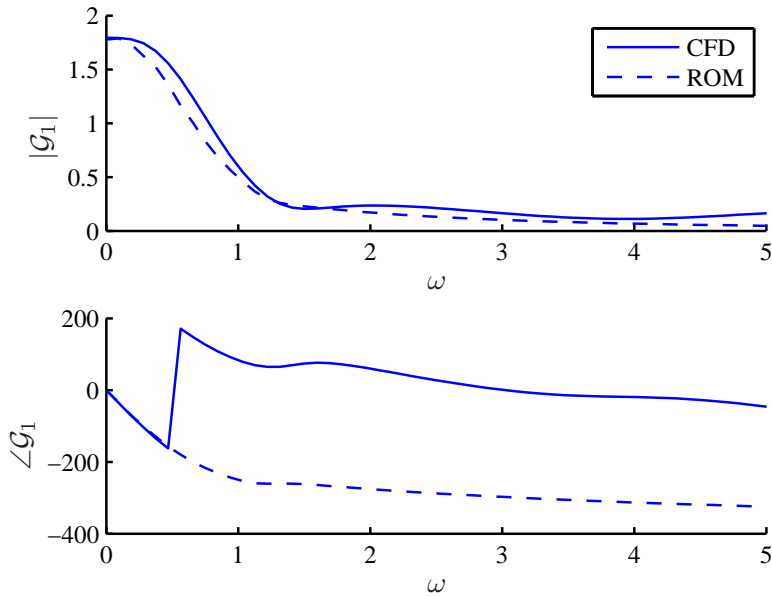
Figure 5.9: Bode diagram comparison of transfer function from bleed b to Mach number y for the CFD model (11,730 states) and the reduced model of order $r = 10$.

interest. It is interesting to note that the reduced-order model obtained by POD performs reasonably well within the range in which the snapshots were collected, i.e. during the first 2 seconds. After that, the output of the reduced-order model from POD diverges, illustrating the instability of the model.
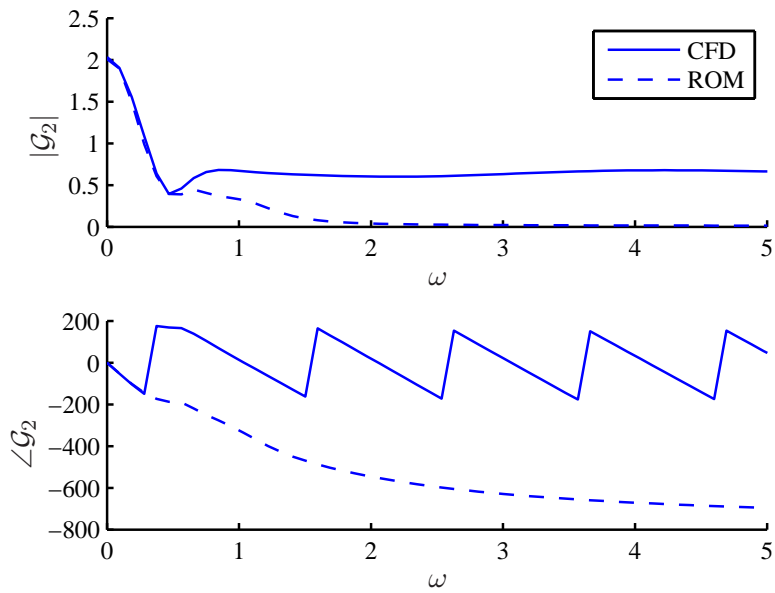
Figure 5.10: Bode diagram comparison of transfer function from disturbance d to Mach number y for the CFD model (11,730 states) and the reduced model of order $r = 10$.
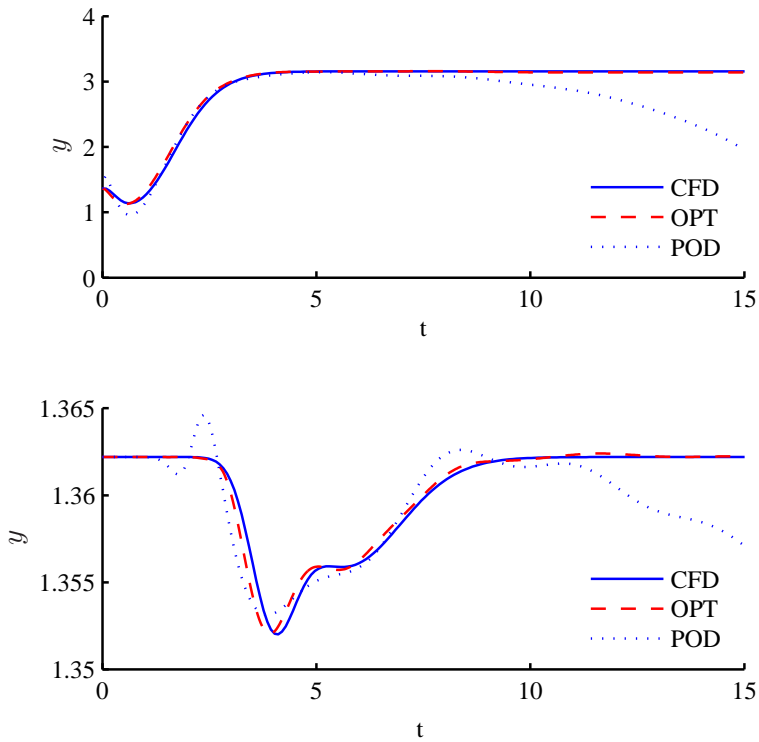
Figure 5.11: Top: Response in Mach number y to a step in bleed input b for the CFD model (11,730 states) and the reduced model of order $r = 10$. Bottom: Response in Mach number y to Gaussian disturbance input d for the CFD model and a reduced model of order $r = 10$.

