Linköping Studies in Science and Technology. Dissertations.
No. 812

Minimax approaches to robust model predictive control

Johan Löfberg

Department of Electrical Engineering
Linköping University, SE–581 83 Linköping, Sweden

Linköping 2003

Minimax approaches to robust model predictive control

johanl@isy.liu.se
http://www.control.isy.liu.se
Department of Electrical Engineering
Linköping University
SE–581 83 Linköping
Sweden

ISBN 91-7373-622-8 ISSN 0345-7524

Printed by UniTryck, Linköping, Sweden 2003

D'oh!

Abstract

Controlling a system with control and state constraints is one of the most important problems in control theory, but also one of the most challenging. Another important but just as demanding topic is robustness against uncertainties in a controlled system. One of the most successful approaches, both in theory and practice, to control constrained systems is model predictive control (MPC). The basic idea in MPC is to repeatedly solve optimization problems on-line to find an optimal input to the controlled system. In recent years, much effort has been spent to incorporate the robustness problem into this framework.

The main part of the thesis revolves around minimax formulations of MPC for uncertain constrained linear discrete-time systems. A minimax strategy in MPC means that worst-case performance with respect to uncertainties is optimized. Unfortunately, many minimax MPC formulations yield intractable optimization problems with exponential complexity.

Minimax algorithms for a number of uncertainty models are derived in the thesis. These include systems with bounded external additive disturbances, systems with uncertain gain, and systems described with linear fractional transformations. The central theme in the different algorithms is semidefinite relaxations. This means that the minimax problems are written as uncertain semidefinite programs, and then conservatively approximated using robust optimization theory. The result is an optimization problem with polynomial complexity.

The use of semidefinite relaxations enables a framework that allows extensions of the basic algorithms, such as joint minimax control and estimation, and approximation of closed-loop minimax MPC using a convex programming framework. Additional topics include development of an efficient optimization algorithm to solve the resulting semidefinite programs and connections between deterministic minimax MPC and stochastic risk-sensitive control.

The remaining part of the thesis is devoted to stability issues in MPC for continuous-time nonlinear unconstrained systems. While stability of MPC for unconstrained linear systems essentially is solved with the linear quadratic controller, no such simple solution exists in the nonlinear case. It is shown how tools from modern nonlinear control theory can be used to synthesize finite horizon MPC controllers with guaranteed stability, and more importantly, how some of the technical assumptions in the literature can be dispensed with by using a slightly more complex controller.

Acknowledgments

First of all, my deepest gratitude goes to my supervisors Professor Torkel Glad and Professor Lennart Ljung for their guidance and support. My journey towards this thesis has not been a straight path, but Torkel and Lennart have kept their trust in me during this five-year period. For this, I am utmost grateful.

Spending five years on a thesis would have been a miserable experience, if it were not for the excellent working conditions and atmosphere in the group. No question is too stupid and everyone is keen on giving a helping hand. However, some persons have probably had more than their fair share of strange questions. Special thanks go to Ragnar Wallin, Måns Östring and Dr. Fredrik Tjärnström. Their knowledge in everything from Mexican painting, water-starts and fish-fermentation to convex optimization, LaTeX and linear algebra has been invaluable.

Ragnar Wallin, Thomas Schön, Jonas Gillberg, Daniel Axehill and Johan Sjöberg have proofread various parts of the thesis. Their comments were of great help and have hopefully made this thesis a bit more readable.

The research was supported within the Vinnova competence center ISIS (Information Systems for Industrial Control and Supervision), which is gratefully acknowledged.

Finally, friends and family. I guess you all are a bit disappointed now. From my complete absence during the last months, I guess you expected a 500-page thesis. Sorry, this is all I could come up with. Anyway, thank you for all support! Now, I hope you do me the honor and keep reading this thesis...

Linköping, March 2003

Johan Löfberg

Contents

Notation

Operators and functions

$A \succeq (\succ)\ 0$	A positive (semi)definite matrix, $x^T Ax \geq (>)\ 0\ \forall x \neq 0$
$A \preceq (\prec)\ 0$	A negative (semi)definite matrix, $x^T Ax \leq (<)\ 0\ \forall x \neq 0$
A^T	Transpose of a matrix
A^{-1}	Inverse of a matrix
$\mathrm{Tr}A$	Trace of a matrix
$\det A$	Determinant of a matrix
$\mathrm{diag}(A)$	Vector with diagonal of matrix
$A \oplus B$	Direct sum of matrices (block diagonal concatenation)
$A \circ B$	Hadamard product of matrices (element-wise product)
$\|\|A\|\|$	Induced 2-norm, $\max_{\|\|x\|\|=1} x^T Ax$
$\mathrm{diag}(x)$	Diagonal matrix $\oplus_{i=1}^n x_i$
$\|x\|$	Element-wise absolute value
$\|\|x\|\|$	Euclidean norm of vector, $\|\|x\|\| = \sqrt{x^T x} = \sqrt{\sum_{i=1}^n x_i^2}$
$\|\|x\|\|_2$	Euclidean norm of vector
$\|\|x\|\|_1$	Sum of absolute elements, $\|\|x\|\|_1 = \sum_{i=1}^n \|x_i\|$
$\|\|x\|\|_\infty$	Largest element $\max_i \|x_i\|$

Sets

$\mathbb{X} \times \mathbb{Y}$	Cartesian product, $\mathbb{X} \times \mathbb{Y} = \{(x, y) : x \in \mathbb{X}, y \in \mathbb{Y}\}$
$\mathbb{R}$ ($\mathbb{R}_+$)	Set of (non-negative) real numbers
$\mathbb{R}^n$ ($\mathbb{R}^n_+$)	Set of (non-negative) real vectors with n elements
$\mathbb{R}^{n\times m}$	Set of real matrices with n rows and m columns
$\mathbb{E}_P$	Ellipsoidal set $\mathbb{E}_P = \{x : x^T P x \leq 1\}$ $(P \succ 0)$
$\mathbf{Co}\{X^{(1)}, \ldots, X^{(n)}\}$	Convex hull

Others

$\mathbf{E}$	Mathematical expectation
I	Identity matrix
$\mathbf{1}$	Vector of ones, $1 = \begin{pmatrix}1 & 1 & \ldots & 1\end{pmatrix}^T$
$\mathbf{1}^N$	Vector of N ones
e_i	ith unit vector, $I = \begin{pmatrix}e_1 & e_2 & \ldots & e_n\end{pmatrix}$

Abbreviations

BMI	Bilinear Matrix Inequality
CLF	Control Lyapunov Function
FIR	Finite Impulse Response
HJB	Hamilton-Jacobi-Bellman
KKT	Karush-Kuhn-Tucker
LMI	Linear Matrix Inequality
LP	Linear Program(ming)
LQ	Linear Quadratic
MAXDET	Determinant Maximization
MPC	Model Predictive Control
PDF	Probability Density Function
QP	Quadratic Program(ming)
SDP	Semidefinite Program(ming)
SOCP	Second Order Cone Program(ming)

1

Introduction

Incorporating uncertainty models in model predictive control (MPC) is the main topic of this thesis. The goal is to develop schemes that transparently extend nominal MPC to deal with uncertainty.

The prevailing approach today to incorporate uncertainty in MPC is to optimize worst-case performance. In optimization language, this corresponds to solving minimax problems. Unfortunately, these minimax problems tend to be intractable for many important problem formulations. The main idea in the thesis is to use methods from robust semidefinite optimization theory to solve conservative approximations of these intractable problems.

1.1 Outline

The thesis starts with a number of introductory chapters on MPC, convex and robust optimization, and robust MPC in Chapter 2, 3 and 4 respectively. The main contributions on minimax MPC can be found in Chapters 5 to 11.

Chapters 5 and 6 present methods to deal with external bounded disturbances and state estimation errors in minimax MPC. Semidefinite relaxations are introduced as a tool to solve conservative approximations of minimax MPC problems. The reader is advised to at least skim through Chapter 5, since much of the introduced notation is used on several locations later in the thesis.

The results from Chapter 5 are improved upon in Chapter 7 where a novel approach to deal with conservativeness is introduced. The problem with standard minimax MPC schemes is that they typically deal with open-loop formulations which easily leads to conservative controllers. An extension to minimax MPC that resolves this problem is closed-loop minimax MPC. However, closed-loop minimax MPC is a much harder problem, and leads to completely intractable problems. The result in Chapter 7 can be interpreted as a way to approximate closed-loop minimax MPC.

Chapter 8 presents a short result on a connection between the developed deterministic minimax MPC algorithms and stochastic risk-sensitive control.

The semidefinite relaxations employed in Chapter 5 and 6 give rise to large sparse and structured semidefinite programs. A dedicated solver is developed in Chapter 9 to solve these optimization problems. It is shown that a very simple solver can yield substantial improvements in computational performance, compared to available software, by exploiting the structure.

Chapters 10 and 11 are devoted to systems with uncertainty in the dynamic model, instead of external disturbances. A rather general approach to minimax MPC for systems described with linear fractional transformations is developed in Chapter 11. Although convex and polynomially growing in size, the semidefinite relaxation of the minimax MPC problem in Chapter 11 suffer from complexity problems. A specialization to systems with uncertain gain is therefore developed independently in Chapter 10. The result is a much more efficient formulation of the minimax MPC problem.

Chapter 12 summarizes the material on minimax MPC and points at some possible future extensions and research directions.

Finally, a completely different problem is addressed in Chapter 13. MPC can readily be extended to cope with nonlinear systems, at least conceptually. The stability theory in linear MPC transfers nicely to nonlinear systems, but the results are not as constructive since they require knowledge of a stabilizing controller with certain prescribed properties. It is shown how some of these requirements can be relaxed, thus making synthesis of nonlinear MPC more tractable.

1.2 Contributions

To summarize, the main contributions of the thesis are:

- Semidefinite relaxations are advocated as a tool to solve conservative approximations of minimax MPC problems with finite horizon quadratic performance measures.

- The semidefinite relaxation of minimax MPC for systems with bounded additive external disturbances in Chapter 5.

- The semidefinite relaxation of minimax MPC for systems with uncertainties described with a linear fractional transformation (LFT model) in Chapter 11.

- Introduction of a novel approach in Chapter 7 to deal with conservativeness in minimax MPC. Closed-loop minimax MPC is approximated by parameterizing the control sequence in a novel way.
- Specialization of the results on minimax MPC for LFT models, to systems with uncertain gain. The result in Chapter 10 is an optimization problem with much better complexity than the general solution.
- It is shown in Chapter 6 that a joint estimation and minimax MPC problem can be written as a semidefinite program with a bilinear matrix inequality.
- It is shown in Chapter 8 that there are close connections between a semidefinite relaxation of a minimax MPC problem and stochastic risk-sensitive control.
- A new approach to synthesize MPC controllers with guaranteed stability in nonlinear continuous-time systems is presented in Chapter 13.

1.3 Publications

Parts of the thesis are based on previously published material. Chapters 5, 6 and 7 are based on

Löfberg, J. (2001a). Linear model predictive control: Stability and robustness. Licentiate thesis LIU-TEK-LIC-2001:03, Department of Electrical Engineering, Linköpings universitet, Sweden.

Löfberg, J. (2002b). Towards joint state estimation and control in minimax MPC. In *Proceedings of the 15th IFAC World Congress on Automatic Control*, Barcelona, Spain.

Löfberg, J. (2003). Approximations of closed-loop minimax MPC. Submitted to CDC03.

Chapter 10 builds upon

Löfberg, J. (2002a). Minimax MPC for systems with uncertain gain. In *Proceedings of the 15th IFAC World Congress on Automatic Control*, Barcelona, Spain.

The main result in Chapter 13 is based on

Löfberg, J. (2001b). Nonlinear receding horizon control: Stability without stabilizing constraints. In *Proceedings of the European Control Conference ECC01*, Porto, Portugal.

A related publication is

Löfberg, J. (2000). Backstepping with local LQ performance and global approximation of quadratic performance. In *Proceedings of the American Control Conference 2000*, Chicago, Illinois.

Part I

Background

2

Model Predictive Control

Model predictive control, or MPC, is a control paradigm with a motley background. The underlying ideas for MPC originated already in the sixties as a natural application of optimal control theory. Already in (Propoi, 1963), a controller with close connections to MPC was developed, and a more general optimal control based feedback controller was discussed in (Lee and Markus, 1968)

"*One technique for obtaining a feedback controller synthesis from knowledge of open-loop controllers is to measure the current control process state and then compute very rapidly for the open-loop control function. The first portion of this function is then used during a short time interval, after which a new measurement of the process state is made and a new open-loop control function is computed for this new measurement. The procedure is then repeated*".

As we will see in this and the following chapters, this is the definition of the control method that we today call MPC.

In this chapter, we will give a short historical account of MPC and describe the basics of an MPC algorithm. The admissible systems will be defined and some simple notation will be explained. After the introduction of a standard MPC controller, we show how the obtained problems can be solved with quadratic programming. The chapter ends with a short introduction to stability issues in nominal MPC.

2.1 Historical Background

The true birth of MPC took place in the industry in the mid-seventies to mid-eighties. Advocated by the work on Model Predictive Heuristic Control (MHRC) (Richalet et al., 1978) and Dynamic Matrix Control (DMC) (Cutler and Ramaker, 1980), the MPC strategy became popular in the petro-chemical industry. During this period, there was a flood of new variants of MPC. Without going into details, MAC, DMC, EHAC, EPSAC, GMV, MUSMAR, MURHAC, PFC, UPC and GPC were just some of the algorithms (Camacho and Bordons, 1998). Despite the vast number of abbreviations introduced, not much differed between the algorithms. Typically, they differed in the process model (impulse, step, state-space etc.), disturbance (constant, decaying, filtered white noise etc), and adaptation to time varying models.

During the nineties, the theory of MPC has matured substantially. The main reason is probably the use of state-space models instead of input-output models. This has simplified, unified and generalized much of the theory. In the case of non-accessible states, the Kalman filter (most easily used in a state-space formulation) simplifies the estimation part, the connections to linear quadratic control give a lot of insight (Bitmead et al., 1990), stability theory is almost only possible in a state-space formulation and much recent MPC theory is based on linear matrix inequalities which are most suitable for state-space methods.

2.2 System Setup

In this thesis, we will exclusively use state-space methods. The system we control will in principle be the same throughout the thesis, a linear discrete-time system

$$x_{k+1} = Ax_k + Bu_k \tag{2.1a}$$

$$y_k = Cx_k \tag{2.1b}$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, $y_k \in \mathbb{R}^p$ denote the state, control input and controlled output respectively.

Besides the dynamics, the system has control and state constraints.

$$u \in \mathbb{U} \tag{2.2a}$$

$$x \in \mathbb{X} \tag{2.2b}$$

The sets $\mathbb{U}$ and $\mathbb{X}$ are polyhedrons, i.e., described by linear inequalities.

$$\mathbb{U} = \{u : E_u u \leq f_u\} \tag{2.3a}$$

$$\mathbb{X} = \{x : E_x x \leq f_x\} \tag{2.3b}$$

2.3 A Basic MPC Controller

MPC is an optimization based control law, and the performance measure is almost always quadratic, i.e., based on ℓ_2-norms. By using a quadratic performance measure, connections to linear quadratic control are evident, efficient optimization problems arise (quadratic programming), and mathematical analysis simplifies. A quadratic performance measure is also intuitively what we want from a control perspective, it is much more important to reduce large deviations than reducing small deviations.

By defining positive definite matrices $Q = Q^T \succ 0$ and $R = R^T \succ 0$ (the performance weights), our underlying goal is to find the optimal control input that minimizes an infinite horizon performance measure.

$$J_k = \sum_{j=0}^{\infty} y_{k+j}^T Q y_{k+j} + u_{k+j}^T R u_{k+j} \tag{2.4}$$

The positive definiteness assumptions on Q and R can in most cases be relaxed to semidefiniteness, but we refrain from this to obtain a simple notation.

An alternative is a performance measure based on ℓ_1-norms or ℓ_∞-norms.

$$J_k = \sum_{j=0}^{\infty} ||Q^{1/2} y_{k+j}||_1 + ||R^{1/2} u_{k+j}||_1 \tag{2.5a}$$

$$J_k = \sum_{j=0}^{\infty} ||Q^{1/2} y_{k+j}||_\infty + ||R^{1/2} u_{k+j}||_\infty \tag{2.5b}$$

The common ingredient in these two performance measures is that they can be dealt with using linear programming. They are therefore sometimes called linear performance measures. However, as indicated in (Rao and Rawlings, 2000), strange closed-loop behavior can in some cases be obtained with these performance measures.

The solution to the infinite horizon problem with the quadratic performance measure is in the unconstrained case given by the linear quadratic (LQ) controller (Anderson and Moore, 1971). In the general constrained case, there does not exist any simple closed-form expression for the solution, although it can be shown that the solution is a piece-wise affine state feedback law (Bemporad et al., 2002b). Instead, the first step in MPC is to define a prediction horizon N and approximate the performance measure by using a finite horizon,

$$J_k = \sum_{j=0}^{N-1} y_{k+j|k}^T Q y_{k+j|k} + u_{k+j|k}^T R u_{k+j|k} \tag{2.6}$$

The variables $y_{k+j|k}$ denote predicted outputs, given an input sequence $u_{k+j|k}$, a state estimate $x_{k|k}$ and the model (2.1). It is assumed throughout this thesis, except in Chapter 6, that the true state is available, hence $x_{k|k} = x_k$.

The term finite horizon is crucial. Due to the finite horizon, we are able to minimize the performance measure since it is an optimization problem with a finite number of decision variables and a finite number of constraints, but at the same time, the finite horizon introduce stability problems.

The second idea is to repeatedly resolve finite horizon problems when we obtain new measurements $x_{k|k}$ of the state x_k. Let us define this finite horizon optimal control problem.

$$\boxed{\begin{array}{rlcl} \min\limits_{u} & \sum_{j=0}^{N-1} y_{k+j|k}^T Q y_{k+j|k} + u_{k+j|k}^T R u_{k+j|k} & & \\ \text{subject to} & u_{k+j|k} & \in & \mathbb{U} \\ & x_{k+j|k} & \in & \mathbb{X} \end{array}} \tag{2.7}$$

Putting the conceptual idea into an algorithm yields the following basic MPC controller

---

Algorithm 2.1 (Basic MPC controller)

1. Measure $x_{k|k}$
2. Obtain $u_{\cdot|k}$ by solving a finite horizon optimal control problem (e.g. (2.7))
3. Apply the first element $u_k = u_{k|k}$

The difference between different MPC schemes is the finite horizon optimal control problem used in step 2.

2.4 Quadratic Programming Formulation of MPC

Already in (Propoi, 1963), it was realized that the optimization problem (2.7) is a quadratic program. The earliest reference that takes advantage of this fact in an MPC context is probably (Garcia and Morshedi, 1986), although it had been in use in the industry for quite some time before. Quadratic programming is a classical optimization problem for which there exist efficient solution methods, and this is probably one of the reasons why MPC has become so popular in practice.

To put the optimization problem in a form suitable for quadratic programming, we introduce stacked vectors with future outputs, states and control inputs

$$Y = \begin{pmatrix} y_{k|k}^T & y_{k+1|k}^T & \cdots & y_{k+N-1|k} \end{pmatrix}^T \tag{2.8a}$$

$$X = \begin{pmatrix} x_{k|k}^T & x_{k+1|k}^T & \cdots & x_{k+N-1|k} \end{pmatrix}^T \tag{2.8b}$$

$$U = \begin{pmatrix} u_{k|k}^T & u_{k+1|k}^T & \cdots & u_{k+N-1|k}^T \end{pmatrix}^T \tag{2.8c}$$

The predicted states and outputs can be conveniently written as

$$Y = \mathcal{C}X \tag{2.9a}$$

$$X = \mathcal{A}x_{k|k} + \mathcal{B}U \tag{2.9b}$$

where $\mathcal{A} \in \mathbb{R}^{Nn\times n}$, $\mathcal{B} \in \mathbb{R}^{Nn\times Nm}$ and $\mathcal{C} \in \mathbb{R}^{Np\times Nn}$ are given by

$$\mathcal{A} = \begin{pmatrix} I \\ A \\ A^2 \\ \vdots \\ A^{N-1} \end{pmatrix}, \quad \mathcal{B} = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ B & 0 & 0 & \dots & 0 \\ AB & B & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A^{N-2}B & \dots & AB & \mathcal{B} & 0 \end{pmatrix}, \quad \mathcal{C} = \oplus_{j=1}^{N} C \tag{2.10}$$

The operator $\oplus$ (direct sum) will be used throughout this thesis to define block diagonal matrices, as in the definition of $\mathcal{Q} \in \mathbb{R}^{Nn\times Nn}$ and $\mathcal{R} \in \mathbb{R}^{Nm\times Nm}$.

$$\mathcal{Q} = \oplus_{j=1}^{N} Q = \begin{pmatrix} Q & 0 & \dots & 0 \\ 0 & Q & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & Q \end{pmatrix}, \quad \mathcal{R} = \oplus_{j=1}^{N} R = \begin{pmatrix} R & 0 & \dots & 0 \\ 0 & R & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & R \end{pmatrix} \tag{2.11}$$

Since the constraints are defined by linear inequalities, they can be written as $U \in \mathbb{U}^N = \{U : \mathcal{E}_u U \leq \mathcal{F}_u\}$ and $X \in \mathbb{X}^N = \{X : \mathcal{E}_x X \leq \mathcal{F}_x\}$ where

$$\mathcal{E}_u = \oplus_{j=1}^{N} E_u, \quad \mathcal{F}_u = \begin{pmatrix} f_u^T & f_u^T & \dots & f_u^T \end{pmatrix}^T \tag{2.12a}$$

$$\mathcal{E}_x = \oplus_{j=1}^{N} E_x, \quad \mathcal{F}_x = \begin{pmatrix} f_x^T & f_x^T & \dots & f_x^T \end{pmatrix}^T \tag{2.12b}$$

The optimization problem (2.7) can now be written as

$$\boxed{\begin{array}{rl} \min\limits_{U} & Y^T \mathcal{Q} Y + U^T \mathcal{R} U \\ \text{subject to} & \mathcal{E}_u U \leq \mathcal{F}_u \\ & \mathcal{E}_x X \leq \mathcal{F}_x \end{array}} \tag{2.13}$$

Inserting the definition of X and simplifying yields the final quadratic program in the decision variable U

$$\boxed{\begin{array}{rl} \min\limits_{U} & U^T(\mathcal{A}^T \mathcal{C}^T \mathcal{Q} \mathcal{C}^T \mathcal{A} + \mathcal{R})U + 2U^T \mathcal{B}^T \mathcal{Q} \mathcal{C} \mathcal{A} x_{k|k} \\ \text{subject to} & \begin{pmatrix} \mathcal{E}_u \\ \mathcal{E}_x \mathcal{B} \end{pmatrix} U \leq \begin{pmatrix} \mathcal{F}_u \\ \mathcal{F}_x - \mathcal{E}_x \mathcal{A} x_{k|k} \end{pmatrix} \end{array}} \tag{2.14}$$

For small to medium-scale systems with reasonably long sample-time, standard quadratic programming algorithms (Nocedal and Wright, 1999) will probably suffice to solve the on-line optimization problems, but if performance is crucial, there exist algorithms that can exploit structure in large-scale MPC problems to reduce the computational complexity (Wright, 1997).

Notice that if the problem is unconstrained, we can easily find the optimal solution analytically.

$$U = -(\mathcal{A}^T \mathcal{C}^T \mathcal{Q} \mathcal{C}^T \mathcal{A} + \mathcal{R})^{-1} \mathcal{B}^T \mathcal{Q} \mathcal{C} \mathcal{A} x_{k|k} \tag{2.15}$$

By isolating the first m rows in this solution, we see that we can write the optimal control law as a linear time-invariant state feedback.

$$u_k = Lx_{k|k} \tag{2.16}$$

Of course, this result is rather irrelevant, since there is no reason to apply finite horizon MPC with a quadratic performance measure to an unconstrained linear system. A better solution is to let $N = \infty$ and calculate the optimal LQ controller. However, we will use this unconstrained solution now when we discuss stability of MPC.

2.5 Stability of MPC

From a theoretical point of view, the main problem with linear MPC has been, and maybe still is, the lack of a general and unifying stability theory.

So why is stability theory for linear MPC still a problem, 25 years after the first development of MPC? To begin with, we have to keep in mind that an unstable input-constrained system cannot be globally stabilized (Saberi et al., 2000). Hence, all results have to be local in the general case. Furthermore, although the system we analyze is linear, the constraints introduce nonlinearities, which complicate the stability analysis. Another problem is that the control law is generated by the solution of an optimization problem. These obstacles prevented the development of stability results in the early days of MPC. The situation was even more complicated by the fact that the analysis often was performed in an input-output setting. As state-space formulations became standard in MPC, stability results began appearing in late eighties and early nineties.

The central concept that started to appear was to abandon the idea to analyze the impact of different choices of the parameters Q, R and N, since these parameters in general affect stability in a complicated way (see example below).

Instead, the main trend now is to reformulate the underlying optimization problem in order to guarantee stability for arbitrary Q, R and N. An excellent survey on recent stability theory for MPC can be found in (Mayne et al., 2000).

To motivate the theory in this section, let us start with a simple numerical study to illustrate the problems with stability in MPC.

Example 2.1 (Unstable MPC)
Consider the following unstable system[1]

$$\begin{aligned} x_{k+1} &= \begin{pmatrix} 1.216 & -0.055 \\ 0.221 & 0.9947 \end{pmatrix} x_k + \begin{pmatrix} 0.02763 \\ 0.002673 \end{pmatrix} u_k \\ y_k &= \begin{pmatrix} 0 & 0.2 \end{pmatrix} x_k \end{aligned}$$

Let us assume that the system is unconstrained, and that we wish to control the system using the standard MPC controller defined in Section 2.3. From Section

[1]Zero-order hold discretization of $\frac{0.1}{(s-1)^2}$, sampled at 0.1 seconds

2.4, we know that the resulting MPC controller is a linear feedback $u_k = Lx_k$. This means that we can analyze stability by calculating the eigenvalues of $A + BL$.

The optimal feedback matrix L depends on the weights Q and R, and the prediction horizon N. To study the impact of these variables, we generate a large number of MPC controllers using

$$Q = I, \quad R = 0, 0.01, \ldots, 10, \quad N = 1, 2, \ldots, 50$$

Stability analysis can now easily be performed by calculating the eigenvalues of $A+BL$ for each combination of N and R. The results are illustrated in Figure 2.1.

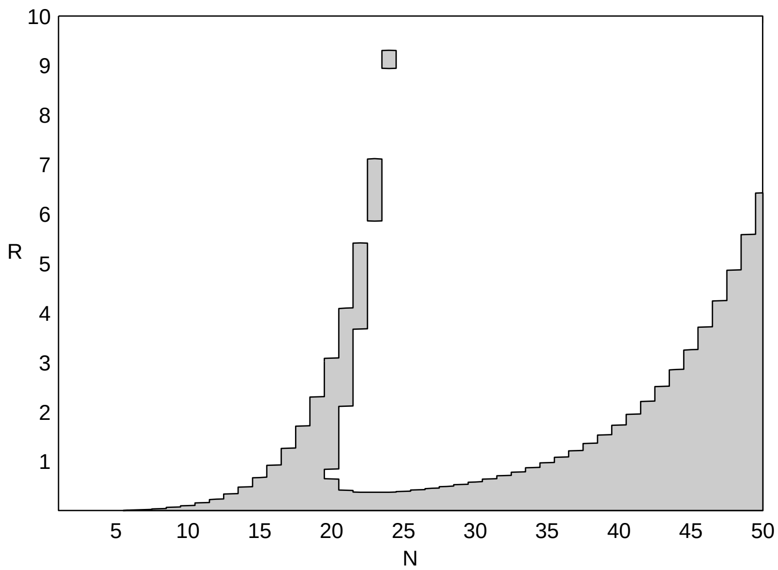

Figure 2.1: The figure shows the effect of the prediction horizon N and the control weight R. Grey regions indicate combinations of N and R that generate stabilizing MPC controllers. White regions indicate unstable control laws. The figure shows that the set of stabilizing controllers for this example is nonconvex in N and R

Combinations of N and R that give a stable closed-loop system, are indicated in the figure with grey regions, whereas combinations yielding an unstable closed-loop are white. As an example, if we look at the slice $R = 1$, the closed-loop is stable for $17 \leq N \leq 20$ and $N \geq 36$. Similarly, for $N = 22$, the system is stable for $R \leq 0.37$ and $3.68 \leq R \leq 5.41$.

The numbers above are not that interesting, but with this simple example, we have numerically found an essential property; the set of stabilizing combinations of Q, N and R can in the general case be nonconvex and disconnected.

2.5.1 A Stabilizing MPC Controller

In this section, a rather general approach to stabilizing MPC will be introduced. The method that follows is based on three ingredients: a nominal controller, a terminal state domain defined by a terminal state constraint, and a terminal state weight. Having these, it is possible to summarize many proposed schemes in the following theorem (Mayne et al., 2000).

Theorem 2.1 (Stabilizing MPC)
Suppose the following assumptions hold for a linear system, a nominal controller $L(x)$, a terminal state domain $\mathbb{X}_T$ and a terminal state weight $\Psi(x)$

A1. $x_{k+1} = Ax_k + Bu_k$

A2. $0 \subseteq \mathbb{X}_T \subseteq \mathbb{X}$

A3. $Ax + BL(x) \in \mathbb{X}_T \ \forall x \in \mathbb{X}_T$

A4. $\Psi(0) = 0, \ \Psi(x) \geq 0 \ \forall \ x \neq 0$

A5. $\Psi(Ax + BL(x)) - \Psi(x) \leq -x^T Qx - L^T(x)RL(x) \quad \forall x \in \mathbb{X}_T$

A6. $L(x) \in \mathbb{U} \ \forall x \in \mathbb{X}_T$

Then, assuming feasibility at the initial state, an MPC controller using the following optimization problem will guarantee asymptotic stability

$$\boxed{\begin{array}{rl} \min\limits_{u} & \sum_{j=0}^{N-1} x_{k+j|k}^T Q x_{k+j|k} + u_{k+j|k}^T R u_{k+j|k} + \Psi(x_{k+N|k}) \\ \text{subject to} & \begin{array}{rcl} u_{k+j|k} & \in & \mathbb{U} \\ x_{k+j|k} & \in & \mathbb{X} \\ x_{k+N|k} & \in & \mathbb{X}_T \end{array} \end{array}} \qquad (2.17)$$

Proof The proof is based on using the optimal cost of the performance measure as a Lyapunov function. Let us denote the cost J_k, and the optimal cost J_k^*, obtained with the optimal control sequence $[u_{k|k}^* \ \ldots \ u_{k+N-1|k}^*]$. The use of a $*$ is a generic notation for variables related to optimal solutions. Introducing the associated optimal state sequence yields

$$J_k^* = \sum_{j=0}^{N-1} x_{k+j|k}^{*T} Q x_{k+j|k}^* + u_{k+j|k}^{*T} R u_{k+j|k}^* + \Psi(x_{k+N|k}^{*T})$$

A feasible solution at time $k+1$ is $[u_{k+1|k}^*, \ \ldots \ , u_{k+N-1|k}^*, L(x_{k+N|k}^*)]$. To see this, we first recall that $x_{k+N|k}^* \in \mathbb{X}_T$. Using Assumption 6, we see that $L(x_{k+N|k}^*)$ satisfies the control constraint, and Assumption 3 assures satisfaction of the terminal state constraint on $x_{k+N+1|k+1}$. Since $\mathbb{X}_T \subseteq \mathbb{X}$ according to Assumption 2 we also have $x_{k+N+1|k+1} \in \mathbb{X}$. The cost using this (sub-optimal) control sequence will be

$$\begin{aligned} J_{k+1} &= \sum_{j=0}^{N-1} [x_{k+1+j|k+1}^T Q x_{k+1+j|k+1} + u_{k+1+j|k+1}^T R u_{k+1+j|k+1}] + \Psi(x_{k+N+1|k+1}) \\ &= \sum_{j=0}^{N-1} [{x^*}_{k+j|k}^T Q x_{k+j|k}^* + {u^*}_{k+j|k}^T R u_{k+j|k}^*] + \Psi(x_{k+N|k}^*) \\ &\quad + \Psi(x_{k+N+1|k+1}) - \Psi(x_{k+N|k}^*) \\ &\quad + x_{k+N|k}^{*T} Q x_{k+N|k}^* + L^T(x_{k+N|k}^*) R L(x_{k+N|k}^*) \\ &\quad - x_{k|k}^{*T} Q x_{k|k}^* - {u^*}_{k|k}^T R u_{k|k}^* \end{aligned}$$

In the equation above, we added and subtracted parts from the optimal cost at time k. This is a standard trick in stability theory of MPC, and the reason is that the first line in the last equality now corresponds to the optimal cost at time k, i.e., J_k^*.

According to Assumption 5, the sum of the second and third row in the last equality is negative. Using this, we obtain

$$J_{k+1} \leq J_k^* - x_{k|k}^{*T} Q x_{k|k}^* - {u^*}_{k|k}^T R u_{k|k}^*$$

Since our new control sequence was chosen without optimization (we only picked a feasible sequence) we know that $J_{k+1} \geq J_{k+1}^*$. In other words, we have

$$J_{k+1}^* \leq J_k^* - x_{k|k}^{*T} Q x_{k|k}^* - u_{k|k}^{*T} R u_{k|k}^*$$

This shows that J_k^* is decreasing as long as $x \neq 0$ ($Q \succ 0$ by assumption). By construction, $J_k \geq 0$, so J_k has to converge to 0. Furthermore, $J_k = 0$ iff $x_k = 0$, hence x_k converge to the origin. □

Details and more rigorous proofs can be found in, e.g., (Lee, 2000) and (Mayne et al., 2000).

The assumptions in the theorem are easily understood heuristically. If we use the controller $u_k = L(x_k)$, and start in $\mathbb{X}_T$, we know that $u_k \in \mathbb{U}$ and $x_{k+1} \in \mathbb{X}_T \subseteq \mathbb{X}$. Furthermore,

$$\Psi(x_{k+1}) - \Psi(x_k) \leq -x_k^T Q x_k - u_k^T R u_k \tag{2.18}$$

By summing the left- and right-hand from time k to infinity, we obtain

$$\Psi(x_\infty) - \Psi(x_k) \leq \sum_{j=0}^{\infty} -x_{k+j}^T Q x_{k+j} - u_{k+j}^T R u_{k+j} \tag{2.19}$$

It follows from the assumptions that $\Psi(x_k)$ is a Lyapunov function when the controller $L(x)$ is used, hence $x_{k+j} \to 0$ for $j \to 0$. Using this, we obtain

$$\sum_{j=0}^{\infty} x_{k+j}^T Q x_{k+j} + u_{k+j}^T R u_{k+j} \leq \Psi(x_k) \tag{2.20}$$

In other words, $\Psi(x)$ is an upper bound of the infinite horizon cost, when we use the (sub-optimal) controller $L(x_k)$. Obviously, the optimal cost is even lower, so $\Psi(x_k)$ is an upper bound of the optimal cost also. This is the most intuitive way to interpret the assumptions. *The terminal state weight $\Psi(x)$ is an upper bound of the optimal infinite horizon cost in the terminal state domain $\mathbb{X}_T$.*

Methods to Choose $\{\mathbb{X}_T, L(x), \Psi(x)\}$

So, having a fairly general theorem for stabilizing MPC controllers, what is the catch? The problem is of course to find the triple $\{\mathbb{X}_T, L(x), \Psi(x)\}$. A number of methods have been proposed over the years.

A very simple method, generalizing the basic idea in, e.g., (Kleinman, 1974), was proposed and analyzed in the seminal paper (Keerthi and Gilbert, 1988). The method holds for a large class of systems, performance measures and constraints, and in order to guarantee stability, a terminal state equality is added to the optimization

$$\boxed{\begin{array}{rrcl} \min\limits_{u} & \multicolumn{3}{l}{\sum_{j=0}^{N-1} x_{k+j|k}^T Q x_{k+j|k} + u_{k+j|k}^T R u_{k+j|k}} \\ \text{subject to} & u_{k+j|k} & \in & \mathbb{U} \\ & x_{k+j|k} & \in & \mathbb{X} \\ & x_{k+N|k} & = & 0 \end{array}} \tag{2.21}$$

In terms of Theorem 2.1, this corresponds to $\mathbb{X}_T = \{0\}$, $L(x) = 0$ and $\Psi(x) = 0$. The set-up is successful since $L(x)$ trivially satisfies all assumptions of Theorem 2.1 in $\mathbb{X}_T$. Notice that the equality constraint is artificial in the sense that the state will not reach the origin at time $k + N$, since the constraint continuously is shifted forward in time.

The second approach, applicable to stable systems without state constraints, can be considered as the complete opposite to the previous approach. Since the system is stable, a stabilizing controller is $L(x) = 0$, and this controller satisfies the control constraints trivially for all states, hence $\mathbb{X}_T = \mathbb{R}^n$. If we chose a quadratic terminal state weight $\Psi(x) = x^T P x$, Assumption 5 in Theorem 2.1 simplifies to

$$x_k^T A^T P A x_k - x_k^T P x_k \preceq -x_k^T Q x_k \tag{2.22}$$

This is guaranteed if

$$A^T P A - P \preceq -Q \tag{2.23}$$

Hence, all we have to do is to solve a Lyapunov equation to find P, and use the following formulation of the finite horizon problem.

$$\boxed{\begin{array}{rrcl} \min\limits_{u} & \multicolumn{3}{l}{\sum_{j=0}^{N-1} x_{k+j|k}^T Q x_{k+j|k} + u_{k+j|k}^T R u_{k+j|k} + x_{k+N|k}^T P x_{k+N|k}} \\ \text{subject to} & u_{k+j|k} & \in & \mathbb{U} \end{array}} \tag{2.24}$$

This was essentially the idea used in (Rawlings and Muske, 1993).

More general schemes can be obtained by combining ideas from the two previous approaches. In (Rawlings and Muske, 1993), only the unstable modes of the system were forced to the origin in the prediction, and a quadratic terminal state weight was applied to the stable modes. Later, generalizing the ideas in, e.g., (Rawlings and Muske, 1993), the following idea emerged.

We use a linear feedback $u_k = Lx_k$ as the nominal controller, an ellipsoidal terminal state domain $\mathbb{X}_T = \mathbb{E}_W = \{x : x^T W x \leq 1\}$ and a quadratic terminal state weight $\Psi(x) = x^T P x$. With these choices, we obtain the following MPC problem from Theorem 2.1.

$$\boxed{\begin{array}{rrcl} \min\limits_{u} & \multicolumn{3}{l}{\sum_{j=0}^{N-1} x_{k+j|k}^T Q x_{k+j|k} + u_{k+j|k}^T R u_{k+j|k} + x_{k+N|k}^T P x_{k+N|k}} \\ \text{subject to} & u_{k+j|k} & \in & \mathbb{U} \\ & x_{k+N|k}^T W x_{k+N|k} & \leq & 1 \end{array}} \tag{2.25}$$

There is one appealing property of this approach. If we chose L as the LQ controller obtained using the weights Q and R, the matrix P will be the Riccati solution to the LQ problem and satisfy

$$x_{k+N|k}^T P x_{k+N|k} = \min_{u \in \mathbb{R}^m} \sum_{j=N}^{\infty} x_{k+j|k}^T Q x_{k+j|k} + u_{k+j|k}^T R u_{k+j|k} \tag{2.26}$$

Now, if control and state constraints are inactive for $i \geq k+N$ in the solution to the constrained infinite horizon problem, and the terminal state constraint is satisfied in this solution, we must have

$$\begin{array}{cl} \min\limits_{u \in \mathbb{U}, x \in \mathbb{X}} & \sum_{j=0}^{\infty} x_{k+j|k}^T Q x_{k+j|k} + u_{k+j|k}^T R u_{k+j|k} \\ & \Leftrightarrow \\ \min\limits_{u \in \mathbb{U}, x \in \mathbb{X}} & \sum_{j=0}^{N-1} x_{k+j|k}^T Q x_{k+j|k} + u_{k+j|k}^T R u_{k+j|k} + \\ & \qquad \min\limits_{u \in \mathbb{R}^m} \sum_{j=N}^{\infty} x_{k+j|k}^T Q x_{k+j|k} + u_{k+j|k}^T R u_{k+j|k} \\ & \Leftrightarrow \\ \min\limits_{\substack{u \in \mathbb{U}, x \in \mathbb{X} \\ x_{k+N|k} \in \mathbb{X}_T}} & \sum_{j=0}^{N-1} x_{k+j|k}^T Q x_{k+j|k} + u_{k+j|k}^T R u_{k+j|k} + x_{k+N|k}^T P x_{k+N|k} \end{array}$$

The stabilized finite horizon MPC solution will thus coincide with the infinite horizon problem whenever control constraints are inactive beyond the prediction horizon and the terminal state constraint is inactive.

This last approach in the cornerstone in many algorithms for stabilizing MPC (Zheng, 1995; Chen, 1997; Bemporad and Mosca, 1997; Scokaert and Rawlings, 1998; Lee and Kouvaritakis, 1999; De Doná, 2000; Lee, 2000).

3

Convex and Robust Optimization

A considerable part of the material in this thesis is based on convex optimization. The idea in almost all chapters is to formulate an optimization problem, and then show that the problem can be cast as some standard convex optimization problem.

The appealing property of convex optimization problems is that we can, under fairly weak conditions, construct algorithms that are guaranteed to find an optimal solution, if one exists, or construct a certificate that there exist no solution (Nesterov and Nemirovskii, 1993; Ben-Tal and Nemirovski, 2001).

Moreover, it is possible to construct algorithms with *polynomial complexity*. This essentially means that the computational effort required to find the solution grows polynomially with respect to the problem dimensions (typically the number of variables and constraints) and the desired accuracy on the solution. In other words, we can construct efficient algorithms that scale well with problem size.

The introduction of convex optimization, and semidefinite programming in particular, as a standard mathematical tool has had, and still has, a profound impact on control and systems theory. In the same sense that we earlier considered a solution in terms of a Riccati equation as a closed form expression, problems with solutions described by a convex program can now in many situations be considered to be "analytically" solved.

There is a vast amount of literature on convex optimization, so let us just mention a few references that fit as reference reading to this thesis. The forthcoming book (Boyd and Vandenberghe, 2002) serves as an excellent introduction to both mathematical and engineering aspects of convex optimization. Another book in the same vein, but with a slight bias towards semidefinite programming, is (Ben-Tal and Nemirovski, 2001). A rather detailed, yet easily accessible, look at some basic solution strategies, so called interior-point methods, can be found in (den Hertog, 1994).

3.1 Standard Convex Optimization Problems

For the sake of completeness, let us begin with the definition of a general convex optimization problem (Boyd and Vandenberghe, 2002).