5.4.2   Closed-Loop Results

Implementing MPC or eMPC directly on the high-fidelity model is infeasible in large-scale settings, for instance when working with models obtained from CFD analysis. We therefore use reduced-order control, where reduced-order models are used to design output-feedback explicit model predictive controllers for the high-fidelity model.

The eMPC framework that uses the reduced-order model was illustrated in Figure 2.4.

The control is implemented as shown in Figure 5.12. In nominal flow conditions, a strong shock sits downstream of the inlet throat. In order to stabilize the shock position in the presence of incoming flow disturbances, and thus prevent engine unstart, active flow control is effected through flow bleeding upstream of the throat.
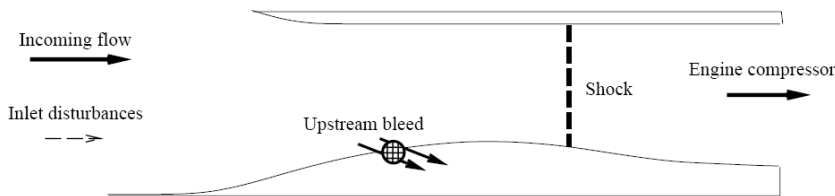

Figure 5.12: Active flow control setup for the supersonic inlet (Willcox and Lassaux, 2005).

The high order of the inlet model is prohibitive for optimal and model-based control, which motivates the use of model reduction. It should be noted that this benchmark is relatively difficult to approximate. Various model reduction methods have been applied to this problem with varying degrees of success. As shown by Willcox and Megretski (2005), POD and Krylov-based methods yield reduced models that are unstable, unless great care is taken during the model reduction process. One reason for this may be that there are inverse responses from the inputs to the output, suggesting non-minimum phase. Non-minimum phase systems are harder to approxi-

mate than minimum phase systems (Antoulas et al., 2002). Balanced truncation is guaranteed to produce stable models, but is difficult to apply in this case due to the singular descriptor matrix E. Good results were shown using the Fourier model reduction approach in Willcox and Megretski (2005); however, that method is applicable only to linear models, in contrast to the optimized-basis algorithm that we are using.

The eMPC framework can be extended naturally to handle disturbances such as the density disturbance. In the controller, we obtain a reduced-order prediction model of the form

$$
\begin{aligned}
\hat{x}_{r_{k+i+1}} &= A_r\hat{x}_{r_{k+i}} + B_r^b b_{k+i} + B_r^d d_{k+i|k} + L\left(y - \hat{y}_{r_{k+i}}\right) &\text{(5.8a)} \\
\hat{y}_{r_{k+i}} &= C_r\hat{x}_{r_{k+i}};\ i \geq 0, &\text{(5.8b)}
\end{aligned}
$$

where B_r^b and B_r^d are the columns of B_r corresponding to the inputs b and d, respectively, and $i = 1,\ldots,\mathcal{N}$ is the ith step on the prediction horizon. We assume that the disturbance d_k is measured, and we use the notation $d_{k+i|k}$ to emphasize that the disturbance d_{k+i}, given the measured value at time step k, is predicted based on an assumption on the future behavior of the disturbance. If we assume that the disturbance is constant over the prediction horizon, one straightforward way to implement the prediction model (5.8) is to augment the state vector and the system matrices as follows:

$$
\hat{x}_{r_k}^a = \begin{bmatrix} \hat{x}_{r_k} \\ d_k \end{bmatrix}, \tag{5.9}
$$

$$
A_r^a = \begin{bmatrix} A_r & B_r^d \\ 0 & 1 \end{bmatrix}, \tag{5.10}
$$

and

$$
C_r^a = \begin{bmatrix} C_r & 0 \end{bmatrix}. \tag{5.11}
$$

To avoid numerical difficulties (the augmented system is marginally stable if we set $d_{k+1} = d_k$), we replace the 1 in equation (5.10) with a scalar δ, and typically choose $\delta = 0.99$.

Now, the control structure can be summarized as follows:

Case	Λ	α	t_p
1	0.01	$2f_0^2$	5
2	0.02	$2f_0^2$	5
3	0.04	$2f_0^2$	5

Table 5.3: Disturbance parameter values for different simulation cases. $f_0 = 3.426$ is related to the steady-state for which the nonlinear model is linearized.

- The Mach number is measured using the output equation

$$y_k = Cx_k. \tag{5.12}$$

- The reduced state is estimated using a Kalman filter based on the reduced-order model and the output of the CFD model.

- The reduced state estimate is fed to the explicit model predictive controller along with the measured disturbance, where the bleed input b_k is found as an explicit function of the augmented state (5.9).

- Control is effected through upstream bleed.