Definition 3.1 (Convex program, CP)

$$\boxed{\begin{array}{rlll} \min\limits_{x} & f_0(x) & & \\ \text{subject to} & f_i(x) & \leq & 0 \quad i = 1\ldots m \\ & Ax & = & b \end{array}} \tag{3.1}$$

The objective function $f_0(x) : \mathbb{R}^n \to \mathbb{R}$ and the constraint functions $f_i(x) : \mathbb{R}^n \to \mathbb{R}$ are assumed convex in the decision variable x, hence the term convex programming.

Although the general problem can be solved relatively efficiently as discussed above, it is always advantageous to use dedicated solvers that exploit structure in the objective function and the constraints. In order to do so, a number of standard problems can be defined, and we will now introduce the most important ones, all of them used in this thesis.

3.1.1 Linear and Quadratic Programming

The first and perhaps most commonly known problem class is linear programming.

Definition 3.2 (Linear program, LP)

$$\boxed{\begin{array}{rlll} \min\limits_{x} & c^T x & & \\ \text{subject to} & Cx & \geq & d \\ & Ax & = & b \end{array}} \tag{3.2}$$

Another problem class that most readers should be familiar with is quadratic programming.

Definition 3.3 (Quadratic program, QP)

$$\boxed{\begin{array}{rrcl} \min\limits_{x} & \frac{1}{2}x^T Q x + c^T x & & \\ \text{subject to} & Cx & \geq & d \\ & Ax & = & b \end{array}} \tag{3.3}$$

Since we are addressing convex optimization problems, it is assumed that the matrix Q is positive semidefinite. This yields a convex problem.

3.1.2 Second Order Cone Programming

Generalization of quadratic programming leads to second order cone programming.

Definition 3.4 (Second order cone program, SOCP)

$$\boxed{\begin{array}{rrcll} \min\limits_{x} & c^T x & & & \\ \text{subject to} & \left\|A_i x + b_i\right\| & \leq & c_i^T x + d_i & i = 1 \ldots m \\ & Ax & = & b & \end{array}} \tag{3.4}$$

A constraint in the form $\left\|A_i x + b_i\right\| \leq c_i^T x + d_i$ is called a second order cone constraint, hence the term second order cone programming. This is a surprisingly expressive problem class with many applications, such as robust least squares, sum-of-norms minimization and quadratically constrained quadratic programming, just to mention a few (Lobo et al., 1998; Ben-Tal and Nemirovski, 2001)

3.1.3 Semidefinite Programming

Linear and quadratic programming is optimization over the nonnegative orthant, and second order cone programming is optimization over the second order cone (the ice-cream cone or Lorenz cone). Further generalization leads us to optimization over the semidefinite cone, described with linear matrix inequalities (LMIs).

Definition 3.5 (Linear matrix inequality, LMI)

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i \succeq 0, \quad x \in \mathbb{R}^m,\ F_i = F_i^T \in \mathbb{R}^{n\times n}$$

Optimization over the semidefinite cone is called semidefinite programming.

Definition 3.6 (Semidefinite program, SDP)

$$\begin{array}{|rrcl|}\hline \min\limits_{x} & c^T x & & \\ \text{subject to} & F(x) & \succeq & 0 \\ & Ax & = & b \\ \hline\end{array} \tag{3.5}$$

An excellent introduction to semidefinite programming and LMIs, with special attention to problems arising in systems theory, can be found in the already classic (Boyd et al., 1994). More recent material can be found in, e.g., (El Ghaoui and Niculescu, 2000) and (Ben-Tal and Nemirovski, 2001).

The semidefinite programming problem can be generalized further[1] and we end up with a determinant maximization problem.

Definition 3.7 (Determinant maximization, MAXDET)

$$\begin{array}{|rrcl|}\hline \min\limits_{x} & c^T x - \log\det G(x) & & \\ \text{subject to} & F(x) & \succeq & 0 \\ & G(x) & \succeq & 0 \\ & Ax & = & b \\ \hline\end{array} \tag{3.6}$$

This awkward looking optimization problem has many applications in, e.g., control theory (invariant set theory), geometry (bounding polytopes with ellipsoids and similar problems) and optimal experiment design (Vandenberghe et al., 1998).

3.2 Robust Optimization

Robust optimization means that we have an optimization problem with uncertainty in the problem data. The uncertainty, here denoted Δ, is only known to belong to some set $\boldsymbol{\Delta}$. The goal is to minimize some objective function, while guaranteeing a set of constraints to be satisfied for all possible uncertainties.

Definition 3.8 (Robust programming)

$$\begin{array}{|rrclll|}\hline \min\limits_{x} & f_0(x) & & & & \\ \text{subject to} & f_i(\Delta, x) & \leq & 0 & i = 1 \ldots m & \forall \Delta \in \boldsymbol{\Delta} \\ & Ax & = & b & & \\ \hline\end{array} \tag{3.7}$$

[1] The determinant maximization problem can actually be stated as a standard semidefinite programming problem but the conversion is highly intricate (Nesterov and Nemirovskii, 1993)

Note that there is no loss of generality to assume the objective function to be known, since an uncertain objective function can be taken care of by an epigraph formulation, i.e., introducing a new variable t and constraint $f_0(\Delta, x) \leq t$, and changing the objective function to t.

From the definition above, we see that robust programming is all about ensuring robust satisfaction of constraints. This means that we must be able to perform a maximization of $f_i(\Delta, x)$ over the admissible uncertainties $\Delta \in \boldsymbol{\Delta}$. The following results will be used throughout this thesis to facilitate this.

The first case involves maximization of a linear function in the unit cube.

Theorem 3.1 (Maximum of linear function in the unit cube)

$$\max_{|x|\leq 1} c^T x = ||c||_1 = |c^T|\mathbf{1} \tag{3.8}$$

Proof Follows immediately since $\max_{|x|\leq 1} c^T x = \max_{|x|\leq 1} \sum c_i x_i = \sum c_i \text{sign}(c_i) = ||c||_1$ □

Changing the uncertainty set to an ellipsoid gives us our second case.

Theorem 3.2 (Maximum of linear function in ellipsoid)
Let $W \succ 0$. It then holds that

$$\max_{x^T W x \leq 1} c^T x = \sqrt{c^T W^{-1} c} \tag{3.9}$$

Proof Let $y = W^{1/2}x$. The objective is now maximization of $c^T W^{-1/2} y$ subject to $y^T y \leq 1$. The optimal choice y is the parallel vector $y = \left(\frac{c^T W^{-1/2}}{||c^T W^{-1/2}||}\right)^T$ which yields the objective $\frac{c^T W^{-1} c}{||c^T W^{-1/2}||} = \sqrt{c^T W^{-1} c}$. □

The third case is a bit more complex and involves robust satisfaction of quadratic constraints (Boyd et al., 1994).

Theorem 3.3 (The S-procedure)
Let $T_i(x)$ be quadratic functions,

$$T_i(x) = x^T P_i x, \quad P_i = P_i^T, \quad i = 0, \ldots, m \tag{3.10}$$

A sufficient condition for

$$T_0(x) \leq 0 \ \forall x \text{ such that } T_i(x) \leq 0 \quad i = 1, \ldots, m \tag{3.11}$$

to hold is that there exist a $\tau \in \mathbb{R}_+^m$ such that

$$P_0 - \sum_{i=1}^{m} \tau_i P_i \preceq 0 \tag{3.12}$$

When the condition is both sufficient and necessary, the S-procedure is said to be lossless. This is the case when $m = 1$.

The S-procedure can be used to derive a number of theorems related to uncertain LMIs. The theorems that follow, on so-called semidefinite relaxations, can be found in (El Ghaoui et al., 1998). The proofs are taken from (El Ghaoui and Lebret, 1997) (minor changes are made in the proof to suit our notation better).

Theorem 3.4 (Robust linear matrix inequality for affine uncertainty) *Let F, L, R and Δ be real matrices of appropriate size. The uncertain LMI*

$$F + L\Delta R + R^T\Delta^T L^T \succeq 0 \tag{3.13}$$

holds for all $\Delta \in \{\Delta : \Delta = \oplus_1^m \Delta_i, \Delta_i \in \mathbb{R}^{r\times p}, ||\Delta_i|| \leq 1,\ i = 1\dots m\}$ if there exist a $\tau \in \mathbb{R}_+^m$, $T = \oplus_{i=1}^m \tau_i I^{p\times p}$ and $S = \oplus_{i=1}^m \tau_i I^{r\times r}$ such that

$$\begin{pmatrix} F - LSL^T & R^T \\ R & T \end{pmatrix} \succeq 0 \tag{3.14}$$

The condition is sufficient and necessary when $m = 1$.

Proof To begin with, we introduce matrices $T_i \in \mathbb{R}^{mp\times p}$ and $S_i \in \mathbb{R}^{mr\times r}$

$$T_1 = \begin{pmatrix} I^{p\times p} \\ 0^{p\times p} \\ \vdots \\ 0^{p\times p} \end{pmatrix}, \dots, T_m = \begin{pmatrix} 0^{p\times p} \\ 0^{p\times p} \\ \vdots \\ I^{p\times p} \end{pmatrix}, S_1 = \begin{pmatrix} I^{r\times r} \\ 0^{r\times r} \\ \vdots \\ 0^{r\times r} \end{pmatrix}, \dots, S_m = \begin{pmatrix} 0^{r\times r} \\ 0^{r\times r} \\ \vdots \\ I^{r\times r} \end{pmatrix} \tag{3.15}$$

The constraint (3.13) is equivalent to $x^T(F + L\Delta R + R^T\Delta^T L^T)x \geq 0\ \forall x$. Introduce $y = \Delta^T L^T x$ and we have $x^T F x + y^T R x + x^T R^T y \geq 0$. Multiply y with T_i^T to obtain $T_i^T y = T_i^T \Delta^T L^T x$. Notice now that $T_i^T \Delta^T = T_i^T \Delta^T S_i S_i^T$, hence $T_i^T y = T_i^T \Delta^T S_i S_i^T L^T x$. The LMI should hold when $||\Delta T_i|| \leq 1$ which implies $y^T T_i T_i^T y \leq x^T L S_i S_i^T S_i S_i^T L^T x$. We notice that $S_i^T S_i = I$, hence we know that $y^T T_i T_i^T y \leq x^T L S_i S_i^T L^T x$. We can now write our uncertain LMI as

$$\begin{pmatrix} x \\ y \end{pmatrix}^T \begin{pmatrix} F & R^T \\ R & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \geq 0 \text{ when } \begin{pmatrix} x \\ y \end{pmatrix}^T \begin{pmatrix} LS_iS_i^TL^T & 0 \\ 0 & -T_iT_i^T \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \geq 0 \tag{3.16}$$

Application of the S-procedure gives a sufficient condition.

$$\begin{pmatrix} F & R^T \\ R & 0 \end{pmatrix} \geq \sum_{i=1}^m \tau_i \begin{pmatrix} LS_iS_i^TL^T & 0 \\ 0 & -T_iT_i^T \end{pmatrix} \tag{3.17}$$

Defining the matrices $T = \sum_{i=1}^m \tau_i T_i T_i^T = \oplus_{i=1}^m \tau_i I^{p\times p}$ and $S = \sum_{i=1}^m \tau_i S_i S_i^T = \oplus_{i=1}^m \tau_i I^{r\times r}$ gives the desired result. □

The following theorem is a slight generalization and deals with uncertainty in a linear fractional transformation model (LFT).

Theorem 3.5 (Robust linear matrix inequality for LFT uncertainty) *Let $F = F^T$, L, R, D and Δ be real matrices of appropriate size. The uncertain LMI*

$$F + L\Delta(I - D\Delta)^{-1}R + R^T(I - D\Delta)^{-T}\Delta^T L^T \succeq 0 \tag{3.18}$$

holds for all $\Delta \in \{\Delta : \Delta = \oplus_1^m \Delta_i, \Delta_i \in \mathbb{R}^{r\times p}, ||\Delta_i|| \leq 1,\ i = 1\ldots m\}$ if there exist a $\tau \in \mathbb{R}_+^m$, $T = \oplus_{i=1}^m \tau_i I^{p\times p}$ and $S = \oplus_{i=1}^m \tau_i I^{r\times r}$ such that that

$$\begin{pmatrix} F - LSL^T & R^T - LSD^T \\ R - DSL^T & T - DSD^T \end{pmatrix} \succeq 0 \tag{3.19}$$

The condition is sufficient and necessary when $m = 1$.

Proof Define the matrices S_i and T_i as in the proof for Theorem 3.4. We require $x^T(F + L\Delta(I - D\Delta)^{-1}R + R^T(I - D\Delta)^{-1}\Delta^T L^T)x \geq 0\ \forall x$. Introduce $y = (I - D\Delta)^{-T}\Delta^T L^T x$ and we have $x^T Fx + y^T Rx + x^T R^T y \geq 0$. Trivial manipulations yield $y = \Delta^T(L^T x + D^T y)$. Multiply with T_i^T to obtain $T_i^T y = T_i^T \Delta^T(L^T x + D^T y)$. Use the fact that $T_i^T \Delta^T = T_i^T \Delta^T S_i S_i^T$ to obtain $T_i^T y = T_i^T \Delta^T S_i S_i^T(L^T x + D^T y)$. Use the norm bound on $T_i\Delta$ to conclude that $y^T T_i T_i^T y \leq (x^T L + y^T D)S_i S_i^T S_i S_i^T(L^T x + D^T y)$. Exploiting $S_i^T S_i = I$ finally enables us to write our uncertain LMI as

$$\begin{pmatrix} x \\ y \end{pmatrix}^T \begin{pmatrix} F & R^T \\ R & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \geq 0 \text{ when } \begin{pmatrix} x \\ y \end{pmatrix}^T \begin{pmatrix} LS_iS_i^T L^T & LS_iS_i^T D^T \\ DS_iS_i^T L^T & DS_iS_i^T D^T - T_iT_i^T \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \geq 0$$

Application of the S-procedure yields a sufficient condition.

$$\begin{pmatrix} F & R^T \\ R & 0 \end{pmatrix} \succeq \sum_{i=1}^m \tau_i \begin{pmatrix} LS_iS_i^T L^T & LS_iS_i^T D^T \\ DS_iS_i^T L^T & DS_iS_i^T D^T - T_iT_i^T \end{pmatrix} \tag{3.20}$$

Defining the matrices $T = \sum_{i=1}^m \tau_i T_i T_i^T = \oplus_{i=1}^m \tau_i I^{p\times p}$ and $S = \sum_{i=1}^m \tau_i S_i S_i^T = \oplus_{i=1}^m \tau_i I^{r\times r}$ gives the desired result. □

3.3 Software

The great success of convex programming lies in the fact that the problems can be solved with high efficiency. Since convex programming is a highly active research area, there exist a plethora of free software packages for solving various classes of convex optimization problems.

The major part of this thesis relies on second order cone and semidefinite programming, so we concentrate on solvers for these problems.

For standard semidefinite problems, DSDP (Benson and Ye, 2001), CSDP (Borchers, 1999) and SDPA (Yamashita et al., 2002) stand out as some of the fastest and most robust solvers. Somewhat more general solvers are SeDuMi (Sturm, 1999) and SDPT3 (Toh et al., 1999). These solvers address mixed semidefinite and second order cone programming problems, i.e., problems with both semidefinite and second order cone constraints. Finally, the determinant maximization problem can be solved using MAXDET (Wu et al., 1996).

A comprehensive computational survey on semidefinite and second order cone programming solvers can be found in (Mittelmann, 2002).

The solvers above only solve the optimization problem, but do not support definition of problems. In order to simplify algorithm development and implementation of optimization problems, some kind of interface to the solver is needed. During the writing of this thesis, the MATLAB toolbox YALMIP (Löfberg, 2002c) was developed to facilitate this. YALMIP supports all of the solvers above, and can be used to define linear, second order cone, semidefinite and determinant maximization problems.

Optimization problems in this thesis have been solved, if not otherwise is stated, with YALMIP and SeDuMi.

4

Robust MPC

Controlling a system with control and state constraints is one of the most important problems in control theory, but also one of the most challenging. Another important but just as demanding topic is robustness against uncertainties in a controlled system. Solving a control problem with both constraints and uncertainties can thus feel like a daunting task. Nevertheless, this is the main topic in this thesis.

The crucial question in robust control is *how to exploit knowledge about uncertainty.* Typical knowledge can be bounds on uncertain parameters in the system, such as the weight of a robot arm, or bounds on external disturbances, such as the load on the robot arm. Of course, we can never capture all uncertainties in practice, but we can hopefully capture the most important and account for these.

We saw in the introductory chapter on MPC that the control law is based on an on-line optimization problem. Adding uncertainties to the model thus requires some way to incorporate this information into the optimization problem. This thesis is to a large extent devoted to an approach for robust MPC called minimax MPC. The purpose of this chapter is to give a background to the topic and summarize the main results.

4.1 Uncertainty Models

The systems we address in this chapter are constrained uncertain linear systems. Loosely speaking, we have

$$x_{k+1} = A(\Delta_k)x_k + B(\Delta_k)u_k + G(\Delta_k), \quad \Delta_k \in \mathbf{\Delta} \tag{4.1a}$$
$$y_k = Cx_k \tag{4.1b}$$

The models that are introduced here cover almost all models used in minimax MPC schemes. For notational simplicity, we do not present the most general models possible, but settle with a simple description that suffices for our purpose.

The simplest model is to assume a bounded unknown external additive disturbance.

$$x_{k+1} = A_k x_k + B_k u_k + G w_k, \quad w_k \in \mathbb{W} \tag{4.2a}$$
$$y_k = Cx_k \tag{4.2b}$$

The disturbance set $\mathbb{W}$ is typically a polytope, but not necessarily. A standard assumption however is that $\mathbb{W}$ is convex and compact.

The perhaps most often used model is the polytopic model.

$$x_{k+1} = A_k x_k + B_k u_k \tag{4.3a}$$
$$y_k = Cx_k \tag{4.3b}$$
$$\begin{pmatrix} A_k & B_k \end{pmatrix} \in \mathbf{Co}\{\begin{pmatrix} A^{(1)} & B^{(1)} \end{pmatrix}, \ldots, \begin{pmatrix} A^{(q)} & B^{(q)} \end{pmatrix}\} \tag{4.3c}$$

The notation $\mathbf{Co}\{\cdot\}$ is used to denote the convex hull.

$$\mathbf{Co}\{x^{(1)}, x^{(2)}, \ldots, x^{(q)}\} = \{x : x = \sum_{j=1}^{q} \lambda_j x^{(j)}, \sum_{j=1}^{q} \lambda_j = 1, \lambda \geq 0\} \tag{4.4}$$

Another frequently used model is the linear fractional transformation uncertainty model, or LFT model for short.

$$x_{k+1} = Ax_k + Bu_k + Gp_k \tag{4.5a}$$
$$y_k = Cx_k \tag{4.5b}$$
$$h_k = D_x x_k + D_u u_k + D_p p_k \tag{4.5c}$$
$$p_k = \Delta_k h_k, \quad ||\Delta_k|| \leq 1 \tag{4.5d}$$

Here, we assumed the uncertainty Δ to be unstructured, but this can be extended to allow for more complex uncertainty structures.

The crucial difference now compared to the nominal case in Chapter 2 is that the predictions $y_{k+j|k}$ and $x_{k+j|k}$ define sets instead of exactly known values. Let us exemplify this.

Example 4.1 (Prediction sets)
Consider an autonomous system with a polytopic uncertainty.

$$x_{k+1} = A_k x_k \quad A_k \in \mathbf{Co}\{A^{(1)}, A^{(2)}, A^{(3)}\} \tag{4.6a}$$

$$A^{(1)} = \begin{pmatrix} 0.25 & 0 \\ 0.5 & 0.25 \end{pmatrix}, A^{(2)} = \begin{pmatrix} 0.5 & 0 \\ 0.25 & 0.25 \end{pmatrix}, A^{(3)} = \begin{pmatrix} 0.25 & 0 \\ 0.25 & 0.25 \end{pmatrix} \tag{4.6b}$$

With an initial condition $x_{k|k}$ we obtain the one-step prediction

$$x_{k+1|k} = A_k x_{k|k} \in \mathbf{Co}\{A^{(1)}x_{k|k}, A^{(2)}x_{k|k}, A^{(3)}x_{k|k}\} = \mathbf{Co}\{x^{(1)}_{k+1|k}, x^{(2)}_{k+1|k}, x^{(3)}_{k+1|k}\}$$

It can readily be shown, see e.g. (Schuurmans and Rossiter, 2000), that the procedure can be recursed and the two-step prediction is

$$\begin{aligned}
\begin{pmatrix} x_{k+1|k} \\ x_{k+2|k} \end{pmatrix} &= \begin{pmatrix} x_{k+1|k} \\ A_k x_{k+1|k} \end{pmatrix} \\
&\in \mathbf{Co}\{\begin{pmatrix} x^{(1)}_{k+1|k} \\ A^{(1)}x^{(1)}_{k+1|k} \end{pmatrix}, \begin{pmatrix} x^{(1)}_{k+1|k} \\ A^{(2)}x^{(1)}_{k+1|k} \end{pmatrix}, \begin{pmatrix} x^{(1)}_{k+1|k} \\ A^{(3)}x^{(1)}_{k+1|k} \end{pmatrix}, \\
&\qquad \begin{pmatrix} x^{(2)}_{k+1|k} \\ A^{(1)}x^{(2)}_{k+1|k} \end{pmatrix}, \ldots, \begin{pmatrix} x^{(3)}_{k+1|k} \\ A^{(2)}x^{(3)}_{k+1|k} \end{pmatrix}, \begin{pmatrix} x^{(3)}_{k+1|k} \\ A^{(3)}x^{(3)}_{k+1|k} \end{pmatrix}\} \qquad (4.7) \\
&= \mathbf{Co}\{\begin{pmatrix} x^{(1)}_{k+1|k} \\ x^{(11)}_{k+2|k} \end{pmatrix}, \begin{pmatrix} x^{(1)}_{k+1|k} \\ x^{(21)}_{k+2|k} \end{pmatrix}, \ldots, \begin{pmatrix} x^{(3)}_{k+1|k} \\ x^{(23)}_{k+2|k} \end{pmatrix}, \begin{pmatrix} x^{(3)}_{k+1|k} \\ x^{(33)}_{k+2|k} \end{pmatrix}\} \qquad (4.8)
\end{aligned}$$

Continuing this process allows us to define the predicted set for arbitrary prediction horizons. See Figure 4.1 for an illustration with $x_{k|k} = \begin{pmatrix} 1 & 4 \end{pmatrix}$. Notice that the number of possible vertices in the description grows exponentially in the horizon length. With q models in the polytopic description and a prediction length of N, there will be q^N elements in the description of the predicted states. This is one of the main obstacles in minimax MPC algorithms.

Our next step is to use these sets of predicted states in some way. The prevailing approach in robust MPC, and robust control in general, is to solve minimax problems, i.e., solve problems where worst-case scenarios are accounted for. Clearly, this is a conservative approach, but it is one of the few ways we have to place the word robustness in a well-defined mathematical framework.

4.2 Minimax MPC

In this section, we will try to give an historical background to minimax MPC and categorize the different approaches under a couple of headers. The categories are not exact, some approaches fit under several headers, but it should give a flavor

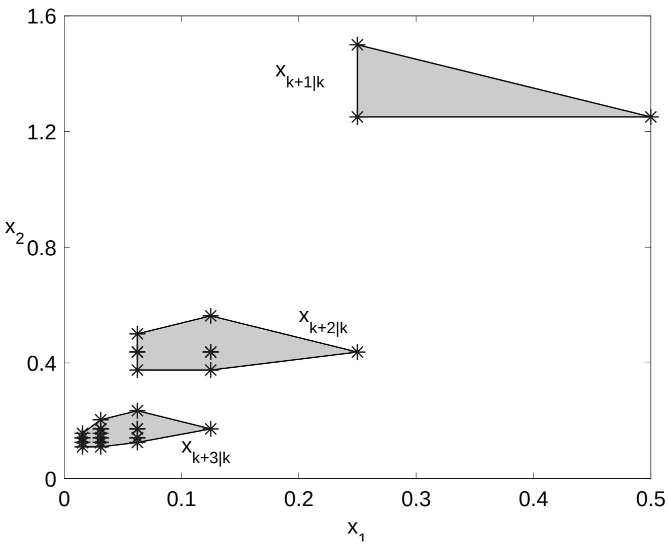

Figure 4.1: State predictions for system with a polytopic uncertainty. The predictions are no longer points but sets. The marker '*' indicate states obtained using the recursive procedure where we create predictions using all the vertices of the uncertainty set. The predicted states are guaranteed to be contained in the convex hull of these predictions, here indicated by the gray-shaded regions.

of what methods we have available today, and what the main differences are. The interested reader might find the surveys in (Bemporad and Morari, 1999) and (Mayne et al., 2000) suitable for additional reading.

A minimax MPC problem can typically be written as

$$\begin{array}{lrcl} \min\limits_{u} \max\limits_{\Delta} & \multicolumn{3}{l}{\ell(x_{k|k}, u_{k|k}, x_{k+1|k}, u_{k+1|k}, \ldots, x_{k+N-1|k}, u_{k+N-1|k})} \\ \text{subject to} & u_{k+j|k} & \in & \mathbb{U} \quad \forall \Delta \in \boldsymbol{\Delta} \\ & x_{k+j|k} & \in & \mathbb{X} \quad \forall \Delta \in \boldsymbol{\Delta} \\ & \Delta_{k+j|k} & \in & \boldsymbol{\Delta} \end{array} \tag{4.9}$$

Notice that the control input works in open-loop. Schemes based on (4.9) are therefore sometimes called open-loop minimax MPC.

A typical and reasonable assumption in minimax MPC is to assume the performance measure ℓ to be convex in $x_{k+j|k}$ and $u_{k+j|k}$. This enables the use of following central theorem.

Theorem 4.1 ((Bertsekas, 1999))
Let $\mathbb{C}$ be a closed convex set and let $f : \mathbb{C} \rightarrow \mathbb{R}$ be a convex function. Then if f attains a maximum over $\mathbb{C}$, it attains a maximum at some extreme point of $\mathbb{C}$.

To us, this means that if ℓ is convex and the uncertainties Δ_k generate a convex set of predictions, we only need to check the extreme points of this set. In the example above, we could have concentrated our effort on the predicted states obtained using the vertex matrices $A^{(1)}$, $A^{(2)}$ and $A^{(3)}$, since the extreme points are among these predictions. Notice that an extreme point of the uncertainty set does not necessarily generate an extreme point of the predicted set. The converse is however true, an extreme point of the predicted set is defined by an extreme point of the polytopic model. We will be sloppy and sometimes talk about extreme points (vertices) of the predicted sets, when we actually mean the states generated by the extreme points of the uncertainty set.

FIR models and Minimum Peak Problems

Minimax formulations for robust MPC date back to the work of (Campo and Morari, 1987). They used an uncertain finite impulse response (FIR) model.

$$y_k = \sum_{i=1}^{M} \theta_i u_{k-i}, \quad \theta \in \Theta \tag{4.10}$$

The uncertainty set Θ was a polytope. It should be mentioned that they also recognized that the model easily could be extended to a polytopic model to deal with parametric uncertainties in the B (or C) matrix of a state space model.

The minimax problem they solved was minimization of the worst-case deviation from a known reference trajectory $r_{k+j|k}$ (the minimum peak problem).

$$\boxed{\begin{array}{ll} \min\limits_{u} \max\limits_{j} \max\limits_{\theta} & \|y_{k+j|k} - r_{k+j|k}\|_\infty \\ \text{subject to} & u_{k+j|k} \in \mathbb{U} \end{array}} \tag{4.11}$$

It was shown that this problem can be cast as a linear program. Unfortunately, the size of the linear program grew exponentially in the number of uncertain parameters, since a brute force enumeration scheme was used, i.e., a constraint for every vertex of the polytope Θ was added to a linear program.

The complexity of the approach in (Campo and Morari, 1987) was later improved upon in (Allwright and Papavasiliou, 1992) where it was recognized that maximization over θ could be performed analytically, and the result was a problem formulation with polynomial complexity, not much larger than the nominal problem. Further work along the same ideas include (Zheng and Morari, 1993; Zheng, 1995) where stability issues are analyzed and slightly more general performance measures are introduced. More recent work include generalization to FIR models with norm-bounded errors in (Boyd et al., 1997), (Oliviera et al., 2000) with emphasis on the uncertainty modeling, and (Vandenberghe et al., 2002) who exploit the inherent structure in the linear programs to develop an efficient solver.

The common ingredients in the references above are that the minimax problems are cast as linear programs, and the maximization with respect to the uncertainty is performed analytically, resulting in a reasonably small linear program.

LMI Based Minimization and Linear State Feedback Parameterizations

A major theoretical breakthrough in MPC came with the seminal paper (Kothare et al., 1994). The idea was to solve a robust linear state feedback problem repeatedly on-line using semidefinite programming and linear matrix inequalities (LMIs).

The system (Kothare et al., 1994) addressed in the original paper was a polytopic model.

$$x_{k+1} = A_k x_k + B_k u_k \tag{4.12}$$

Before we present the MPC scheme, let us look at the robust linear state feedback problem. Finding a linear state feedback $u_k = Lx_k$ that minimizes an upper bound of the worst-case infinite horizon quadratic cost

$$\max_{\Delta} \sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k \tag{4.13}$$

given a polytopic uncertainty model, can be done using LMIs. The function $x_k^T P x_k$ with $P \succ 0$ is an upper bound on the worst-case cost if it holds for all possible models A_k and B_k that

$$x_{k+1}^T P x_{k+1} - x_k^T P x_k \leq -x_k^T Q x_k - u_k^T R u_k \tag{4.14}$$

This is easily seen by adding up the left- and right-hand side from 0 to ∞. After inserting a linear feedback $u_k = Lx_k$, we obtain a matrix inequality.

$$(A_k + B_k L)^T P (A_k + B_k L) - P \preceq -Q - L^T R L \tag{4.15}$$

The problem of finding P and L can be transformed to a semidefinite program by performing a clever variable change (Boyd et al., 1994). Performing a congruence transformation with $W = P^{-1}$, defining $K = LP^{-1}$ and applying a Schur complement yields an LMI in W and K.

$$\begin{pmatrix} W & (A_k W + B_k K)^T & W & K^T \\ A_k W + B_k K & W & 0 & 0 \\ W & 0 & Q^{-1} & 0 \\ K & 0 & 0 & R^{-1} \end{pmatrix} \succeq 0 \tag{4.16}$$

The LMI should hold for all possible models, but due to the polytopic model, we only need to check the vertices (Boyd et al., 1994)

$$\begin{pmatrix} W & (A^{(i)} W + B^{(i)} K)^T & W & K^T \\ A^{(i)} W + B^{(i)} K & W & 0 & 0 \\ W & 0 & Q^{-1} & 0 \\ K & 0 & 0 & R^{-1} \end{pmatrix} \succeq 0 \tag{4.17}$$

This off-line solution does not honor any constraints in the system, and this is where (Kothare et al., 1994) enter the scene. Instead of solving the problem off-line, a related semidefinite program is solved on-line for each $x_{k|k}$. At time k, we calculate a feedback to be used in the future $u_{k+j} = L_k x_{k+j}$. A natural performance measure is the upper bound on the infinite horizon cost using this feedback law, $\gamma_k = x_{k|k}^T P_k x_{k|k}$. We now have to ensure that the control law $L_k x_k$ is feasible, and will remain feasible, with respect to control and state constraints. To do this, we first note that P_k defines an invariant ellipsoid for the (unconstrained) control law $u_{k+j} = L_k x_{k+j}$, since $x_k^T P_k x_k$ is a Lyapunov function according to (4.14). We thus know that the control law will keep us in the ellipsoid $x^T P_k x \leq \gamma_k$. Ensuring that the constraints are satisfied in this ellipsoid can be written as

$$\max_{x^T P_k x \leq \gamma_k} E_x x \;\leq\; f_x \tag{4.18a}$$

$$\max_{x^T P_k x \leq \gamma_k} E_u L_k x \;\leq\; f_u \tag{4.18b}$$

It can be shown that the conservative choice $W_k = \gamma_k P_k^{-1}$ together with Theorem 3.2 enables us to write the complete problem as the following semidefinite program (Kothare et al., 1994) (the notation $(\cdot)_i$ means the ith row).

$$\begin{aligned}
&\min_{W_k, K_k, \gamma_k} \quad \gamma_k \\
&\text{subject to} \begin{pmatrix} W_k & (A^{(i)} W_k + B^{(i)} K_k)^T & W_k & K_k^T \\ A^{(i)} W + B^{(i)} K_k & W_k & 0 & 0 \\ W_k & 0 & \gamma_k Q^{-1} & 0 \\ K_k & 0 & 0 & \gamma_k R^{-1} \end{pmatrix} \succeq 0 \\
&\begin{pmatrix} ((f_u)_i)^2 & (E_u K_k)_i \\ ((E_u K_k)_i)^T & W_k \end{pmatrix} \succeq 0 \\
&\begin{pmatrix} ((f_x)_i)^2 & (E_x)_i \\ ((E_x)_i)^T & W_k \end{pmatrix} \succeq 0 \\
&\begin{pmatrix} 1 & x_{k|k}^T \\ x_{k|k} & W_k \end{pmatrix} \succeq 0
\end{aligned}$$

This semidefinite program is solved at each sample, and the control input to the system is $u_{k|k} = L_k x_{k|k} = K_k W_k^{-1} x_{k|k}$. A strong feature of the approach is that stability follows by construction, but there are some drawbacks. To begin with, we are parameterizing the control sequence in terms of a linear state feedback. This is clearly a conservative choice. Consider for instance when the true optimal solution is constant over a number of future samples. This can never be obtained with a linear state feedback. Another major problem is state and control constraints. These are treated using ellipsoidal arguments, and the outcome of this is that we cannot treat asymmetric constraints such as $u_k \geq 0$.

The results were later extended in various directions with LFT models, tracking formulations, observer-based control, vanishing disturbances and more (Kothare et al., 1996).

The ideas lay the foundation for a number of extensions, including systems with measurable uncertainties (Lu and Arkan, 2000) and uncertainties with a bounded rate of change (Casavola et al., 2002a). Conservativeness of the linear state feedback parameterization is one of the main arguments against the framework. One approach towards a stronger parameterization is (Cuzzola et al., 2002) where a more flexible Lyapunov function is used. Another reason for conservativeness of the original approach in (Kothare et al., 1994) is that the method essentially uses a horizon $N = 0$ and calculates a terminal state weight on-line. This has been improved upon in (Casavola et al., 2000) and (Schuurmans and Rossiter, 2000) where a finite horizon cost has been appended to the problem, and the methods in (Kothare et al., 1994) are used for optimizing terminal state weight and constraint. This gives substantial improvements in some cases, but leads to problems with exponential complexity. Finally, we should also mention that the framework has been used as a tool for nonlinear MPC (Bloemen and van den Boom, 1999; Bloemen et al., 2000; Wu, 2001).

Enumerative Schemes and Dynamic Programming

The original minimax formulation with worst-case peak minimization in (Campo and Morari, 1987) was based on a pure enumerative scheme. The same has been done also for more general problems. As we saw above, if it can be shown that the set of possible future states defines a convex set, we only need to check the extreme points of this set to find the maximum.

For polytopic models, this means that we can solve any minimax problem with a convex performance measure by minimizing a variable t subject to the constraints

$$\ell(x_{k|k}, u_{k|k}, x^{(i)}_{k+1|k}, u_{k+1|k}, \dots, x^{(i)}_{k+N-1|k}, u_{k+N-1|k}) \leq t \tag{4.19}$$

where $x^{(i)}_{k+N-1|k}$ denote the (exponentially many) extreme point vertices. This scheme has been used in (Casavola et al., 2000) and (Schuurmans and Rossiter, 2000) for quadratic performance measures. Due to the exponential complexity however, the methods are only applicable to problems with short horizons and a small number of uncertain parameters.

A slightly more general minimax problem has been studied in, for example, (Lee and Yu, 1997) and (Scokaert and Mayne, 1998). The idea is to assume that measurements will be available in the future, and solve

$$\min_{u_{k|k}} \max_{w_{k|k}} \cdots \min_{u_{k+N-1|k}} \max_{w_{k+N-1|k}} \ell(x_{k|k}, u_{k|k}, \dots, x_{k+N-1|k}, u_{k+N-1|k}) \tag{4.20}$$

This type of problems, called closed-loop minimax MPC, was proposed and analyzed for polytopic systems and a quadratic performance measure in (Lee and Yu, 1997) using dynamic programming. Systems with additive disturbances and general performance measures was addressed in (Scokaert and Mayne, 1998) using a straightforward enumerative approach. Unfortunately, the results in (Lee and Yu, 1997) and (Scokaert and Mayne, 1998) are optimization problems with exponential complexity.

Feedback Predictions

One class of methods that has received attention due to its simplicity is algorithms based on feedback predictions. The idea is that we pre-stabilize the system with some suitably chosen feedback law $Lx_{k|k}$ and then optimize a bias, or perturbation, $u_{k+j|k} = Lx_{k+j|k} + v_{k+j|k}$, so that the constraints are robustly satisfied. A prototype algorithm for this scheme would be

$$\boxed{\begin{array}{llll} \min\limits_{v} & \sum_{j=0}^{N-1} v_{k+j|k}^T v_{k+j|k} & & \\ \text{subject to} & u_{k+j|k} \in \mathbb{U} & \forall\Delta \in \mathbf{\Delta} \\ & x_{k+j|k} \in \mathbb{X} & \forall\Delta \in \mathbf{\Delta} \end{array}} \tag{4.21}$$

We see immediately that when the constraints can be guaranteed with the feedback law $u_{k+j|k} = Lx_{k+j|k}$, the bias $v_{k+j|k}$ will be zero, hence giving us our pre-defined linear state feedback. One of the benefits with this formulation is that the performance measure is detached from the minimax problem. All uncertainty lies in the control and state constraints. The idea has been used for polytopic uncertainties in (Lee and Kouvaritakis, 2000; Chisci et al., 2001), and additive disturbances (Bemporad, 1998; Chisci and Zappa, 1999; Bemporad and Garulli, 2000).

Explicit off-line Solutions

One of the most interesting recent ideas is to calculate explicit solutions to the minimax problems off-line. It has recently been shown that nominal MPC controllers can be written explicitly as piecewise affine feedback laws (Bemporad et al., 2002a; Bemporad et al., 2002b). These affine control laws are found using multi-parametric programming, i.e., parameterization of the solution of an optimization problem with respect to some variable, in our case the state $x_{k|k}$. These ideas can in some cases be extended to minimax formulations.

Systems with external disturbances and linear performance measures are addressed in (Bemporad et al., 2001), (Borrelli, 2002), (Kerrigan and Mayne, 2002) and (Sato et al., 2002). A quadratic performance measure and external disturbances are used in (Kakalis et al., 2002), however, only a nominal performance measure is minimized, and the uncertainties are only addressed in the constraints. Explicit solutions of minimax MPC with quadratic performance measures seem to be an unresolved problem.

4.3 Stability

The discussion in the previous section focused on the choice of uncertainty models, performance measures, and the related minimax formulations. As in nominal MPC, special considerations have to be taken to actually guarantee robust stability.

The following theorem is a direct extension of Theorem 2.1. The proof, and some additional technical assumptions, can be found in (Mayne et al., 2000).

Theorem 4.2 (Robustly stabilizing minimax MPC)
Suppose the following assumptions hold for a system with uncertainty $\Delta \in \mathbf{\Delta}$, a nominal controller $L(x)$, a terminal state domain $\mathbb{X}_T$, a stage cost $\ell(x_k, u_k)$ and a terminal state weight $\Psi(x)$

A1. $x_{k+1} = f(x_k, u_k, \Delta_k)$

A2. $0 \in \mathbb{X}_T \subseteq \mathbb{X}$

A3. $f(x_k, L(x_k), \Delta_k) \in \mathbb{X}_T \ \forall x_k \in \mathbb{X}_T, \forall \Delta \in \mathbf{\Delta}$

A4. $\ell(0,0) = 0,\ \ell(x,u) > 0 \ \forall \ (x,u) \neq 0$

A5. $\Psi(0) = 0,\ \Psi(x) > 0 \ \forall \ x \neq 0$

A6. $\Psi(f(x, L(x), \Delta)) - \Psi(x) \leq -\ell(x, L(x)) \ \forall x \in \mathbb{X}_T,\ \forall \Delta \in \mathbf{\Delta}$

A7. $L(x) \in \mathbb{U} \ \forall x \in \mathbb{X}_T$

Then, assuming feasibility at the initial state, an MPC controller using the following minimax formulation will guarantee asymptotic stability

$$\boxed{\begin{array}{ll} \min\limits_{u} \max\limits_{\Delta} & \sum_{j=0}^{N-1} \ell(x_{k+j|k}, u_{k+j|k}) + \Psi(x_{k+N|k}) \\ \text{subject to} & u_{k+j|k} \in \mathbb{U} \quad \forall \Delta \in \mathbf{\Delta} \\ & x_{k+j|k} \in \mathbb{X} \quad \forall \Delta \in \mathbf{\Delta} \\ & x_{k+N|k} \in \mathbb{X}_T \quad \forall \Delta \in \mathbf{\Delta} \end{array}} \tag{4.22}$$

The interpretation is the same as in the nominal case. We append the performance measure with a terminal state weight that serves as an upper bound on the worst-case infinite horizon cost using the controller $L(x)$. This bound holds only if the nominal controller is unconstrained, hence the optimization problem has to be appended with a terminal state constraint, to ensure that the end state reaches a positively invariant domain $\mathbb{X}_T$ where all constraints indeed are satisfied.

Unfortunately, we will not be able to use this theorem in this thesis. The reasons for this will be discussed later. However, the basic idea in the theorem (terminal state weight based on worst-case infinite horizon cost) will be used to create controllers with good practical performance. For guaranteed stability, we will be forced to employ more conservative approaches based on contraction constraints, as in, e.g., (Zheng, 1995; Badgwell, 1997).

4.4 Summary

From the brief survey, we see that there is a considerable amount of proposed methods to deal with uncertainty in MPC. Algorithms exist for the most important uncertainty structures such as additive bounded external disturbances, polytopic uncertainties and LFT models. The tools to solve the problems range from simple linear programs, almost as efficiently formulated as the nominal problems, to LMI based methods and off-line solutions using multi-parametric optimization.

An open problem seems to be minimax formulations with quadratic performance measures. The approaches available are essentially approximate methods based on upper bounds using linear state feedback parameterizations, and exact but enumerative schemes with exponential complexity. The problems with the quadratic performance measure are reasonable, since we in many cases try to solve special cases of well known NP-hard problems. Consider for instance a minimax formulation with a quadratic performance measure and an external disturbance $||w_k||_\infty \leq 1$. Solving the maximization part boils down to maximization of a convex quadratic function in the unit-cube. This is a well-known NP-hard problem (Vavasis, 1991). The same holds for the case $||w_k||_2 \leq 1$. This leads to maximization of a convex quadratic function over the intersection of ellipsoids. Also this is a NP-hard problem (follows immediately since $w_k \in \mathbb{R}$ gives maximization over the unit-cube).

Despite these obviously inherent problems with the quadratic performance measures, we will devote the main part of this thesis to this formulation.

Part II

Minimax MPC

5

MPC FOR SYSTEMS WITH ADDITIVE DISTURBANCES

This chapter focuses on MPC applied to systems with bounded external disturbances. These disturbances can have a physical meaning, such as worst-case loads on a robot arm, leakage in a tank system or similar. An alternative is to consider fictitious disturbances used in a more pragmatic way to regularize the solutions in the optimization problems in MPC to obtain a less sensitive control law.

The main contribution in this chapter is that we show how a minimax MPC problem for systems with external disturbances, and a finite horizon quadratic performance measure, can be efficiently addressed using semidefinite relaxations. The novelty compared to earlier work is that we can work with the classical quadratic performance measure in a rather general framework. Previous approaches have typically abandoned the quadratic performance measure and introduced new cost functions for which the minimax problems become tractable (Chisci and Zappa, 1999; Bemporad and Garulli, 2000; Bemporad et al., 2001; Sato et al., 2002), or resorted to computationally intractable solutions (Scokaert and Mayne, 1998).

Since this is the first chapter in this thesis where we encounter the concept semidefinite relaxations, and its use in minimax MPC, the material in this chapter will be slightly more detailed compared to the forthcoming chapters where similar ideas are used. Much notation introduced in this chapter will be used throughout the thesis.

5.1 Uncertainty Model

The class of systems we address in this chapter is linear discrete-time systems with external disturbances.

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Gw_k && (5.1a) \\ y_k &= Cx_k && (5.1b) \end{aligned}$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, $y_k \in \mathbb{R}^p$ and $w_k \in \mathbb{R}^r$ denote the state, control input, controlled output and external disturbance respectively. Furthermore, the system is constrained, $u_k \in \mathbb{U}$ and $x_k \in \mathbb{X}$. The constraint sets $\mathbb{U}$ and $\mathbb{X}$ are assumed to be polyhedrons.

The disturbance w_k is only known to be bounded in some measure, but otherwise unknown. The set of possible disturbances is denoted $\mathbb{W}$.

$$w_k \in \mathbb{W} \qquad (5.2)$$

The disturbance set $\mathbb{W}$ is one of the ingredients that determine the type of optimization problem we end up with. The two models that will be used in this chapter are ball-constrained and box-constrained disturbances.

$$\begin{aligned} \mathbb{W}_2 &= \{w : ||w||_2 \le 1\} && (5.3a) \\ \mathbb{W}_\infty &= \{w : ||w||_\infty \le 1\} && (5.3b) \end{aligned}$$

These two standard models can be used to define more complex disturbance models, and most results in this chapter can be extended to those models without too much effort. For the sake of clarity and notational simplicity, we omit these extensions.

5.2 Minimax MPC

The goal in this chapter is to derive an MPC controller that explicitly considers the external disturbances. As we saw in the previous chapter, the standard way to do this is to look at worst-case scenarios, which translates into solving a minimax problem.

One of the main ideas with the algorithms that are developed in this thesis is that they should be as close as possible to the original nominal MPC formulation. Changing from a nominal to a worst-case performance measure should not force you to leave the classical framework with finite horizon quadratic performance measures. Hence, the following minimax problem is used.

$$\boxed{\begin{array}{llll} \min\limits_{u} \max\limits_{w} & \sum_{j=0}^{N-1} y_{k+j|k}^T Q y_{k+j|k} + u_{k+j|k}^T R u_{k+j|k} & & \\ \text{subject to} & u_{k+j|k} \in \mathbb{U} & \forall w \in \mathbb{W} & \\ & x_{k+j|k} \in \mathbb{X} & \forall w \in \mathbb{W} & \\ & w_{k+j|k} \in \mathbb{W} & & \end{array}} \qquad (5.4)$$

To obtain a convenient notation, define stacked versions of the predicted outputs, states, inputs, and unknown disturbance realization.

$$Y = \begin{pmatrix} y_{k|k}^T & y_{k+1|k}^T & \dots & y_{k+N-1|k}^T \end{pmatrix}^T \tag{5.5a}$$
$$X = \begin{pmatrix} x_{k|k}^T & x_{k+1|k}^T & \dots & x_{k+N-1|k}^T \end{pmatrix}^T \tag{5.5b}$$
$$U = \begin{pmatrix} u_{k|k}^T & u_{k+1|k}^T & \dots & u_{k+N-1|k}^T \end{pmatrix}^T \tag{5.5c}$$
$$W = \begin{pmatrix} w_{k|k}^T & w_{k+1|k}^T & \dots & w_{k+N-1|k}^T \end{pmatrix}^T \tag{5.5d}$$

Since $w \in \mathbb{W}$, we readily obtain

$$W \in \mathbb{W}^N = \mathbb{W} \times \mathbb{W} \times \cdots \times \mathbb{W} \tag{5.6}$$

The predicted states and outputs depend linearly on the current state, the future control input and the disturbance. Hence, the following relations hold.

$$Y = \mathcal{C}X \tag{5.7a}$$
$$X = \mathcal{A}x_{k|k} + \mathcal{B}U + \mathcal{G}W \tag{5.7b}$$

The matrices $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are given by Equation (2.10), whereas $\mathcal{G} \in \mathbb{R}^{Nn \times Nr}$ is defined as

$$\mathcal{G} = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ G & 0 & 0 & \dots & 0 \\ AG & G & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A^{N-2}G & \dots & AG & G & 0 \end{pmatrix} \tag{5.8}$$

Vectorization of the performance measure in (5.4) gives a more compact definition of the minimax problem (recall the definition of $\mathcal{Q}$ and $\mathcal{R}$ in Equation (2.11)).

$$\boxed{\begin{array}{llcll} \min\limits_{U} \max\limits_{W} & Y^T \mathcal{Q} Y + U^T \mathcal{R} U & & & \\ \text{subject to} & U & \in & \mathbb{U}^N & \forall W \in \mathbb{W}^N \\ & X & \in & \mathbb{X}^N & \forall W \in \mathbb{W}^N \\ & W & \in & \mathbb{W}^N & \end{array}} \tag{5.9}$$

This is the optimization problem we will attempt to solve in this chapter.

5.2.1 Semidefinite Relaxation of Minimax MPC

Guided by the discussion in Section 4.4, it is clear that one should not set ones hope too high on the prospect of solving the minimax problem exactly. Instead, we will show that replacing the original minimax problem with a conservative approximation yields a problem that can be solved efficiently.

To do this, we begin with an epigraph formulation of the objective function

$$\boxed{\begin{array}{lrcll} \min\limits_{U,t} & t & & & \\ \text{subject to} & Y^T\mathcal{Q}Y + U^T\mathcal{R}U & \leq & t & \forall W \in \mathbb{W}^N \\ & U & \in & \mathbb{U}^N & \forall W \in \mathbb{W}^N \\ & X & \in & \mathbb{X}^N & \forall W \in \mathbb{W}^N \end{array}} \tag{5.10}$$

The derivation of the main result will now be divided in two steps. We will first show how the first, performance related, constraint in (5.10) can be dealt with, and then address state and control constraints.

Bounding the Performance Measure

The following theorem will be very useful in this and forthcoming chapters (Zhang, 1999).

Theorem 5.1 (Non-strict Schur complement)
If $W \succ 0$, then for any $X \succeq 0$

$$X - Z^TW^{-1}Z \succeq 0 \Leftrightarrow \begin{pmatrix} X & Z^T \\ Z & W \end{pmatrix} \succeq 0$$

Applying a Schur complement on the first constraint in (5.10) transforms the uncertain quadratic constraint to an uncertain LMI in t and U (remember that Y is linearly parameterized in U and W).

$$\begin{pmatrix} t & Y^T & U^T \\ Y & \mathcal{Q}^{-1} & 0 \\ U & 0 & \mathcal{R}^{-1} \end{pmatrix} \succeq 0 \quad \forall W \in \mathbb{W}^N \tag{5.11}$$

Inserting the definition of Y and separating certain and uncertain terms shows that the uncertain LMI can be written as[1]

$$\begin{pmatrix} t & (\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U))^T & U^T \\ \star & \mathcal{Q}^{-1} & 0 \\ \star & 0 & \mathcal{R}^{-1} \end{pmatrix} + \begin{pmatrix} 0 \\ \mathcal{C}\mathcal{G} \\ 0 \end{pmatrix} W \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} + (\star) \succeq 0 \tag{5.12}$$

[1] The notation $P + S + (\star)$ denotes $P + S + S^T$. This convention will be used throughout this thesis in order to save space. In a similar fashion, a $\star$ inside a matrix is short for an element defined by symmetry.

The next step is to apply Theorem 3.4 which deals with robust satisfaction of uncertain LMIs. To do this, we have to transform our problem into the format used in the theorem.

Let us begin with the case $w_k \in \mathbb{W}_2$. First, write the vector W in terms of a block diagonal uncertainty which we call Δ.

$$W = (\oplus_{j=0}^{N-1} w_{k+j|k})\mathbf{1}^N = ((\oplus_{j=0}^{N-1} w_{k+j|k}^T))^T\mathbf{1}^N = \Delta^T\mathbf{1}^N \tag{5.13}$$

This means that we define Δ as a block diagonal matrix with $w_{k+j|k}^T$ as the diagonal blocks[2]. Furthermore, define

$$\tilde{F} = \begin{pmatrix} t & (\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U))^T & U^T \\ \star & \mathcal{Q}^{-1} & 0 \\ \star & 0 & \mathcal{R}^{-1} \end{pmatrix}, \tilde{R} = \begin{pmatrix} 0 & \mathcal{G}^T\mathcal{C}^T & 0 \end{pmatrix}, \tilde{L} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} (\mathbf{1}^N)^T$$

This enables us to write the LMI in (5.12) as

$$\tilde{F} + \tilde{L}\Delta\tilde{R} + \tilde{R}^T\Delta^T\tilde{L}^T \succeq 0 \tag{5.14}$$

Theorem 3.4 can now be used directly and we obtain a multiplier $\tau \in \mathbb{R}_+^N$ and associated matrices $\mathcal{S} = \oplus_1^N \tau_i$ and $\mathcal{T} = \oplus_1^N \tau_i I^{r\times r}$. According to the theorem, feasibility of the following LMI is a sufficient condition for (5.12) to be robustly satisfied.

$$\begin{pmatrix} \tilde{F} - \tilde{L}\mathcal{S}\tilde{L}^T & \tilde{R}^T \\ \tilde{R} & \mathcal{T} \end{pmatrix} \succeq 0 \tag{5.15}$$

Straightforward calculations shows that the matrix $\tilde{L}\mathcal{S}\tilde{L}^T$ evaluates to

$$\tilde{L}\mathcal{S}\tilde{L}^T = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} (\mathbf{1}^N)^T(\oplus_1^N \tau_i)\mathbf{1}^N \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^N \tau_j & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{5.16}$$

Inserting the definition of $\tilde{F}$ and $\tilde{R}$ gives us our final LMI.

$$\begin{pmatrix} t - \sum_{i=1}^N \tau_i & (\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U))^T & U^T & 0 \\ \mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U) & \mathcal{Q}^{-1} & 0 & \mathcal{C}\mathcal{G} \\ U & 0 & \mathcal{R}^{-1} & 0 \\ 0 & (\mathcal{C}\mathcal{G})^T & 0 & \mathcal{T} \end{pmatrix} \succeq 0 \tag{5.17}$$

The case with $w_k \in \mathbb{W}_\infty$ is treated in a similar fashion. The only difference is that we now obtain a diagonal matrix Δ when we place the independent disturbances in W in a diagonal matrix.

$$W = (\oplus_{j=1}^{Nr} W_i)\mathbf{1}^{Nr} = \Delta^T\mathbf{1}^{Nr} \tag{5.18}$$

[2]The reason we have transposed Δ related to W is that this will give us a semidefinite relaxation with a simple structure. Transposing the definition of the uncertainty decides the choice of L and R in Theorem 3.4.

The changes in the semidefinite relaxation using Theorem 3.4 is that we now have $\tau \in \mathbb{R}_+^{Nr}$ and $\mathcal{S} = \mathcal{T} = \oplus_1^{Nr}\tau_i$. The semidefinite relaxation is now

$$\begin{pmatrix} t - \sum_{i=1}^{Nr}\tau_i & (\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U))^T & U^T & 0 \\ \mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U) & \mathcal{Q}^{-1} & 0 & \mathcal{C}\mathcal{G} \\ U & 0 & \mathcal{R}^{-1} & 0 \\ 0 & (\mathcal{C}\mathcal{G})^T & 0 & \mathcal{T} \end{pmatrix} \succeq 0 \tag{5.19}$$

Hence, the only difference is the structure of the matrix $\mathcal{T}$ and the number of τ variables. For notational convenience in the remainder of this chapter, we introduce a variable s to denote the number of multipliers, i.e., s is either N or Nr depending on the uncertainty structure. Furthermore, we will not explicitly derive the semidefinite relaxations or describe the structure of $\mathcal{T}$. The relaxations will look almost the same throughout this chapter and s, τ, $\mathcal{S}$ and $\mathcal{T}$ follow immediately from the uncertainty structure.

Robust Constraint Satisfaction

First note that the control constraints $E_u u_{k+j|k} \leq f_u$ are unaffected by the disturbances and do not have to be addressed at the moment.

The predicted states however, depend on the disturbances and have to be addressed. Robust constraint satisfaction means that (definitions of $\mathcal{E}_x$ and $\mathcal{F}_x$ are given in (2.12))

$$\mathcal{E}_x X \leq \mathcal{F}_x \quad \forall W \in \mathbb{W}^N \tag{5.20}$$

Inserting the definition (5.7) of X into (5.20) yields

$$\mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}U) + \mathcal{E}_x\mathcal{G}W \leq \mathcal{F}_x \quad \forall W \in \mathbb{W} \tag{5.21}$$

What have to be done now is to maximize each row in the uncertain term $\mathcal{E}_x\mathcal{G}W$. Partition the rows of the matrix $\mathcal{E}_x\mathcal{G}$ as

$$\mathcal{E}_x\mathcal{G} = \begin{pmatrix} \omega_1^T \\ \omega_2^T \\ \vdots \end{pmatrix}, \quad \omega_i^T = \begin{pmatrix} \omega_{i1}^T & \omega_{i2}^T & \dots & \omega_{iN}^T \end{pmatrix} \tag{5.22}$$

The dimension of $\omega_{ij}^T \in \mathbb{R}^{1\times r}$ is chosen so that the ith row of $\mathcal{E}_x\mathcal{G}W$ can be written

$$(\mathcal{E}_x\mathcal{G}W)_i = \omega_i^T W = \sum_{j=1}^{N} \omega_{ij}^T w_{k+j-1|k} \tag{5.23}$$

Introduce a vector γ and let

$$\gamma_i \quad = \quad \max_{W\in\mathbb{W}^N} \omega_i^T W = \max_{w\in\mathbb{W}} \sum_{j=1}^{N} \omega_{ij}^T w_{k+j-1|k} \tag{5.24}$$

The state constraints are then robustly satisfied if

$$\mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}U) + \gamma \leq \mathcal{F}_x \tag{5.25}$$

Notice that the computational complexity is unchanged, since the nominal linear state constraints are transformed to another set of linear constraints.

The constant vector γ depends on the uncertainty structure and the gain $\mathcal{E}_x\mathcal{G}$. For the element-wise bounded disturbances $w \in \mathbb{W}_\infty$, the maximization is performed using Theorem 3.1.

$$\begin{aligned} \gamma_i &= \max_{w \in \mathbb{W}_\infty} \omega_i^T W & (5.26)\\ &= ||\omega_i||_1 & (5.27)\end{aligned}$$

The maximization for $w \in \mathbb{W}_2$ is done with Theorem 3.2.

$$\begin{aligned} \gamma_i &= \max_{w \in \mathbb{W}_2} \sum_{j=1}^{N} \omega_{ij}^T w_{k+j-1|k} \\ &= \sum_{j=1}^{N} ||\omega_{ij}||_2 & (5.28)\end{aligned}$$

The Complete Semidefinite Program

At this point, we are able to summarize our findings in an optimization problem that solves a conservative approximation of the minimax problem (5.4)

$$\begin{array}{ll} \min\limits_{U,t,\tau} & t \\ \text{subject to} & \begin{pmatrix} t - \sum_{i=1}^{s} \tau_i & (\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U))^T & U^T & 0 \\ \mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U) & \mathcal{Q}^{-1} & 0 & \mathcal{C}\mathcal{G} \\ U & 0 & \mathcal{R}^{-1} & 0 \\ 0 & (\mathcal{C}\mathcal{G})^T & 0 & \mathcal{T} \end{pmatrix} \succeq 0 \\ & \mathcal{E}_u U \leq \mathcal{F}_u \\ & \mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}U) + \gamma \leq \mathcal{F}_x \end{array} \tag{5.29}$$

This is a rather clean result, and we will now take matters one step further by looking at possible (and necessary) extensions.

5.3 Extensions

One of the main features of the framework that we develop in this thesis is that it easily allows extensions. The most important ones are discussed below.

5.3.1 Feedback Predictions

The minimax controller we have developed can easily become conservative. The reason is that we are optimizing an open-loop control sequence that has cope with all possible future disturbance realizations, without taking future measurements into account. In other words, we have thrown away the knowledge that we are applying a receding horizon, and will recalculate the control sequence at the next sample instant.

What we really would like to solve is the closed-loop minimax program where we incorporate the notion that measurements will be obtained (Scokaert and Mayne, 1998).

$$\min_{u_{k|k}} \max_{w_{k|k}} \cdots \min_{u_{k+N-1|k}} \max_{w_{k+N-1|k}} \sum_{j=0}^{N-1} y_{k+j|k}^T Q y_{k+j|k} + u_{k+j|k}^T R u_{k+j|k} \tag{5.30}$$

Instead of solving this (intractable) problem, the idea in feedback predictions, sometimes referred to as closed-loop predictions, is to a introduce new decision variables $v_{k+j|k}$, in some references denoted perturbations or bias terms, and parameterize the future control sequence in the future states and $v_{k+j|k}$.

$$u_{k+j|k} = Lx_{k+j|k} + v_{k+j|k} \tag{5.31}$$

This way, we say that there is at least some kind of feedback in the system, although not optimal. This remedy is a standard concept and is used in several minimax MPC schemes, see, e.g., (Bemporad, 1998; Schuurmans and Rossiter, 2000; Chisci et al., 2001). We will return to feedback predictions in Chapter 7.

To incorporate feedback predictions in our framework, write the feedback predictions in a vectorized form.

$$U = \mathcal{L}X + V \tag{5.32}$$

where V is a stacked version of $v_{k+j|k}$ and $\mathcal{L}$ is a block diagonal matrix

$$V = \begin{pmatrix} v_{k|k}^T & v_{k+1|k}^T & \cdots & v_{k+N-1|k}^T \end{pmatrix}^T \tag{5.33a}$$

$$\mathcal{L} = \oplus_{j=1}^{N} L \tag{5.33b}$$

Of course, nothing prevents us from using a matrix $\mathcal{L}$ with a larger degree of freedom, i.e., different feedback matrices along the diagonal, or feedback terms also in the lower triangular part of the matrix $\mathcal{L}$. The only requirement is that the matrix is causal in the sense that $u_{k+j|k}$ only depends on $x_{k+i|k},\ i \leq j$.

The predictions are now defined by a set of coupled equations.

$$Y = \mathcal{C}X \tag{5.34a}$$

$$X = \mathcal{A}x_{k|k} + \mathcal{B}U \tag{5.34b}$$

$$U = \mathcal{L}X + V \tag{5.34c}$$

Solving this gives a parameterization linear in V and W.

$$Y = \mathcal{C}(I-\mathcal{B}\mathcal{L})^{-1}(\mathcal{A}x_{k|k}+\mathcal{B}V+\mathcal{G}W) \quad (5.35a)$$
$$X = (I-\mathcal{B}\mathcal{L})^{-1}(\mathcal{A}x_{k|k}+\mathcal{B}V+\mathcal{G}W) \quad (5.35b)$$
$$U = \mathcal{L}(I-\mathcal{B}\mathcal{L})^{-1}(\mathcal{A}x_{k|k}+\mathcal{B}V+\mathcal{G}W)+V \quad (5.35c)$$

It will now be shown that feedback predictions fit nicely into our minimax problem in Section 5.2.1. To begin with, define

$$\Omega = (I-\mathcal{B}\mathcal{L})^{-1} \quad (5.36)$$

The LMI in (5.11) can in the new variables be written as

$$\begin{pmatrix} t & (\mathcal{C}\Omega(\mathcal{A}x_{k|k}+\mathcal{B}V+\mathcal{G}W))^T & (\mathcal{L}\Omega(\mathcal{A}x_{k|k}+\mathcal{B}V+\mathcal{G}W)+V)^T \\ \star & \mathcal{Q}^{-1} & 0 \\ \star & 0 & \mathcal{R}^{-1} \end{pmatrix} \succeq 0 \quad (5.37)$$

As before, this is an uncertain LMI due to W. Separate certain and uncertain terms.

$$\begin{pmatrix} t & (\mathcal{C}\Omega(\mathcal{A}x_{k|k}+\mathcal{B}V))^T & (\mathcal{L}\Omega(\mathcal{A}x_{k|k}+\mathcal{B}V))^T \\ \star & \mathcal{Q}^{-1} & 0 \\ \star & 0 & \mathcal{R}^{-1} \end{pmatrix} + \begin{pmatrix} 0 \\ \mathcal{C}\Omega\mathcal{G} \\ \mathcal{L}\Omega\mathcal{G} \end{pmatrix} W \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} + (\star) \succeq 0$$

and apply exactly the same procedure as in Section 5.2.1 to obtain our semidefinite relaxation of the minimax problem.

$$\begin{pmatrix} t-\sum_{i=1}^{s}\tau_i & (\mathcal{C}\Omega(\mathcal{A}x_{k|k}+\mathcal{B}V))^T & (\mathcal{L}\Omega(\mathcal{A}x_{k|k}+\mathcal{B}V)+V)^T & 0 \\ \star & \mathcal{Q}^{-1} & 0 & \mathcal{C}\Omega\mathcal{G} \\ \star & 0 & \mathcal{R}^{-1} & \mathcal{L}\Omega\mathcal{G} \\ 0 & \star & \star & \mathcal{T} \end{pmatrix} \succeq 0 \quad (5.38)$$

We see that we have the same type of LMI as in the case without feedback predictions. Of course, s, τ and $\mathcal{T}$ are defined exactly as in Section 5.2.1 and follow immediately from $\mathbb{W}$.

One thing is however fundamentally different compared to the case without feedback predictions. From (5.35), we see that the future control sequence is uncertain. The reason is that the control constraints are mapped into state constraints due to the feedback term. The control and state constraints can be written as

$$\begin{pmatrix} \mathcal{E}_x\Omega(\mathcal{A}x_{k|k}+\mathcal{B}V+\mathcal{G}W) \\ \mathcal{E}_u(\mathcal{L}\Omega(\mathcal{A}x_{k|k}+\mathcal{B}V+\mathcal{G}W)+V) \end{pmatrix} \leq \begin{pmatrix} \mathcal{F}_x \\ \mathcal{F}_u \end{pmatrix} \quad \forall W \in \mathbb{W} \quad (5.39)$$

Or equivalently,

$$\begin{pmatrix} \mathcal{E}_x\Omega(\mathcal{A}x_{k|k}+\mathcal{B}V) \\ \mathcal{E}_u(\mathcal{L}\Omega(\mathcal{A}x_{k|k}+\mathcal{B}V)+V) \end{pmatrix} + \begin{pmatrix} \mathcal{E}_x\Omega\mathcal{G} \\ \mathcal{E}_u\mathcal{L}\Omega\mathcal{G} \end{pmatrix} W \leq \begin{pmatrix} \mathcal{F}_x \\ \mathcal{F}_u \end{pmatrix} \quad \forall W \in \mathbb{W} \quad (5.40)$$

This uncertain constraint can be dealt with in the same way as the state constraint in the previous section, and the result is a new set of linear constraints.

$$\begin{pmatrix} \mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}V) \\ \mathcal{E}_u(\mathcal{L}\Omega(\mathcal{A}x_{k|k} + \mathcal{B}V) + V) \end{pmatrix} + \gamma \leq \begin{pmatrix} \mathcal{F}_x \\ \mathcal{F}_u \end{pmatrix} \tag{5.41}$$

The only difference is that γ now is derived using $\begin{pmatrix} \mathcal{E}_x\Omega\mathcal{G} \\ \mathcal{E}_u\mathcal{L}\Omega\mathcal{G} \end{pmatrix}$ instead of $\mathcal{E}_x\mathcal{G}$.

For future reference, define the semidefinite program to solve the relaxation of the minimax MPC problem with feedback predictions.

$$\boxed{\begin{array}{ll} \min\limits_{U,t,\tau} & t \\ \text{subject to} & (5.38), (5.41) \end{array}} \tag{5.42}$$

We should mention that feedback predictions introduce a new tuning knob in minimax MPC, the feedback matrix L. The choice of L is not obvious, and this will be discussed in detail in the examples, and even more in Chapter 7 where we try to incorporate L as a free variable in the optimization problem.

5.3.2 Tracking

In practice, the control objective is often to have the output y_k follow a desired reference trajectory. This can also be addressed in our minimax framework, and the derivation is straightforward so we just state the results for future reference.

As a performance measure, we use the deviation from the desired future output $y^r_{k+j|k}$ and control input $u^r_{k+j|k}$.

$$\min_u \max_w \sum_{j=0}^{N-1} ||y_{k+j|k} - y^r_{k+j|k}||^2_Q + ||u_{k+j|k} - u^r_{k+j|k}||^2_R \tag{5.43}$$

As usual, we vectorize everything

$$Y_r = \begin{pmatrix} y^r_{k|k} & y^r_{k+1|k} & \dots & y^r_{k+N-1|k} \end{pmatrix}^T \tag{5.44a}$$

$$U_r = \begin{pmatrix} u^r_{k|k} & u^r_{k+1|k} & \dots & u^r_{k+N-1|k} \end{pmatrix}^T \tag{5.44b}$$

Applying the same techniques as before gives the following semidefinite program (assuming for notational simplicity the case without feedback predictions).

$$\boxed{\begin{array}{lrcl} \min\limits_{U,t,\tau} & t & & \\ \text{subject to} & \begin{pmatrix} t - \sum_{i=1}^{s}\tau_i & (\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U) - Y_r)^T & (U - U_r)^T & 0 \\ \star & \mathcal{Q}^{-1} & 0 & \mathcal{C}\mathcal{G} \\ \star & 0 & \mathcal{R}^{-1} & 0 \\ 0 & \star & 0 & \mathcal{T} \end{pmatrix} & \succeq & 0 \\ & \mathcal{E}_u U & \leq & \mathcal{F}_u \\ & \mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}U) + \gamma & \leq & \mathcal{F}_x \end{array}}$$

5.3.3 Stability Constraints

Nominal MPC controllers are not stabilizing *per se*, but require additional tricks to guarantee stability. Of course, uncertainty in the system will add additional difficulties when we try to guarantee or establish stability.

Weakness of the Proposed Framework

Unfortunately, the proposed framework has a weakness that makes it hard to obtain stability guarantees. Of course, to begin with, asymptotic stability can never be asked for, since there are external unknown non-vanishing disturbances acting on the system. The main flaw however is the use of a sufficient, not necessary, condition to deal with the maximization in the minimax problems.

The stability theorem (4.2) is based on the idea that if we are able to satisfy a terminal state constraint at time k, we will be able to satisfy it also at $k+1$, if the terminal state domain is suitably chosen according to the assumptions in the theorem. Consider now the most common terminal state domain, an ellipsoid. This means that we have a terminal state constraint of the type

$$x_{k+N|k}^T P x_{k+N|k} \leq 1 \quad \forall W \in \mathbb{W}^N$$

In our framework, we would take care of this constraint with a Schur complement and application of Theorem 3.4, and obtain a sufficient condition.

The problem now is that feasibility of the semidefinite relaxation of the terminal state constraint at time k, although it indeed implies feasibility at $k+1$ for the original problem, says nothing about whether our sufficient condition based on the semidefinite relaxation will be feasible. The same problem applies to the use of the minimax objective function as a Lyapunov function to prove stability.

Towards Stability Guarantees Anyway

Let us at least show that it is possible to derive a theorem to guarantee stability, i.e., boundness of the states. The theorem is clearly overly conservative and should probably not be used in practice. It should mostly be seen upon as a crowd pleaser for the theoretically inclined readers.

Theorem 5.2 (Guaranteed stability)
Assume that there exists a linear state feedback $u_k = Lx_k$ and an ellipsoid[3] $\mathbb{E}_P$ such that, with $u_k = Lx_k$ and $x_k \in \mathbb{E}_P$, it holds that $x_{k+1} \in \mathbb{E}_P \ \forall w_k \in \mathbb{W}$. Furthermore, $Lx_k \in \mathbb{U} \ \forall x_k \in \mathbb{E}_P$ and $\mathbb{E}_P \subseteq \mathbb{X}$. Appending the minimax problem (5.42) with the constraint $x_{k+1|k} \in \mathbb{E}_P$ and using feedback predictions based on L guarantees stability if the problem is initially feasible for $x_{0|0}$.

Proof The proof follows by induction. Assume the problem was feasible for $x_{k-1|k-1}$. Since the problem was feasible for $x_{k-1|k-1}$, we know that $x_{k|k} \in \mathbb{E}_P$. A feasible solution

[3] $\mathbb{E}_P$ is used throughout this thesis to denote ellipsoidal sets $\mathbb{E}_P = \{x : x^T Px \leq 1\}$ $(P \succ 0)$.

at time k is $V = 0$. To see this, recall that the state predictions with feedback predictions and $V = 0$ are given by $x_{k+j+1|k} = (A + BL)x_{k+j|k} + Gw_{k+j|k}$. From the invariance assumption on L and P, these predictions are contained in $\mathbb{E}_P$, guaranteeing feasibility of state constraints and control constraints. The new constraint $x_{k+1|k} \in \mathbb{E}_P$ holds for $u_{k|k} = Lx_{k|k}$ by construction. Hence, the problem is feasible at time k. □

To use this theorem, we append our semidefinite relaxation (5.42) with

$$x_{k+1|k}^T P x_{k+1|k} \leq 1 \quad \forall w_{k|k} \in \mathbb{W} \tag{5.45}$$

The way to take care of this constraint depends on the disturbance structure.

For $w_{k|k} \in \mathbb{W}_\infty$, it is both sufficient and necessary to check all the vertices $\{w^{(1)}, w^{(2)}, \ldots, w^{(2^r)}\}$ of the unit cube $\mathbb{W}^N$. This is easily seen using Theorem 4.1. We have $w_{k|k} \in \mathbf{Co}\{w^{(i)}\}$, hence $x_{k+1|k} \in \mathbf{Co}\{Ax_{k|k} + Bu_{k|k} + Gw^{(i)}\}$. The function $x_{k+1|k}^T P x_{k+1|k}$ is convex, so the theorem applies.

The condition (5.45) is efficiently written as 2^r second order cone constraints.

$$||P^{1/2}(Ax_{k|k} + Bu_{k|k} + Gw^{(i)})|| \leq 1 \tag{5.46}$$

Exponentially many constraints is of course a drawback, so the method only applies to systems with few scalar disturbances. This is the price we have to pay to obtain an exact condition.

The second case is $w_k \in \mathbb{W}_2$. Apply a Schur complement on (5.45) and the result is an uncertain LMI.

$$\begin{pmatrix} 1 & (Ax_{k|k} + Bu_{k|k} + Gw_{k|k})^T \\ \star & P^{-1} \end{pmatrix} \succeq 0 \quad \forall w_k \in \mathbb{W}_2 \tag{5.47}$$

Apply Theorem 3.4 which now is both necessary and sufficient. Introduce a scalar $\tau \in \mathbb{R}_+$ and the theorem tells us that (5.45) is equivalent to feasibility of

$$\begin{pmatrix} 1-\tau & (Ax_{k|k} + Bu_{k|k})^T & 0 \\ \star & P^{-1} & G \\ 0 & \star & \tau I \end{pmatrix} \succeq 0 \tag{5.48}$$

To summarize; appending the semidefinite program (5.42) with the constraint (5.48) (or (5.46) depending on the uncertainty structure) will guarantee stability, assuming the problem is initially feasible for $x_{0|0}$ and the matrix L is used to define the feedback predictions.

All that has to be done now is to find an ellipsoid $\mathbb{E}_P$ and a linear state feedback matrix L satisfying the assumptions. This is a standard robust linear state feedback problem, and the details can be found in Appendix 5.A.

5.4 Simulation Results

We conclude this chapter with a couple of numerical experiments to study the behavior of the proposed minimax controller. All controllers are implemented using YALMIP and the semidefinite programs are solved using SEDUMI.

We start with an example to show how the minimax controller compares to a nominal controller.

Example 5.1 (Comparison with nominal MPC)
This example is adapted from (Bemporad and Garulli, 2000). The system is described with the following model.

$$\begin{aligned} x_{k+1} &= \begin{pmatrix} 1.64 & -0.79 \\ 1 & 0 \end{pmatrix} x_k + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u_k + \begin{pmatrix} 0.25 \\ 0 \end{pmatrix} w_k \\ y_k &= \begin{pmatrix} 0.14 & 0 \end{pmatrix} x_k \end{aligned}$$

The disturbance $w_k \in \mathbb{W}_\infty$ can be interpreted as an actuator disturbance. The control objective is to transfer the output y_k to the reference level 1, under the control constraint $|u_k| \leq 2$ and an output constraint $-1 \leq p_k \leq 3$ where

$$p_k = \begin{pmatrix} -1.93 & 2.21 \end{pmatrix} x_k$$

The open-loop step-response of the system can be seen in Figure 5.1. The system

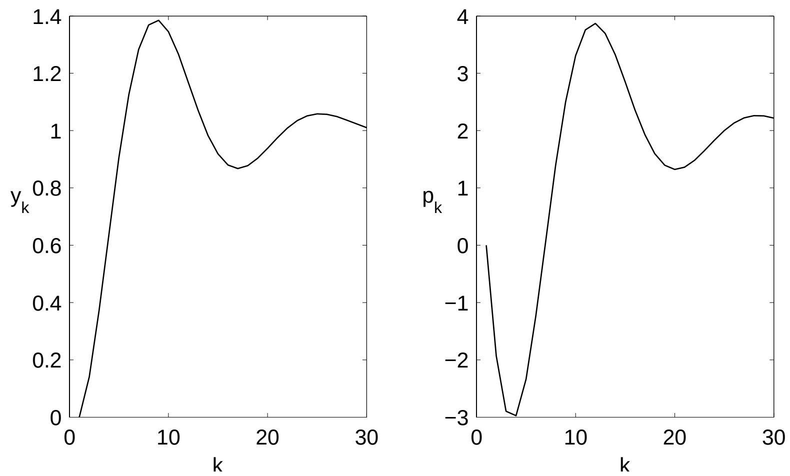

Figure 5.1: Open-loop step-response for the system. The output y_k has an oscillatory response, but the most important observation is the severe non-minimum phase behavior of the constrained output p_k.

exhibits a severe non-minimum phase behavior to the constrained output p_k, and this is the main difficulty when the system is controlled.

From Figure 5.1, we conclude that a reasonable prediction horizon is $N = 10$. The main objective is to control the output y_k, so it was decided to use $Q = 1$ and $R = 0.01$. The minimax controller (5.42) is implemented, with obvious changes for the tracking formulation in Section 5.3.2. The feedback matrix L in (5.31) is chosen as an LQ controller calculated using $Q = R = 1$. The motivation for this choice will be discussed in Example 5.2. The simulation starts from $x_0 = 0$, and the disturbance w_k was a random signal uniformly distributed between -1 and 1.

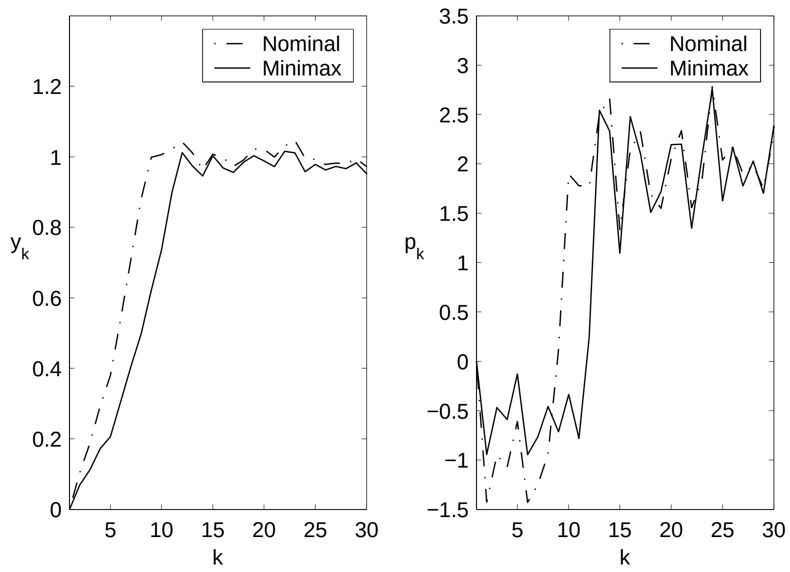

Figure 5.2: Closed-loop response for nominal MPC and the proposed minimax controller. The minimax controller satisfies the output constraints during the whole step-response, in contrast to the nominal controller.

The closed-loop responses for the proposed minimax controller and a nominal MPC controller are given in Figure 5.2. The minimax controller is successful in keeping the constrained output p_k within its limits, in contrast to the nominal controller[4]. The price paid is a slower step-response in the controlled output y_k.

Feedback predictions were added to the minimax controller in Section 5.3.1 to reduce the level of conservativeness. Unfortunately, the choice of the feedback matrix L in (5.31) is not obvious, and no simple guidelines are available. One might believe that the feedback matrix from the related LQ problem would be a good candidate, but this is not necessarily the case. The reason is that feedback predictions introduce a trade-off between uncertainty in the future states and uncertainty in the future control inputs. We illustrate these problems with an example.

[4]Feasibility was recovered by removing the first output constraint (i.e. the constraint on $p_{k|k}$)

Example 5.2 (Feedback predictions)

We continue on the same system as in Example 5.1. To see the impact of feedback predictions, we create a number of minimax controllers, using feedback predictions based on LQ controllers calculated with $Q = 1$ and R ranging from 0.001 to 10. All other numerical data are the same as in Example 5.1

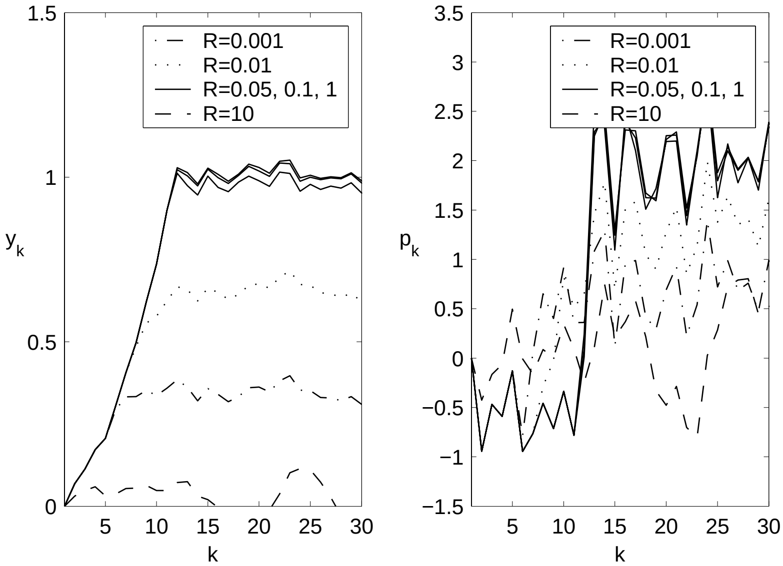

Figure 5.3: The figure closed-loop responses for the minimax controller, using different feedback predictions. The feedback predictions are based on LQ controllers, calculated using $Q = 1$ and different values on the control weight.

Figure 5.3 illustrates the problems with feedback predictions. The minimax controllers using feedback predictions based on aggressive LQ controllers (R small), fails miserably. The reason is that the future control sequence becomes too uncertain in the predictions. The steady-state control at the reference level $y_k = 1$ is $u_k = 1$. When the minimax controller approaches the steady-state level, the predicted control input is approximately 1 plus uncertainty due to uncertain predictions. If the feedback controller has too large gain, the additional uncertainty will be large and the worst-case control will exceed the upper bound on 2. The only solution for the minimax controller is to position the output y_k at a lower level where the steady-state control input is lower.

On the other hand, making R large enough does not solve the problem either. If R is chosen too large, the feedback controller will be cautious, and the result is too uncertain state predictions. The predictions of the constrained variable p_k

becomes too uncertain, and no control action can be taken without violating the output constraints. The result is that the system gets stuck in the origin. This can seen in Figure 5.3 for $R = 10$.

The problems with the parameterization of the feedback predictions have motivated the work in Chapter 7.

For $w_k \in \mathbb{W}_\infty$, we can solve the minimax problem (5.10) exactly by brute-force enumeration. Introducing the vertices of the set $\mathbb{W}^N$ allows us to write $W \in \mathbf{Co}\{W^{(1)}, W^{(2)}, \ldots, W^{(2^{N_r})}\}$. Our predictions can then be written as

$$X = \mathcal{A}x_{k|k} + \mathcal{B}U + \mathcal{G}W \in \mathbf{Co}\{\mathcal{A}x_{k|k} + \mathcal{B}U + \mathcal{G}W^{(i)}\} \tag{5.49}$$

Hence, the prediction set is convex. Since our quadratic performance measure is convex, Theorem 4.1 tells us that we can find the maximum by looking at the vertices of the prediction set. Our minimax problem (5.10) can thus be solved exactly with the following program (for efficient implementation in SeDuMi, we write the performance constraints using second order cone constraints)

$$\begin{array}{lrcl}
\min\limits_{U,t} & t & & \\
\text{subject to} & \left\| \begin{array}{c} 2\mathcal{Q}^{1/2}\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U + \mathcal{G}W^{(i)}) \\ 2\mathcal{R}^{1/2}U \\ 1-t \end{array} \right\| & \leq & 1+t \\
& U & \in & \mathbb{U}^N \quad \forall W \in \mathbb{W}^N \\
& X & \in & \mathbb{X}^N \quad \forall W \in \mathbb{W}^N
\end{array} \tag{5.50}$$

Control and state constraints, feedback predictions and tracking formulations are dealt with easily using the same methods as for the proposed minimax controller. The details are omitted for brevity. Let us use this exact solution to evaluate the quality of the approximation obtained using semidefinite relaxations.