For all results presented in the following, the inlet model is discretized with a time step of $\Delta_t = 0.025\,\mathrm{s}$. The controllers are verified to be sufficiently fast for this example.

The disturbance input is set to be a Gaussian distribution, which is described by its amplitude Λ, rise time α and peak time t_p through the relation

$$d = \rho\left(t\right) = -\Lambda\rho_0 e^{-\alpha(t-t_p)^2}. \tag{5.13}$$

In the following, we address the controller robustness by tuning its performance for a set of disturbances for which the linear model is a good representation of the nonlinear CFD model. (The linearized CFD model is only valid for small perturbations from steady-state conditions.) The parameter values for the disturbance inputs are shown in Table 5.3, and the
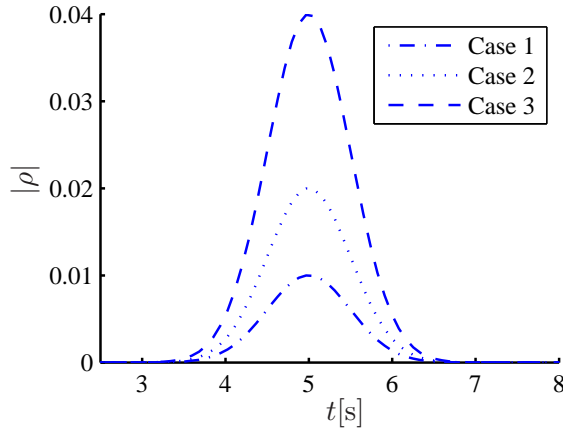
Figure 5.13: Magnitude of disturbance inputs used in Cases 1-3.

different disturbance cases are shown in Figure 5.13.

The computed control input b_k is in fact a perturbation about the nominal steady state bleed b^{ss} of 1% of the inlet mass flow,

$$b^{total} = b^{ss} + b_k. \tag{5.14}$$

We therefore require that the total bleed b^{total} is non-negative, i.e.

$$b_{k+i} \geq -0.01; \ i \geq 0. \tag{5.15}$$

We also put an upper bound on the control action,

$$b_{k+i} < b_{\max}; \ i \geq 0, \tag{5.16}$$

and we bound the Mach number at the throat

$$y_{\min} < y_{r_{k+i}} < y_{\max}; \ i \geq 0. \tag{5.17}$$

Since our objective is to prevent the shock from moving upstream causing engine unstart, we will set $y_{\min} > 1$, e.g. $y_{\min} = 1.1$. The controller tuning parameters are the weighting matrices, the prediction horizon, and the

control horizon in the MPC formulation. Good performance is obtained by setting $\mathcal{M} = \mathcal{N} = 10$, $Q = C_r^T C_r$, $R = 0.05$ and P to the solution to the algebraic Riccati equation. The resulting closed-loop performance is shown for the different disturbance cases in Figure 5.14. It is seen that the con-
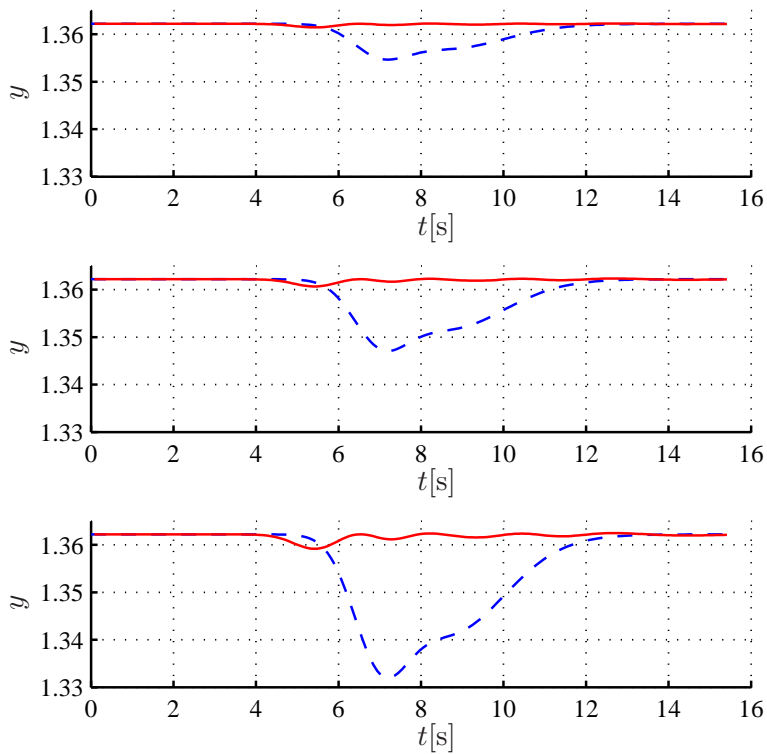

Figure 5.14: Uncontrolled (dashed) and controlled (solid) Mach number for Case 1 (top), Case 2 (middle) and Case 3 (bottom).

troller gives good performance in all three cases. There are, however, some minor oscillations in the closed-loop response, which are attributed to full model/reduced model mismatch and inexact modeling of the disturbance in the prediction model. Recall that we assume that the disturbance is constant over the prediction horizon, while it in fact increases or decreases, corresponding to the shape of the Gaussian distribution. Also, the horizon $\mathcal{M} = \mathcal{N} = 10$ is somewhat short, especially since there is an inverse response from inputs to output.