Example 5.3 (Quality of semidefinite relaxations)

We continue on the same system as in the previous example, and implement both our minimax controller (5.42), and a minimax controller based on the exact solution (5.50). The numerical data is the same as in Example 5.1, except for one crucial difference. The prediction horizon is now reduced to $N = 6$. The reason is that the original horizon $N = 10$ gives a too large problem (recall that we have to introduce 2^N second order cone constraints in the exact solution given by (5.50)). The closed-loop responses for the two controllers are depicted in Figure 5.4. From the closed-loop responses, it seems like the semidefinite relaxation does a very god job. The two controllers give essentially identical step-responses.

An alternative way to evaluate the quality of the semidefinite relaxation is to study the value of the upper bound t in the minimax problem (5.42), and compare it to the exact value obtained by solving (5.50). Let us denote the optimal value from

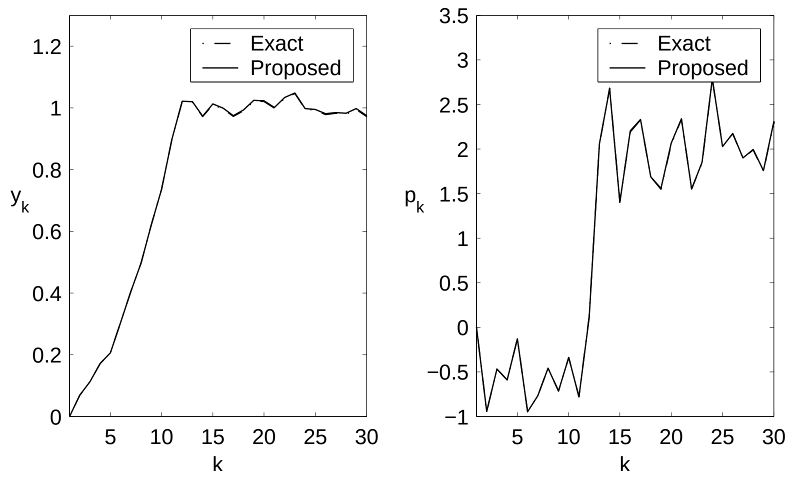

Figure 5.4: Closed-loop response for the exact minimax MPC controller and the proposed minimax controller based on semidefinite relaxations. The responses are essentially indistinguishable.

the semidefinite relaxation t_{SDP} and the value from the exact solution t_{exact}. A reasonable way to compare these numbers is to look at the ratio $\alpha = \frac{t_{SDP}}{t_{exact}}$. In theory, this measure is always larger than or equal to 1. A small α indicates that the semidefinite relaxation introduces little conservativeness, while a large α indicates a poor approximation. When the closed-loop response for the proposed minimax controller was calculated, the exact problem was also solved at each state. This enabled us to calculate the quality measure α, and the result is depicted in Figure 5.5. The semidefinite relaxation never produced an upper bound on the worst-case finite horizon cost that was more than 5 percent higher than the true cost. Notice the extremely good quality on the approximation for the first 7 samples.

Of course, no general conclusions can be drawn from this single experiment. However, the numbers are consistent with other simulations that have been performed. The semidefinite relaxation tends to give an upper bound somewhere between 0 and 10 percent higher than the true worst-case cost.

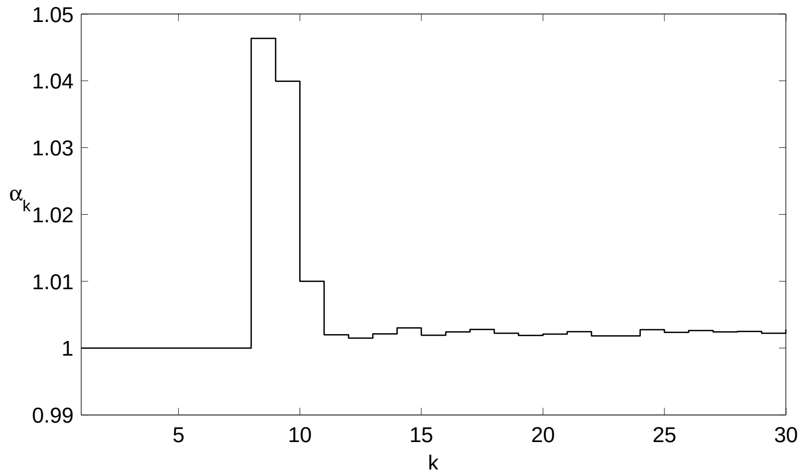

Figure 5.5: The figure shows the ratio between the upper bound on the worst-case finite horizon cost obtained with the semidefinite relaxation, and the exact worst-case finite horizon cost

Appendix

5.A Robustly Invariant Ellipsoid

An ellipsoid $\mathbb{E}_P$ is said to be robustly invariant with respect to disturbances $w_k \in \mathbb{W}$ if $x_{k+1} \in \mathbb{E}_P \ \forall x_k \in \mathbb{E}_P, w_k \in \mathbb{W}$. In other words

$$\begin{aligned} x_{k+1}^T P x_{k+1} &\leq 1 && \text{(5.A.51a)} \\ \text{when} & \\ x_k^T P x_k &\leq 1 && \text{(5.A.51b)} \\ w_k &\in \mathbb{W} && \text{(5.A.51c)} \end{aligned}$$

Assume to begin with that $\mathbb{W} = \mathbb{W}_2$. Inserting the control law $u_k = Lx_k$ gives the closed-loop model $x_{k+1} = (A + BL)x_k + Gw_k$. The invariance constraint (5.A.51) can be written as

$$\begin{pmatrix} x_k \\ w_k \\ 1 \end{pmatrix}^T \begin{pmatrix} (A+BL)^T P (A+BL) & (A+BL)^T PG & 0 \\ G^T P (A+BL) & G^T PG & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_k \\ w_k \\ 1 \end{pmatrix} \leq 0 \quad \text{(5.A.52a)}$$

$$\text{when} \quad \begin{pmatrix} x_k \\ w_k \\ 1 \end{pmatrix}^T \begin{pmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_k \\ w_k \\ 1 \end{pmatrix} \leq 0 \quad \text{(5.A.52b)}$$

$$\begin{pmatrix} x_k \\ w_k \\ 1 \end{pmatrix}^T \begin{pmatrix} P & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_k \\ w_k \\ 1 \end{pmatrix} \leq 0 \quad \text{(5.A.52c)}$$

Straightforward application of the S-procedure (Theorem 3.3) yields a sufficient condition for L to render the ellipsoid $\mathbb{E}_P$ robustly invariant. We introduce two scalars $\tau_1, \tau_2 \in \mathbb{R}_+$ and obtain

$$\begin{pmatrix} (A+BL)^T P(A+BL) & (A+BL)^T PG & 0 \\ G^T P(A+BL) & G^T PG & 0 \\ 0 & 0 & -1 \end{pmatrix} \preceq \tau_1 \begin{pmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & -1 \end{pmatrix} + \tau_2 \begin{pmatrix} P & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \preceq 0 \quad (5.A.53)$$

This is a bilinear matrix inequality (BMI) due to the multiplication of τ_2 and P, but by fixating τ_2, an LMI is obtained (if we assume for the moment that the feedback matrix L is given). Optimization of P, or more precisely some measure of P related to the size of the invariant ellipsoid $\mathbb{E}_P$, can therefore be done with bisection in τ_2. However, the BMI can be written in a better format. First, notice that we can split the BMI into two constraints

$$\tau_1 + \tau_2 \leq 1 \quad (5.A.54a)$$

$$\begin{pmatrix} \tau_2 P & 0 \\ 0 & \tau_1 I \end{pmatrix} - \begin{pmatrix} (A+BL)^T \\ G^T \end{pmatrix} P \begin{pmatrix} (A+BL) & G \end{pmatrix} \succeq 0 \quad (5.A.54b)$$

A congruence transformation with $\begin{pmatrix} P^{-1} & 0 \\ 0 & I \end{pmatrix}$ turns (5.A.54b) into

$$\begin{pmatrix} \tau_2 P^{-1} & 0 \\ 0 & \tau_1 I \end{pmatrix} - \begin{pmatrix} P^{-1}(A+BL)^T \\ G^T \end{pmatrix} P \begin{pmatrix} (A+BL)P^{-1} & G \end{pmatrix} \succeq 0 \quad (5.A.55)$$

Define $W = P^{-1}$ and $K = LP^{-1}$ and perform a Schur complement

$$\begin{pmatrix} \tau_2 W & 0 & WA^T + K^T B^T \\ 0 & \tau_1 I & G^T \\ AW + BK & G & W \end{pmatrix} \succeq 0 \quad (5.A.56)$$

Obviously, we still have a BMI due to the product $\tau_2 W$, but in the W variable two things are gained. To begin with, a MAXDET problem can be solved (for fixed τ_2) to maximize the volume of the robustly invariant ellipsoid. Furthermore, the matrix K can be a decision variable, hence allowing us to optimize the feedback L.

Control and state constraints have to be satisfied in the invariant ellipsoid. This can be solved with Theorem 3.2. Consider the ith row in $E_u u_k \leq f_u$

$$\begin{aligned} \max_{x_k^T P x_k \leq 1} (E_u)_i u_k &= \max_{x_k^T P x_k \leq 1} (E_u L)_i x_k \\ &= \sqrt{(E_u L)_i P^{-1}((E_u L)_i)^T} \leq f_i \end{aligned} \quad (5.A.57)$$

Squaring the constraint[5] and applying a Schur complement yields

$$\begin{pmatrix} ((f_u)_i)^2 & (E_u L)_i \\ ((E_u L)_i)^T & P \end{pmatrix} \succeq 0 \quad (5.A.58)$$

[5]We assume that we have symmetric constraints, which ensures $f_u \geq 0$. Otherwise, ellipsoidal methods do not apply.

A congruence transformation with $\begin{pmatrix} I & 0 \\ 0 & P^{-1} \end{pmatrix}$ gives us an LMI in W and K

$$\begin{pmatrix} ((f_u)_i)^2 & ((E_u K)_i)^T \\ (E_u K)_i & W \end{pmatrix} \succeq 0 \tag{5.A.59}$$

Doing the same procedure for the state constraints also yield an LMI. To summarize, finding, e.g., a maximum volume[6] invariant ellipsoid, with corresponding linear feedback matrix L, is solved with

$$\begin{array}{ll} \max\limits_{W,K,\tau_1,(\tau_2)} & \det W \\ \text{subject to} & \begin{pmatrix} \tau_2 W & 0 & WA^T + K^T B^T \\ 0 & \tau_1 I & G^T \\ AW + BK & G & W \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} ((f_u)_i)^2 & (E_u K)_i \\ ((E_u K)_i)^T & W \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} ((f_x)_i)^2 & (E_x)_i \\ ((E_x)_i)^T & W \end{pmatrix} \succeq 0 \end{array} \tag{5.A.60}$$

The difference when we have $w_k \in \mathbb{W}_\infty$ instead is that we replace the term $\tau_1 I$ with $\mathrm{diag}(\tau)$ where $\tau \in \mathbb{R}^r_+$.

[6] The volume of $\mathbb{E}_P$ is proportional to $\det P^{-1/2} = \sqrt{\det P^{-1}}$ (Vandenberghe et al., 1998)

6

Joint state estimation and control in minimax MPC

In practice, the state estimate $x_{k|k}$ and the true state x_k never coincide. Instead, disturbed measurements are used to obtain a state estimate $x_{k|k}$. This motivates an extension of the minimax MPC controllers from Chapter 5.

Incorporating the state estimate in a minimax controller can essentially be done in three ways. The first and most straightforward solution is to neglect the estimation errors and just use the estimate $x_{k|k}$ as if it was the true state. If explicit bounds on the state estimate errors can be derived, more advanced methods are possible. The idea is to pose the minimax problem over both the disturbances and the bounded estimation error. Using this strategy, the control input will depend on how certain the current state estimate is, certainly a reasonable feature in a robust control problem. The error bounds can be incorporated in the controller in two ways, as described below.

In this chapter, the material in Chapter 5 is extended to cope with bounded state estimate errors. The results are two-fold. To begin with, it is shown that *a priori* bounds on estimation errors easily can be incorporated in the proposed minimax MPC schemes. It is also shown how joint state estimation and minimax MPC can be cast as a semidefinite program involving a (unfortunately) quadratic matrix inequality. The nonconvex quadratic matrix inequality can be conservatively approximated as a linear matrix inequality and thus enable an algorithm to approximately solve the joint problem using semidefinite programming.

A related approach for minimax MPC with both estimation errors and disturbances can be found in (Bemporad and Garulli, 2000). The difference compared to the work in this chapter is the choice of state estimator and the performance measure in the MPC controller, and the extension to joint state estimation and control. The joint estimation and control problem does not seem to have been addressed elsewhere in a minimax MPC framework.

6.1 Uncertainty Model

The class of systems addressed is linear discrete-time systems with external system and measurement disturbances.

$$x_{k+1} = Ax_k + Bu_k + Gw_k \tag{6.1a}$$
$$y_k = Cx_k \tag{6.1b}$$
$$h_k = C_h x_k + D_h \xi_k \tag{6.1c}$$

Hence, the difference compared to Chapter 5 is the measurement h_k. The disturbances are assumed to be unknown but bounded.

$$\xi_k \in \Xi = \{\xi : \xi^T \xi \leq 1\} \tag{6.2a}$$
$$w_k \in \mathbb{W}_2 = \{w : w^T w \leq 1\} \tag{6.2b}$$

This can be generalized to other models, such as $w_k \in \mathbb{W}_\infty$, but these generalizations are omitted for brevity.

Since only a disturbed output variable is measured, a state estimator has to be used. Regardless of how this is done, the true state x_k, the estimated state $x_{k|k}$ and the state estimation error e_k are related according to

$$x_k = x_{k|k} + e_k \tag{6.3}$$

The estimator used in this chapter gives a state estimate with a guaranteed ellipsoidal bound on the estimation error.

$$e_k^T P_k e_k \leq 1 \tag{6.4}$$

The matrix P_k is termed the confidence matrix and is an output from the state estimation procedure, which now will be discussed in detail.

6.2 Deterministic State Estimation

What is an optimal state estimate in a minimax framework? Clearly, the best choice is to find the smallest set $\mathbb{X}_k$ such that

$$x_k \in \mathbb{X}_k \tag{6.5}$$

can be guaranteed, given all inputs and measurements obtained since startup, and some prior knowledge on the initial state $x_0 \in \mathbb{X}_0$. Loosely speaking, the problem is to find the solution to

$$\min \text{Size}(\mathbb{X}_k) \text{ given } \mathbb{X}_0, u_0, h_1, u_1, h_2, \dots u_{k-1}, h_k$$

The crux is that this is not practically implementable, even for our simple model. The problem is that the complexity of the set $\mathbb{X}_k$ grows when more measurements are obtained (Schweppe, 1973).

The standard way to overcome this problem is to restrict $\mathbb{X}_k$ to have some predefined geometry, such as ellipsoidal (Schweppe, 1968; Schweppe, 1973; El Ghaoui and Calafiore, 1999) or parallelotopic (Chisci et al., 1996). Furthermore, a recursive scheme is employed. Unfortunately, assuming $\mathbb{X}_{k-1}$ to have a particular geometry does not imply that $\mathbb{X}_k$ has the same geometry. Hence, by forcing $\mathbb{X}_k$ to be, e.g., an ellipsoid, some approximation will inevitably occur.

When this approximation is introduced, there will be some degree of freedom, and it will be shown in this chapter how this degree of freedom can be used to improve the performance in a minimax MPC controller.

6.2.1 Ellipsoidal State Estimates

An ellipsoidal approximation of the set $\mathbb{X}_k$ is used in this chapter. The ellipsoidal state estimation problem can be stated as follows; given a guaranteed ellipsoidal bound on the previous estimation error $e_{k-1} = x_{k-1} - x_{k-1|k-1}$

$$x_{k-1} \in \mathbb{X}_{k-1} = \{x_{k-1} : e_{k-1}^T P_{k-1} e_{k-1} \leq 1\} \tag{6.6}$$

and a new measurement h_k, use the model (6.1) and the disturbance bounds (6.2) to find a new state estimate guaranteed to satisfy

$$x_k \in \mathbb{X}_k = \{x_k : e_k^T P_k e_k \leq 1\} \tag{6.7}$$

The estimation problem can conceptually be cast as

$$\begin{array}{lrcl} \min\limits_{P_k, x_{k|k}} & \text{Size}(\mathbb{X}_k) & & \\ \text{subject to} & x_k & = & Ax_{k-1} + Bu_{k-1} + Gw_{k-1} \\ & x_k & \in & \mathbb{X}_k \quad \forall x_{k-1} \in \mathbb{X}_{k-1},\ w_{k-1} \in \mathbb{W}_2 \\ & h_k & = & C_h x_k + D_h \xi_k,\ \forall \xi_k \in \Xi \end{array} \tag{6.8}$$

This problem has been addressed in essentially two different ways in the literature.

Ellipsoidal State Estimation using Ellipsoidal Calculus

Deterministic state estimation with ellipsoidal confidence regions can be viewed upon as a geometric problem and solved using ellipsoidal calculus (Schweppe, 1968; Schweppe, 1973; Kurzhanski and Vályi, 1997). Explicit expressions for feasible $x_{k|k}$ and P_k can be derived using these methods, and some analytic expressions for optimal choices in various measures are available.

Ellipsoidal State Estimation using Semidefinite Programming

An alternative to ellipsoidal calculus is to address the optimization problem (6.8) directly. The advantage with this approach is that more general results are possible, such as extensions to more advanced uncertainty models. Moreover, the optimization approach fits better in our framework.

It will now be shown how (6.8) can be cast as a semidefinite program, and to do this, we use the ideas in (El Ghaoui and Calafiore, 1999), with straightforward modifications to fit the model (6.1).

The constraints in (6.8) are first written as an implication. A state estimate $x_{k|k}$ and confidence matrix P_k are feasible if

$$\begin{aligned}
e_k^T P_k e_k &\leq 1 && \text{(6.9a)}\\
&\text{when} \\
e_{k-1}^T P_{k-1} e_{k-1} &\leq 1 && \text{(6.9b)}\\
w_{k-1}^T w_{k-1} &\leq 1 && \text{(6.9c)}\\
\xi_k^T \xi_k &\leq 1 && \text{(6.9d)}\\
h_k &= C_h x_k + D_h \xi_k && \text{(6.9e)}\\
x_k &= A x_{k-1} + B u_{k-1} + G w_{k-1} && \text{(6.9f)}
\end{aligned}$$

As a first step towards a semidefinite program, a basis for all the involved variables is defined. To this end, define the auxiliary variable z.

$$z = \begin{pmatrix} x_{k-1}^T & w_{k-1}^T & \xi_k^T & 1 \end{pmatrix}^T \tag{6.10}$$

This enables us to write

$$\begin{aligned}
e_k &= T_{e_k} z = \begin{pmatrix} A & G & 0 & Bu_{k-1} - x_{k|k} \end{pmatrix} z && \text{(6.11a)}\\
e_{k-1} &= T_{e_{k-1}} z = \begin{pmatrix} I & 0 & 0 & -x_{k-1|k-1} \end{pmatrix} z && \text{(6.11b)}\\
h_k - (C_h x_k + D_h \xi_k) &= T_{h_k} z = \begin{pmatrix} -C_h A & -C_h G & -E & y_k - C_h B u_{k-1} \end{pmatrix} && \text{(6.11c)}\\
w_k &= T_{w_{k-1}} z = \begin{pmatrix} 0 & I & 0 & 0 \end{pmatrix} && \text{(6.11d)}\\
\xi_k &= T_{\xi_k} z = \begin{pmatrix} 0 & 0 & I & 0 \end{pmatrix} && \text{(6.11e)}\\
1 &= T_1 z = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix} && \text{(6.11f)}
\end{aligned}$$

The implication (6.9) can now be written as

$$\begin{aligned}
z^T T_{e_k}^T P_k T_{e_k} z &\leq z^T T_1^T T_1 z && \text{(6.12a)}\\
&\text{when} \\
z^T T_{e_{k-1}}^T P_{k-1} T_{e_{k-1}} z &\leq z^T T_1^T T_1 z && \text{(6.12b)}\\
z^T T_{w_{k-1}}^T T_{w_{k-1}} z &\leq z^T T_1^T T_1 z && \text{(6.12c)}\\
z^T T_{\xi_k}^T T_{\xi_k} z &\leq z^T T_1^T T_1 z && \text{(6.12d)}\\
z^T T_{h_k}^T T_{h_k} z &= 0 && \text{(6.12e)}
\end{aligned}$$

Application of the S-procedure (Theorem 3.3) gives a sufficient condition for this to hold. Introduce three non-negative scalars θ_e, θ_w and θ_ξ to relax the three inequalities, and the indefinite scalar θ_h is to relax the equality constraint[1].

$$(T_1^T T_1 - T_{e_k}^T P_k T_{e_k}) \geq \theta_e (T_1^T T_1 - T_{e_{k-1}}^T P_{k-1} T_{e_{k-1}}) + \theta_w (T_1^T T_1 - T_{w_{k-1}}^T T_{w_{k-1}}) + \theta_\xi (T_1^T T_1 - T_{\xi_k}^T T_{\xi_k}) + \theta_h T_{h_k}^T T_{h_k} \tag{6.13}$$

Define a matrix Γ to simplify notation.

$$\Gamma = T_1^T T_1 - \theta_e (T_1^T T_1 - T_{e_{k-1}}^T P_{k-1} T_{e_{k-1}}) - \theta_w (T_1^T T_1 - T_{w_{k-1}}^T T_{w_{k-1}}) - \theta_\xi (T_1^T T_1 - T_{\xi_k}^T T_{\xi_k}) - \theta_h T_{h_k}^T T_{h_k} \tag{6.14}$$

The matrix equality (6.13) simplifies to

$$\Gamma - T_{e_k}^T P_k T_{e_k} \succeq 0 \tag{6.15}$$

A Schur complement gives the final expression

$$\begin{pmatrix} \Gamma & T_{e_k}^T \\ T_{e_k} & P_k^{-1} \end{pmatrix} \succeq 0 \tag{6.16}$$

Having this sufficient condition is the first step in a state estimation procedure. The next step is to select a particular solution $x_{k|k}$ and P_k^{-1}. To do this, some performance measure on P_k^{-1} is minimized under the constraint (6.16). A typical choice (El Ghaoui and Calafiore, 1999) is the trace, $\text{Tr}P_k^{-1}$.

$$\begin{array}{ll} \min\limits_{P_k^{-1}, x_{k|k}, \theta} & \text{Tr}P_k^{-1} \\ \text{subject to} & \begin{pmatrix} \Gamma & T_{e_k}^T \\ T_{e_k} & P_k^{-1} \end{pmatrix} \succeq 0 \end{array} \tag{6.17}$$

The optimization problem can be simplified by eliminating the variable θ_h according to (Boyd et al., 1994). For notational convenience, we omit this simplification.

6.3 Minimax MPC with State Estimation Errors

The goal in this section is to explain how the ellipsoidal state estimate described in the previous section can be used in a minimax MPC controller.

The minimax problem can be cast in two different ways, depending on whether it is assumed that the state estimate is available when the minimax problem is solved, or if the estimation problem is solved jointly with the control problem.

The two problems are derived in a similar way and the obtained optimization problems look almost the same, but the complexity in solving them differs substantially. We begin with the simple where the estimation problem and the control problem are solved separately, and then extend this to the joint problems.

[1] Multipliers for equalities in the S-procedure do not have to be positive.

Separate Estimation and Control

Assume that the estimation problem (6.17) has been solved and returned a state estimate $x_{k|k}$ and a matrix P_k such that $e_k^T P_k e_k \leq 1$. Using the same notation as in Chapter 5, the predicted output is given by

$$\begin{aligned} Y &= \mathcal{C}X &(6.18\text{a})\\ X &= \mathcal{C}(\mathcal{A}x_k + \mathcal{B}U + \mathcal{G}W) \\ &= \mathcal{C}(\mathcal{A}(x_{k|k} + e_k) + \mathcal{B}U + \mathcal{G}W) &(6.18\text{b})\end{aligned}$$

The state estimation error can be written in terms of a normalized estimation error z_k ($P_k^{-1/2}$ denotes the symmetric square root of P_k^{-1})

$$e_k = P_k^{-1/2} z_k, \quad z_k \in \mathbb{Z}_2 = \{z_k : z_k^T z_k \leq 1\} \tag{6.19}$$

Insert this definition of e_k in (6.18) to obtain

$$X \;=\; \mathcal{A}x_{k|k} + \mathcal{B}U + \mathcal{A}P_k^{-1/2} z_k + \mathcal{G}W \tag{6.20}$$

The minimax MPC problem is almost the same as the problem in Chapter 5, except that estimation errors should be accounted for in the minimax formulation.

$$\boxed{\begin{array}{llll} \min\limits_{U} \max\limits_{W,z_k} & Y^T \mathcal{Q} Y + U^T \mathcal{R} U & & \\ \text{subject to} & U \in \mathbb{U}^N & \forall W \in \mathbb{W}^N, z_k \in \mathbb{Z}_2 \\ & X \in \mathbb{X}^N & \forall W \in \mathbb{W}^N, z_k \in \mathbb{Z}_2 \\ & W \in \mathbb{W}^N & \\ & z_k \in \mathbb{Z}_2 & \end{array}} \tag{6.21}$$

This can be solved using the same techniques as in Chapter 5. Rewriting the problem with an epigraph formulation, applying a Schur complement on the performance constraint, and separating uncertain and certain terms yield an uncertain LMI.

$$\begin{pmatrix} t & (\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U))^T & U^T \\ \star & \mathcal{Q}^{-1} & 0 \\ \star & 0 & \mathcal{R}^{-1} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ \mathcal{C}\mathcal{G} & \mathcal{C}\mathcal{A}P_k^{-1/2} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} W \\ z_k \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} + (\star) \succeq 0 \tag{6.22}$$

Doing exactly as in Chapter 5, but replacing the matrix $\mathcal{C}\mathcal{G}$ with $\begin{pmatrix} \mathcal{C}\mathcal{G} & \mathcal{C}\mathcal{A}P_k^{-1/2} \end{pmatrix}$ and performing the relaxation with respect to $\begin{pmatrix} W^T & z_k^T \end{pmatrix}^T \in \mathbb{W}^N \times \mathbb{Z}_2$, gives the following sufficient LMI.

$$\begin{pmatrix} t - \sum_{i=1}^{s+1} \tau_i & (\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U))^T & U^T & 0 & 0 \\ \star & \mathcal{Q}^{-1} & 0 & \mathcal{C}\mathcal{G} & \mathcal{C}\mathcal{A}P_k^{-1/2} \\ \star & 0 & \mathcal{R}^{-1} & 0 & 0 \\ 0 & \star & 0 & \mathcal{T} & 0 \\ 0 & \star & 0 & 0 & \mathcal{T}_z \end{pmatrix} \succeq 0 \tag{6.23}$$

Notice the new multiplier τ_{s+1}, and the matrix $\mathcal{T}_z = \tau_{s+1} I^{n \times n}$, introduced during the semidefinite relaxation of the normalized state estimation error z_k.

The next step concerns the state constraints. These must be satisfied for all possible disturbances and state estimation errors.

$$\mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}U + \mathcal{G}W) + \mathcal{E}_x \mathcal{A} P_k^{-1/2} z_k \leq \mathcal{F}_x \quad \forall W \in \mathbb{W}^N, z_k \in \mathbb{Z}_2 \tag{6.24}$$

This can also be solved using the methods in Chapter 5. To begin with, maximize the left-hand side with respect to W. As in (5.20)-(5.25), a constant γ is calculated and a new robustified constraint is obtained.

$$\mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}U) + \gamma + \mathcal{E}_x \mathcal{A} P_k^{-1/2} z_k \leq \mathcal{F}_x \quad \forall z_k \in \mathbb{Z}_2 \tag{6.25}$$

The next step is maximization of the left-hand side with respect to the estimation error, i.e., the uncertain term $\mathcal{E}_x \mathcal{A} P_k^{-1/2} z_k$. Let $(\mathcal{E}_x \mathcal{A})_i$ denote the ith row of the matrix $\mathcal{E}_x \mathcal{A}$, and Theorem 3.2 yields

$$\max_{||z_k|| \leq 1} (\mathcal{E}_x \mathcal{A})_i P_k^{-1/2} z_k = \sqrt{(\mathcal{E}_x \mathcal{A})_i P_k^{-1} ((\mathcal{E}_x \mathcal{A})_i)^T} = \nu_i \tag{6.26}$$

To summarize, the semidefinite relaxation of the minimax problem for systems with external disturbances and a known ellipsoidal state estimation error bound is given by the following semidefinite program.

$$\begin{array}{rl} \min\limits_{U,t,\tau,\tau_z} & t \\ \text{subject to} & \begin{pmatrix} t - \sum_{i=1}^{s} \tau_i & (\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U))^T & U^T & 0 & 0 \\ \star & \mathcal{Q}^{-1} & 0 & \mathcal{C}\mathcal{G} & \mathcal{C}\mathcal{A}P_k^{-1/2} \\ \star & 0 & \mathcal{R}^{-1} & 0 & 0 \\ 0 & \star & 0 & \mathcal{T} & 0 \\ 0 & \star & 0 & 0 & \mathcal{T}_z \end{pmatrix} \succeq 0 \\ & \mathcal{E}_u U \leq \mathcal{F}_u \\ & \mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}U) + \gamma + \nu \leq \mathcal{F}_x \end{array} \tag{6.27}$$

Joint Estimation and Control

The focus will now be turned to the main problem of this chapter, the joint solution of the estimation and the control problem. The equations involved in this problem can readily be obtained from the results above.

To begin with, the matrix inequality (6.23) is linear in $x_{k|k}$ and the matrix $P_k^{-1/2}$. Furthermore, combining (6.26) and (6.25) shows that robust satisfaction of the ith state constraint is guaranteed if

$$(\mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}U))_i + \gamma_i + \sqrt{(\mathcal{E}_x \mathcal{A})_i P_k^{-1} ((\mathcal{E}_x \mathcal{A})_i)^T} \leq (\mathcal{F}_x)_i \tag{6.28}$$

This can be written as a second order cone constraint, linearly parameterized in $x_{k|k}$, $P_k^{-1/2}$ and U.

$$||P_k^{-1/2} ((\mathcal{E}_x \mathcal{A})_i)^T|| \leq (\mathcal{F}_x)_i - (\mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}U))_i - \gamma_i \tag{6.29}$$

Finally, the state estimation procedure constrains the feasible choices of $x_{k|k}$ and P_k with the following matrix inequality.

$$\begin{pmatrix} \Gamma & T_{e_k}^T \\ T_{e_k} & P_k^{-1} \end{pmatrix} = \begin{pmatrix} \Gamma & T_{e_k}^T \\ T_{e_k} & P_k^{-1/2} P_k^{-1/2} \end{pmatrix} \succeq 0 \tag{6.30}$$

The matrix T_{e_k} is linearly parameterized in $x_{k|k}$ according to (6.11a). Unfortunately, the state estimation inequality (6.30) is not linear in $P_k^{-1/2}$, but quadratic. Anyway, define the joint estimation and minimax MPC problem (the variable ν is only introduced for notational reasons and is of course eliminated in practice).

$$\begin{array}{rrcl} \min\limits_{U,t,\tau,x_{k|k},P_k^{-1/2},\theta} & t & & \\ \text{subject to} & (6.23) & & \\ & ||P_k^{-1/2}((\mathcal{E}_x\mathcal{A})_i)^T|| & \leq & \nu_i \\ & \mathcal{F}_x - (\mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}U)) - \gamma & = & \nu \\ & \mathcal{E}_u U & \leq & \mathcal{F}_u \\ & \begin{pmatrix} \Gamma & T_{e_k}^T \\ T_{e_k} & P_k^{-1} \end{pmatrix} & \succeq & 0 \end{array} \tag{6.31}$$

This is a semidefinite problem with a bilinear matrix inequality (BMI). It is well known that general semidefinite problems with BMIs are NP-hard (Safonov et al., 1994), hence intractable for anything but trivial problems. The standard approach is to resort to local schemes based on linearizations (Goh et al., 1994).

A Tractable Approximation of the Joint Problem

To obtain a tractable problem, a scheme based on successive linearizations is proposed. Consider the following trivial inequality which holds for all compatible matrices X and Y.

$$(X - Y)^T(X - Y) \succeq 0 \tag{6.32}$$

Expanding the square gives a lower bound on the squared matrix X^TX.

$$X^TX \succeq X^TY + Y^TX - Y^TY^T \tag{6.33}$$

This can be interpreted as a linearization of the bilinear matrix expression X^TX at the point Y.

Linearize $P_k^{-1} = P_k^{-1/2}P_k^{-1/2}$ in an arbitrary point S ($S^T = S \in \mathbb{R}^{n\times n}$)

$$P_k^{-1} \succeq P_k^{-1/2}S + SP_k^{-1/2} - S^2 \tag{6.34}$$

Replacing the estimation BMI in (6.31) with the conservative approximation

$$\begin{pmatrix} \Gamma & T_{e_k}^T \\ T_{e_k} & P_k^{-1/2}S + SP_k^{-1/2} - S^2 \end{pmatrix} \succeq 0 \tag{6.35}$$

will guarantee the original BMI to be satisfied. This follows immediately from the lower bound (6.34).

$$\begin{pmatrix} \Gamma & T_{e_k}^T \\ T_{e_k} & P_k^{-1} \end{pmatrix} \succeq \begin{pmatrix} \Gamma & T_{e_k}^T \\ T_{e_k} & P_k^{-1/2}S + SP_k^{-1/2} - S^2 \end{pmatrix} \succeq 0 \tag{6.36}$$

The linearization is now used to define the following approximation of the joint estimation and control problem.

$$\begin{array}{lrcl} \min\limits_{U,t,\tau,x_{k|k},P_k^{-1/2},\theta} & t & & \\ \text{subject to} & (6.23), (6.35) & & \\ & ||P_k^{-1/2}((\mathcal{E}_x\mathcal{A})_i)^T|| & \leq & \nu_i \\ & \mathcal{F}_x - (\mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}U)) - \gamma & = & \nu \\ & \mathcal{E}_uU & \leq & \mathcal{F}_u \end{array} \tag{6.37}$$

Clearly, the main problem now is to select the linearization point S. The perhaps easiest solution is to solve the problem (6.17), and use the solution to define S. Of course, this can be repeated in order to find a local minimum. Let us define this with an algorithm.

Algorithm 6.1 (Local solution of joint estimation and control)

begin
Solve the estimation problem (6.17) to obtain an initial $P_k^{-1/2}$
repeat
Let $S := P_k^{-1/2}$
Linearize the BMI (6.30) using (6.35)
Solve semidefinite program (6.37)
until suitable stopping criteria satisfied
end

In practice, it is not be possible to perform many iterations, since each step consists of solving a semidefinite program. However, simulations indicate that one iteration suffice and give clear improvements compared to a minimax scheme where the estimation and control problems are solved separately.

6.4 Simulation Results

A simple numerical experiment is conducted to analyze the performance of the proposed minimax controllers.

Example 6.1 (Joint estimation and control)

We continue with the same system as in Example 5.1, originally taken from (Bemporad and Garulli, 2000). Since this chapter focus on how the state estimation error influence the minimax controller, we reduce the size of the disturbance significantly, in order to isolate the effects from the state estimate errors. The model, now with measurement errors, is given by

$$\begin{aligned} x_{k+1} &= \begin{pmatrix} 1.64 & -0.79 \\ 1 & 0 \end{pmatrix} x_k + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u_k + \begin{pmatrix} 0.05 \\ 0 \end{pmatrix} w_k \\ y_k &= \begin{pmatrix} 0.14 & 0 \end{pmatrix} x_k \\ h_k &= \begin{pmatrix} 0.14 & 0 \end{pmatrix} x_k + 0.05\xi_k \end{aligned}$$

All other numerical data and control objectives are the same as in Example 5.1, except that no feedback predictions are used. The reason is that we try to make the controllers as simple as possible to reduce the possible factors that influence the results.

Three different controllers were implemented. In the first controller, denoted $\mathcal{C}_0$, the state estimation is performed by solving (6.17), and the estimate is then used in the minimax controller defined by (6.27). In controller $\mathcal{C}_1$, an initial state estimate is found by solving (6.17), and the obtained confidence matrix is used to linearize the joint problem in (6.37). In other words, $\mathcal{C}_1$ implements Algorithm 6.1 with one iteration. A third controller $\mathcal{C}_2$ implements Algorithm 6.1 with two iterations. Extension of the algorithms in this chapter to incorporate reference tracking follow easily as in Section 5.3.2.

The three controllers were simulated 100 times for 25 samples, with different initial conditions, measurement error and disturbance realizations. The initial state estimate was $x_{0|0} = 0$ and $P_0 = 4I$, while the true initial state was uniformly distributed in the ellipsoid $||x_0|| \leq 0.25$ (the numbers are chosen to enable comparison with (Bemporad and Garulli, 2000)). Disturbances and measurement errors were uniformly distributed.

The mean of the accumulated quadratic performance measure,

$$\sum_{k=0}^{25} (Cx_k - 1)^T Q (Cx_k - 1) + (u_k - 1)^T R (u_k - 1)$$

was calculated and gave a cost 6.70 for $\mathcal{C}_1$, 5.63 for $\mathcal{C}_1$ and 5.39 for $\mathcal{C}_2$. The numbers are not very impressive, but it should be mentioned that the improvements are more pronounced in a model with larger external disturbances. However, $\mathcal{C}_0$ often became infeasible for larger disturbances, so a comparison would not make any sense.

Figure 6.1 shows a simulation where the proposed joint estimation and control strategy has improved the step response.

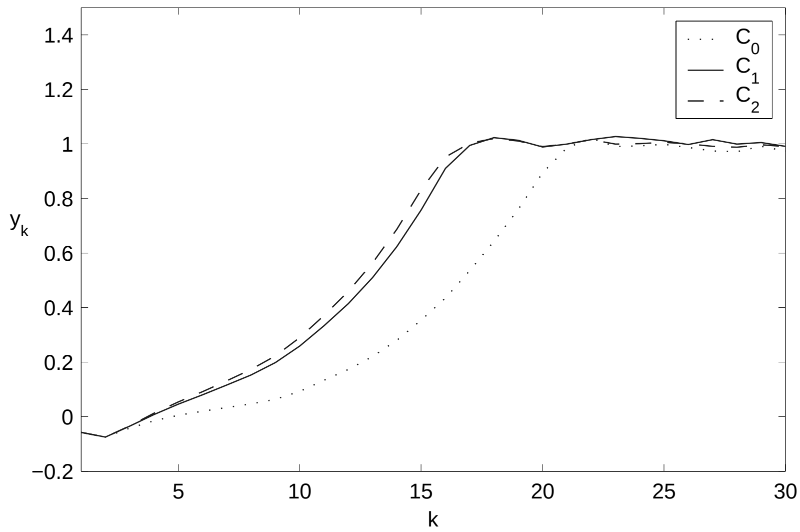

Figure 6.1: Step-responses for different minimax controllers.

The reason for the improvement in this example is the output constraint. The constrained output has a severe non-minimum phase behavior, as demonstrated in Example 5.1. If the uncertainty in the state estimate is too large, the uncertainty in the constrained output will force the controller to be cautious. Since the limiting factor for performance is the output constraint, it is important that the measurements are used to obtain an estimate that is certain along the constrained output directions. This is done automatically in the joint approach, hence leading to improved performance.

Figure 6.2 shows the evolution of the state estimate confidence regions for $\mathcal{C}_1$ (left) and $\mathcal{C}_0$ (right). The soft-shaded slab indicates the admissible region for the output constraint $-1 \leq \begin{pmatrix}-1.93 & 2.21\end{pmatrix} x_k \leq 3$. The difference is not particularly large, but one can see than the state estimates to the left are more elongated along the constrained direction. This is the reason why $\mathcal{C}_1$ and $\mathcal{C}_2$ have managed to improve the performance. Note that the state estimation procedure is not guaranteed to generate a state estimate ellipsoid contained in the admissible set for the output constraint.

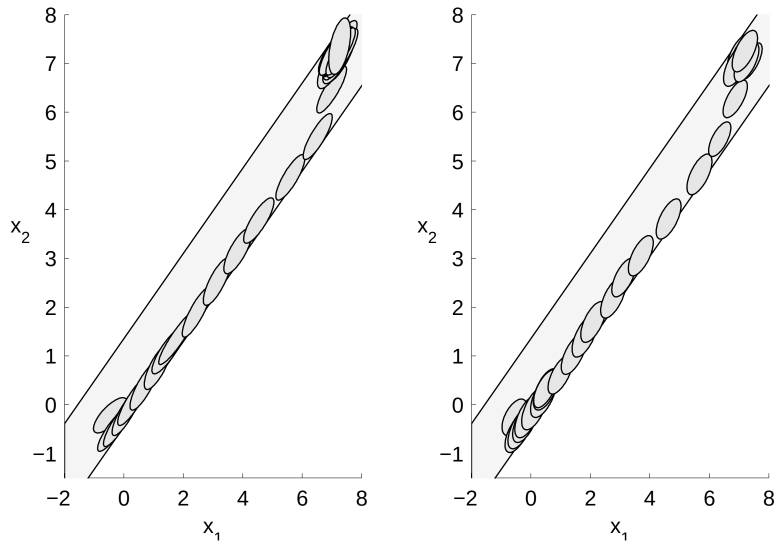

Figure 6.2: Confidence regions for state estimate, using joint state estimation and control ($\mathcal{C}_1$, left) and separate estimation and control ($\mathcal{C}_0$, right).

7

On convexity of feedback predictions in minimax MPC

Feedback predictions were introduced in Chapter 5.1 to reduce the level of conservativeness in the proposed minimax controller. They were introduced as a tuning knob in the controller to be decided off-line, but no guidelines were given, or exist in the literature, on how to select these. The simulation examples indicated that the choice was crucial for good performance of the minimax controller. Obviously, something has to be done.

In this chapter, we change the definition of feedback predictions slightly, and show that the new feedback prediction matrix can be incorporated as a decision variable in the on-line optimization problem. The obtained optimization problems are convex and fits nicely into the framework developed in earlier chapters. Unfortunately, the optimization problem grows rapidly, although polynomially, in the system dimension, the number of constraints and the prediction horizon. To resolve this problem, it is shown how the general solution can serve as a basis for off-line calculations, and approximations with a reduced degree of freedom, but with much better computational properties. Simulations indicate that these approximations work very well.

To the author's knowledge, on-line optimization of feedback predictions is a completely novel idea and has never been proposed before in the literature.