In order to guarantee feasibility of the MPC problem, we soften the constraints on the outputs.

If we again consider disturbance Case 3, we see from Figure 5.14 that the controlled Mach number falls below 1.36. Now, we set $y_{\min} = 1.36$ as a soft constraint, and penalize constraint violation with an exact penalty function. The resulting Mach number is compared to the simulation from Figure 5.14 which has a hard constraint $y_{\min} = 1.1$ in Figure 5.15. The corresponding control inputs are shown in Figure 5.16.

To further address the question of robustness, we add noise to the measured Mach number y. For that purpose we add Gaussian white noise of different intensities to the output of the CFD model during the simulation, and study the effect in closed loop.

Figure 5.17 shows a simulation run without noise, compared to three simulation runs with Gaussian white noise. It can be seen that in the presence of noise, particularly at the two lower levels, the controller performance remains good.
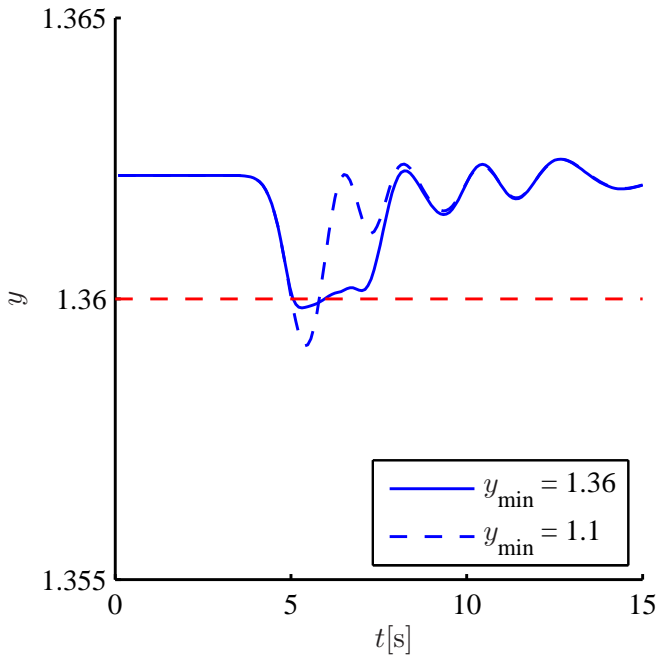
Figure 5.15: Mach number at inlet throat for two simulations with disturbance Case 3, with a soft constraint $y_{k+i} > 1.36$ and a hard constraint $y_{k+i} > 1.1$. The horizontal line indicates the soft lower bound for the soft-constrained case.
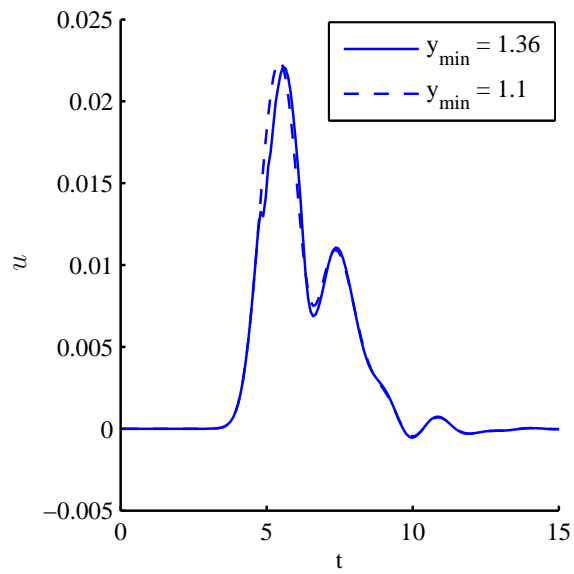
Figure 5.16: Control input for two simulations with disturbance Case 3, with a soft constraint $y_{\min} = 1.36$ and a hard constraint $y_{\min} = 1.1$.
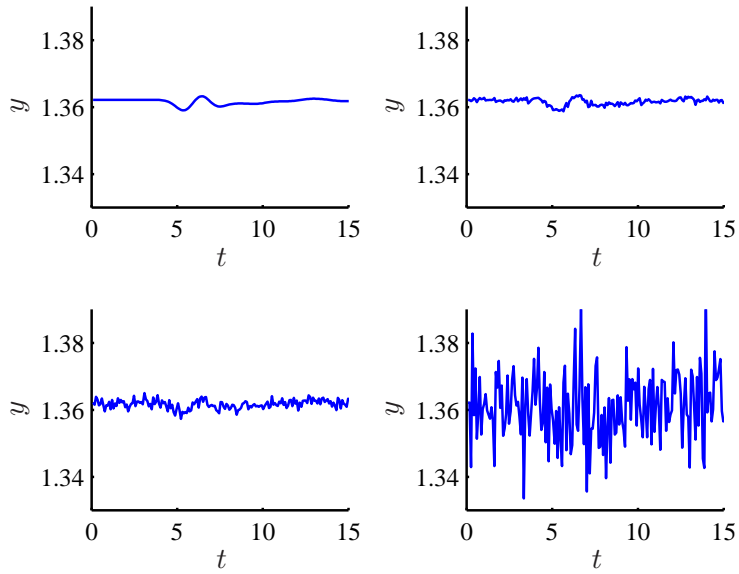
Figure 5.17: Controlled Mach number with measurement noise. Top left: No noise. Top right: Gaussian white noise of intensity 2.5×10^{-7}. Bottom left: Gaussian white noise of intensity 10^{-6}. Bottom right: Gaussian white noise of intensity 10^{-4}, corresponding to Mach number measurement accuracy within $\pm 0.01M$.

5.5   Concluding Remarks

This chapter presented a new framework for constrained optimal control of fast, large-scale systems, such as those arising in aerospace flow control applications. This is an important step towards achieving and actually implementing real-time, constrained optimal, control for such systems. The methodology, which combines eMPC with model reduction, is demonstrated for an example that considers control of a supersonic inlet. This example presents a significant challenge to model reduction methods. First, POD reduced models suffer from instability and thus cannot be used in a control setting. Further, obtaining models of very low dimensional is critical in order for the eMPC scheme to be viable for real-time control. Using a goal-oriented reduction methodology, we were able to derive a reduced model with ten states that yields acceptable approximation quality and is within the capacity of the eMPC scheme.

The proposed methodology is also applicable for more complicated control tasks, such as nonlinear MPC and reference tracking, for which the explicit solution of the MPC problem can still be found, although approximately, in some cases.

Chapter 6

Conclusions and Further Work

The results presented in this thesis are a step towards achieving advanced model-based real-time control for systems described by CFD-models. A framework is established for achieving constrained optimal control for large-scale systems with fast dynamics, through the use of model reduction, state estimation and low order controller design. Even with the considerable recent progress in model reduction to enable flow control, achieving real-time control in a constrained setting—which is crucial if these methods are to be adopted in actual systems—has not previously been possible. It is only the combination of recently developed model reduction methodology, along with state estimation and explicit model predictive control, that makes the approach feasible in this setting. To our knowledge, this is the first time that model reduction has been used in an explicit MPC setting to address the issue of constraints.

Moreover, it is demonstrated how model reduction techniques can significantly reduce the complexity of explicit model predictive control. This is essential, since it allows the control methodology to be applied for a larger number of systems, and for a wider range of controller parameters.

We develop a novel robust model predictive control design procedure which facilitates the design of model predictive control based on reduced-order models. The procedure guarantees closed-loop stability when the

reduced-order model predictive controller is attached to the high-fidelity model, and relies on solving a semi-definite program. The design uses the original plant model in an offline phase of determining cost function parameters, thereby making use of both the reduced-order model and the original model in the design. Since the main objective is to design an efficient *online* controller, it is reasonable to put some extra work in the offline phase. For large-scale systems, however, this procedure is too computationally demanding. Future work should investigate the possibility of treating parts of the dynamics as model uncertainty, or other ways to make the design applicable to larger systems.

More general stability analysis of closed-loop systems consisting of controllers based on reduced-order models of CFD-models should also be considered. In particular, stability of *explicit* MPC based on reduced-order models would be an interesting result. Moreover, development of model-based reduction methodology targeted at control applications for large-scale systems is needed. Many of the model reduction methods that are used frequently to design low order controllers, do not take into account the *outputs* of the system, but considers all states in the state space.

Further, model reduction of nonlinear systems entailed by reduced-order control is still very much an open research field. More rigorous methods are needed that are applicable to large-scale systems, and do not require an excessive amount of computations. Nonlinear control theory should then be applied, to achieve robust nonlinear control with low-order controllers.

Bibliography

V. M. Adamjan, D. Z. Arov, and M. G. Krein. Analytic properties of schmidt pairs for a Hankel operator and the generalized Schur-Takagi problem. *Math. USSR Sbornik*, 15(1):31–73, 1971.

K. Afanasiev and M. Hinze. Adaptive control of a wake flow using proper orthogonal decomposition. In J. Cagnol, M.P. Polis, and J.-P. Zolesio, editors, *Lecture Notes in Pure and Applied Mathematics*, volume 216 of *Shape Optimization and Optimal Design*, pages 317–332. CRC Press, 2001.

S. Ahuja, C. W. Rowley, I. G. Kevrekidis, M. Wei, T. Colonius, and G. Tadmor. Low-dimensional models for control of leading-edge vortices: Equilibria and linearized models. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, January 2007. AIAA Paper 2007-709.

V. Akcelik, J. Bielak, G. Biros, I. Epanomeritakis, A. Fernandez, O. Ghattas, E. J. Kim, J. Lopez, D. O'Hallaron, T. Tu, and J. Urbanic. High resolution forward and inverse earthquake modeling on terascale computers. In *Proceedings of the ACM/IEEE conference on Supercomputing*, 2003.

F. Allgöwer and A. Zheng, editors. *Nonlinear Model Predictive Control*. Birkhäuser, 2000.

J. D. Anderson. *Computational Fluid Dynamics*. McGraw-Hill International Editions, 1995.

A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. Advances in Design and Control, SIAM, Philadelphia, 2005a.

A. C. Antoulas. A new result on passivity preserving model reduction. *Systems and Control Letters*, 54(4):361–374, 2005b.