7.1 Feedback Predictions

The system we analyze is the same as in the previous chapters.

$$x_{k+1} = Ax_k + Bu_k + Gw_k \tag{7.1a}$$
$$y_k = Cx_k \tag{7.1b}$$

The dominating approach in robust MPC synthesis is as we have seen to employ a minimax strategy, i.e., minimization of a worst-case performance

$$\min_u \max_w \ell(x_{k|k}, u_{k|k}, x_{k+1|k}, u_{k+1|k}, \ldots, x_{k+N-1|k}, u_{k+N-1|k}) \tag{7.2}$$

A fundamental flaw with this formulation is the fact that the MPC controller in reality applies feedback. This will make the minimax approach unnecessarily conservative, since it has to find a single control sequence that works well in open-loop for all admissible disturbance realizations.

A standard trick to reduce conservatism in minimax schemes is feedback predictions[1](Bemporad, 1998; Schuurmans and Rossiter, 2000; Chisci et al., 2001). The idea is to assume that at least some feedback will be employed. This can be done by parameterizing the future control sequence in terms of the future states and a new decision variable $v_{k+j|k} \in \mathbb{R}^m$,

$$u_{k+j|k} = Lx_{k+j|k} + v_{k+j|k} \tag{7.3}$$

The feedback matrix L is typically chosen off-line, or some heuristic procedure is used to find a suitable L on-line. The predicted states when feedback predications are used are given by

$$x_{k+j|k} = (A+BL)^j x_{k|k} + \sum_{i=1}^{j} (A+BL)^{j-i}(Bv_{k+i-1|k} + Gw_{k+i-1|k}) \tag{7.4}$$

The influence from the disturbances to the predicted states can be reduced by placing the eigenvalues of $(A + BL)$ appropriately. However, the feedback predictions will transfer uncertainty to the predicted control sequence.

$$u_{k+j|k} = L((A+BL)^j x_{k|k} + \sum_{i=1}^{j} L(A+BL)^{j-i}(Bv_{k+i-1|k} + Gw_{k+i-1|k})) + v_{k+j|k} \tag{7.5}$$

A small example illustrates this trade-off.

[1]Feedback predictions can also be used in nominal MPC in order to obtain better conditioned problems. If perfect optimization however is assumed, feedback predictions cannot influence the solution in the nominal case.

Example 7.1 (Trade-off in feedback predictions)

The system is the same as in Example 5.1.

Examples of open-loop predicted step-responses with disturbances are shown in the top left part of Figure 7.1. The predictions are highly uncertain and the output fluctuates between 0.75 and 1.25 after the initial transient.

Closed-loop predicted step responses with a feedback matrix based on an LQ controller with $Q = 1$ and $R = 0.01$ is shown in the bottom left figure. The uncertainty in the output prediction has been significantly reduced. The steady-state output can now be found between 0.95 and 1.05. The price paid for the reduced output uncertainty is uncertainty in the input instead. The control input after the transient now fluctuates between 0.7 and 1.3.

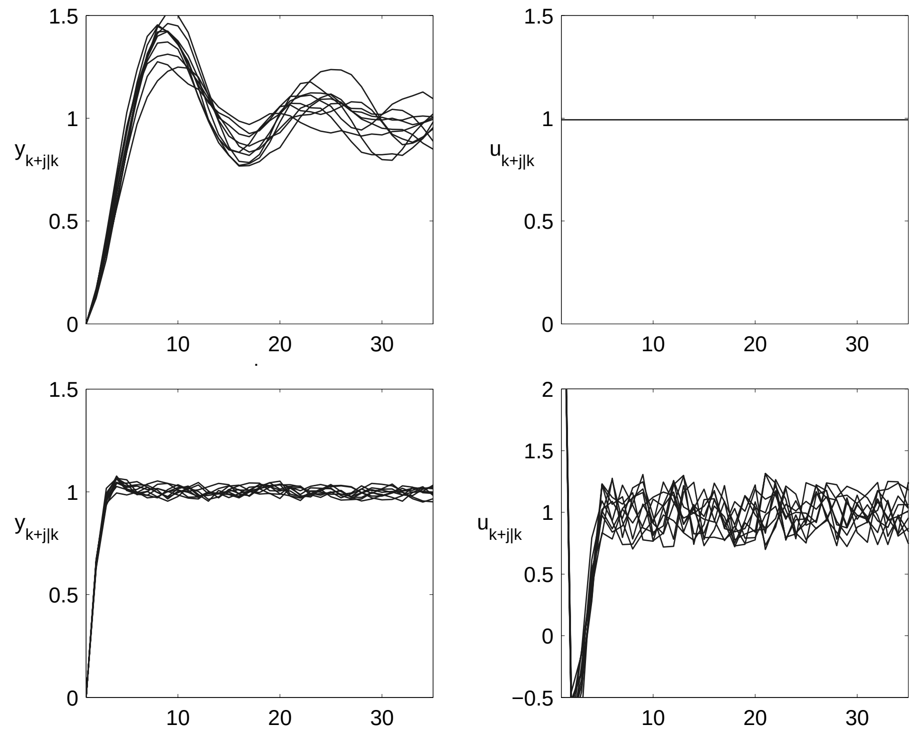

Figure 7.1: Predicted step-responses for disturbed system. Top left figure shows the uncertain output predictions when no feedback is applied (only the constant input shown in top right figure). Bottom left figure shows the predictions when feedback predictions are applied. The responses are much more certain, but the price paid is uncertainty in the control signal (bottom right figure).

The problems with feedback predictions can essentially be addressed in four different ways.

- Select L off-line by trial and error and numerous simulations.
- Use heuristics to find suitable L on-line at each sample instant.
- Optimize L on-line.
- Abandon feedback predictions and solve the underlying closed-loop minimax MPC problem instead (see below).

It will be shown in this chapter that with a slightly different parameterization of the control sequence, some minimax problems with on-line optimization of L can be cast as convex programs.

7.2 Closed-loop Minimax MPC

As we stated earlier, finding an open-loop control sequence for the finite horizon problem is a bad idea, since this control sequence has to cope with all possible disturbance realizations. In a closed-loop minimax MPC approach, one would assume instead that the future control u_{k+1} is calculated optimally over the horizon $N-1$ first when x_{k+1} is available. The problem to solve in closed-loop minimax MPC is thus

$$\min_{u_{k|k}} \max_{w_{k|k}} \cdots \min_{u_{k+N-1|k}} \max_{w_{k+N-1|k}} \ell(x_{k|k}, u_{k|k}, \ldots, x_{k+N-1|k}, u_{k+N-1|k}) \tag{7.6}$$

This type of minimax MPC has been addressed in, e.g., (Lee and Yu, 1997) and (Scokaert and Mayne, 1998). Some formulations can be solved with enumerative schemes. An example illustrates these methods best.

Example 7.2 (Closed-loop minimax MPC)

We are given a linear system with one scalar disturbance $w_k \in \mathbb{W}_\infty$, i.e., $|w_k| \leq 1$. Moreover, the performance measure ℓ is convex. Theorem 4.1 implies that the maximum of ℓ is attained at a vertex of the disturbance set. This means that only predictions obtained with $w_k = \pm 1$ are interesting. These disturbances will be denoted extreme disturbances.

The minimax problem can be solved using direct enumeration as follows. No matter what the disturbance w_k will be, the first term in the control sequence has to be decided at time k, so the extreme disturbance and control sequences begin like

$$\{1\}\ ,\ \{u_{k|k}\} \tag{7.7a}$$

$$\{-1\}\ ,\ \{u_{k|k}\} \tag{7.7b}$$

There are two possible values of x_{k+1} (depending on w_k). Therefore, we have two possible optimal control inputs at $k+1$. Hence, the possible extreme disturbances and control sequences are

$$\{1,1\}\ ,\ \{u_{k|k}, u^{(1)}_{k+1|k}\} \tag{7.8a}$$

$$\{1,-1\}\ ,\ \{u_{k|k}, u^{(1)}_{k+1|k}\} \tag{7.8b}$$

$$\{-1,1\}\ ,\ \{u_{k|k}, u^{(2)}_{k+1|k}\} \tag{7.8c}$$

$$\{-1,-1\}\ ,\ \{u_{k|k}, u^{(2)}_{k+1|k}\} \tag{7.8d}$$

Now, u_{k+2} is calculated first when x_{k+2} is available. Once again, we have 2 different transitions from x_{k+1} to x_{k+2}, and have to define a control element for each possible x_{k+2} (see Figure 7.2).

$$\{1,1,1\}\ ,\ \{u_{k|k}, u^{(1)}_{k+1|k}, u^{(1)}_{k+2|k}\} \tag{7.9a}$$

$$\{1,-1,1\}\ ,\ \{u_{k|k}, u^{(1)}_{k+1|k}, u^{(2)}_{k+2|k}\} \tag{7.9b}$$

$$\{-1,1,1\}\ ,\ \{u_{k|k}, u^{(2)}_{k+1|k}, u^{(3)}_{k+2|k}\} \tag{7.9c}$$

$$\{-1,-1,1\}\ ,\ \{u_{k|k}, u^{(2)}_{k+1|k}, u^{(4)}_{k+2|k}\} \tag{7.9d}$$

$$\{1,1,-1\}\ ,\ \{u_{k|k}, u^{(1)}_{k+1|k}, u^{(1)}_{k+2|k}\} \tag{7.9e}$$

$$\{1,-1,-1\}\ ,\ \{u_{k|k}, u^{(1)}_{k+1|k}, u^{(2)}_{k+2|k}\} \tag{7.9f}$$

$$\{-1,1,-1\}\ ,\ \{u_{k|k}, u^{(2)}_{k+1|k}, u^{(3)}_{k+2|k}\} \tag{7.9g}$$

$$\{-1,-1,-1\}\ ,\ \{u_{k|k}, u^{(2)}_{k+1|k}, u^{(4)}_{k+2|k}\} \tag{7.9h}$$

The maximum of the performance measure is found along some of these worst-case predictions. A closed-loop minimax MPC problem with $N=3$ can therefore be solved with the following program.

$$\begin{array}{ll} \displaystyle\min_{u,t} & t \\ \text{subject to} & \ell(x_{k|k}, u_{k|k}, u^{(1)}_{k+1|k}, x^{(1)}_{k+1|k}, u^{(1)}_{k+2|k}, x^{(1)}_{k+2|k}) \leq t \\ & \ell(x_{k|k}, u_{k|k}, u^{(1)}_{k+1|k}, x^{(1)}_{k+1|k}, u^{(2)}_{k+2|k}, x^{(2)}_{k+2|k}) \leq t \\ & \ell(x_{k|k}, u_{k|k}, u^{(2)}_{k+1|k}, x^{(2)}_{k+1|k}, u^{(3)}_{k+2|k}, x^{(3)}_{k+2|k}) \leq t \\ & \ell(x_{k|k}, u_{k|k}, u^{(2)}_{k+1|k}, x^{(2)}_{k+1|k}, u^{(4)}_{k+2|k}, x^{(4)}_{k+2|k}) \leq t \end{array} \tag{7.10}$$

To summarize; for the closed-loop minimax MPC problem with $N=3$, a control sequence has to be assigned for every possible extreme disturbance realization $\{w_k, w_{k+1}\}$ (the last disturbance w_{k+2} cannot influence the control sequence, and $x_{k+3|k}$ is not used in the performance measure). The result is 4 possible control sequences (7.9a-7.9d) with 7 control variables ($u_{k|k}$, $u^{(1)}_{k+1|k}$, $u^{(2)}_{k+1|k}$, $u^{(1)}_{k+2|k}$, $u^{(2)}_{k+2|k}$,

$u^{(3)}_{k+2|k}$ and $u^{(4)}_{k+2|k}$). Furthermore, since there are 4 possible worst-case disturbance realizations, there will be 4 state trajectories. Figure 7.2 illustrates the explosion in possible trajectories and introduced control variables.

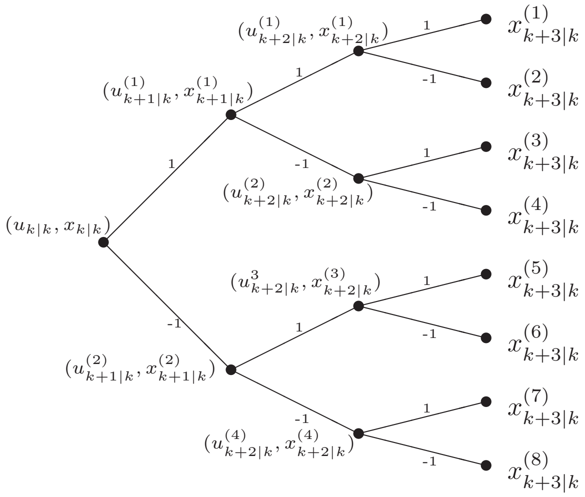

Figure 7.2: The figure shows the exponential explosion in the number of disturbance realizations and corresponding control sequences that have to be defined to solve closed-loop minimax MPC using straightforward enumeration.

In the general case with q extreme values on an additive disturbance w_k, there will be q^{N-1} disturbance sequences and $(1 + q + q^2 + \ldots + q^{N-1})$ control variables (Scokaert and Mayne, 1998). To solve the general closed-loop minimax problem, we would enumerate all disturbance realizations with the corresponding control sequences and state trajectories, and pick the control sequence that minimized the worst-case cost.

It is easy to realize that this will only work for small p and N. As an example, assuming $p = 4$ (w_k two-dimensional with upper and lower bound) and $N = 10$ would give us 262144 different realizations with 349525 different control variables of dimension m. Clearly an intractable problem already for this small example.

The reason we looked at this solution to the minimax problem for the finite horizon MPC problem is that we will see some similarities with this solution and the feedback parameterization we are going to propose, *the optimal solution is parameterized in the future unknown disturbance sequence*. Keep this is mind and the results in the next section will seem much more intuitive.

7.3 A Linear Parameterization of U

In this section, we introduce a new parameterization of the control sequence. Recall the definition of the predicted states from Section 5.3.1.

$$X = \mathcal{A}x_{k|k} + \mathcal{B}U + \mathcal{G}W \tag{7.11}$$

Feedback predictions in the form $u_{k+j|k} = Lx_{k+j|k} + v_{k+j|k}$ were written in a more compact form using a vectorized variable V and a block diagonal matrix $\mathcal{L}$ (see (5.33)).

$$U = \mathcal{L}X + V \tag{7.12}$$

This standard parameterization will not be used in this chapter and is only introduced to simplify the interpretation of the parameterization that will be defined later. To motivate the introduction of a new parameterization, recall that we were able to solve for U and X, with $\Omega = (I - \mathcal{B}\mathcal{L})^{-1}$.

$$X = \Omega(\mathcal{A}x_{k|k} + \mathcal{B}V + \mathcal{G}W) \tag{7.13a}$$

$$U = \mathcal{L}\Omega(\mathcal{A}x_{k|k} + \mathcal{B}V + \mathcal{G}W) + V \tag{7.13b}$$

The problem is that the mapping from $\mathcal{L}$ and V to X and U is nonlinear, hence optimization over both $\mathcal{L}$ and V is likely to cause problem. At least, it is not obvious how this parameterization can be incorporated in a standard convex optimization problem.

Let us look at bit closer on the parameterized control sequence.

$$U = (\mathcal{L}\Omega(\mathcal{A}x_{k|k} + \mathcal{B}V) + V) + \mathcal{L}\Omega\mathcal{G}W \tag{7.14}$$

It is composed of one certain part, $\mathcal{L}\Omega(\mathcal{A}x_{k|k} + \mathcal{B}V) + V$, and one mapping from the disturbances to the control sequence, $\mathcal{L}\Omega\mathcal{G}W$. Important to remember is that the matrix $\mathcal{L}\Omega\mathcal{G}$ has a causal structure, i.e. $u_{k+j|k}$ is only affected by $w_{k+i|k}$, $i < j$.

The findings motivate us to try an alternative parameterization.

$$U = \mathcal{L}W + V \tag{7.15a}$$

$$\mathcal{L} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ L_{10} & 0 & \dots & 0 \\ L_{20} & L_{21} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{(N-1)0} & L_{(N-1)1} & L_{(N-1)(N-2)} & 0 \end{pmatrix} \tag{7.15b}$$

The control sequence is now parameterized directly in the uncertainty. The matrix $L_{ij} \in \mathbb{R}^{m\times r}$ describes how $u_{k+i|k}$ uses $w_{k+j|k}$. Note that the parameterization is causal in the same sense as the standard parameterization (7.12). Inserting the parameterization yields

$$X = \mathcal{A}x_{k|k} + \mathcal{B}V + (\mathcal{G} + \mathcal{B}\mathcal{L})W \tag{7.16a}$$

$$U = \mathcal{L}W + V \tag{7.16b}$$

The mapping from $\mathcal{L}$ and V to X and U is now bilinear. This is the main idea in this work, and it will allow us to formulate a number of convex minimax MPC problems (with polynomial complexity in the prediction horizon N and system dimensions).

7.3.1 Connections to Closed-loop Minimax MPC

At a first look, (7.15) might seem suspicious. How can we parameterize the control sequence in terms of an unknown future disturbance realization?

Compare it to the closed-loop minimax MPC solution. In that approach, we have one control sequence for each disturbance realization. The basic idea is that we assume an optimal choice over the reduced horizon once x_{k+1} is available. What we have here is basically a sub-optimal version of the closed-loop minimax solution. Instead of having an optimal control sequence for each worst-case disturbance realization, we assume that we will not be optimal in the future over the reduced horizon, but will at least have different solutions (linearly dependent) for different disturbance realizations. In other words, our minimax MPC algorithm is based on a reduced degree of freedom solution of the closed-loop solution.

7.4 Minimax MPC is Convex in $\mathcal{L}$ and V

In the previous section, a parameterization of the future control sequence was proposed. It will now be shown that this parameterization allows us to solve some minimax MPC problems with on-line optimization of feedback predictions using convex programming.

7.4.1 Minimum Peak Performance Measure

A commonly used performance measure in minimax MPC is minimization of the worst-case deviation along the predicted trajectory (Campo and Morari, 1987; Zheng, 1995; Oliviera et al., 2000). Of course, the results here can be generalized to deviation from a reference trajectory, but this is omitted to keep notation simple. The problem can be stated as

$$\boxed{\begin{array}{llll} \min\limits_{u} \max\limits_{w} \max\limits_{j} & ||y_{k+j|k}||_\infty & & \\ \text{subject to} & u_{k+j|k} \in \mathbb{U} & \forall w \in \mathbb{W} \\ & x_{k+j|k} \in \mathbb{X} & \forall w \in \mathbb{W} \\ & w_{k+j|k} \in \mathbb{W} & \end{array}} \tag{7.17}$$

Rewrite to a compact epigraph formulation by noting that $\max\limits_{j} ||y_{k+j|k}||_\infty = ||Y||_\infty$. Furthermore, introduce the proposed feedback predictions and vectorize the control and state constraints.

$$\begin{array}{lrcl}\min_{V,\mathcal{L},t} & t & & \\ \text{subject to} & ||\mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}V+(\mathcal{G}+\mathcal{B}\mathcal{L})W)||_\infty & \leq & t \quad \forall W \in \mathbb{W}^N \\ & \mathcal{E}_u(V+\mathcal{L}W) & \leq & \mathcal{F}_u \quad \forall W \in \mathbb{W}^N \\ & \mathcal{E}_x(\mathcal{A}x_{k|k}+\mathcal{B}V+(\mathcal{G}+\mathcal{B}\mathcal{L})W) & \leq & \mathcal{F}_x \quad \forall W \in \mathbb{W}^N\end{array} \tag{7.18}$$

The peak constraint is equivalent to two sets of linear inequalities

$$\mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}V)+\mathcal{C}(\mathcal{G}+\mathcal{B}\mathcal{L})W \leq t\mathbf{1} \quad \forall W \in \mathbb{W}^N \tag{7.19a}$$
$$-\mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}V)-\mathcal{C}(\mathcal{G}+\mathcal{B}\mathcal{L})W \leq t\mathbf{1} \quad \forall W \in \mathbb{W}^N \tag{7.19b}$$

To satisfy these uncertain linear inequalities, the same approach as in Section 5.2.1 can be used. For $w \in \mathbb{W}_\infty$, Theorem 3.1 states (in a vectorized form) that $\max_{|x|\leq 1} Px = |P|\mathbf{1}$. Hence, the uncertain constraints with $w \in \mathbb{W}_\infty$ are satisfied if

$$\mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}V)+|\mathcal{C}(\mathcal{G}+\mathcal{B}\mathcal{L})|\mathbf{1} \leq t\mathbf{1} \tag{7.20a}$$
$$-\mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}V)+|\mathcal{C}(\mathcal{G}+\mathcal{B}\mathcal{L})|\mathbf{1} \leq t\mathbf{1} \tag{7.20b}$$

To take care of these constraints using linear programming, we need to bound the absolute value of the matrix $\mathcal{C}(\mathcal{G}+\mathcal{B}\mathcal{L})$ from above. This is done by defining a matrix variable Υ

$$\mathcal{C}(\mathcal{G}+\mathcal{B}\mathcal{L}) \leq \Upsilon \tag{7.21a}$$
$$-\mathcal{C}(\mathcal{G}+\mathcal{B}\mathcal{L}) \leq \Upsilon \tag{7.21b}$$

and the peak constraint is equivalent to

$$\mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}V)+\Upsilon\mathbf{1} \leq t\mathbf{1} \tag{7.22a}$$
$$-\mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}V)+\Upsilon\mathbf{1} \leq t\mathbf{1} \tag{7.22b}$$

The same method can be applied to state and control constraints and gives a new matrix variable Ω and the constraints

$$\begin{pmatrix}\mathcal{E}_x(\mathcal{A}x_{k|k}+\mathcal{B}V)\\ \mathcal{E}_uV\end{pmatrix}+\Omega\mathbf{1} \leq \begin{pmatrix}\mathcal{F}_x\\ \mathcal{F}_u\end{pmatrix} \tag{7.23a}$$
$$\begin{pmatrix}\mathcal{E}_x(\mathcal{G}+\mathcal{B}\mathcal{L})\\ \mathcal{E}_u\mathcal{L}\end{pmatrix} \leq \Omega \tag{7.23b}$$
$$-\begin{pmatrix}\mathcal{E}_x(\mathcal{G}+\mathcal{B}\mathcal{L})\\ \mathcal{E}_u\mathcal{L}\end{pmatrix} \leq \Omega \tag{7.23c}$$

To summarize, the minimum peak problem with element-wise bounded disturbances and feedback predictions is solved with the following linear program.

$$
\boxed{\begin{array}{lrcl}
\min\limits_{V,\mathcal{L},t,\Omega,\Upsilon} & t & & \\
\text{subject to} & \mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}V)+\Upsilon\mathbf{1} & \leq & t\mathbf{1} \\
& -\mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}V)+\Upsilon\mathbf{1} & \leq & t\mathbf{1} \\
& \mathcal{C}(\mathcal{G}+\mathcal{B}\mathcal{L}) & \leq & \Upsilon \\
& -\mathcal{C}(\mathcal{G}+\mathcal{B}\mathcal{L}) & \leq & \Upsilon \\
& \begin{pmatrix}\mathcal{E}_x(\mathcal{A}x_{k|k}+\mathcal{B}V)\\ \mathcal{E}_uV\end{pmatrix}+\Omega\mathbf{1} & \leq & \begin{pmatrix}\mathcal{F}_x\\ \mathcal{F}_u\end{pmatrix} \\
& \begin{pmatrix}\mathcal{E}_x(\mathcal{G}+\mathcal{B}\mathcal{L})\\ \mathcal{E}_u\mathcal{L}\end{pmatrix} & \leq & \Omega \\
& -\begin{pmatrix}\mathcal{E}_x(\mathcal{G}+\mathcal{B}\mathcal{L})\\ \mathcal{E}_u\mathcal{L}\end{pmatrix} & \leq & \Omega
\end{array}} \tag{7.24}
$$

The case $w \in \mathbb{W}_2$ can also be dealt with analytically, but require some additional notation, and will generate a more complex optimization problem. Define to begin with a partitioning in the same way as in Section 5.2.1.

$$
\mathcal{C}(\mathcal{G}+\mathcal{B}\mathcal{L}) = \begin{pmatrix}\upsilon_1^T\\ \upsilon_2^T\\ \vdots\end{pmatrix}, \quad \upsilon_i^T = \begin{pmatrix}\upsilon_{i1}^T & \upsilon_{i2}^T & \dots & \upsilon_{iN}^T\end{pmatrix} \tag{7.25a}
$$

$$
\begin{pmatrix}\mathcal{E}_x(\mathcal{G}+\mathcal{B}\mathcal{L})\\ \mathcal{E}_u\mathcal{L}\end{pmatrix} = \begin{pmatrix}\omega_1^T\\ \omega_2^T\\ \vdots\end{pmatrix}, \quad \omega_i^T = \begin{pmatrix}\omega_{i1}^T & \omega_{i2}^T & \dots & \omega_{iN}^T\end{pmatrix} \tag{7.25b}
$$

Maximizing the uncertainties in the linear inequalities can now be done using the same technique as in Section 5.2.1. However, the vectors ω_{ij} and υ_{ij} depend on $\mathcal{L}$ so the expression (5.28) cannot be evaluated. Instead, we are forced to introduce matrices Ω and Υ together with large number of second order cone constraints.

$$
||\omega_{ij}|| \leq \Omega_{ij} \tag{7.26a}
$$

$$
||\upsilon_{ij}|| \leq \Upsilon_{ij} \tag{7.26b}
$$

The robustified linear inequalities can with these variables be written as

$$
\mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}V)+\Upsilon\mathbf{1} \leq t\mathbf{1} \tag{7.27a}
$$

$$
-\mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}V)+\Upsilon\mathbf{1} \leq t\mathbf{1} \tag{7.27b}
$$

$$
\begin{pmatrix}\mathcal{E}_x(\mathcal{A}x_{k|k}+\mathcal{B}V)\\ \mathcal{E}_uV\end{pmatrix}+\Omega\mathbf{1} \leq \begin{pmatrix}\mathcal{F}_x\\ \mathcal{F}_u\end{pmatrix} \tag{7.27c}
$$

The minimum peak problem with $w \in \mathbb{W}_2$ and feedback predictions is thus solved with the following second order cone program.

$$
\boxed{\begin{array}{lrcl}
\min\limits_{V,\mathcal{L},t,\Omega,\Upsilon} & t & & \\
\text{subject to} & \mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}V)+\Upsilon\mathbf{1} & \leq & t\mathbf{1}\\
& -\mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}V)+\Upsilon\mathbf{1} & \leq & t\mathbf{1}\\
& \begin{pmatrix}\mathcal{E}_x(\mathcal{A}x_{k|k}+\mathcal{B}V)\\ \mathcal{E}_uV\end{pmatrix}+\Omega\mathbf{1} & \leq & \begin{pmatrix}\mathcal{F}_x\\ \mathcal{F}_u\end{pmatrix}\\
& ||\omega_{ij}|| & \leq & \Omega_{ij}\\
& ||\upsilon_{ij}|| & \leq & \Upsilon_{ij}
\end{array}} \tag{7.28}
$$

The derived problems (7.24) and (7.28) can be solved with standard linear programming or second order cone programming solvers, just as when $\mathcal{L}$ is fixed. This means that the introduction of $\mathcal{L}$ as a free variable not has complicated the problem in terms of conceptual complexity. However, the number of introduced variables and constraints is huge. This will be discussed further in Section 7.5 where cheaper parameterizations are introduced.

7.4.2 Quadratic Performance Measure

The second class of minimax MPC we address is a formulation with the standard quadratic performance measure. This problem was addressed in Chapter 5, and the optimization problem is formulated in Equation (5.4).

The solution in Chapter 5 was based on a semidefinite relaxation of the following uncertain LMI.

$$
\begin{pmatrix} t & Y^T & U^T\\ Y & \mathcal{Q}^{-1} & 0\\ U & 0 & \mathcal{R}^{-1}\end{pmatrix}\succeq 0 \quad \forall W\in\mathbb{W}^N \tag{7.29}
$$

Insert our new definition of Y and U, and separate certain and uncertain terms.

$$
\begin{pmatrix} t & (\mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}V))^T & V^T\\ \star & \mathcal{Q}^{-1} & 0\\ \star & 0 & \mathcal{R}^{-1}\end{pmatrix}+\begin{pmatrix}0\\ \mathcal{C}(\mathcal{G}+\mathcal{B}\mathcal{L})\\ \mathcal{L}\end{pmatrix}W\begin{pmatrix}1 & 0 & 0\end{pmatrix}+(\star)\succeq 0 \tag{7.30}
$$

This uncertain LMI be dealt with using the same semidefinite relaxation as in Chapter 5. The outcome is a set of multipliers $\tau\in\mathbb{R}^s_+$ (where s is N or Nr depending on the uncertainty structure $\mathbb{W}$), the associated diagonal matrix $\mathcal{T}$, and a sufficient condition for the uncertain LMI (7.29) to hold.

$$
\begin{pmatrix} t-\sum_{i=1}^s\tau_i & (\mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}V))^T & V^T & 0\\ \star & \mathcal{Q}^{-1} & 0 & \mathcal{C}(\mathcal{G}+\mathcal{B}\mathcal{L})\\ \star & 0 & \mathcal{R}^{-1} & \mathcal{L}\\ 0 & \star & \star & \mathcal{T}\end{pmatrix}\succeq 0 \tag{7.31}
$$

Remarkably, the semidefinite relaxation of the minimax problem is linear in $\mathcal{L}$.

We note that any linear state or control constraint can be taken care of using the methods described in the previous section, i.e., the problem will only be augmented with a set of linear or second order cone inequalities. Our solution to the minimax problem is thus given by

$$\boxed{\begin{array}{ll} \displaystyle\min_{V,\mathcal{L},t,\Omega,\tau} & t \\ \text{subject to} & (7.31) \\ & (7.23) \text{ or } (7.26a), (7.27c) \end{array}} \tag{7.32}$$

The semidefinite program to solve the semidefinite relaxation of a minimax MPC problem with feedback predictions has obviously not changed much, compared to the results in Chapter 5. The main difference lies in the many linear or second order constraints that are added, and the huge amount of additional variables that have to be introduced.

7.5 Alternative Parameterizations

The main problem with the minimax formulations (7.24), (7.28) and (7.32) is the excessive amount of decision variables and constraints.

The reason is, to begin with, the high-dimensional parameterization of the matrix $\mathcal{L}$. The full parameterization is given by

$$\mathcal{L} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ L_{10} & 0 & \dots & 0 \\ L_{20} & L_{32} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{(N-1)0} & L_{(N-1)2} & L_{(N-2)(N-1)} & 0 \end{pmatrix} \tag{7.33}$$

The number of free variables in this matrix alone is

$$mr(N-1) + mr(N-2) + \ldots + mr = \frac{mr}{2}N(N-1) \tag{7.34}$$

In addition to this, the matrices Ω and Υ introduce another large set of variables.

Note however that many of the variables and constraints are redundant. For instance, the matrix $\mathcal{C}(\mathcal{G} + \mathcal{B}\mathcal{L})$ is lower block triangular, so Ω can also be lower block triangular. These issues will not be dealt with here. The software used to implement the algorithms, YALMIP and SEDUMI, take care of redundant variables and constraints automatically. Furthermore, the notation would be unnecessarily detailed.

Clearly, something has to be done if we want to obtain problems that can be used in practice. It could be argued that the minimax problem for $w \in \mathbb{W}_\infty$ in Section 7.4.1 is tractable since extremely efficient solvers for large-scale linear programs are available, and solvers exploiting structure in related MPC and minimax MPC problems have been developed (Rao et al., 1998; Vandenberghe et al., 2002).

This does not hold for the problem in Section 7.4.2, since a huge mixed semidefinite and second order cone problem has to be solved. Note though, that the dimension of the matrix constraint (7.31) is unchanged compared to Chapter 5.

To find a cheaper parameterization, let us first recall a standard feedback prediction (i.e. parameterization in X, which we denote with subscript X)).

$$\mathcal{L}_X = \begin{pmatrix} L & 0 & \dots & 0 \\ 0 & L & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & L \end{pmatrix} = \oplus_{j=1}^{N} L \tag{7.35}$$

It was previously shown that this gives the following nonlinear parameterization

$$U = \mathcal{L}_X(I - \mathcal{B}\mathcal{L}_X)^{-1}(\mathcal{A}x_{k|k} + \mathcal{B}V + \mathcal{G}W) + V \tag{7.36}$$

Comparing this with our parameterization $U = \mathcal{L}W + V$, we see that we would have had to obtain the following parameterization to have the same feedback from the disturbances.

$$\mathcal{L} = \mathcal{L}_X(I - \mathcal{B}\mathcal{L}_X)^{-1}\mathcal{G} \tag{7.37}$$

This matrix does not have the simple block diagonal structure as the matrix $\mathcal{L}_X$, so we are forced to over-parameterize $\mathcal{L}$ in order to obtain standard feedback predictions. Motivated by this, let us look at some ways to obtain a cheaper parameterization of $\mathcal{L}$.

Exploiting Toeplitz Structure in $\mathcal{L}_X(I - \mathcal{B}\mathcal{L}_X)^{-1}\mathcal{G}$

The first thing to notice is that the matrix $\mathcal{L}_X(I - \mathcal{B}\mathcal{L}_X)^{-1}\mathcal{G}$, with $\mathcal{L}_X$ defined according to (7.35), has a lower block triangular Toeplitz structure. This follows from (7.5). Comparing (7.33) and (7.5) shows that $L_{ij} = L(A+BL)^{i-1-j}G$ makes $\mathcal{L}$ and $\mathcal{L}_X(I - \mathcal{B}\mathcal{L}_X)^{-1}\mathcal{G}$ identical.

This means that only $N-1$ different matrices L_{ij} are needed to recover the matrix $\mathcal{L}_X(I - \mathcal{B}\mathcal{L}_X)^{-1}\mathcal{G}$. A more efficient parameterization is thus given by (with the new decision variable $L_j \in \mathbb{R}^{m \times r}$)

$$\mathcal{L} = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ L_1 & 0 & 0 & \dots & 0 \\ L_2 & L_1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ L_{N-1} & L_{N-2} & \dots & L_1 & 0 \end{pmatrix} \tag{7.38}$$

This parameterization requires only $mr(N-1)$ variables. Obviously, a clear improvement to the initial parameterization, but still an over-parameterization compared to mn variables needed to define $\mathcal{L}_X$ in a standard state feedback parameterization (7.35).

Linearization of $\mathcal{L}_X(I-\mathcal{B}\mathcal{L}_X)^{-1}\mathcal{G}$

Assuming a reasonably well working feedback matrix $\mathcal{L}_X^0$ is available, a viable approach might to linearize the nonlinear expression (7.37) and use a first order approximation as a new feedback matrix.

The new decision variable is a standard feedback matrix $L \in \mathbb{R}^{m\times n}$. This matrix defines the block diagonal feedback matrix $\mathcal{L}_X = \oplus_1^N L$. A first order approximation of (7.37) is given by[2]

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_X(I-\mathcal{B}\mathcal{L}_X)^{-1}\mathcal{G}\\ &\simeq \mathcal{L}_X^0\Omega_0^{-1}\mathcal{G} + (\mathcal{L}_X-\mathcal{L}_X^0)\Omega_0^{-1}\mathcal{G} + \mathcal{L}_X^0\Omega_0^{-1}\mathcal{B}(\mathcal{L}_X-\mathcal{L}_X^0)\Omega_0^{-1}\mathcal{G} \qquad (7.39)\end{aligned}$$

The parameterization (7.39) is, by construction, linear in $\mathcal{L}_X$, hence the methods derived in this chapter are applicable.

Linear Combinations

Another approach worth trying is to determine a sequence of feedback predictions off-line,

$$\mathcal{L}^i = \mathcal{L}_X^i(I-\mathcal{B}\mathcal{L}_X^i)^{-1}\mathcal{G} \quad i = 1\ldots q \qquad (7.40)$$

and then parameterize the on-line feedback prediction as a linear combination of these pre-calculated feedback predictions

$$\mathcal{L} = \sum_{i=1}^{q}\alpha_i\mathcal{L}^i \qquad (7.41)$$

The weights $\alpha \in \mathbb{R}^q$ are the only free variables in the feedback prediction matrix.

A related idea was used in (Bemporad, 1998). A set of feedback predictions was defined off-line and the on-line MPC problem was augmented with an outer loop. For each pre-defined feedback prediction, a minimax MPC problem was solved, and the solution to the problem that gave lowest worst-case cost was applied to the system.

Structure in Ω and Υ

The lower block triangular Toeplitz structure revealed in $\mathcal{L}_X(I-\mathcal{B}\mathcal{L}_X)^{-1}\mathcal{G}$ can be found also in the matrices $\mathcal{C}(\mathcal{G}+\mathcal{B}\mathcal{L})$ and $\mathcal{E}_x(\mathcal{G}+\mathcal{B}\mathcal{L})$ (assuming $\mathcal{L}$ has a block diagonal Toeplitz structure as in (7.38), (7.39) and (7.41)). This means that much cheaper parameterizations of the matrices Ω and Υ are possible. The details are omitted for brevity.

[2]Use Sherman-Morrison-Woodbury formula, $(I-(X+\Delta))^{-1} = ((I-X)-\Delta)^{-1}$
$= (I-X)^{-1} + (I-X)^{-1}\Delta(I-(I-X)^{-1}\Delta)^{-1}(I-X)^{-1} \simeq (I-X)^{-1} + (I-X)^{-1}\Delta(I-X)^{-1}$ for small Δ

7.6 Simulation Results

To start with, we return to the first example in Chapter 5 to see if we could have done any better.

Example 7.3 (Example 5.1 revisited)
In Example 5.2, a feedback matrix was chosen rather arbitrarily as an LQ controller designed with $Q = R = 1$. The question that now arise is whether this could have been done better, or if the LQ controller does a good job.

A minimax controller based on (7.32), with obvious changes for a reference tracking formulation, was implemented and applied to the system in Example (5.1). The original controller from the example was also implemented and simulated using the same disturbance realization. The results are given in Figure 7.3.

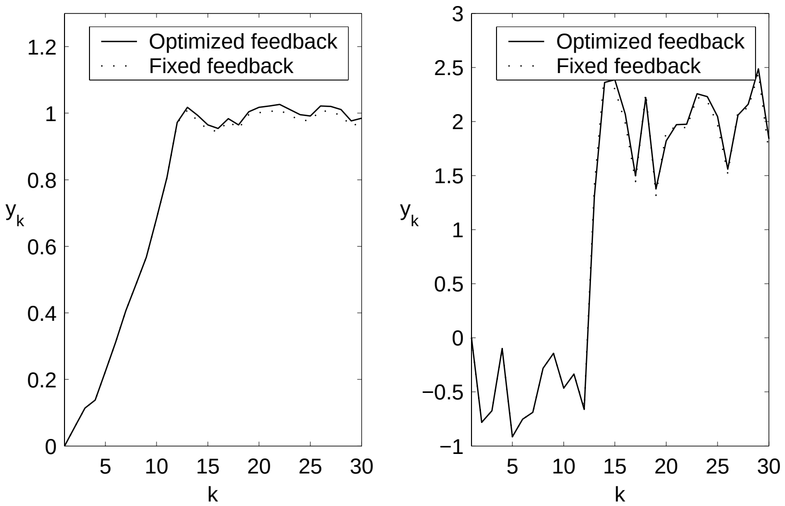

Figure 7.3: The optimized feedback matrix does not improve performance.

The results indicate that no improvements have been obtained. This is both good news and bad news. It is good news in the sense that we obviously made a good choice in Example 5.2, and it seems like fixed feedback predictions indeed can do a good job. On the other hand, it is bad news in light of this chapter. The results developed here are seemingly to no use.

Of course, the results in the example above cannot be generalized. Practice has shown that feedback predictions calculated from LQ controllers often work well for stable low-order systems, but it is easy to construct examples where fixed feedback predictions are hard to choose off-line.

Example 7.4 (Unstable third-order system)
The system is given by [3]

$$\begin{aligned} x_{k+1} &= \begin{pmatrix} 2.938 & -0.7345 & 0.25 \\ 4 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} x_k + \begin{pmatrix} 0.25 \\ 0 \\ 0 \end{pmatrix} u_k + \begin{pmatrix} 0.0625 \\ 0 \\ 0 \end{pmatrix} w_k \\ y_k &= \begin{pmatrix} -0.2072 & 0.04141 & 0.07256 \end{pmatrix} x_k \end{aligned}$$

The input disturbance satisfies $|w_k| \leq 1$.
The control objective is to keep the output y_k as close as possible to the reference value 1, but at the same time never exceed this level. The input is also constrained $|u_k| \leq 1$. Since the main objective is to balance the output close to the constraint level, the weight matrices are chosen as $Q = 1$ and $R = 0.01$. The prediction horizon was $N = 10$. The proposed minimax controller with free feedback predictions was implemented, together with a number of standard minimax controllers from Chapter 5.1. These controllers used fixed feedback matrices defined using LQ controllers with control weights ranging from 10^{-3} to 10^4. The results are given in Figure 7.4. The performance is substantially improved for the new minimax con-

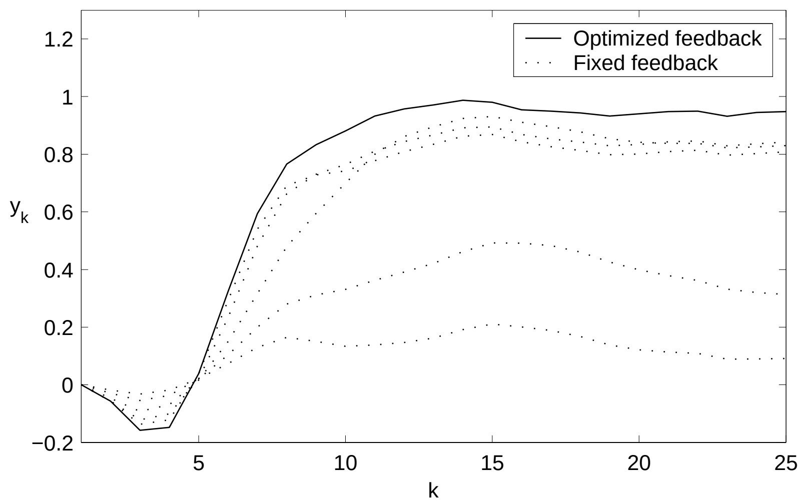

Figure 7.4: Closed-loop response with the proposed parameterization of the feedback prediction. Dashed lines represent closed-loop responses from minimax controllers using fixed feedback matrices defined using LQ controllers with different control weights ranging from 10^{-3} to 10^4

[3]Discretized version (zero-order hold, sample-time 1 second) of the system $\frac{0.25^2(-2s+1)}{s(s^2+0.25^2)}$.

troller. The output y_k is stationed around 0.95, while the best minimax controller with a fixed feedback matrix settles at an output level around 0.85.