A. C. Antoulas, D. C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *V. Olshevsky (Ed.), Contemporary Mathematics*, 280:193–219, 2001.

A. C. Antoulas, D. C. Sorensen, and Y. Zhou. On the decay rate of Hankel singular values and related issues. *Systems & Control Letters*, 46(5):323–342, 2002.

P. Astrid and S. Weiland. POD based model approximation for an industrial glass feeder. In *16th IFAC World Congress*, Prague, 2005.

P. Astrid, L. Huisman, S. Weiland, and A. C. P. M. Backx. Reduction and predictive control design for a computational fluid dynamics model. In *Proc. 41st IEEE Conf. on Decision and Control*, volume 3, pages 3378–3383, Las Vegas, NV, 2002.

J. A. Atwell and B. B. King. Reduced order controllers for spatially distributed systems via proper orthogonal decomposition. *SIAM Journal on Scientific Computing*, 26(1):128–964, 2005.

J. A. Atwell, J. T. Borggaard, and B. B. King. Reduced order controllers for Burgers' equation with a nonlinear observer. *Applied Mathematics and Computational Science*, 11(6):1311–1330, 2001.

Z. Bai and R. W. Freund. A partial Padé-via-Lanczos method for reduced-order modeling. *Linear Algebra and Its Applications*, 332:139–164, 2001.

A. Bemporad and M. Morari. Robust model predictive control: A survey. In A. Garulli, A. Tesi, and A. Vicino, editors, *Robustness in Identification and Control*, number 245 in Lecture Notes in Control and Information Sciences, pages 207–226. Springer, 1999.

A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1): 3–20, 2002.

P. Benner. Model reduction at ICIAM '07. *SIAM News*, 40(8), October 2007.

P. Benner and J. Saak. A semi-discretized heat transfer model for optimal cooling of steel profiles. In P. Benner, V. Mehrmann, and D. C. Sorensen, editors, *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes in Computational Science and Engineering*, pages 353–356. Springer-Verlag, Berlin/Heidelberg, Germany, 2005.

P. Benner, E. S. Quintana-Ortí, and G. Quintana-Ortí. Balanced truncation model reduction of large-scale dense systems on parallel computers. *Mathematical and Computer Modelling of Dynamical Systems*, 6(4):383–405, 2000.

P. Benner, V. Mehrmann, and D.C. Sorensen, editors. *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin/Heidelberg, Germany, 2005.

D. S. Bernstein and W. M. Haddad. Lqg control with an $\mathcal{H}^\infty$ performance bound: a riccati equation approach. *IEEE Transactions on Automatic Control*, 34(3):293–305, 1989.

L. G. Bleris and M. V. Kothare. Real-time implementation of model predictive control. In *American Control Conference*, pages 4166–4171, 2005.

J. Borggaard. Optimal reduced-order modeling for nonlinear distributed parameter systems. In *American Control Conference*, 2006.

W. L. Briggs and S. F. McCormick. *A Multigrid Tutorial*. Society for Industrial Mathematics, 2000.

T. Bui-Thanh, K. E. Willcox, O. Ghattas, and B. van Bloemen Waanders. Goal-oriented, model-constrained optimization for reduction of large-scale systems. *Journal of Computational Physics*, 224:880–896, June 2007.

Y. Chahlaoui and P. van Dooren. A collection of benchmarks examples for model reduction of linear time invariant dynamical systems. SLICOT Working Note, 2002.

B. Codrons, P. Bendotti, C.M. Falinower, and M. Gevers. A comparison between model reduction and controller reduction: application to a PWR nuclear plant. volume 5, 1999.

K. Cohen, S. Siegel, J. Seidel, and T. McLaughlin. Reduced order modeling for closed-loop control of three dimensional wakes. In *3rd AIAA Flow Control Conference*, San Francisco, California, June 2006. AIAA Paper 2006-3356.

T.F. Coleman and Y. Li. An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6:418–445, 1996.

K.A. Evans. *Reduced Order Controllers for Distributed Parameter Systems*. PhD thesis, Virginia Tech., 2003.

J. H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer Verlag, 3rd edition, 2002.

E. G. Gilbert and K. T. Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9):1008–1020, 1991.

K. Glover. All optimal hankel-norm approximations of linear multivariable systems and their L_∞-error bounds. *International Journal of Control*, 39 (6):1115–1193, 1984.

D. W. Gu., B. W. Choi, and I. Postlethwaite. Low-order stabilizing controllers. *IEEE Transactions on Automatic Control*, 38(11):1713–1717, 1993.

D. W. Gu, B. W. Choi, and I. Postlethwaite. Low-order $\mathcal{H}_\infty$ sub-optimal controllers. In *Proceedings of the 12nd world congress of IFAC*, pages 347–350, 1993.

G. Gu. All optimal Hankel-norm approximations and their error bounds in discrete-time. *International Journal of Control*, 78(6):408–423, 2005.

S. Gugercin and A. C. Antoulas. A survey of model reduction by balanced truncation and some new results. *International Journal of Control*, 77 (8):748–766, 2004.

W. P. Heath and A. G. Wills. The inherent robustness of constrained linear model predictive control. In *16th IFAC World Congress*, Prague, 2005.