The improvement using the proposed parameterization of U is obvious, but the price is high in computational burden, due to the large number of decision variables. Let us therefore evaluate the approximations introduced in Section 7.5.

Example 7.5 (Alternative parameterizations)
Four different ways to parameterize the feedback prediction matrix have been proposed in this chapter. To test these methods, four controllers are implemented.

- $\mathcal{C}_1$: Full parameterization (7.15b).
- $\mathcal{C}_2$: Toeplitz-like parameterization (7.38)
- $\mathcal{C}_3$: Linearized parameterization (7.39). The feedback prediction matrix $\mathcal{L}_X$ is linearized around an LQ controller with $Q = R = 1$.
- $\mathcal{C}_4$: Weighted parameterization (7.41). The matrices used to define the feedback matrices are LQ controllers with $Q = 1$ and $R = 0.01, 0.1, 1, 10$ and 100.

Applying these controllers on the system in Example 7.4 gives the step-responses shown in Figure (7.5). The results are encouraging. The reduced degree of freedom in the parameterizations has not influenced the performance particularly.

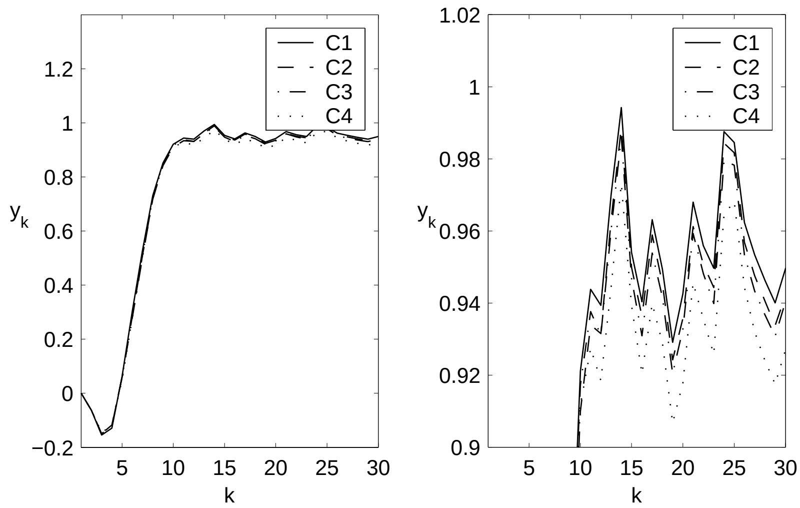

Figure 7.5: Closed-loop responses with different parameterizations. Complete step-responses in the left figure while the right figure is zoomed around the reference level $y_k = 1$. The full parameterization performs best, but the difference is very small compared to the approximations.

The ultimate test is to see how the proposed controller compares to the exact solution to the closed-loop minimax MPC problem, given by the optimization problem (7.6). Define the extreme realizations $W \in \mathbf{Co}\{W^{(1)}, W^{(2)}, \ldots, W^{(2^{N-1})}\}$ and corresponding control realizations $U^{(i)}$ (see Figure 7.2). The closed-loop minimax problem (7.6), applied to our minimax formulation, can then be written as.

$$\begin{array}{lrcl} \min\limits_{U^i,t} & t & & \\ \text{subject to} & \left\| \begin{matrix} \mathcal{Q}^{1/2}\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U^{(i)} + \mathcal{G}W^{(i)}) \\ \mathcal{R}^{1/2}U^{(i)} \end{matrix} \right\| & \leq & t \\ & \mathcal{E}_u U^{(i)} & \leq & \mathcal{F}_u \\ & \mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}U^{(i)} + \mathcal{G}W^{(i)}) & \leq & \mathcal{F}_x \end{array} \tag{7.42}$$

Example 7.6 (Comparison with closed-loop minimax solution)
Once again, we return to the system in Example 7.4. All numerical data is the same, except for the prediction horizon N which now decreased to $N = 7$. The reason is the exponential explosion of the number of variables and constraints, discussed in Section 7.2. The closed-loop step responses are given in Figure 7.6.

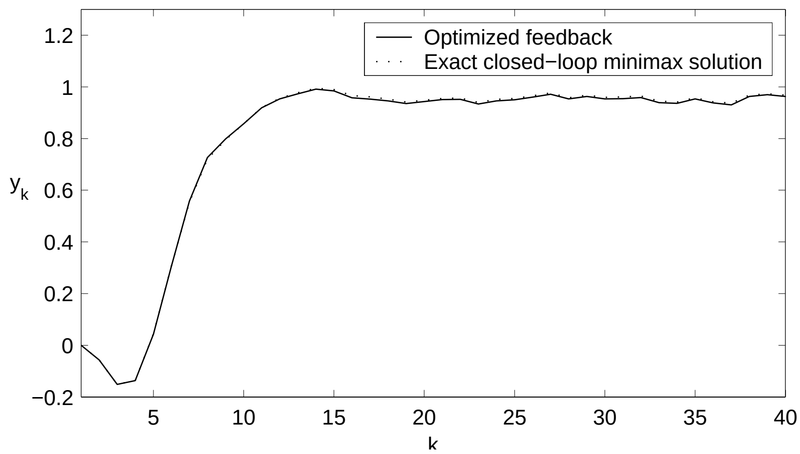

Figure 7.6: Step-responses for proposed minimax controller with on-line optimized feedback predictions, and the exact closed-loop minimax MPC controller. The closed-loop behavior is essentially the same.

As one can see in the figure, the performance of the two controllers are almost identical. The results are very encouraging, considering that two approximations are involved in our minimax controller (the semidefinite relaxation and the linear parameterization of U).

8

A stochastic interpretation of minimax MPC

The aim of this chapter is to show a structural similarity between two, at a first glance very different, approaches to MPC. To be more precise, to show a connection between stochastic risk-sensitive controllers and the deterministic minimax controllers introduced in previous chapters.

So why is this of interest? To begin with, it is intriguing from a mathematical point of view that the connections exist, and the connections can hopefully, in the end, help to deepen our understanding of the two approaches. A more aesthetic reason is that the results might calm the reader with a preference for stochastic and an aversion against deterministic unknown-but-bounded assumptions, or the other way around. All results in the previous chapters can just as well be interpreted in a stochastic framework.

Connections between deterministic worst-case and stochastic risk-sensitive approaches have been discovered for various problems and are well known properties. Connections between H_∞-control and risk-sensitive control are firmly established in (Glover and Doyle, 1988), and were discussed already in (Jacobson, 1977). The results are however quite different to ours since they address (mainly) infinite horizon problems with induced energy norms, and design linear feedback controllers, in contrast to the framework here with amplitude bounded disturbances, control and state constraints, finite horizon performance measures, and semidefinite relaxations.

8.1 Deterministic and Stochastic Models

The class of systems we address is the same as before, constrained linear discrete-time systems with external disturbances.

$$x_{k+1} = Ax_k + Bu_k + Gw_k \tag{8.1a}$$
$$y_k = Cx_k \tag{8.1b}$$

The two approaches we compare use different ways to model the external disturbance w_k. Basically, we are comparing a stochastic framework with independent Gaussian disturbances $w_k \in \mathbb{W}_{\mathcal{N}} = \mathcal{N}(0, I)$ and the deterministic, unknown-but-bounded, assumption $w_k \in \mathbb{W}_2$ or $w_k \in \mathbb{W}_\infty$.

Of course, it is assumed that the measurement error model, if there is any, is of corresponding type. However, we will not explicitly work with the estimation part so we do not address this issue. Instead, we assume the following model of the estimation error.

$$x_k = x_{k|k} + P_k^{-1/2} z_k \tag{8.2}$$

The expression $P_k^{-1/2} z_k$ hence denote the (unknown) state estimation error. The two different models of the state estimate error are

$$z_k \in \mathcal{Z}_{\mathcal{N}} = \mathcal{N}(0, I) \tag{8.3a}$$
$$z_k \in \mathcal{Z}_2 = \{z \ : \ ||z|| \leq 1\} \tag{8.3b}$$

This corresponds to a normal distributed estimation error with covariance matrix P_k^{-1}, typically obtained using a Kalman filter or the corresponding risk-sensitive filter (Hassibi et al., 1999), or a guaranteed estimation error $e_k^T P_k e_k \leq 1$, obtained using ellipsoidal state estimation, see Section 6.2.1.

8.2 Risk and Minimax Performance Objectives

The objective function used in MPC is typically a finite horizon quadratic performance measure.

$$J_k = \sum_{j=0}^{N-1} y_{k+j|k}^T Q y_{k+j|k} + u_{k+j|k}^T R u_{k+j|k} \tag{8.4}$$

Since $y_{\cdot|k}$ is uncertain due to both state estimation error and future disturbances, this has to be addressed in some way. The classical approach is to assume that the estimation error and the external disturbances are Gaussian, and minimize the expected value of (8.4)

$$\min_u \mathbf{E} J_k, \quad z_k \in \mathbb{Z}_{\mathcal{N}}, \ w \in \mathbb{W}_{\mathcal{N}} \tag{8.5}$$

It is well known and easily shown that this is equivalent to a problem without estimation error and external disturbances, i.e., the control law will be identical to the nominal MPC controller in Chapter 2.

The standard approach to robustify nominal MPC is to employ a minimax strategy, i.e., optimize worst-case performance. In the previous chapters, we have derived methods for this and showed that, e.g., the problem

$$\min_{u} \max_{z_k,w} J_k, \quad z_k \in \mathbb{Z}_2,\ w \in \mathbb{W}_2 \tag{8.6}$$

gives rise to a problem that can be solved with semidefinite programming.

It will be shown that the proposed deterministic worst-case approach has connections to a stochastic approach, so called risk-sensitive control (Jacobson, 1977; Whittle, 1981). The idea in risk-sensitive control is to introduce a scalar risk-parameter[1] θ and minimize

$$\min_{u} \frac{2}{\theta} \log \mathbf{E} e^{\frac{\theta}{2}J_k}, \quad z_k \in \mathbb{Z}_{\mathcal{N}},\ w \in \mathbb{W}_{\mathcal{N}} \tag{8.7}$$

A controller using a positive θ is said to be risk-sensitive, or risk-averse. A controller using $\theta = 0$ is called risk-neutral, whereas a negative θ gives an 'optimistic' or risk-willing controller. A risk-willing controller essentially assumes that future disturbances will be benign to the system. Note that if θ is chosen too large, the problem will break down and the expectation will be infinite.

By letting θ approach zero, the nominal MPC problem is recovered. This is most easily seen using a Taylor expansion of the performance measure.

$$\frac{2}{\theta} \log \mathbf{E} e^{\frac{\theta}{2}J_k} \approx \mathbf{E}J_k + \frac{\theta}{4}\mathbf{E}(J_k - \mathbf{E}J_k)^2 \tag{8.8}$$

This expression also reveals a crucial property of the risk-sensitive performance measure; higher order moments are included. The practical implication of this is that one would expect the risk-sensitive controller to give a closed-loop system with smaller peaks on output and input variables.

Despite the awkward looking performance measure (8.7), it can be shown that the risk-sensitive controller for an unconstrained linear system with Gaussian disturbances leads to a linear state feedback controller (Jacobson, 1977). However, in this work, we are interested in the constrained case and the relations to minimax control, so these results are not of much help for us.

Note that we are not addressing exactly the same problem as in (Jacobson, 1977) and (Whittle, 1981). The formulation here assumes that no measurements are obtained in the future, i.e., an open-loop performance criteria is minimized. The reason is that the minimax formulation uses an open-loop formulation.

[1] For notational convenience we have changed the sign on the risk factor compared to standard notation (Whittle, 1981)

8.3 Minimax MPC is Risk-averse

This section will present the main result, a connection between the risk-sensitive MPC controller and the minimax MPC in Chapter 6.

The notation used is the same as in Chapter 5 and 6. To begin with, define an extended disturbance with the stacked future process disturbances W and the current (normalized) state estimation error z_k.

$$Z = \begin{pmatrix} W^T & z_k^T \end{pmatrix}^T \in \mathcal{N}(0, I) \tag{8.9}$$

Define $\mathcal{K} = \begin{pmatrix} \mathcal{G} & \mathcal{A}P^{-1/2} \end{pmatrix}$ and introduce the state and output predictions.

$$\begin{aligned} Y &= \mathcal{C}\mathcal{X} && \text{(8.10a)} \\ X &= \mathcal{A}x_k + \mathcal{B}U + \mathcal{G}W \\ &= \mathcal{A}(x_{k|k} + P^{-1/2}z_k) + \mathcal{B}U + \mathcal{G}W \\ &= \mathcal{A}x_{k|k} + \mathcal{B}U + \mathcal{K}Z && \text{(8.10b)} \end{aligned}$$

To simplify notation in the following sections, let $\mathcal{Q}_C = \mathcal{C}^T\mathcal{Q}\mathcal{C}$. The finite horizon quadratic performance measure (8.4) can then be written as

$$J_k = Y^T\mathcal{Q}Y + U^T\mathcal{R}U = X^T\mathcal{Q}_C X + U^T\mathcal{R}U \tag{8.11}$$

8.3.1 Risk-sensitive MPC

Inserting J_k in the risk-sensitive optimization problem yields

$$U = \arg\min_U \frac{2}{\theta}\log \mathbf{E}e^{\frac{\theta}{2}((\mathcal{A}x_{k|k}+\mathcal{B}U+\mathcal{K}Z)^T\mathcal{Q}_C(\mathcal{A}x_{k|k}+\mathcal{B}U+\mathcal{K}Z)+U^T\mathcal{R}U)} \tag{8.12}$$

The performance measure can be rewritten by pulling out the deterministic part from the expectation, and we obtain

$$\begin{aligned} &(\mathcal{A}x_{k|k} + \mathcal{B}U)^T\mathcal{Q}_C(\mathcal{A}x_{k|k} + \mathcal{B}U) + U^T\mathcal{R}U \\ &\qquad + \frac{2}{\theta}\log \mathbf{E}e^{\frac{\theta}{2}(Z^T\mathcal{K}^T\mathcal{Q}_C\mathcal{K}\mathcal{Z}+2Z^T\mathcal{K}^T\mathcal{Q}_C(\mathcal{A}x_{k|k}+\mathcal{B}U))} \end{aligned} \tag{8.13}$$

The expected value of a function $f(z)$ is given by $\mathbf{E}f(z) = \int_{\mathbb{R}^n} f(z)p_z(z)\,dz$ where $p_z(z)$ is the probability density function (PDF). A Gaussian variable $z \in \mathbb{R}^n$ with mean $\bar{z}$ and covariance P has the following PDF.

$$p_z(z) = \frac{1}{(2\pi)^{n/2}(\det P)^{1/2}} e^{-\frac{1}{2}(z-\bar{z})^T P^{-1}(z-\bar{z})} \tag{8.14}$$

Using this, we can calculate the expectation w.r.t $Z \in \mathcal{N}(0, I)$ (let n momentarily denote the dimension of the vector Z).

$$\begin{aligned}
&\mathbf{E}e^{\frac{\theta}{2}(Z^T\mathcal{K}^T\mathcal{Q}_C\mathcal{K}Z + 2Z^T\mathcal{K}^T\mathcal{Q}_C(\mathcal{A}x_{k|k}+\mathcal{B}U))} \\
&\qquad = \frac{1}{(2\pi)^{n/2}}\int_{\mathbb{R}^n} e^{\frac{\theta}{2}(Z^T\mathcal{K}^T\mathcal{Q}_C\mathcal{K}Z + 2Z^T\mathcal{K}^T\mathcal{Q}_C(\mathcal{A}x_{k|k}+\mathcal{B}U))}e^{-\frac{1}{2}Z^TZ}\,dZ \\
&\qquad = \frac{1}{(2\pi)^{n/2}}\int_{\mathbb{R}^n} e^{-\frac{1}{2}Z^T(I-\theta\mathcal{K}^T\mathcal{Q}_C\mathcal{K})Z + \theta Z^T\mathcal{K}^T\mathcal{Q}_C(\mathcal{A}x_{k|k}+\mathcal{B}U)}\,dZ \qquad (8.15)
\end{aligned}$$

To simplify notation, introduce the matrix $\mathcal{M}$ and the vector S

$$\mathcal{M} = I - \theta\mathcal{K}^T\mathcal{Q}_C\mathcal{K}, \quad S = \theta\mathcal{K}^T\mathcal{Q}_C(\mathcal{A}x_{k|k} + \mathcal{B}U) \tag{8.16}$$

and note that the final expression in (8.15) can be written as

$$\frac{1}{(2\pi)^{n/2}}\int_{\mathbb{R}^n} e^{-\frac{1}{2}Z^T\mathcal{M}Z + Z^TS}\,dZ \tag{8.17}$$

Completing the squares yields

$$\frac{1}{(2\pi)^{n/2}}\int_{\mathbb{R}^n} e^{-\frac{1}{2}(Z-\mathcal{M}^{-1}S)^T\mathcal{M}(Z-\mathcal{M}^{-1}S)+\frac{1}{2}S^T\mathcal{M}^{-1}S}\,dZ \tag{8.18}$$

Pull out the constant term, multiply and divide with $(\det\mathcal{M}^{-1})^{1/2}$

$$\frac{e^{\frac{1}{2}S^T\mathcal{M}^{-1}S}(\det\mathcal{M}^{-1})^{1/2}}{(2\pi)^{n/2}(\det\mathcal{M}^{-1})^{1/2}}\int_{\mathbb{R}^n} e^{-\frac{1}{2}(Z-\mathcal{M}^{-1}S)^T\mathcal{M}(Z-\mathcal{M}^{-1}S)}\,dZ \tag{8.19}$$

This expression contains an integrated PDF for a variable $Z \in \mathcal{N}(\mathcal{M}^{-1}S, \mathcal{M}^{-1})$. A PDF integrates to 1, so the expression simplifies to

$$e^{\frac{1}{2}S^T\mathcal{M}^{-1}S}(\det\mathcal{M}^{-1})^{1/2} \tag{8.20}$$

Insert this expression in (8.13) and the objective function is

$$(\mathcal{A}x_{k|k} + \mathcal{B}U)^T\mathcal{Q}_C(\mathcal{A}x_{k|k} + \mathcal{B}U) + U^T\mathcal{R}U + \frac{2}{\theta}\log(e^{\frac{1}{2}S^T\mathcal{M}^{-1}S}(\det\mathcal{M}^{-1})^{1/2}) \tag{8.21}$$

The following expression is obtained after some simplifications.

$$(\mathcal{A}x_{k|k} + \mathcal{B}U)^T\mathcal{Q}(\mathcal{A}x_{k|k} + \mathcal{B}U) + U^T\mathcal{R}U + \frac{1}{\theta}S^T\mathcal{M}^{-1}S + \frac{1}{\theta}\log\det\mathcal{M}^{-1} \tag{8.22}$$

By inserting the definition of S, removing all constant terms and rearranging the position of θ, minimization of the risk-sensitive performance measure (8.22) boils down to the following optimization problem.

$$\begin{aligned}
U = \arg\min_U\ &(\mathcal{A}x_{k|k} + \mathcal{B}U)^T\mathcal{Q}_C(\mathcal{A}x_{k|k} + \mathcal{B}U) + U^T\mathcal{R}U \\
&+ (\mathcal{A}x_{k|k} + \mathcal{B}U)^T\mathcal{Q}_C\mathcal{K}\big(\frac{1}{\theta}I - \mathcal{K}^T\mathcal{Q}_C\mathcal{K}\big)^{-1}\mathcal{K}^T\mathcal{Q}_C(\mathcal{A}x_{k|k} + \mathcal{B}U) \qquad (8.23)
\end{aligned}$$

Remarkably, we still have a quadratic program with the same complexity as the nominal MPC problem. The difference compared a nominal MPC problem is that the state weight $\mathcal{Q}_C$ has been replaced with

$$\mathcal{Q}_C + \mathcal{Q}_C\mathcal{K}\big(\frac{1}{\theta}I - \mathcal{K}^T\mathcal{Q}_C\mathcal{K}\big)^{-1}\mathcal{K}^T\mathcal{Q}_C \tag{8.24}$$

or equivalently, that the original output weight $\mathcal{Q}$ has been changed to

$$\mathcal{Q} + \mathcal{Q}\mathcal{C}\mathcal{K}\big(\frac{1}{\theta}I - \mathcal{K}^T\mathcal{Q}_C\mathcal{K}\big)^{-1}\mathcal{K}^T\mathcal{C}^T\mathcal{Q} \tag{8.25}$$

The calculations above were performed under the assumption that the matrix $\mathcal{M}$ was positive definite (the expectation would be infinite otherwise). For this to hold, θ has to be strictly smaller than the largest eigenvalue of the matrix $\mathcal{K}^T\mathcal{Q}_C\mathcal{K}$. Note that this is in some sense an indication that the risk-parameter θ not is as universal as one would expect, since it is model dependent.

8.3.2 Minimax MPC

Let us now go back to the minimax MPC problem that we have addressed in previous chapters.

$$U = \arg\min_U \max_Z Y^T\mathcal{Q}Y + U^T\mathcal{R}U$$

A semidefinite relaxation gave a semidefinite program where an upper bound t was minimized, subject to the following LMI (and additional linear constraints).

$$\begin{pmatrix} t - \sum_{i=1}^s \tau_i & (\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U))^T & U^T & 0 \\ \mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U) & \mathcal{Q}^{-1} & 0 & \mathcal{C}\mathcal{K} \\ U & 0 & \mathcal{R}^{-1} & 0 \\ 0 & (\mathcal{C}\mathcal{K})^T & 0 & \mathcal{T} \end{pmatrix} \succeq 0 \tag{8.26}$$

This is the formulation that would be used to solve the optimization problem, since (8.26) is an LMI in U, t and the multipliers τ. However, the aim here is to show connections with risk-sensitive control. To do this, rewrite the LMI using a Schur complement and pose the problem as minimization of t subject to

$$(\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U)^T)(\mathcal{Q}^{-1} - \mathcal{C}\mathcal{K}\mathcal{T}^{-1}(\mathcal{C}\mathcal{K})^T)^{-1}\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U) + U^T\mathcal{R}U \leq t - \sum_{i=1}^s \tau_i$$

Apply the Sherman-Morrison-Woodbury formula (Golub and van Loan, 1996)

$$(\mathcal{Q}^{-1} - \mathcal{C}\mathcal{K}\mathcal{T}^{-1}(\mathcal{C}\mathcal{K})^T)^{-1} = \mathcal{Q} + \mathcal{Q}\mathcal{C}\mathcal{K}\big(\mathcal{T} - (\mathcal{C}\mathcal{K})^T\mathcal{Q}\mathcal{C}\mathcal{K}\big)^{-1}(\mathcal{C}\mathcal{K})^T\mathcal{Q} \tag{8.27}$$

and obtain the equivalent constraint

$$\begin{aligned} &(\mathcal{A}x_{k|k} + \mathcal{B}U)^T\mathcal{Q}_C(\mathcal{A}x_{k|k} + \mathcal{B}U) + U^T\mathcal{R}U \\ &+ (\mathcal{A}x_{k|k} + \mathcal{B}U)^T\mathcal{Q}_C\mathcal{K}\big(\mathcal{T} - \mathcal{K}^T\mathcal{Q}_C\mathcal{K}\big)^{-1}\mathcal{K}^T\mathcal{Q}_C(\mathcal{A}x_{k|k} + \mathcal{B}U) \leq t - \sum_{i=1}^s \tau_i \end{aligned} \tag{8.28}$$

Hence, our minimax controller is defined by the following optimization problem.

$$U = \arg\min_{U,\tau} \sum_{i=1}^{s} \tau_i + (\mathcal{A}x_{k|k} + \mathcal{B}U)^T \mathcal{Q}_C(\mathcal{A}x_{k|k} + \mathcal{B}U) + U^T \mathcal{R}U$$
$$+ (\mathcal{A}x_{k|k} + \mathcal{B}U)^T \mathcal{Q}_C \mathcal{K}\big(\mathcal{T} - \mathcal{K}^T \mathcal{Q}_C \mathcal{K}\big)^{-1} \mathcal{K}^T \mathcal{Q}_C(\mathcal{A}x_{k|k} + \mathcal{B}U) \quad (8.29)$$

After comparing this with the final optimization problem for the risk-sensitive approach (8.23), the connections are obvious.

Only Estimation Error

In the case when we have ellipsoidal estimation errors $z_k \in \mathbb{Z}_2$, and neglect external disturbances, we have $Z = z_k$ and $\mathcal{K} = \mathcal{A}P^{-1/2}$. Furthermore, there is only one multiplier ($s = 1$) and $\mathcal{T}$ is diagonal $\mathcal{T} = \tau I$. If we now compare the risk-sensitive optimization problem (8.23) and the minimax problem (8.29), we see that τ plays the role of an inverse risk-parameter, and the minimax controller can be interpreted as a controller that minimizes a risk-sensitive performance measure, with an additional penalty, the τ-term, against using a too small risk-parameter. (recall that a large τ corresponds to a small θ and a less risk-averse controller.)

General Case

In the general case, the matrix $\mathcal{T}$ is no longer a scaled identity matrix, but a diagonal matrix, with a structure depending on whether state estimation errors are addressed or not, and the set $\mathbb{W}$. In this case, we can interpret the risk-sensitive solution as an approximate solution to the minimax problem where we conservatively have forced all τ-variables to be identical, and fixed this value.

8.3.3 State and Input Constraints

The discussion above has been concentrating on the relationship between the objective functions in nominal MPC, risk-sensitive MPC, and minimax MPC. It turns out that robust constraint satisfaction in minimax MPC also can be given a stochastic interpretation. Let us for notational simplicity concentrate on the case without state estimation error.

It was shown in Section 5.2.1 that robust state constraints $\mathcal{E}_x X \le \mathcal{F}_x \; \forall W \in \mathbb{W}$ could be written as $\mathcal{E}_x X + \gamma \le \mathcal{F}_x$. Depending on the type of uncertainty, γ is calculated as

$$\gamma_i = ||\omega_i||_1 \quad (8.30)$$

or

$$\gamma_i = \sum_{j=1}^{N} ||\omega_{ij}|| \quad (8.31)$$

for $w \in \mathbb{W}_\infty$ and $w \in \mathbb{W}_2$ respectively. The vectors ω_i and ω_{ij} where defined from a partition of the matrix $\mathcal{E}_x\mathcal{G}$

$$\mathcal{E}_x\mathcal{G} = \begin{pmatrix} \omega_1^T \\ \omega_2^T \\ \vdots \end{pmatrix}, \quad \omega_i^T = \begin{pmatrix} \omega_{i1}^T & \omega_{i2}^T & \dots & \omega_{iN}^T \end{pmatrix} \tag{8.32}$$

This partition allowed us to write each row in $\mathcal{E}_x\mathcal{G}W$ as

$$(\mathcal{E}_x\mathcal{G}W)_i = \omega_i^T W = \sum_{j=1}^{N} \omega_{ij}^T w_{k+j-1|k} \tag{8.33}$$

If we would have had a stochastic model $w \in \mathbb{W}_\mathcal{N}$, it holds that the scalar $\omega_{ij}^T w_{k+j-1|k} \in \mathcal{N}(0, \omega_{ij}^T \omega_{ij}) = \mathcal{N}(0, ||\omega_{ij}||^2)$. This means that $||\omega_{ij}||$ is the standard deviation of each stochastic term added to the nominal part of the constraint, hence the term (8.31) corresponds to a sum of the standard deviations from the uncertain term (note, it is not the standard deviation of the sum of the uncertainties). A similar interpretation can be given for (8.30). It can be seen as the sum of the standard deviations of the uncertainties coming from each element of $w \in \mathbb{W}_\mathcal{N}$.

9

Efficient solution of a minimax MPC problem

An underlying idea in the previous chapters has been that once a convex optimization problem is obtained, we rest assured and consider the problem solved, since convex problems can be solved relatively efficiently. While standard general-purpose solvers might suffice for rapid prototyping of algorithms and testing small-scale problems, they loose much performance due to their generality.

The semidefinite programs derived in previous chapters have structure that can be exploited to improve computational efficiency in a solver. This includes sparseness, diagonal blocks, linear constraints and easily obtained initial solutions, just to mention a few.

In this chapter, we study computational aspects of solving the semidefinite programs. A dedicated solver is developed for one of the semidefinite relaxations in Chapter 5, and compared with general-purpose semidefinite solvers. It is found that the developed solver is capable of solving the minimax problems much more efficiently by exploiting the inherent structure. The computational improvements are essentially the result of two features of the proposed solver; efficient calculation of the Hessian, and the decision to work only in the primal space.

9.1 The Semidefinite Program

The following semidefinite program was derived in Section 5.2.1 to solve a semidefinite relaxation of a minimax MPC problem.

$$\begin{array}{ll} \min\limits_{U,t,\tau} & t \\ \text{subject to} & \begin{pmatrix} t-\sum_{i=1}^{s}\tau_i & (\mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}U))^T & U^T & 0 \\ \mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}U) & \mathcal{Q}^{-1} & 0 & \mathcal{C}\mathcal{G} \\ U & 0 & \mathcal{R}^{-1} & 0 \\ 0 & (\mathcal{C}\mathcal{G})^T & 0 & \mathcal{T} \end{pmatrix} \succeq 0 \\ & \mathcal{E}_uU \leq \mathcal{F}_u \\ & \mathcal{E}_x(\mathcal{A}x_{k|k}+\mathcal{B}U)+\gamma \leq \mathcal{F}_x \end{array} \tag{9.1}$$

To simplify notation, it is assumed that $\mathcal{T} = \text{diag}(\tau) = \oplus_{i=1}^{N}\tau_i$. This corresponds to scalar disturbances $w \in \mathbb{W}_\infty$ in the minimax problem.

The first step towards an efficient solver is to rewrite the problem slightly. To begin with, introduce an additional set of decision variables Y, and add the constraint $Y = \mathcal{A}x_{k|k} + \mathcal{B}U$ to the semidefinite program. This might sound inefficient, but it will simplify many calculations and enable us to derive a compact expression for the Hessian later. Furthermore, perform a variable change and introduce $\tilde{t} = t - \sum_{i=1}^{s}\tau_i$. The new problem is

$$\begin{array}{ll} \min\limits_{U,Y,\tilde{t},\tau} & \tilde{t}+\sum_{i=1}^{s}\tau_i \\ \text{subject to} & \begin{pmatrix} \tilde{t} & Y^T & U^T & 0 \\ Y & \mathcal{Q}^{-1} & 0 & \mathcal{C}\mathcal{G} \\ U & 0 & \mathcal{R}^{-1} & 0 \\ 0 & (\mathcal{C}\mathcal{G})^T & 0 & \mathcal{T} \end{pmatrix} \succeq 0 \\ & Y = \mathcal{C}(\mathcal{A}x_{k|k}+\mathcal{B}U) \\ & \mathcal{E}_uU \leq \mathcal{F}_u \\ & \mathcal{E}_x(\mathcal{A}x_{k|k}+\mathcal{B}U)+\gamma \leq \mathcal{F}_x \end{array} \tag{9.2}$$

This is the semidefinite program addressed in this chapter. The most important properties are summarized below.

- The LMI is relatively sparse since $\mathcal{Q}$ and $\mathcal{R}$ typically are diagonal and $\mathcal{T}$ always is diagonal.
- The variables $(Y, U, \tilde{t}, \tau)$ all enter in at most two elements in the semidefinite constraint. This follows from treating both Y and U as variables.
- The free variables enter in an arrow structure (in first row, first column, and in the diagonal).
- The only thing that changes between different sample instants is $x_{k|k}$ in the linear constraints.

The question is if any of these properties can be exploited to develop an efficient solver, and to what extent this will improve computational performance.

Standard Problem

To simplify notation further, introduce the following standard problem.

$$\begin{array}{lrcl} \min\limits_{x} & c^T x & & \\ \text{subject to} & F(x) = \begin{pmatrix} x_1 & x_2^T & 0 \\ x_2 & \Phi & \mathcal{H} \\ 0 & \mathcal{H}^T & \operatorname{diag}(x_3) \end{pmatrix} & \succeq & 0 \\ & Ex - f & = & 0 \\ & Cx - d & \geq & 0 \end{array} \tag{9.3}$$

The matrices Φ and $\mathcal{H}$ are constant matrices, x_1 is a scalar, while x_2 and x_3 are column vectors. Define $x = \begin{pmatrix} x_1 & x_2^T & x_3^T \end{pmatrix}^T$. The problem (9.2), and many of the MPC controllers in the previous chapters, can readily be cast in this form.

9.2 Solving Semidefinite Programs

The huge interest in LMIs and semidefinite programming comes from the fact that the optimization problems are convex, and thus can be solved be solved relatively efficiently. Consider the following standard semidefinite program.

$$\begin{array}{lrcl} \min\limits_{x} & c^T x & & \\ \text{subject to} & F(x) = F_0 + \sum_{i=1}^m F_i x_i & \succeq & 0 \end{array} \tag{9.4}$$

The prevailing approach today to solve this semidefinite program is to use primal-dual algorithms. General-purpose solvers using this type of algorithms include, among others, (Vandenberghe and Boyd, 1998; Borchers, 1999; Sturm, 1999; Toh et al., 1999; Yamashita et al., 2002).

A primal-dual solver works with the Karush-Kuhn-Tucker (KKT) conditions of the semidefinite program. It can be shown, under fairly weak assumptions (Boyd and Vandenberghe, 2002), that the primal variable x is optimal if and only if there exist a dual matrix Z such that

$$F(x) \succeq 0 \tag{9.5a}$$
$$Z \succeq 0 \tag{9.5b}$$
$$ZF(x) = 0 \tag{9.5c}$$
$$\operatorname{Tr} F_i Z = c_i \tag{9.5d}$$

A primal-dual solver typically performs better than a primal solver for general problems, since the dual variable can be used to obtain better search directions. The price paid is the increased number of variables and constraints.

There are however cases when a primal solver is preferred. Our problem is one instance that probably is more efficiently addressed with a primal algorithm. The reason is that the matrix $F(x)$ is structured and relatively sparse, whereas the dual matrix Z is completely dense. This has a large impact on the effectiveness of a primal-dual solver. For instance, sparsity in $F(x)$ makes it easy to check the constraint $F(x) \succeq 0$, whereas the constraint $Z \succeq 0$ requires much more computational effort. This is the motivation for the decision to implement a primal solver.

9.3 Semidefinite Programming with SUMT

A simple way to solve constrained optimization problems is to minimize, for decreasing values of a barrier parameter μ, the original objective function appended with a barrier function on the inequality constraints. This approach is called sequential unconstrained minimization technique (SUMT) (Fiacco and McCormick, 1968). Primal-dual algorithms can often be interpreted in terms of primal barrier algorithms (Nocedal and Wright, 1999), but the difference is that a primal barrier algorithm has no help from a dual variable to, for example, find good search directions and update the barrier parameter μ. Nevertheless, our solver will be based on a simple primal SUMT strategy, for reasons given in the previous sections.

The problem to solve is a mixed linear and semidefinite program with linear equality constraints.

$$\boxed{\begin{array}{rrcl} \min\limits_{x} & c^T x & & \\ \text{subject to} & F(x) & \succeq & 0 \\ & Ex - f & = & 0 \\ & Cx - d & \geq & 0 \end{array}} \tag{9.6}$$

Appending the objective function with the logarithmic barriers $\log\det F(x)$ and $\sum \log(Cx - d)$ yields the merit function (summation operator without index means summation of all elements).

$$f(x,\mu) = c^T x - \mu \log\det F(x) - \mu \sum \log(Cx - d)$$

Some notation is introduced for future reference.

$$\begin{aligned} f(x,\mu) &= c^T x - \mu\phi(x) \\ \phi(x) &= \phi_S(x) + \phi_L(x) \\ \phi_S(x) &= \log\det F(x) \\ \phi_L(x) &= \sum \log(Cx - d) \end{aligned}$$

The problem to solve for decreasing values of μ in the SUMT algorithm is

$$\boxed{\begin{array}{rrcl} \min\limits_{x} & f(x,\mu) & & \\ \text{subject to} & Ex - f & = & 0 \end{array}} \tag{9.7}$$

Minimization (for a fixed μ) is typically done with a Newton method. A Newton method means that the function $f(x,\mu)$ is approximated with a quadratic model

$$f(x+p,\mu) \simeq f(x,\mu) + b^T p + \frac{1}{2} p^T H p \tag{9.8}$$

The quadratic model is obtained from a second order Taylor expansion of the merit function

$$b = \nabla_x f(x,\mu) \tag{9.9a}$$

$$H = \nabla^2_{xx} f(x,\mu) \tag{9.9b}$$

A suitable search direction is found by minimizing the approximate model, subject to the linear equality constraints. This can be stated as an equality constrained quadratic program.

$$\boxed{\begin{array}{rrcl} \min\limits_{p} & b^T p + \frac{1}{2} p^T H p & & \\ \text{subject to} & Ep & = & 0 \end{array}} \tag{9.10}$$

When the search direction p is found, a line search is performed to minimize the merit function along the search direction. A step is then taken and the procedure is repeated until an (approximate) optimum is found for the current μ. The barrier parameter μ is then decreased and a new unconstrained optimization problem is solved. This is repeated until a solution is obtained for a sufficiently small μ.

The verbally explained algorithm can be put in the following algorithmic format (Fiacco and McCormick, 1968; den Hertog, 1994).

Algorithm 9.1 (SUMT)

Given : Strictly feasible initial iterate x, initial barrier parameter $\mu > 0$, barrier update parameter $\theta \in (0,1)$, proximity parameter $\eta \in (0,1)$ and desired accuracy $\epsilon > 0$.

begin
  repeat
    Update barrier parameter $\mu := \theta\mu$
    repeat
      Calculate Hessian H and gradient b
      Calculate search direction p
      Calculate suitable step length r
      Update $x := x + rp$
    until $p^T H p \leq \eta$
  until $\mu \leq \frac{\epsilon}{4n}$
end

This algorithm will be used to solve our semidefinite program. The following sections will go through the different steps in detail, and then show how we can exploit structure in our semidefinite program to perform some of the steps more efficiently.

9.3.1 Analytic Expression of Gradient and Hessian

To find the search direction, the Hessian and gradient of the barrier $\phi(x)$ are needed. For the barrier function $\phi_S(x) = \log\det F(x)$ where

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i, \quad F_i \in \mathbb{R}^{n\times n} \tag{9.11}$$

the gradient and Hessian are given by (Vandenberghe and Boyd, 1996)

$$(\nabla_x \phi_S(x))_i = \operatorname{Tr} F(x)^{-1} F_i \tag{9.12a}$$

$$(\nabla^2_{xx} \phi_S(x))_{ij} = -\operatorname{Tr} F(x)^{-1} F_i F(x)^{-1} F_j \tag{9.12b}$$

For a general dense unstructured problem, computing the Hessian can become a major bottleneck, since the complexity is $O(n^3m+n^2m^2)$[1]. However, by exploiting the simple structure on F_i in our semidefinite constraint, we can significantly reduce the amount of work. In fact, the analytic expression below is one of the main contributor to the performance gains we will see compared to existing software.

Define the inverse of $F(x)$ (the partition means that sizes of corresponding blocks are the same, i.e., ν has the same dimension as $\mathcal{G}$ etc.).

$$F(x)^{-1} = \begin{pmatrix} x_1 & x_2^T & 0 \\ x_2 & \Phi & \mathcal{H} \\ 0 & \mathcal{H}^T & \operatorname{diag}(x_3) \end{pmatrix}^{-1} = \begin{pmatrix} \gamma & \alpha & \tau \\ \alpha^T & \Omega & \nu \\ \tau^T & \nu^T & \Gamma \end{pmatrix} \tag{9.13}$$

By exploiting the structure of the matrices F_i, it is possible to derive (see the appendix) the following expressions for the gradient and Hessian of $\phi_S(x)$ ($\circ$ denotes Hadamard product)

$$\nabla_x \phi_S(x) = \begin{pmatrix} \gamma \\ 2\alpha^T \\ \operatorname{diag}(\Gamma) \end{pmatrix} \tag{9.14a}$$

$$\nabla^2_{xx} \phi_S(x) = -\begin{pmatrix} \gamma^2 & 2\gamma\alpha & \tau\circ\tau \\ 2\gamma\alpha^T & 2(\gamma\Omega + \alpha^T\alpha) & 2\nu \operatorname{diag}(\tau) \\ (\tau\circ\tau)^T & 2\operatorname{diag}(\tau)\nu^T & \Gamma\circ\Gamma \end{pmatrix} \tag{9.14b}$$

Note that the Hessian can be assembled using only $O(n^2)$ operations. It should however be kept in mind that the inverse of $F(x)$ is needed. This is an $O(n^3)$ operation, and we will later see that this is the main bottleneck of the solver.

The gradient and Hessian of the barrier function $\phi_L(x)$ for the linear constraints are easily calculated. Introduce the inverted slacks

$$t_i = \frac{1}{(Cx-d)_i} \tag{9.15a}$$

$$T = \operatorname{diag}(t) \tag{9.15b}$$

[1] m matrix multiplications of $n\times n$ matrices have to be done. Moreover, $m(m+1)/2$ inner products between $n\times n$ matrices are calculated. Since a matrix multiplication is an $O(n^3)$ operation and inner product $O(n^2)$, the total complexity of the compilation is $O(mn^3+n^2m^2)$

The gradient and Hessian of $\phi_L(x)$ are (Nocedal and Wright, 1999)

$$\begin{aligned} \nabla_x(\phi_L(x)) &= C^T t && (9.16a) \\ \nabla^2_{xx}(\phi_L(x)) &= -C^T T^2 C && (9.16b) \end{aligned}$$

Adding the gradient and Hessian of the objective function $c^T x$, $\phi_S(x)$ and ϕ_L gives the gradient b and Hessian H of the merit function $f(x, \mu)$.

$$b = c - \mu \begin{pmatrix} \gamma \\ 2\alpha^T \\ \mathrm{diag}(\Gamma) \end{pmatrix} - \mu C^T t \tag{9.17a}$$

$$H = \mu \begin{pmatrix} \gamma^2 & 2\gamma\alpha & \tau \circ \tau \\ 2\gamma\alpha^T & 2(\gamma\Omega + \alpha^T\alpha) & 2\nu\ \mathrm{diag}(\tau) \\ (\tau \circ \tau)^T & 2\mathrm{diag}(\tau)\nu^T & \Gamma \circ \Gamma \end{pmatrix} + \mu C^T T^2 C \tag{9.17b}$$

9.3.2 Solving the Equality Constrained QP

The Hessian H and gradient b are used to define the equality constrained QP (9.10). By introducing a Lagrange multiplier λ, the KKT conditions for optimality are (Nocedal and Wright, 1999).

$$\begin{pmatrix} H & E^T \\ E & 0 \end{pmatrix} \begin{pmatrix} p \\ \lambda \end{pmatrix} = - \begin{pmatrix} b \\ 0 \end{pmatrix}$$

Depending on the structure on E and H, different strategies can be used to solve this KKT system. In the null space method (Nocedal and Wright, 1999), we first calculate an orthogonal complement[2] $E_\perp$, the null space basis matrix, and solve a linear equation.

$$E_\perp^T H E_\perp v = -E_\perp^T b$$

The Newton step in the original variable x is recovered with

$$p = E_\perp v$$

It will be clear later that the null space strategy indeed is suitable for our problem.

9.3.3 Line Search

Once a search direction p is found, a line search is performed to find a suitable step length $r \geq 0$. A line search means that we try to (approximately) minimize a merit function, here $f(x, \mu)$, along the search direction p, while taking the inequality constraints into account (the linear equality constraint is satisfied by construction

[2] $EE_\perp = 0$ and $E_\perp$ full rank

along the search direction). Hence, the line search problem is

$$\begin{array}{|rrcl|}\hline \min_{r} & c^T(x+rp) - \mu\phi(x+rp) & & \\ \text{subject to} & F(x+rp) & \succeq & 0 \\ & C(x+rp)-d & \geq & 0 \\ \hline \end{array}$$

The line search procedure can typically be divided into two parts. First, the largest possible step length r_{max} with respect to the constraints is calculated. Finding this step length is typically a computationally expensive procedure in a semidefinite solver, due to the semidefinite constraint $F(x+rp) \succeq 0$. In a second step, the merit function is minimized over $r \in (0, r_{max})$. The minimization does not have to be exact. Instead, one can use algorithms that guarantee the step to yield a sufficiently large decrease in the merit function to obtain convergence of the Newton method. One such algorithm is Armijo's backtracking algorithm (Nocedal and Wright, 1999).

Algorithm 9.2 (Armijo backtracking)

Given : $\alpha_A \in (0,1)$ and $\beta_A \in (0,1)$

begin
    $r := r_{max}$
    while $f(x+rp,\mu) - f(x,\mu) > r\alpha_A b^T p$
        $r := \beta_A r$
    end
end

To find the maximal step length and evaluate the function $f(x+rp,\mu)$ during the Armijo backtracking procedure, two approaches have been implemented. The first approach, used in e.g. (Rendl et al., 1995) and (Vandenberghe and Boyd, 1996) is an exact but possibly expensive method based on an eigenvalue decomposition of $F(x)$, while the second approach is an approximate but hopefully efficient method based on Cholesky factorizations of $F(x)$ and backtracking.

Line Search using Eigenvalue Decomposition

From linearity of $F(x)$ and invariance of eigenvalues with respect to a congruence transformation, we have that the following three conditions are equivalent (with R denoting the Cholesky factor of $F(x)$, $F(x) = R^T R$)

$$\begin{aligned} F(x+rp) &\succeq 0 \Leftrightarrow \\ F(x) + r(F(p) - F_0) &\succeq 0 \Leftrightarrow \\ I + rR^{-T}(F(p) - F_0)R^{-1} &\succeq 0 \end{aligned}$$

If we let λ_S denote the eigenvalues of $R^{-T}(x)(F(p) - F_0)R^{-1}$ (or equivalently the generalized eigenvalues of $(F(p) - F_0, F(x))$), it follows that the largest possible step length with respect to the matrix inequality is given by

$$r_S = \begin{cases} \frac{1}{|\min(\lambda_S)|} & \min(\lambda_S) < 0 \\ \infty & \min(\lambda_S) \geq 0 \end{cases}$$

The largest possible step with respect to the linear inequality is easily derived (recall the definition of T in Equation (9.15))

$$\begin{aligned} C(x + rp) - d &\geq 0 \Leftrightarrow \\ Cx - d + rCp &\geq 0 \Leftrightarrow \\ \mathbf{1} + rTCp &\geq 0 \end{aligned}$$

Hence, with $\lambda_L = TCp$ the following step length is obtained.

$$r_L = \begin{cases} \frac{1}{|\min(\lambda_L)|} & \min(\lambda_L) < 0 \\ \infty & \min(\lambda_L) \geq 0 \end{cases}$$

Typically, one should not go all the way to the border of the feasible set (in fact, we must not since the calculations above require $F(x) \succ 0$ and $Cx - d > 0$). Instead, we take a damped step by introducing $\gamma_S \in (0, 1)$ and $\gamma_L \in (0, 1)$. Furthermore, a Newton method should eventually take unit steps, so r is never allowed to be larger than 1. Combining the constraints on r gives us

$$r_{max} = \min(\gamma_S r_S, \gamma_L r_L, 1) \tag{9.18}$$

To evaluate the merit function, or more importantly $f(x + rp) - f(x)$ which is used in the Armijo backtracking algorithm, the previously calculated generalized eigenvalues are re-used.

$$\begin{aligned} \phi_S(x + rp) &= \log\det(F(x) + r(F(p) - F_0)) \\ &= \log\det R^T(I + rR^{-T}(F(p) - F_0)R^{-1})R \\ &= \log\det(I + rR^{-1}(F(p) - F_0)R^{-T}) + \log\det F(x) \\ &= \log\prod(\mathbf{1} + r\lambda_S) + \log\det F(x) \\ &= \sum\log(\mathbf{1} + r\lambda_S) + \log\det F(x) \end{aligned}$$

The barrier for the linear constraints can be simplified in the same way.

$$\begin{aligned} \phi_L(x + rp) &= \sum\log(C(x + rp) - d) \\ &= \sum\log(\mathbf{1} + r\lambda_L) + \sum\log\det(Cx - d) \end{aligned}$$

The decrease $f(x + rp, \mu) - f(x, \mu)$ simplifies to

$$f(x + rp, \mu) - f(x, \mu) = rc^T p - \mu\sum\log(\mathbf{1} + r\lambda_S) - \mu\sum\log(\mathbf{1} + r\lambda_L)$$

This expression can be used in Algorithm 9.2. Notice that $f(x + rp) - f(x)$ can be evaluated with $O(n)$ operations, so the backtracking procedure will be extremely cheap computationally.

Line Search using Cholesky Factorizations

One of the most efficient ways to check if a matrix is positive definite is to check if it is possible to perform a Cholesky factorization. This can be used to determine a feasible step length with respect to the semidefinite constraint. Start with a step length r, try to factorize $F(x + rp)$, if it fails, decrease r and try again.

An algorithm to find a feasible step length r_S with respect to the semidefinite constraint can thus be summarized as follows.

Algorithm 9.3 (Cholesky backtracking)

Given : $\beta_C \in (0, 1)$

begin
  $r_S := \mathbf{min}(1, \gamma_L \lambda_L)$
  $[R, failed] := \mathbf{Cholesky}(F(x + r_S p)$
  while $failed$
    $r_S := \beta_C r_S$
    $[R, failed] := \mathbf{Cholesky}(F(x + r_S p))$
  end
end

Notice that we initialized the step length to the largest step length with respect to the linear inequalities. The reason is that the final step length, when both the semidefinite constraint and the linear constraint are taken into account, cannot be larger than $\min(r_L, r_S)$.

If the Newton step p is a good search direction, both in direction and length, a feasible r_S is hopefully found in a few iterations. The Cholesky factorization is an $O(n^3)$ operation for dense matrices (Golub and van Loan, 1996), so the algorithm will only be efficient if the number of iterations stay low throughout all Newton iterations. Note that the Cholesky factorization does not have be completed when the matrix not is positive definite. Instead, the algorithm can stop immediately when a so-called negative pivot is found. Worst-case complexity is however still $O(n^3)$. For details on how a Cholesky factorization is computed, see (Golub and van Loan, 1996).

After finding a feasible step length r_S with respect to the semidefinite constraint, we set $r_{max} = r_S$. This means that we do not scale r_S with a damping parameter λ_S as we did before. The reason is that the Cholesky factor of $F(x + r_S p)$ will be used later in the Armijo backtracking algorithm. With $r_{max} = \lambda_S r_S$, we would have been forced to calculate a new factorization of $F(x+\lambda_S r_S p)$. By construction, $F(x + r_S p)$ is positive definite, so the step length will never take us all the way to the border of the feasible set.

Finally, $f(x+rp, \mu) - f(x, \mu)$ is needed the Armijo backtracking. The problem is to calculate the term $\log\det F(x + rp)$. In the previous section, we solved this efficiently by re-using the generalized eigenvalues. This cannot be done now.

Instead, we are forced to calculate the determinant explicitly. An efficient way to calculate the determinant of a matrix is to use triangular factorizations (Higham, 1996). For a Cholesky factorization $F(x+rp) = R^T R$, the determinant is given by

$$\begin{aligned}\log\det(F+rp) &= \log\det R^T R \\ &= \log(\det R)^2 \\ &= \log(\prod_{i=1}^{n} R_{ii})^2 \\ &= 2\sum_{i=1}^{n}\log R_{ii} \qquad (9.19)\end{aligned}$$

This can be used in the Armijo backtracking algorithm. Notice that $f(x,\mu)$ can be saved from the previous Newton iteration, and does not have to be re-computed.

Algorithm 9.4 (Cholesky based Armijo backtracking)

Given : $\alpha_A \in (0,1)$ and $\beta_A \in (0,1)$
begin
  $r := r_{max}$
  while $c^T(x+pr) - \mu 2\sum_{i=1}^{n}\log R_{ii} - \mu\sum\log(C(x+rp)-d) - f(x,\mu) > r\alpha_A b^T p$
    $r := r\beta_A$
    $R :=$ **Cholesky**$(F(x+rp))$
  end
end

If the sufficient decrease condition is fulfilled in the first iteration, the computational cost will be $O(n)$, since the already computed Cholesky factor R is used. If it fails, backtracking is needed and new Cholesky factorizations are calculated until the step is sufficiently small to yield a sufficient decrease.

The final Cholesky factor can be used to calculate the inverse of $F(x+rp)$ which is needed in the next iterate to calculate the gradient and Hessian of the logarithmic barrier. The standard way to calculate the inverse of a positive definite matrix F is to calculate the Cholesky factorization $F = R^T R$, solve the triangular system $RX = I$, and set $F^{-1} = XX^T$ (Higham, 1996). The performance gain compared to the command `inv` in MATLAB is however small. The main computational burden lies in the backsolve and matrix multiplication step, and not in the factorization of the sparse matrix $F(x)$.

9.4 Tailoring the Code for MPC

The semidefinite solver derived in the previous section can be improved upon, by exploiting structure in the problem that comes from the underlying problem formulation, minimax MPC.

Recall that the variables in the MPC problem (9.2) relate to the variables in the general problem (9.3) as

$$\begin{aligned} x_1 &= \tilde{t},\ x_2 = \begin{pmatrix} Y \\ U \end{pmatrix},\ x_3 = \tau \\ \Phi &= \begin{pmatrix} \mathcal{Q}^{-1} & 0 \\ 0 & \mathcal{R}^{-1} \end{pmatrix},\ \mathcal{H} = \begin{pmatrix} \mathcal{C}\mathcal{G} \\ 0 \end{pmatrix} \end{aligned}$$

9.4.1 Feasible Initial Iterate

Finding a feasible initial iterate can be done at a low computational cost. A Schur complement on the LMI in (9.3) transforms the constraints to

$$\begin{aligned} x_1 &\geq x_2^T(\Phi - \mathcal{H}\mathrm{diag}(x_3)^{-1}\mathcal{H}^T)^{-1}x_2 \\ \Phi - \mathcal{H}\mathrm{diag}(x_3)^{-1}\mathcal{H}^T &\succeq 0 \\ x_3 &\geq 0 \\ Cx - d &\geq 0 \\ Ex - f &= 0 \end{aligned}$$

In the minimax MPC problem, there are only linear constraints on x_2 (since x_1 and x_3 correspond to the auxiliary variables $\tilde{t}$ and τ.) Due to this separation, the following scheme can be used to find a feasible starting point

---

Algorithm 9.5 (Feasible initial iterate in minimax MPC)

Given : $\rho > 1$

Off-line

Find a vector $x_3 > 0$ such that $\Phi - \mathcal{H}\mathrm{diag}(x_3)^{-1}\mathcal{H}^T \succ 0$. This is always possible by choosing x_3 large enough since $\Phi \succ 0$.

On-line

1 Find x_2 strictly satisfying the linear constraints. The cost to do this is relatively small since we can use linear programming. Keep in mind that this step is necessary also in a nominal MPC formulation when a standard quadratic programming solver is used.

2 Given x_2 and x_3, let $x_1 > \rho(x_2^T(\Phi - \mathcal{H}\mathrm{diag}(x_3)^{-1}\mathcal{H}^T)^{-1}x_2)$.

9.4.2 Finding $E_\perp$

The matrix $E_\perp$ can essentially be obtained in two ways. The first and most obvious approach is to perform, e.g., a singular value decomposition and use this to calculate an orthonormal basis $E_\perp$. However, in some cases it is beneficial to exploit the structure in E.

The equality constraint $Ex = f$ is introduced to take care of the constraint $Y = \mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U)$, so E is given by

$$E = \begin{pmatrix} 0 & I & -\mathcal{CB} & 0 \end{pmatrix}$$

A natural choice of independent variables are $\tilde{t}$, U and τ and a basis for the null space of E is then given by

$$E_\perp = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \mathcal{CB} & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix} \tag{9.20}$$

The advantage with this choice is that $E_\perp$ tends to be sparser than the orthonormal basis, see Figure 9.1 for a typical example.

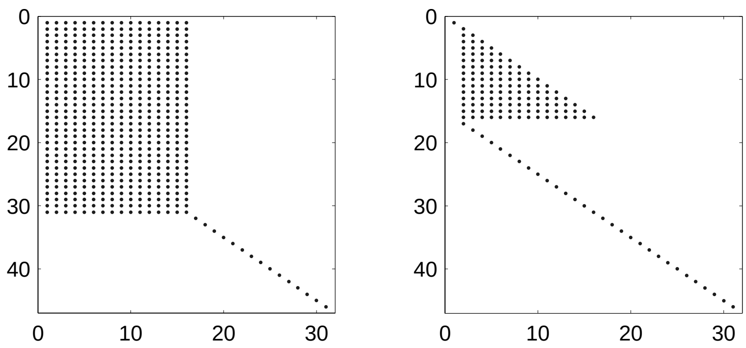

Figure 9.1: The figure shows the sparsity pattern for $E_\perp$ calculated with a singular value decomposition (left) and Equation (9.20) (right).

The block structure on $E_\perp$ defined with (9.20) allows us to calculate the projected Hessian $E_\perp^T H E_\perp$ and gradient $E_\perp^T b$ more efficiently.

$$\begin{aligned} E_\perp^T \nabla_{xx}^2 \phi_S E_\perp &= -(\star)^T \begin{pmatrix} \gamma^2 & 2\gamma\alpha & \tau \circ \tau \\ \star & 2(\gamma\Omega + \alpha^T\alpha) & 2\nu \operatorname{diag}(\tau) \\ \star & \star & \Gamma \circ \Gamma \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \begin{pmatrix} \mathcal{CB} \\ I \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ 0 & 0 & I \end{pmatrix} \\ &= -\begin{pmatrix} \gamma^2 & 2\gamma\alpha \begin{pmatrix} \mathcal{CB} \\ I \end{pmatrix} & \tau \circ \tau \\ \star & 2\begin{pmatrix} \mathcal{CB} \\ I \end{pmatrix}^T (\gamma\Omega + \alpha^T\alpha) \begin{pmatrix} \mathcal{CB} \\ I \end{pmatrix} & 2\begin{pmatrix} \mathcal{CB} \\ I \end{pmatrix}^T \nu \operatorname{diag}(\tau) \\ \star & \star & \Gamma \circ \Gamma \end{pmatrix} \end{aligned} \tag{9.21}$$

The projected gradient simplifies to

$$\nabla_x \phi_S = \begin{pmatrix} \gamma \\ 2\begin{pmatrix} \mathcal{CB} \\ I \end{pmatrix}^T \alpha^T \\ \text{diag}(\Gamma) \end{pmatrix} \tag{9.22}$$

The calculation of the projected Hessian and gradient originating from the linear constraints, $E_\perp^T C^T T^2 C E_\perp$ and $E_\perp^T C^T t$, can be simplified by calculating the matrix $E_\perp^T C^T$ off-line.

Note that $E_\perp$ does not depend on the current state x_k, so if the orthonormal basis is chosen, it can be calculated off-line.

9.4.3 Exploiting Sparseness

Needless to say, sparseness should be exploited whenever possible. For the MPC application, sparseness arises in $E_\perp$ (as we saw above) and in C, which typically is extremely sparse. It is of paramount importance to exploit this in the calculations.

Additional performance gains can be obtained by applying a row and column permutation on the sparse matrix $F(x)$ before the Cholesky factorizations are performed in Algorithms 9.3 and 9.4. A straightforward Cholesky factorization of $F(x)$ gives a Cholesky factor R with a large number of nonzero elements (so-called fill-in). A reordering reduces the number of nonzero elements, and thus the computational cost, substantially. Figure 9.2 shows a typical example of the impact of a reordering.

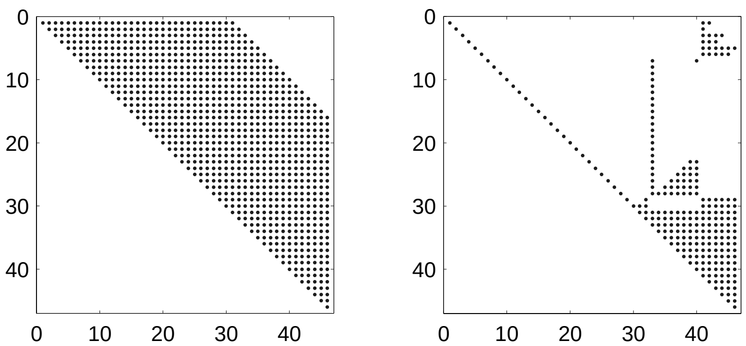

Figure 9.2: The figure shows the sparsity pattern for the Cholesky factor of $F(x)$ without reordering (left) and with symmetric minimum degree reordering (right).

The permutation can be calculated off-line since the sparsity pattern of $F(x)$ is the same all the time (assuming the decision variables to be nonzero). There are numerous ways to derive a permutation. The algorithm that worked best for our purpose turned out to be symmetric minimum degree permutation, called `symmmd` in MATLAB.

9.4.4 Scaling Barrier for Linear Inequalities

During the development of the code, it turned out that the algorithm sometimes suffered from numerical problems (yielding slow convergence in the Newton iterations) when many linear control and state constraints were included.

To solve these issues, the following simple but surprisingly efficient heuristic was employed. Instead of using the same barrier parameter on the semidefinite constraint and the linear constraints, a parameter $\gamma \in (0, 1)$ was introduced and the merit function was changed to

$$f(x, \mu) \quad = \quad c^T x - \mu\phi_S(x) - \mu\gamma\phi_L(x)$$

In other words, the influence of the linear constraints is decreased in the logarithmic barrier function. At a first glance, it might be believed that the same effect should be possible to obtain by multiplying C and d with some small (<< 1) constant. This is however not the case since $\log\det\gamma(Cx - d) = \log(\gamma) + \log\det(Cx - d)$, so a simple constraint scaling does not have any effect. The changes in the code for this heuristic are immediate, so the details are omitted for brevity.

9.4.5 Exploiting Diagonal Terms

The matrix $F(x)$ has free variables along the diagonal (x_1 and x_3). This can be exploited when we are looking for a suitable step length since we now that these diagonal elements have to be positive for $F(x)$ to be positive definite.

To this end, partition the step p in the same way as x, i.e. $p = \begin{pmatrix} p_1 & p_2^T & p_3^T \end{pmatrix}^T$ and define $\lambda_{D1} = \frac{p_1}{x_1}$ and $\lambda_{D3} = \frac{p_3}{x_3}$ (element-wise division). Maximal step length with respect to the diagonal elements is $r_D = \min(r_{D1}, r_{D3})$ where

$$r_{D1} \quad = \quad \begin{cases} \frac{1}{|\min(\lambda_{D1})|} & \lambda_{D1} < 0 \\ \infty & \min(\lambda_L) \geq 0 \end{cases} \tag{9.23a}$$

$$r_{D3} \quad = \quad \begin{cases} \frac{1}{|\min(\lambda_{D3})|} & \min(\lambda_{D1}) < 0 \\ \infty & \min(\lambda_L) \geq 0 \end{cases} \tag{9.23b}$$

The new bound on r can be used in the Algorithm 9.3 by changing the first step to $r_S = \min(1, \gamma_L r_D, \gamma_L r_L)$.

9.5 Application to Other Minimax Problems

The solver has been developed for the simplest minimax problem possible, but it can easily be extended to solve some of the extensions in Chapter 5.

Feedback Predictions

The changes when feedback predictions are applied are that Y now contains the matrix Ω, $Y = \mathcal{C}\Omega(\mathcal{A}x_{k|k} + \mathcal{B}V)$, and U is replaced with $\mathcal{L}\Omega(\mathcal{A}x_{k|k} + \mathcal{B}V) + V)$.

Define the variables Y, V and U, and the LMI in the semidefinte program (5.42) can be written in the standard form with

$$x_2 = \begin{pmatrix} Y \\ U \end{pmatrix}, \mathcal{H} = \begin{pmatrix} \mathcal{C}\Omega\mathcal{G} \\ \mathcal{L}\Omega\mathcal{G} \end{pmatrix}$$

The equality constraints are

$$\begin{aligned} Y &= \mathcal{C}\Omega(\mathcal{A}x_{k|k} + \mathcal{B}V) \\ U &= \mathcal{L}\Omega(\mathcal{A}x_{k|k} + \mathcal{B}V) + V \end{aligned}$$

Having both U and V is unnecessary (and does not fit the standard model). The variable V can be eliminated

$$V = (I + \mathcal{L}\Omega\mathcal{B})^{-1}(U - \mathcal{L}\mathcal{A}x_{k|k}) \tag{9.24}$$

and we obtain the original single linear equality constraint between Y and U

$$Y = \mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}U)$$

Hence, the only difference lies in the constant matrix $\mathcal{H}$.

Tracking

The only difference in the tracking formulation (5.45) is the constant terms Y_r and U_r. These do not affect any crucial parts of the solver since they only enter in the term F_0 of the matrix $F(x)$.

Norm-bounded Disturbance

The largest difference occurs if we want to use the solver for the disturbance model $w \in \mathbb{W}_2$. This model gives a matrix $\mathcal{T}$ with blocks of scaled identity matrices, $\mathcal{T} = \oplus_{i=1}^{N} \tau_i I^{r\times r}$. A simple way to account for this extension is to work with a matrix $\mathcal{T} = \oplus_{i=1}^{Nr} \tau_i$ and add additional linear equality constraints on τ (i.e. x_3) to impose the required structure.

Of course, if maximal performance is wanted, the only solution is to derive explicit expressions for the Hessian and gradient with the new structure on $\mathcal{T}$.

9.6 Computational Results

To evaluate the performance of the proposed solution strategy, a number of test problems will be solved. To be able to draw any conclusions from the results, we will also solve the problems using standard SDP solvers, both primal-dual solvers and a primal solver.

9.6.1 Test Strategy

The systems that will be used to create the minimax problems is discretized versions (zero-order hold, sampling time $1s$) of a second order system, parameterized in a damping parameter ξ.

$$y(t) = \frac{0.04}{p^2 + 0.4\xi p + 0.04}(u(t) + 0.25w(t)), \quad w \in \mathbb{W}_\infty \tag{9.25}$$

The minimax MPC problem (5.4) is solved using the semidefinite program (9.1). The performance weights in the MPC controller are $Q = R = 1$, and the system is constrained with $|u_k| \leq 1$ and $|y_k| \leq 5$.

The test problems are generated by choosing the damping parameter ξ randomly between 0 and 1. The state x_k is also generated randomly, with the restriction that the minimax problem is feasible. Infeasible problems are not of interest since infeasibility detection is taken care of by the linear programming solver used in Algorithm 9.5.

For each horizon length $N = 5, 10, \ldots, 70$ we create 10 test cases with different initial states x_k and damping parameters ξ, and calculate the average CPU-time[3] required to solve each minimax problem. The solver is implemented in MATLAB.

If not stated otherwise, the parameters in the solver are $\mu_0 = 1$, $\theta = 0.1$, $\eta = 0.9$, $\gamma = 0.1$, $\epsilon = 10^{-4}$, $\gamma_S = 0.9$, $\gamma_L = 0.9$, $\alpha_A = 0.3$, $\beta_A = 0.7$ and $\beta_C = 0.5$.

A feasible initial iterate was found using Algorithm 9.5. The algorithm used $\rho = 2$ and the initial value on τ (x_3) was $\tau_i = 10$ (turned out to be feasible for all generated systems). These values were chosen without any deeper thought and more problem-dependent choices are likely to give better performance.

9.6.2 Impact of Line Search Method

The first and most important experiment is to study which of the two line search methods that perform best; the method based on an eigenvalue decomposition, or the algorithm based on Cholesky factorizations.

Figure 9.3 reveals that the approximate method using Cholesky factorizations outperforms the exact method based on eigenvalues. The solutions obtained from the two methods are almost always identical up to 8-9 digits, so there is no loss from an accuracy point of view to use the approximate method based on Cholesky factorizations. Since the Cholesky based algorithm performed so well, it is used in the remaining experiments.

9.6.3 Impact of $E_\perp$

As a second experiment, we study the impact of the matrix $E_\perp$. The results in Figure 9.4 reveal that the performance gains are minor for small problems. Large-scale problems do however benefit from the structured and sparser null space matrix.

[3] MatLab 6.5, SUN ULTRA10 with a 300MHz processor and 128MB memory

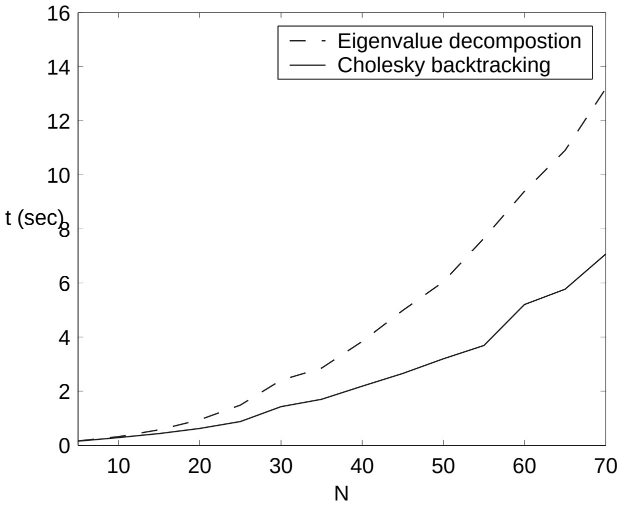

Figure 9.3: The line search procedure based on Cholesky factorizations is approximately 50 percent faster than the approach based on an eigenvalue decomposition.

9.6.4 Impact of γ

Without any theoretical motivation, a slightly nonstandard barrier was introduced to account for the linear constraints. The idea was to reduce the size of the barrier function for the linear constraints, compared to the barrier for semidefinite constraint, by multiplying the barrier for the linear constraints with a constant $\gamma < 1$. The same experiments as above were conducted, but now comparing the situation $\gamma = 1$ (standard) and $\gamma = 0.1$ (scaled barrier). The results are presented in Figure 9.5. The modified barrier function gave an average improvement of approximately 25 percent.

9.6.5 Profiling the Code

To find the most computationally expensive parts of the algorithm, the code was profiled in MATLAB.

A test case with $N = 50$ was solved and profiled. Since we earlier found that best performance was obtained with the Cholesky based line search, the structured null space matrix $E_{\perp}$ and the scaled barrier for the linear inequalities, this setup was used also here.

The relative CPU-time spent during the iterations is illustrated in the pie chart in Figure 9.6. Approximately 40 percent of the CPU-time is spent on calculating the inverse of $F(x)$. Hence, this has to be considered the main bottleneck in the code. As expected, assembling the Hessian and gradient (H and b) is relatively cheap. Note that this part includes the projection (9.21) on the equality constraints. The

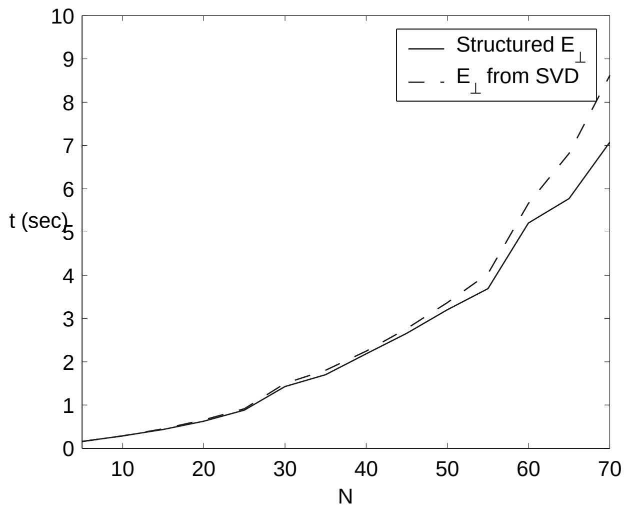

Figure 9.4: The figure shows the performance gains of a structures null space matrix $E_\perp$ derived from the problem formulation instead of using an SVD decomposition of E to calculate $E_\perp$. The gain is minor for short horizons, but increase with increasing problem size.

large portion of unspecified CPU-time in the field *Others* can mainly be attributed to code overhead in MATLAB, such as function calls.

9.6.6 Comparison with DSDP, SeDuMi and SDPT3

Finally, let us compare the computational results with those obtained with three of the most efficient general-purpose SDP solvers available, DSDP (Benson and Ye, 2001), SeDuMi (Sturm, 1999) and SDPT3 (Toh et al., 1999).

The tolerance in the solvers was set to 10^{-4}. Other settings were left unaltered in all solvers. No uniform improvement could be obtained, neither in performance or in robustness, by changing the default parameter choices. Furthermore, DSDP, SDPT3 and SeDuMi can use warm-starts. However, this was not used since initial experiments, using the initial solution from Algorithm 9.5, showed no uniform performance gains.

The same problems as above were solved and the computational results are reported in Figure 9.7. The poor performance of SeDuMi (version 1.05R4) and SDPT3 (version 3.0) were expected. SeDuMi and SDPT3 are based on primal-dual algorithms, and consequently have to introduce a dual variable for the semidefinite constraint. This variable is dense, hence dense algebra has to be used. Another reason for the poor performance is the Hessian assembly. SeDuMi and SDPT3 do not exploit the low rank properties of the involved matrices, but only exploit sparseness.

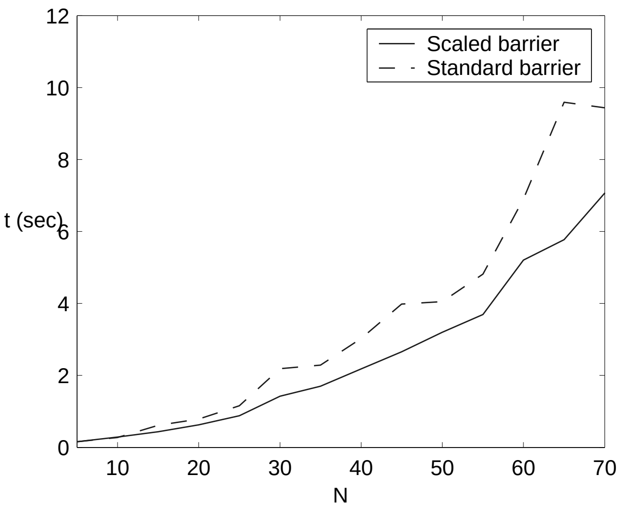

Figure 9.5: The CPU-time was uniformly lower when a scaled barrier was used on the linear constraints.

The poor performance of DSDP (version 4.5) is a surprise. DSDP is a primal algorithm (in our notation) and does not have to introduce any dense dual variable. Furthermore, the main idea in DSDP is to exploit low rank structure of the matrices F_i. Despite these promising features, DSDP was not competitive for the problems that were solved. The reason for the poor performance seems to lie in the linear constraints. Without linear constraints, the performance was excellent and DSDP was most often as fast or even faster than the proposed SUMT solver. However, when linear constraints were added, the performance often deteriorated. Recall that the linear constraints caused some trouble also in our SUMT solver, but not by far as drastic, and the problem was essentially resolved by the scaled barrier.

Finally, it should be mentioned that DSDP, SEDUMI and SDPT3 are (at least partially) implemented in C, whereas the proposed solver was implemented entirely in MATLAB.

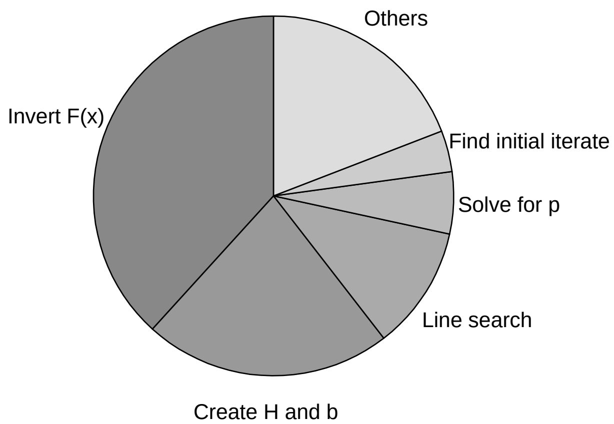

Figure 9.6: Relative CPU-time spent on different steps in the solver. The main bottleneck in the code is the inversion of the matrix $F(x)$.

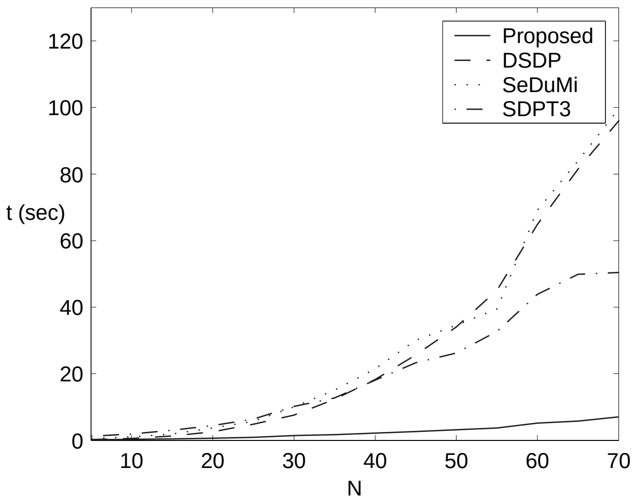

Figure 9.7: Comparison between the proposed solver, the primal solver DSDP and the primal-dual solvers SeDuMi and SDPT3.

Appendix

9.A Hessian and Gradient of $\log\det F(x)$

To obtain a simple notation in the derivation of the gradient and Hessian of the logarithmic barrier function $\log\det F(x)$, we introduce a matrix S with

$$S = F(x)^{-1} = \begin{pmatrix} x_1 & x_2^T & 0 \\ x_2 & \Phi & \mathcal{H} \\ 0 & \mathcal{H}^T & \mathrm{diag}(x_3) \end{pmatrix}^{-1} = \begin{pmatrix} \gamma & \alpha & \tau \\ \alpha^T & \Omega & \nu \\ \tau^T & \nu^T & \Gamma \end{pmatrix}$$

Furthermore, $x = \begin{pmatrix} x_1 & x_2^T & x_3^T \end{pmatrix}^T$ where $x_1 \in \mathbb{R}$, $x_2 \in \mathbb{R}^{n_{x_2}}$ and $x_3 \in \mathbb{R}^{n_{x_3}}$. Since x_1 and x_3 are located on the diagonal, and x_2 is located in the first row and column of $F(x)$, we have that the basis matrices of $F(x)$ is given by

$$F_0 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \Phi & \mathcal{H} \\ 0 & \mathcal{H}^T & 0 \end{pmatrix} \tag{9.A.26a}$$

$$F_1 = e_1 e_1^T \tag{9.A.26b}$$

$$F_i = e_1 e_i^T + e_i e_1^T \quad (2 \le i \le 1 + n_{x_2}) \tag{9.A.26c}$$

$$F_i = e_i e_i^T \quad (2 + n_{x_2} \le i \le 1 + n_{x_2} + n_{x_3}) \tag{9.A.26d}$$

Evaluating (9.12) gives us the gradient and the Hessian of $\phi_S(x) = \log\det F(x)$.

9.A.1 Gradient

Gradient w.r.t x_1

$$\mathrm{Tr} S F_1 = \mathrm{Tr} S e_1 e_1^T = e_1^T S e_1$$

Gradient w.r.t x_2

$$\mathrm{Tr} S F_i = \mathrm{Tr} S (e_1 e_i^T + e_i e_1^T) = 2 e_1^T S e_i, \quad (2 \le i \le 1 + n_{x_2})$$

Gradient w.r.t x_3

$$\text{Tr}SF_j = \text{Tr}S(e_j e_j^T) = e_j^T S e_j, \quad (2 + n_{x_2} \le j \le 1 + n_{x_2} + n_{x_3})$$

Compiling the Gradient

Careful inspection of the indices and relating to the partition of S reveals that the gradient can be written compactly as

$$\nabla\phi_S(x) = \begin{pmatrix} -\gamma \\ -2\alpha^T \\ -\text{diag}(\Gamma) \end{pmatrix}$$

9.A.2 Hessian

Hessian w.r.t x_1

$$-\text{Tr}SF_1SF_1 = -Se_1e_1^TSe_1e_1^T = -(e_1^TSe_1)^2$$

Hessian w.r.t x_1 and x_2

$$\begin{aligned} -\text{Tr}SF_1SF_i &= -\text{Tr}Se_1e_1^TS(e_1e_i^T + e_ie_1^T) \quad \big(2 \le i \le 1 + n_{x_2}\big) \\ &= -2(e_i^TSe_1)(e_1^TSe_1) \end{aligned}$$

Hessian w.r.t x_1 and x_3

$$\begin{aligned} -\text{Tr}SF_1SF_j &= -\text{Tr}Se_1e_1^TSe_je_j^T \quad \big(2 + n_{x_2} \le j \le 1 + n_{x_2} + n_{x_3}\big) \\ &= -(e_1^TSe_j)^2 \end{aligned}$$

Hessian w.r.t x_2

$$\begin{aligned} -\text{Tr}SF_iSF_j &= -\text{Tr}S(e_1e_i^T + e_ie_1^T)S(e_1e_j^T + e_je_1^T) \quad \begin{pmatrix} 2 \le i \le 1 + n_{x_2} \\ 2 \le j \le 1 + n_{x_2} \end{pmatrix} \\ &= -2((e_1^TSe_1)(e_i^TSe_j) + (e_1^TSe_i)(e_1^TSe_j)) \end{aligned}$$

Hessian w.r.t x_2 and x_3

$$\begin{aligned} -\text{Tr}SF_iSF_j &= -\text{Tr}S(e_1e_i^T + e_ie_1^T)S(e_je_j^T), \quad \begin{pmatrix} 2 \le i \le 1 + n_{x_2} \\ 2 + n_{x_2} \le j \le 1 + n_{x_2} + n_{x_3} \end{pmatrix} \\ &= -2(e_i^TSe_1)(e_jSe_i) \end{aligned}$$

Hessian w.r.t x_3

$$\begin{aligned} -\mathrm{Tr} S F_i S F_j &= -\mathrm{Tr} S(e_i e_i^T) S(e_j e_j^T), \quad \begin{pmatrix} 2 + n_{x_2} \leq i \leq 1 + n_{x_2} + n_{x_3} \\ 2 + n_{x_2} \leq j \leq 1 + n_{x_2} + n_{x_3} \end{pmatrix} \\ &= -(e_i^T S e_j)^2 \end{aligned}$$

Compiling the Hessian

Straightforward but tedious compilation yields the following expression for the Hessian ($\circ$ denotes the element-wise Hadamard product)

$$\nabla^2 \phi_S(x) = - \begin{pmatrix} \gamma^2 & 2\gamma\alpha & \tau \circ \tau \\ 2\gamma\alpha^T & 2(\gamma\Omega + \alpha^T \alpha) & 2\nu \ \mathrm{diag}(\tau) \\ (\tau \circ \tau)^T & 2\mathrm{diag}(\tau)\nu^T & \Gamma \circ \Gamma \end{pmatrix}$$

10

MPC for systems with uncertain gain

In this chapter, we do not use the disturbance model that has been addressed in previous chapters. Instead of bounded external disturbances, we now focus on systems with uncertainties in the system model. To be more specific, this chapter assumes uncertainties in the gain, i.e., in the B matrix.

The main goal of this chapter is the same as in Chapter 5. It is shown how a minimax MPC problem with quadratic performance measure can be addressed using robust semidefinite programming.

The model and the main result in this chapter are in principle special cases of the more general theory that will be presented in the next chapter. However, several simplifications can be done due to the special structure in the model used in this chapter. This will enable much more efficient implementations.

10.1 Uncertainty Model

A problem setup that has been used in many approaches to robust MPC is models with an uncertain B matrix[1]. Introduce a time-varying uncertainty Δ_k and write

[1]An alternative formulation is to use an uncertain C matrix. The differences are minor.

this as

$$x_{k+1} = Ax_k + B(\Delta_k)u_k \tag{10.1a}$$
$$y_k = Cx_k \tag{10.1b}$$

The model has most often been polytopic, $B \in \mathbf{Co}\{B^{(1)}, \ldots, B^{(q)}\}$ (Campo and Morari, 1987; Zheng and Morari, 1993; Oliviera et al., 2000). In this work however, we turn our attention to a norm-bounded uncertainty model (Boyd et al., 1994).

$$B(\Delta_k) = B_0 + B_1\Delta_k B_2, \quad \Delta_k \in \boldsymbol{\Delta} \tag{10.2a}$$
$$\boldsymbol{\Delta} = \{\Delta \in \mathbb{R}^{n_\Delta \times m_\Delta} : \left\|\Delta\right\| \leq 1\} \tag{10.2b}$$

The uncertainty is thus the norm-bounded matrix Δ_k. More details and discussion about this model can be found in Section 10.4.

10.2 Minimax MPC

The goal in this chapter is to solve the minimax MPC problem with a quadratic performance measure and robust satisfaction of constraint.

$$\boxed{\begin{array}{lrcl} \min\limits_u \max\limits_\Delta & \multicolumn{3}{l}{\sum_{j=0}^{N-1} y_{k+j|k}^T Q y_{k+j|k} + u_{k+j|k}^T R u_{k+j|k}} \\ \text{subject to} & u_{k+j|k} & \in & \mathbb{U} \quad \forall \Delta \in \boldsymbol{\Delta} \\ & x_{k+j|k} & \in & \mathbb{X} \quad \forall \Delta \in \boldsymbol{\Delta} \\ & \Delta_{k+j|k} & \in & \boldsymbol{\Delta} \end{array}} \tag{10.3}$$

Earlier work related to (10.3) differ from our approach in that they typically assume linear performance measures and polytopic models. The reason is that the maximization in the optimization problem then can be done analytically, and the result is a linear program, see , e.g., (Campo and Morari, 1987; Zheng and Morari, 1993; Oliviera et al., 2000). Of course, as we saw in Chapter 4, there are approaches that can cope with the above minimax problem, but they do not exploit the inherent structure that arise from the gain uncertainty assumption. Instead, they concentrate on general uncertain systems.