W. P. Heath, G. Li, A. G. Wills, and B. Lennox. IQC analysis of linear constrained MPC. In *IEEE sponsored Colloquium on Predictive Control*, University of Sheffield, 2005a.

W. P. Heath, A. G. Wills, and J. A. G. Akkermans. A sufficient stability condition for optimizing controllers with saturating actuators. *International Journal for Robust and Nonlinear Control*, 15(12):515–529, 2005b.

Ø. Hegrenæs, J. T. Gravdahl, and P. Tøndel. Spacecraft attitude control using explicit model predictive control. *Automatica*, 41(12):2107–2114, 2005.

M. Hinze and S. Volkwein. Proper orthognonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control. In P. Benner, V. Mehrmann, and D.C. Sorensen, editors, *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes in Computational Science and Engineering*, pages 261–306. Springer-Verlag, Berlin/Heidelberg, Germany, 2005.

S. Hovland. Soft constraints in explicit MPC. Master's thesis, Dept. of Engineering Cybernetics, Norwegian University of Science and Technology, 2004.

S. Hovland and J. T. Gravdahl. Order reduction and output feedback stabilization of an unstable CFD model. In *American Control Conference*, Minneapolis, June 14-16 2006a.

S. Hovland and J. T. Gravdahl. Complexity reduction in explicit MPC through model reduction. In *the 17th IFAC World Congress*, Seoul, Korea, July 2008.

S. Hovland and J. T. Gravdahl. Stabilizing a CFD model of an unstable system through model reduction and feedback control. *Modeling, Identification and Control*, 27(3):171–180, 2006b.

S. Hovland and J. T. Gravdahl. Stabilizing a CFD model of an unstable system through model reduction and feedback control. In *46th Scandinavian Conference on Simulation and Modeling*, Trondheim, Norway, October 2006c.

S. Hovland, K. E. Willcox, and J. T. Gravdahl. MPC for Large-Scale Systems via Model Reduction and Multiparametric Quadratic Programming. In *Proc. 45th IEEE Conf. on Decision and Control*, San Diego, CA, 2006.

S. Hovland, C. Løvaas, J. T. Gravdahl, and G. C. Goodwin. Stability of MPC based on reduced-order models. In *the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008a. IEEE.

S. Hovland, C. Løvaas, J. T. Gravdahl, and G. C. Goodwin. Stability of MPC based on reduced-order models. *Submitted to Control Engineering Practice*, 2008b.

S. Hovland, K. E. Willcox, and J. T. Gravdahl. Explicit model predictive control for large-scale systems via model reduction. *Journal of Guidance, Control, and Dynamics*, 31(4), August 2008c.

C. S. Hsu, X. Yu, H. H. Yeh, and S. S. Banda. $\mathcal{H}_\infty$ compensator design with minimal order observers. *IEEE Transactions on Automatic Control*, 39(8):1679–1681, 1994.

T.. Iwasaki and R. E. Skelton. All low order $\mathcal{H}_\infty$ controllers with covariance upper bound. In *Proceedings of the American Control Conference*, pages 2180–2184, San Francisco, CA, 1993.

T. A. Johansen, W. Jackson, R. Schreiber, and P. Tøndel. Hardware architecture design for explicit model predictive control. *American Control Conference*, pages 1924–1929, 2006.

E. Jonckheere and L. Silverman. A new set of invariants for linear systems: Application to reduced order compensator design. *IEEE Transactions on Automatic Control*, 28(10):953–964, 1983.

K. Karhunen. Zur spektral theorie stochasticher prozesse. *Ann Acad Sci Fennicae, Ser A1 Math Phys*, 34:1–7, 1946.

E. C. Kerrigan. Matlab invariant set toolbox. www-control.eng.cam.ac.uk/eck21/matlab/invsetbox/, 2005.

E. C. Kerrigan and J. M. Maciejowski. Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control. In *Proc. 39th IEEE Conf. on Decision and Control*, Sydney, 2000a.

E. C. Kerrigan and J. M. Maciejowski. Robust feasibility in model predictive control: Necessary and sufficient conditions. In *Proc. 40th IEEE Conf. on Decision and Control*, Orlando, FL, 2001.

E. C. Kerrigan and J.M. Maciejowski. Soft constraints and exact penalty functions in model predictive control. In *Proc. UKACC International Conference (Control 2000)*, 2000b.

D. Kristiansen and O. Egeland. Time and spatial discretization methods that preserve passivity properties for systems described by partial differential equations. *Proceedings of the American Control Conference*, 2: 1255–1259, 2000. doi: 10.1109/ACC.2000.876701.

K. Kunisch and S. Volkwein. Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition. *Journal of Optimization Theory and Applications*, 102(2):345–371, August 1999.

K. Kunisch and S. Volkwein. Proper orthogonal decomposition for optimality systems. Technical Report 10, Institute for Mathematics and Scientific Computing, University of Graz, 2006.

H. Kwakernaak and R. Sivan. *Linear optimal control systems*. Wiley-Interscience, New York, 1972.

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

J. Li and J. White. Low rank solution of Lyapunov equations. *Journal on Matrix Analysis and Applications*, 24(1):260–280, 2002.

A. Linnemann. Existence of controllers stabilizing the reduced-order model and not the plant. *Automatica*, 24(5):719–719, 1988.

M. Loève. Functiona aleatoire de second ordre. *Revue Science*, 84:195–206, 1946.

J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. URL `http://control.ee.ethz.ch/~joloef/yalmip.php`.

C. Løvaas. *Dissipativity, Optimality and Robustness of Model Predictive Control Policies*. PhD thesis, School of Electrical Engineering and Computer Science, University of Newcastle, 2008.

C. Løvaas, M. M. Seron, and G. C. Goodwin. A dissipativity approach to robustness in constrained model predictive control. In *the 46th IEEE Conference on Decision and Control*, 2007a.

C. Løvaas, M. M. Seron, and G. C. Goodwin. Robust model predictive control of input-constrained stable systems with unstructured uncertainties. In *Proc. European Control Conference*, 2007b.

C. Løvaas, M. M. Seron, and G. C. Goodwin. Robust output-feedback model predictive control for systems with unstructured uncertainty. *Automatica*, 44(8):1933–1943, 2008a.

C. Løvaas, M. M. Seron, and G. C. Goodwin. Robust output-feedback MPC with soft state constraints. In *17th IFAC World Congress*, Seoul, Korea, July 2008b.

J. L. Lumley. The structure of inhomogeneous turbulent flow. In *Atmospheric Turbulence and Wave Propagation*, 1967.

J. M. Maciejowski. *Predictive Control with Constraints*. Pearson Education, 2001.

D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6): 789–814, 2000.

M. Meyer and H. G. Matthies. Efficient model reduction in non-linear dynamics using the karhunen-loève expansion and dual-weighted-residual methods. *Computational Mechanics*, 31:179–191, 2003.

B. C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. on Automatic Control*, 26(1):17–32, 1981.

K. R. Muske and J. B. Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2):262–287, 1993. doi: 10.1002/aic.690390208.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Verlag, 1999.

G. Obinata and B. D.O. Anderson. *Model Reduction for Control System Design*. Applied Mechanics Reviews. Springer, 2001.

G. Pannocchia, J. B. Rawlings, and S. J. Wright. Fast, large-scale model predictive control by partial enumeration. *Automatica*, 43(5):852–860, 2007.

S. Prajna. POD model reduction with stability guarantee. In *Proc. 42nd IEEE Conf. on Decision and Control*, Hawaii, 2003.

J. T. Pukrushpan, A. G. Stefanopoulou, and H. Peng. Control of fuel cell breathing: Initial results on the oxygen starvation problem. *IEEE Control Systems Magazine*, 24(2):30–46, 2004.

S. S. Ravindran. A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids*, 34(5):425–448, 2000.

J. B. Rawlings. Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, 20(3):38–52, 2000.

J.M.A. Scherpen. Balancing for nonlinear systems. *Systems & Control Letters*, 21(2):143–153, 1993.

P. O. M. Scokaert and J. B. Rawlings. Feasibility issues in linear model predictive control. *AIChE Journal*, 45(8):1649–1659, 1999.

D. C. Sorensen. Passivity preserving model reduction via interpolation of spectral zeros. *Systems & Control Letters*, 54(4):347–360, 2004.

D. C. Sorensen and A. C. Antoulas. The Sylvester equation and approximate balanced reduction. *Linear Algebra and Its Applications*, 351–352:671–700, 2002.

E. Storkaas, S. Skogestad, and J.-M. Godhavn. A low-dimensional dynamic model of severe slugging for control design and analysis. In *Proceedings of MultiPhase*, San Remo, Italy, June 2003.

J. F. Sturm. Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1):625–653, 1999.

J. F. Thompson, B. Soni, and N. P. Weatherrill. *Handbook of Grid Generation*. CRC Press, 1998.

P. Tøndel. *Constrained Optimal Control via Multiparametric Quadratic Programming*. PhD thesis, Department of Engineering Cybernetics, Norw. University of Sci. and Techn., 2003.

P. Tøndel and T. A. Johansen. Complexity reduction in explicit linear model predictive control. In *XV IFAC World Congress*, 2002.

P. Tøndel, T. A. Johansen, and A. Bemporad. Computation of piecewise affine control via binary search tree. *Automatica*, 39(5):945–950, 2003.

J. Vada, O. Slupphaug, T. A. Johansen, and B. A. Foss. Linear MPC with optimal prioritized infeasibility handling: Application, computational issues and stability. *Automatica*, 37:1835–1843, 2001.

H. K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics*. Longman, 1995.

P. Wesseling. *Principles of Computational Fluid Dynamics*. Springer, 2001.

K. E. Willcox and G. Lassaux. Model reduction of an actively controlled supersonic diffuser. In P. Benner, V. Mehrmann, and D. C. Sorensen, editors, *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes in Computational Science and Engineering*, pages 357–361. Springer-Verlag, Berlin/Heidelberg, 2005.

K. E. Willcox and A. Megretski. Fourier series for accurate, stable, reduced-order models in large-scale applications. *SIAM Journal for Scientific Computing*, 26(3):944–962, 2005.

P. M. R. Wortelboer, M. Steinbuch, and O. H. Bosgra. Iterative model and controller reduction using closed-loop balancing, with application to a compact disc mechanism. *Int. J. Robust Nonlinear Control*, 9(3):123–142, 1999.

K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1996.