10.2.1 Semidefinite Relaxation of Minimax MPC

We proceed as usual and introduce a vectorized notation. To do this, define

$$\mathcal{B}_0 = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 \\ B_0 & 0 & 0 & \ldots & 0 \\ AB_0 & B_0 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A^{N-2}B_0 & \ldots & AB_0 & B_0 & 0 \end{pmatrix}, \; \mathcal{B}_1 = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 \\ B_1 & 0 & 0 & \ldots & 0 \\ AB_1 & B_1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A^{N-2}B_1 & \ldots & AB_1 & B_1 & 0 \end{pmatrix}$$

$$\mathcal{B}_2 = \oplus_{j=1}^{N} B_2$$

$$\Delta^N = \oplus_{j=0}^{N-1} \Delta_{k+j|k} \in \boldsymbol{\Delta}^N = \boldsymbol{\Delta} \times \ldots \times \boldsymbol{\Delta}$$

These matrices and the block diagonal structure of Δ^N enables the following compact notation (with additional variables defined as in Chapter 5).

$$\begin{aligned} Y &= \mathcal{C}X && (10.4a)\\ X &= \mathcal{A}x_{k|k} + (\mathcal{B}_0 + \mathcal{B}_1\Delta^N\mathcal{B}_2)U && (10.4b)\end{aligned}$$

Bounding the Performance Measure

As a first step, introduce the epigraph form of the optimization problem and define the problem that we wish to address with a semidefinite relaxation.

$$\boxed{\begin{array}{lrcll} \min\limits_{U,t} & t & & & \\ \text{subject to} & Y^T\mathcal{Q}Y + U^T\mathcal{R}U & \leq & t & \forall\Delta^N \in \boldsymbol{\Delta}^N \\ & U & \in & \mathbb{U}^N & \forall\Delta^N \in \boldsymbol{\Delta}^N \\ & X & \in & \mathbb{X}^N & \forall\Delta^N \in \boldsymbol{\Delta}^N \end{array}} \qquad (10.5)$$

Applying as Schur complement on the first constraint takes us to the matrix space.

$$\begin{pmatrix} t & Y^T & U^T \\ Y & \mathcal{Q}^{-1} & 0 \\ U & 0 & \mathcal{R}^{-1}\end{pmatrix} \succeq 0 \quad \forall\Delta^N \in \boldsymbol{\Delta}^N \qquad (10.6)$$

Insert the definition $Y = \mathcal{C}(\mathcal{A}x_{k|k} + (\mathcal{B}_0 + \mathcal{B}_1\Delta^N\mathcal{B}_2)U)$, and separate certain and uncertain terms in the matrix inequality

$$\begin{pmatrix} t & (\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}_0U))^T & U^T \\ \star & \mathcal{Q}^{-1} & 0 \\ \star & 0 & \mathcal{R}^{-1}\end{pmatrix} + \begin{pmatrix} 0 \\ \mathcal{C}\mathcal{B}_1 \\ 0 \end{pmatrix} \Delta^N \begin{pmatrix} \mathcal{B}_2U & 0 & 0 \end{pmatrix} + (\star) \succeq 0 \; \forall\Delta^N \in \boldsymbol{\Delta}^N$$

Direct application of Theorem 3.4 gives a multiplier $\tau \in \mathbb{R}_+^N$ and the matrices $\mathcal{T} = \oplus_1^N \tau_j I^{m_\Delta \times m_\Delta}$ and $\mathcal{S} = \oplus_1^N \tau_j I^{n_\Delta \times n_\Delta}$, and a sufficient condition for (10.6) to hold.

$$\begin{pmatrix} t & (\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}_0U))^T & U^T & U^T\mathcal{B}_2^T \\ \star & \mathcal{Q}^{-1} - \mathcal{C}\mathcal{B}_1\mathcal{S}\mathcal{B}_1^T\mathcal{C}^T & 0 & 0 \\ \star & 0 & \mathcal{R}^{-1} & 0 \\ \star & 0 & 0 & \mathcal{T}\end{pmatrix} \succeq 0 \qquad (10.7)$$

The next step is the state constraints.

Robust Constraint Satisfaction

Robust satisfaction of state constraints is also easy to handle, although the matter is a bit more intricate than robust constraint satisfaction for the models with external disturbances in Chapter 5

Writing the linear constraints with U and Δ^N inserted gives the linear inequalities that should hold for all possible uncertainty realizations.

$$\mathcal{E}_x(\mathcal{A}x_{k|k} + (\mathcal{B}_0 + \mathcal{B}_1\Delta^N\mathcal{B}_2)U) \leq \mathcal{F}_x \quad \forall\Delta^N \in \boldsymbol{\Delta}^N \tag{10.8}$$

Partition $\mathcal{B}_1$ into N block rows

$$\mathcal{B}_1 = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ B_1 & 0 & 0 & \dots & 0 \\ AB_1 & B_1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A^{N-2}B_1 & \dots & AB_1 & B_1 & 0 \end{pmatrix} = \begin{pmatrix} H_0 & H_1 & \dots H_{N-1} \end{pmatrix} \tag{10.9}$$

Separate the certain and the uncertain terms of the constraint, and use the matrix H and the structure of $\mathcal{B}_2$ and Δ^N to expand the vectorized uncertainty

$$\mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}_0 U) + \sum_{j=0}^{N-1} \mathcal{E}_x H_j \Delta_{k+j|k} B_2 u_{k+j|k} \leq \mathcal{F}_x \tag{10.10}$$

The complicating thing here is that the expression is bilinear in the uncertainty and the decision variable U.

Let us for the moment look at a single row in the constraint (with $(\cdot)_i$ denoting the ith row).

$$(\mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}_0 U))_i + \sum_{j=0}^{N-1} (\mathcal{E}_x H_j)_i \Delta_{k+j|k} B_2 u_{k+j|k} \leq (\mathcal{F}_x)_i \tag{10.11}$$

The problem now is to maximize the sum with respect to Δ^N. The following theorem enables this.

Theorem 10.1
Let the vectors x and y and the matrix Δ be of compatible sizes. It then holds that

$$\max_{||\Delta|| \leq 1} x^T \Delta y = ||x|| ||y|| \tag{10.12}$$

Proof Follows from Schwarz inequality $(a^T b)^2 \leq ||a||^2||b||^2$, with equality if a and b are parallel. Hence, equality when Δ is chosen so that x and Δy are parallel. □

Maximization of (10.11) can now be done by letting $(\mathcal{E}_x H_j)_i$ play the role of x^T and $B_2 u_{k+j|k}$ the role of y. Application of the theorem requires the Euclidean norm of the vectors $B_2 u_{k+j|k}$. This is solved by introducing a new decision variable $\kappa \in \mathbb{R}^N$ and N second order cone constraints.

$$||B_2 u_{k+j|k}|| \leq \kappa_{j+1} \tag{10.13}$$

Since $||B_2u_{k+j|k}||$ now is smaller than κ_{j+1}, the ith state constraint (10.11) is guaranteed to hold if

$$(\mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}_0U))_i + \sum_{j=0}^{N-1} ||(\mathcal{E}_xH_j)_i||\kappa_{j+1} \leq (\mathcal{F}_x)_i \tag{10.14}$$

Define a matrix $\Omega_{i(j+1)} = ||(\mathcal{E}_xH_j)_i||$, and all robustified state constraints are efficiently written as

$$\mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}_0U) + \Omega\kappa \leq \mathcal{F}_x \tag{10.15}$$

Hence, robust satisfaction of the linear state constraints requires the same amount of linear constraints as original nominal constraints, but need N additional second order cone constraints and N new variables. The last conclusion is important; no matter how many nominal linear constraints there are, the number of additional variables and second order cone constraints remain constant. Notice also that the second order cone constraints are of low dimension. This is important from a computational efficiency point of view.

Final Problem

The results obtained at this point enable us to state the main result of this chapter. The semidefinite relaxation of the minimax MPC problem (10.5) is given by the following semidefinite program.

$$\begin{array}{lrcl}
\min\limits_{U,t,\tau,\kappa} & t & & \\
\text{subject to} & \begin{pmatrix} t & (\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}_0U))^T & U^T & U^T\mathcal{B}_2^T \\ \star & \mathcal{Q}^{-1} - \mathcal{C}\mathcal{B}_1\mathcal{S}\mathcal{B}_1^T\mathcal{C}^T & 0 & 0 \\ \star & 0 & \mathcal{R}^{-1} & 0 \\ \star & 0 & 0 & \mathcal{T} \end{pmatrix} & \succeq & 0 \\
 & \mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}_0U) + \Omega\kappa & \leq & \mathcal{F}_x \\
 & ||B_2u_{k+j|k}|| & \leq & \kappa_{j+1} \\
 & \mathcal{E}_uU & \leq & \mathcal{F}_u
\end{array} \tag{10.16}$$

Simplification of the Semidefinite Program

The optimization problem just derived can be simplified, from a computational efficiency point of view[2].

Assume for the moment that $\mathcal{Q}^{-1} - \mathcal{C}\mathcal{B}_1\mathcal{S}\mathcal{B}_1^T\mathcal{C}^T$ is positive definite. A Schur complement on the LMI in (10.16) gives the constraint (to save space, we locally

[2]The statements that follow are valid if we assume that second order cone constraints are to be preferred against the corresponding formulation using LMIs. This is the case for the solvers used in this thesis to solve the mixed semidefinite and second order cone programs (Toh et al., 1999; Sturm, 1999)

use the incorrect notation $Y = \mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}_0 U))$

$$Y^T(\mathcal{Q}^{-1} - \mathcal{C}\mathcal{B}_1\mathcal{S}\mathcal{B}_1^T\mathcal{C}^T)^{-1}Y + U^T\mathcal{R}U + U^T\mathcal{B}_2^T\mathcal{T}^{-1}\mathcal{B}_2U \leq t \tag{10.17}$$

This can, with the new scalar decision variables t_x, t_u and t_δ, be written as

$$t_x + t_u + t_\delta \leq t \tag{10.18a}$$
$$Y^T(\mathcal{Q}^{-1} - \mathcal{C}\mathcal{B}_1\mathcal{S}\mathcal{B}_1^T\mathcal{C}^T)^{-1}Y \leq t_x \tag{10.18b}$$
$$U^T\mathcal{R}U \leq t_u \tag{10.18c}$$
$$U^T\mathcal{B}_2^T\mathcal{T}^{-1}\mathcal{B}_2U \leq t_\delta \tag{10.18d}$$

The most efficient way to implement the constraint (10.18b) is probably in the original LMI format.[3]

$$\begin{pmatrix} t_x & (\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}_0U))^T \\ \star & \mathcal{Q}^{-1} - \mathcal{C}\mathcal{B}_1\mathcal{S}\mathcal{B}_1^T\mathcal{C}^T \end{pmatrix} \succeq 0 \tag{10.19}$$

The control cost constraint (10.18c) can be implemented as a second order cone constraint

$$\left\| \begin{matrix} 2\mathcal{R}^{1/2}U \\ 1 - t_u \end{matrix} \right\| \leq 1 + t_u \tag{10.20a}$$

The third constraint (10.18d) requires a bit more thought. Of course, one simple solution is to apply a Schur complement and go back to an LMI.

$$\begin{pmatrix} t_\delta & U^T\mathcal{B}_2^T \\ \mathcal{B}_2U & \mathcal{T} \end{pmatrix} \succeq 0 \tag{10.21}$$

However, it can be implemented more efficiently. To begin with, note that the block diagonal structures of $\mathcal{T}$ and $\mathcal{B}_2$ allow the following expansion of the left-hand side of (10.18d).

$$U^T\mathcal{B}_2^T\mathcal{T}^{-1}\mathcal{B}_2U = \sum_{j=0}^{N-1} u_{k+j|k}^T B_2^T(\tau_{j+1}I)^{-1}B_2u_{k+j|k} \tag{10.22}$$

Add an upper bound on the terms in the sum (redefine the scalar t_δ to a vector of length N instead, $t_\delta \in \mathbb{R}^N$)

$$u_{k+j|k}^T \frac{B_2^TB_2}{\tau_{j+1}} u_{k+j|k} \leq t_{\delta_{j+1}} \tag{10.23}$$

[3] The constraint is referred to as a matrix fractional constraint in the literature (Nesterov and Nemirovskii, 1993; Lobo et al., 1998). These constraints can in some cases be rewritten to second order cone constraints. The problem we have does however not seem to fulfill the requirements for this transformation to be applicable.

Multiply with the positive scalar τ_j to obtain a so called rotated Lorentz cone constraint (sometimes referred to as a hyperbolic constraint (Lobo et al., 1998))

$$\left\|B_2 u_{k+j|k}\right\|^2 \leq t_{\delta_{j+1}} \tau_{j+1} \tag{10.24}$$

This can easily be shown to be equivalent to the following second order cone constraint

$$\left\|\begin{matrix} 2B_2 u_{k+j|j} \\ t_{\delta_{j+1}} - \tau_{j+1} \end{matrix}\right\| \leq t_{\delta_{j+1}} + \tau_{j+1} \tag{10.25}$$

If we once again summarize our results, we find that a more efficient formulation of the problem (10.16) is given by

$$\begin{array}{lrcl} \min\limits_{U, t_x, t_u, t_\delta, \tau, \kappa} & t_x + t_u + \sum_{j=1}^{N} t_{\delta_j} & & \\ \text{subject to} & \begin{pmatrix} t_x & (\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}_0 U))^T \\ \star & \mathcal{Q}^{-1} - \mathcal{C}\mathcal{B}_1\mathcal{S}\mathcal{B}_1^T\mathcal{C}^T \end{pmatrix} & \succeq & 0 \\ & \left\|\begin{matrix} 2\mathcal{R}^{1/2}U \\ 1 - t_u \end{matrix}\right\| & \leq & 1 + t_u \\ & \left\|\begin{matrix} 2B_2 u_{k+j|j} \\ t_{\delta_{j+1}} - \tau_{j+1} \end{matrix}\right\| & \leq & t_{\delta_{j+1}} + \tau_{j+1} \\ & \mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}_0 U) + \Omega\kappa & \leq & \mathcal{F}_x \\ & ||B_2 u_{k+j|k}|| & \leq & \kappa_j \\ & \mathcal{E}_u U & \leq & \mathcal{F}_u \end{array} \tag{10.26}$$

Connections to Nominal MPC

One of the nice features with the proposed solution is that the obtained optimization problem can be intuitively interpreted to some extent.

Recall that our optimization program minimized the following expression, with Y once again denoting the nominal predictions $\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}_0 U)$,

$$Y^T(\mathcal{Q}^{-1} - \mathcal{C}\mathcal{B}_1\mathcal{S}\mathcal{B}_1^T\mathcal{C}^T)^{-1}Y + U^T\mathcal{R}U + U^T\mathcal{B}_2^T\mathcal{T}^{-1}\mathcal{B}_2 U \tag{10.27}$$

This means that for *fixed* $\mathcal{T}$, we are solving a nominal problem with new weights

$$\mathcal{Q} \leftarrow (\mathcal{Q}^{-1} - \mathcal{C}\mathcal{B}_1\mathcal{S}\mathcal{B}_1^T\mathcal{C}^T)^{-1} \tag{10.28a}$$

$$\mathcal{R} \leftarrow \mathcal{R} + \mathcal{B}_2^T\mathcal{T}^{-1}\mathcal{B}_2 \tag{10.28b}$$

Hence, the difference is, to begin with, the additional weight $\mathcal{B}_2^T\mathcal{T}^{-1}\mathcal{B}_2$ on the control sequence. By recalling the block diagonal definition of $\mathcal{B}_2$ and $\mathcal{T}$, it follows that there is an additional weight on $u_{k+j|k}$ proportional to τ_j^{-1}.

The modified state weight is a bit less intuitive to analyze. However, we begin by applying the Sherman-Morrison-Woodbury formula to obtain

$$(\mathcal{Q}^{-1} - \mathcal{C}\mathcal{B}_1\mathcal{S}\mathcal{B}_1^T\mathcal{C}^T)^{-1} = \mathcal{Q} + \mathcal{Q}\mathcal{C}\mathcal{B}_1(\mathcal{S}^{-1} - \mathcal{B}_1^T\mathcal{C}^T Q\mathcal{C}\mathcal{B})^{-1}\mathcal{B}_1^T\mathcal{C}^T\mathcal{Q} \tag{10.29}$$

For small τ (large $\mathcal{S}^{-1}$), this is approximately equal to $\mathcal{Q} + \mathcal{Q}\mathcal{C}\mathcal{B}_1\mathcal{S}\mathcal{B}_1^T\mathcal{C}^T\mathcal{Q}$. Of course, these calculations only hold if the optimal τ indeed is small. However, the purpose of these calculations is only to give a flavor of what happens. The control weight is inversely proportional to τ, while the output weight is proportional to τ (this can be seen immediately in (10.28a), but the first order expression is more intuitive).

Important to realize is that both weights are increased, i.e., the regularization of the MPC solution is not done merely by, e.g., increasing the control weight and effectively turning of the controller. We can also see that the adjustment of the weights depend on both the system model and the original weights. Moreover, the new output weight $\mathcal{Q}$ will not be block diagonal anymore, i.e., cross terms are introduced in the output weight. The control weight $\mathcal{R}$ will however remain block diagonal.

10.3 Extensions

For the proposed framework to be interesting, it is important that standard extensions to nominal MPC can be applied also to our minimax controller. Indeed, this is the case, as we will show with a couple of examples.

First, it should be mentioned that feedback predictions are impossible in the proposed minimax controller. The reason is that feedback predictions destroy the main feature exploited in this chapter; linearity between the uncertainty and predicted states. Linearity enabled the compact expression for the predictions (10.4) and the analytic maximization of the uncertainty in the state constraints (10.11). This drawback is not a particular problem for our solution, but holds for all approaches that exploit the linearity, e.g., (Campo and Morari, 1987; Zheng and Morari, 1993; Oliviera et al., 2000).

10.3.1 Output Gain Uncertainty

Changing the location of the uncertainty to the output gain can be done without much effort. Consider a system of the form

$$x_{k+1} = Ax_k + Bu_k \tag{10.30a}$$

$$y_k = (C_0 + C_1\Delta_k C_2)x_k \quad ||\Delta_k|| \leq 1 \tag{10.30b}$$

Defining matrices $\mathcal{C}_0 = \oplus_{j=1}^N C_0$, $\mathcal{C}_1 = \oplus_{j=1}^N C_1$ and $\mathcal{C}_2 = \oplus_{j=1}^N C_2$ gives the prediction $Y = (\mathcal{C}_0 + \mathcal{C}_1\Delta^N\mathcal{C}_2)(\mathcal{A}x_{k|k} + \mathcal{B}U)$. All results in the chapter can then be recovered by merely replacing $\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}_0U)$ with $\mathcal{C}_0(\mathcal{A}x_{k|k} + \mathcal{B}U)$, $\mathcal{B}_2U$ with $\mathcal{C}_2(\mathcal{A}x_{k|k} + \mathcal{B}U)$ and $\mathcal{B}_1$ with $\mathcal{C}_1$. Some minor changes have to be done to take care of output constraints, but the details are omitted for brevity.

10.3.2 Disturbances and Estimation Errors

The results in this chapter can be extended to incorporate the material in Chapter 5 and 6. Adding external disturbances to the model yields

$$Y = \mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}_0 U + \mathcal{B}_1 \Delta^N \mathcal{B}_2 + \mathcal{G}W) \tag{10.31}$$

Combining (5.24), (5.29), (10.13) and (10.16) leads to the following semidefinite program to solve the semidefinite relaxation of a minimax problem over the uncertainties Δ^N and W (with $\mathcal{T}_W$, $\mathcal{S}_\Delta$ and $\mathcal{T}_\Delta$ being the diagonal matrices defined by the vectors τ_W and τ_Δ, obtained during the relaxation procedure)

$$\begin{array}{lll}
\displaystyle\min_{U,t,\tau_W,\tau_\Delta,\kappa} & t & \\
\text{subject to} & \begin{pmatrix} t - \mathbf{1}^T\tau_W & (\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}_0 U))^T & U^T & U^T\mathcal{B}_2^T & 0 \\ \star & \mathcal{Q}^{-1} - \mathcal{C}\mathcal{B}_1\mathcal{S}_\Delta\mathcal{B}_1^T\mathcal{C}^T & 0 & 0 & (\mathcal{C}\mathcal{G})^T \\ \star & 0 & \mathcal{R}^{-1} & 0 & 0 \\ \star & 0 & 0 & \mathcal{T}_\Delta & 0 \\ 0 & \star & 0 & 0 & \mathcal{T}_W \end{pmatrix} \succeq & 0 \\
& \mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}_0 U) + \Omega\kappa + \gamma \leq & \mathcal{F}_x \\
& ||B_2 u_{k+j|k}|| \leq & \kappa_{j+1} \\
& \mathcal{E}_u U \leq & \mathcal{F}_u
\end{array}$$

An ellipsoidal state estimator for this class of systems can be designed using the theory in, e.g., (El Ghaoui and Calafiore, 1999), hence allowing us to extend the results to incorporate a bounded state estimation error.

10.3.3 Stability Constraints

The minimax scheme in this chapter suffers from the same problems as the controller in Chapter 5. Due to the sufficient but not necessary nature of the semidefinite relaxation, guaranteeing stability turns out to be much harder than in the nominal case, or in an approach based on the exact minimax solution. See Section 5.3.3 for a more detailed discussion on why the relaxation complicates matters.

Instead of developing a stability theory based on Theorem 4.2, we revert to a weaker result using explicit contraction constraints. As in Chapter 5, we give a hint on how stability can be obtained, but the results are overly conservative and only included for completeness. No state constraints are allowed, and to begin with, an asymptotically stable open-loop system is assumed.

Theorem 10.2 (Guaranteed stability, open-loop stable systems)
Assume that there exists matrices $P,\ S \succ 0$ such that

$$A^T P A - P \preceq -S \tag{10.32}$$

Furthermore, $\mathbb{X} = \mathbb{R}^n$ and $0 \in \mathbb{U}$. Appending the minimax MPC problem (10.16) with the contraction constraint

$$x_{k+1|k}^T P x_{k+1|k} - x_{k|k}^T P x_{k|k} \leq -x_{k|k}^T S x_{k|k}\ \forall \Delta_k \in \mathbf{\Delta} \tag{10.33}$$

will guarantee asymptotic stability.

Proof Follows trivially. The assumption (10.32) implies that $u_{k|k} = 0$ satisfies the contraction constraint 10.33. Since $0 \in \mathbb{U}$ and there are no state constraints, this is always a feasible solution. The remaining $N-1$ control inputs can be chosen arbitrarily with $U \in \mathbb{U}^N$. Asymptotic stability follows from the contraction constraint. □

Generalization to unstable systems can be done by introducing an additional constraint on the initial state.

Theorem 10.3 (Guaranteed stability, general systems)
Assume that there exists matrices $P,\ S \succ 0$ and a linear state feedback L such that

$$(A + (B_0 + B_1\Delta B_2 L))^T P(A + (B_0 + B_1\Delta B_2 L)) - P \preceq -S\ \forall\Delta \in \mathbf{\Delta} \quad (10.34)$$

Furthermore, assume $\mathbb{X} = \mathbb{R}^n$ and $Lx_k \in \mathbb{U}\ \forall x_k \in \mathbb{E}_P$. Appending the minimax MPC problem (10.16) with the contraction constraint

$$x_{k+1|k}^T P x_{k+1|k} - x_{k|k}^T P x_{k|k} \leq -x_{k|k}^T S x_{k|k}\ \forall\Delta_k \in \mathbf{\Delta} \quad (10.35)$$

will guarantee asymptotic stability if the initial state satisfies $x_{0|0} \in \mathbb{E}_P$.

Proof Follows by induction. Assume the problem was feasible for $x_{k-1|k-1}$. The contraction constraint then guarantees $x_{k|k} \in \mathbb{E}_P$. The choice $u_{k|k} = Lx_{k|k}$ is feasible with respect to the control constraints from the assumptions. This choice also satisfies the contraction constraint (10.35) due to (10.34). Hence, the problem is feasible for $x_{k|k}$. Asymptotic stability follows from the contraction constraint. □

The contraction constraints (10.33) and (10.35) can be incorporated into our framework by first applying a Schur complement

$$\begin{pmatrix} x_{k|k}^T P x_{k|k} - x_{k|k}^T S x_{k|k} & (Ax_{k|k} + (B_0 + B_1\Delta_k B_2)u_{k|k})^T \\ \star & P^{-1} \end{pmatrix} \succeq 0 \quad \forall\Delta_k \in \mathbf{\Delta}$$

Robust satisfaction of this LMI can be solved using Theorem 3.4. Crucial to note is that Theorem 3.4 is both sufficient and necessary in this case, since we only have one unstructured uncertainty. Introduce the scalar multiplier τ, and the semidefinite relaxation gives (compare (10.6) and (10.7))

$$\begin{pmatrix} x_{k|k}^T P x_{k|k} - x_{k|k}^T S x_{k|k} & (Ax_{k|k} + B_0 u_{k|k})^T & (B_2 u_{k|k})^T \\ \star & P^{-1} - \tau B_1 B_1^T & 0 \\ \star & 0 & \tau I \end{pmatrix} \succeq 0$$

Adding this LMI to the minimax problem (10.16) will guarantee asymptotic stability, since the constraint is equivalent to the contraction constraints (10.33) and (10.35) which always are feasible.

All that is left is to find the matrices P, S and L. This is a standard robust feedback problem. Multiply (10.34) from left and right with the matrix $W = P^{-1}$, define $K = LW$, and apply a Schur complement. This gives us the following uncertain LMI.

$$\begin{pmatrix} W & (AW + (B_0 + B_1\Delta_k B_2)K)^T & W \\ \star & W & 0 \\ \star & 0 & S^{-1} \end{pmatrix} \succeq 0 \quad \Delta_k \in \mathbf{\Delta} \tag{10.36}$$

Separating certain and uncertain terms, and applying Theorem 3.4 for the single unstructured uncertainty Δ_k gives a necessary and sufficient LMI in W, K and τ

$$\begin{pmatrix} W & (AW + B_0K)^T & W & (B_2K)^T \\ \star & W - \tau B_1B_1^T & 0 & 0 \\ \star & 0 & S^{-1} & 0 \\ \star & 0 & 0 & \tau I \end{pmatrix} \succeq 0 \tag{10.37}$$

Satisfaction of the control constraints in $\mathbb{E}_P$ is solved using the same methods as in Appendix 5.A and the result is a set of LMIs

$$\begin{pmatrix} ((f_u)_i)^2 & (E_uK)_i \\ ((E_uK)_i)^T & W \end{pmatrix} \succeq 0 \tag{10.38}$$

A semidefinite program can now be defined to find, e.g., a maximum volume ellipsoid $\mathbb{E}_P$. See Appendix 5.A for a similar result.

10.4 Models with Uncertain Gain

In this section, we will try to convince the reader that the framework with a norm-bounded uncertainty indeed is a reasonable model. Some extensions are pointed out and it is briefly discussed how the models can be obtained.

Several Uncertainty Blocks

The framework in this chapter can without too much effort be extended to models with several uncertainty blocks.

$$B(\Delta_k) = B_0 + \sum_{i=1}^{q} B_1^i \Delta_k^i B_2^i, \quad ||\Delta_k^i|| \leq 1 \tag{10.39}$$

The only complication with this model is that the stability theory in Section 10.3.3 fail. The reason is that there are several uncertainty blocks in the prediction of $x_{k+1|k}$, hence, the semidefinite relaxation of the contraction constraint will not be necessary for the contraction constraint to be feasible. The extended model can be used to model some parametric (scalar) uncertainties.

Example 10.1 (Parametric model)
Consider the following model with scalar parametric uncertainties.

$$B(\Delta_k) = \begin{pmatrix} 1+\Delta_k^1 & 2+\Delta_k^1+\Delta_k^2 \\ 3 & 4+\Delta_k^2 \end{pmatrix}, \quad |\Delta_k^i| \leq 1 \tag{10.40}$$

This can be written as

$$B(\Delta_k) = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Delta_k^1 \begin{pmatrix} 1 & 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \Delta_k^2 \begin{pmatrix} 0 & 1 \end{pmatrix} \tag{10.41}$$

Unfortunately, general parametric models cannot be modeled using this type of model. A simple counter-example is the following model

$$B(\Delta_k) = \begin{pmatrix} 1+\Delta_k & 0 \\ 0 & 1+\Delta_k \end{pmatrix}, \quad |\Delta_k| \leq 1 \tag{10.42}$$

Since Δ_k^i are scalars in the parametric models, the products $B_1^i \Delta_k^i B_2^i$ all have rank 1. Hence, the uncertainties must enter as rank-1 perturbations.

Approximating Polytopic Models with Norm-bounded Models

An alternative to formulating parametric models in the form (10.39) is to conservatively approximate them as a simple norm-bounded model. The following algorithm has been proposed (Boyd et al., 1994). Assume a polytopic model.

$$B \in \mathbf{Co}\{B^{(1)}, B^{(2)}, \ldots, B^{(q)}\} \tag{10.43}$$

For fixed x, the matrix B maps x to a vector y

$$y = Bx \in \mathbf{Co}\{B^{(1)}x, B^{(2)}x, \ldots, B^{(q)}x\} = \mathbf{Co}\{y^{(1)}, y^{(2)}, \ldots, y^{(q)}\} \tag{10.44}$$

This set should be contained in the set that x would be mapped to if a norm-bounded model had been used instead.

$$y = (B_0 + B_1 \Delta B_2)x, \quad ||\Delta|| \leq 1 \tag{10.45}$$

With $\eta = \Delta B_2 x$, this can be written as

$$y = B_0 x + B_1 \eta \tag{10.46}$$

The polytopic set is contained if each vertex, $B^{(i)}x$, is in the set defined by the norm-bounded uncertainty (which is convex). In other words, for each vertex $y^{(i)}$, it should be possible to write

$$y^{(i)} = B_0 x + B_1 \eta, \quad \eta^T \eta \leq x^T B_2^T B_2 x \tag{10.47}$$

If B_1 is assumed to be invertible, it holds that $\eta = B_1^{-1}(B^{(i)} - B_0)x$. The inequality in (10.47) thus implies

$$(B^{(i)} - B_0)^T B_1^{-T} B_1^{-1} (B^{(i)} - B_0) \preceq B_2^T B_2 \tag{10.48}$$

Define $Z = B_1 B_1^T$ and $Y = B_2^T B_2$, and a Schur complement yields

$$\begin{pmatrix} Y & (B^{(j)} - B_0)^T \\ (B^{(j)} - B_0) & Z \end{pmatrix} \succeq 0 \tag{10.49}$$

At this point, any size or gain-related measure (Tr(·), $||\cdot||$,...) on Z and Y can be minimized. Notice that the procedure does not uniquely define B_1 and B_2. As an example, if B_2 is a column-vector, Y will be a scalar and there will be infinitely many B_2 such that $B_2^T B_2 = Y$.

Identifying B_0, B_1 and B_2 from experimental data

Let us just sketch how a model of our type can be obtained from input-output data. Given a data-set $z^N = \{u_1, u_2, \ldots, u_N, y_1, y_2, \ldots, y_N\}$, the problem is to identify suitable matrices A, B_0, B_1, B_2 and C.

To begin with, identify a nominal model A, B_0 and C using any preferred method (Ljung, 1999), and calculate the output error residuals e_k. for this model.

$$x_{k+1} = Ax_k + Bu_k \tag{10.50a}$$
$$y_k = Cx_k + e_k \tag{10.50b}$$

In a second step, assume that the residuals can be modeled by an FIR model of order M with a norm-bounded uncertainty.

$$e_k = (B_0 + B_1 \Delta_k B_2) U_k, \quad ||\Delta_k|| \leq 1 \tag{10.51a}$$
$$U_k = \begin{pmatrix} u_{k-1}^T & u_{k-2}^T & \ldots & u_{k-M}^T \end{pmatrix}^T \tag{10.51b}$$

Identifying a model of this type is outlined in (Calafiore et al., 2002). When this has been completed, a system with a known A matrix and norm-bounded uncertainty in the B matrix can be obtained with a suitable parallel connection of the two linear systems.

10.5 Simulation Results

To evaluate the performance of the minimax controller, we conclude the chapter with a couple of numerical experiments.

Example 10.2 (Comparison with nominal MPC)
The system under consideration is a double integrator with an uncertain gain and zero location.

$$\begin{aligned} x_{k+1} &= \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} x_k + Bu_k \\ y_k &= \begin{pmatrix} 0 & 1 \end{pmatrix} x_k \end{aligned}$$

where

$$B \in \mathbf{Co}\{B^{(1)}, B^{(2)}, B^{(3)}, B^{(4)}\} = \mathbf{Co}\{\begin{pmatrix} 1.50 \\ 0.75 \end{pmatrix}, \begin{pmatrix} 1.50 \\ 0.25 \end{pmatrix}, \begin{pmatrix} 0.50 \\ 0.75 \end{pmatrix}, \begin{pmatrix} 0.50 \\ 0.25 \end{pmatrix}\} \tag{10.52}$$

The polytopic model on B has to be converted to a norm-bounded model. This is done using the approach described in Section 10.4 and gives

$$B_0 = \begin{pmatrix} 1.0 \\ 0.5 \end{pmatrix}, \; B_1 = \begin{pmatrix} 0.61 & 0 \\ 0 & 0.43 \end{pmatrix}, \; B_2 = \begin{pmatrix} 0.71 \\ 0.71 \end{pmatrix} \tag{10.53}$$

Our goal is to control the output y_k under the control constraint $|u_k| \leq 1$. A natural tuning is thus to put a substantial weight on y_k in the performance measure. It was decided to use $Q = 1$ and $R = 0.01$, and a prediction horizon $N = 15$.

As a first experiment, we test a severely upsetting uncertainty realization.

$$B = \begin{cases} B^{(4)} & u_k > 0 \\ \\ B^{(1)} & u_k \leq 0 \end{cases} \tag{10.54}$$

The can be interpreted as a system with a sign dependent gain,

$$B = B_0(1 - 0.5\text{sign}(u_k)) \tag{10.55}$$

Closed-loop responses for the proposed minimax controller and a nominal MPC controller, with initial condition $x_0 = \begin{pmatrix} 0 & 5 \end{pmatrix}^T$ and the uncertainty realization (10.54), are given in Figure 10.1. The nominal MPC controller fails completely, while the minimax controller gives a reasonable response, albeit slow.

Of course, the aggressive tuning is doomed to give a nominal controller with poor robustness. A natural solution is thus to detune the nominal controller. After some trial and error, $R = 100$ gives a nominal controller with performance comparable to the minimax controller. The response is given Figure 10.2, together with the response for the minimax controller using $R = 0.01$ as before.

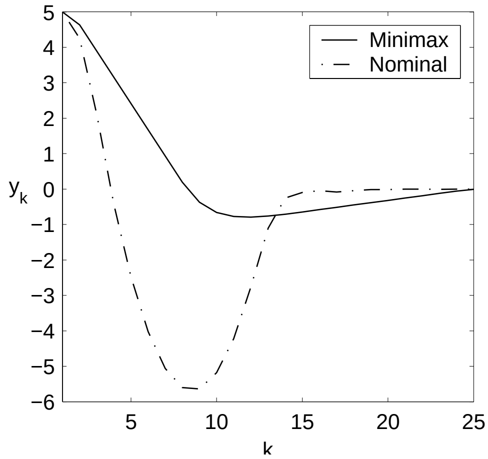

Figure 10.1: Closed-loop responses for minimax MPC controller and nominal MPC controller with control weight $R = 0.01$. The aggressive tuning gives poor robustness in the nominal controller.

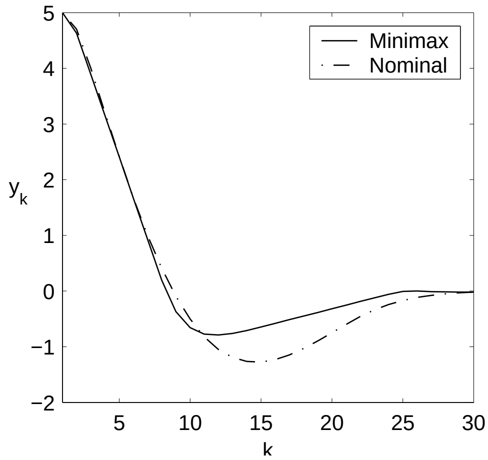

Figure 10.2: The performance of the nominal controller can be improved by increasing the control weight to $R = 100$.

Hence, by tuning the nominal controller carefully, it is possible to obtain acceptable performance. This might seem to invalidate the work in this chapter. However, the idea with robust control is that the tuning variables should reflect the control objective, and robustness should be built-in. Given a new uncertainty model, it

should not be necessary to redesign the controller.
Finally, Figure 10.3 shows the response for a random uncertainty realization.

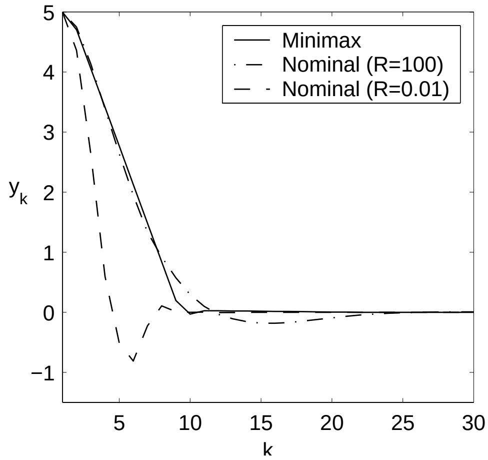

Figure 10.3: Closed-loop responses under random uncertainty realization. The price paid for improved robustness against worst-case uncertainties is a slower system.

A minimax problem with a polytopic uncertainty model can be solved with a straightforward enumeration technique. For $B \in \mathbf{Co}\{B^{(1)}, \dots B^{(q)}\}$, it is readily shown that the predictions are given by $X \in \mathcal{A}x_{k|k} + \mathbf{Co}\{\mathcal{B}^{(1)}U, \dots, \mathcal{B}^{(q^N)}U\}$, with $\mathcal{B}^{(i)}$ defined as in (2.10), but with different combinations of B matrices along the prediction horizon. This means that the minimax problem (10.5) can be solved with (write the problem as a second order cone program to allow for efficient implementation).

$$
\begin{array}{lrcl}
\min\limits_{U,t} & t & & \\
\text{subject to} & \left\| \begin{matrix} 2\mathcal{Q}^{1/2}\mathcal{C}(\mathcal{A}x_{k|k} + \mathcal{B}^{(i)}U) \\ 2\mathcal{R}^{1/2}U \\ 1-t \end{matrix} \right\| & \leq & 1+t \\
 & \mathcal{E}_u U & \leq & \mathcal{F}_u \\
 & \mathcal{E}_x(\mathcal{A}x_{k|k} + \mathcal{B}^{(i)}U) & \leq & \mathcal{F}_x
\end{array}
\tag{10.56}
$$

An interesting question now is how the proposed minimax controller compares to the exact solution. In other words, how efficient are the semidefinite relaxations that have been used in this chapter. A simple example illustrates this.

Example 10.3 (Quality of relaxations)
The system is essentially the same as in Example 10.2, except for a different uncertainty structure.

$$B \in \mathbf{Co}\{\begin{pmatrix} 1.5 \\ 0.75 \end{pmatrix}, \begin{pmatrix} 0.5 \\ 0.25 \end{pmatrix}\} \tag{10.57}$$

Moreover, the horizon was reduced to $N = 8$. The motivation for these changes is that the original problem is intractable for an enumerative solution approach (it would require $4^{15} \simeq 10^9$ second order cone constraints).
The polytopic model (10.57) can be written as a norm-bounded model with

$$B_0 = \begin{pmatrix} 1 \\ 0.5 \end{pmatrix}, \; B_1 = \begin{pmatrix} 0.5 \\ 0.25 \end{pmatrix}, \; B_2 = 1 \tag{10.58}$$

The same numerical experiments as in the previous example were carried out, using both the proposed minimax controller based on semidefinite relaxations, and a controller based on the exact solution (10.56). Surprisingly, the responses were identical in all simulations. The quality measure $\alpha = \frac{t_{sdp}}{t_{exact}}$ (see Example 5.3 for a detailed explanation) was evaluated during the simulation of the proposed minimax controller, and the results from one of the simulations is presented[4] in Figure 10.4. The numbers indicate that the upper bound on the worst-case cost, obtained using the semidefinite relaxation, never was more than 2 percent larger than the true worst-case cost. Note that the upper bound was tight in 13 out of 15 samples.

[4] Only the numbers for the first 15 samples are presented. The state had then converged to the origin, and the values of t_{sdp} and t_{exact} were so small that the numerical precision in the solver SDPT3 made a comparison irrelevant.

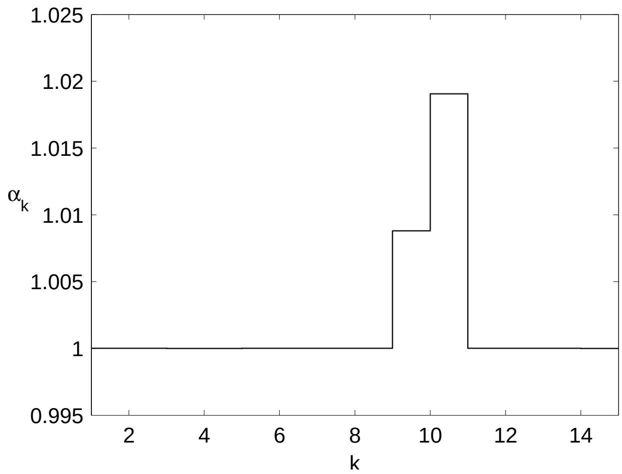

Figure 10.4: The figure shows the ratio between the upper bound on the worst-case finite horizon cost obtained with the semidefinite relaxation, and the exact worst-case finite horizon cost. The upper bound is never more than 2 percent larger than the true value.

11

MPC for LFT systems

To incorporate uncertainties in the A matrix, the simple model in Chapter 10 with an uncertain B matrix is extended to a more general linear fractional transformation (LFT) model.

The minimax finite horizon problem with a quadratic performance has been addressed before, but most often with polytopic models (Schuurmans and Rossiter, 2000; Casavola et al., 2000). The motivation for using polytopic models is that the problem can be solved exactly by straightforward enumeration. The drawback is that the resulting optimization problems have exponential complexity in the number of uncertainties and the prediction horizon. LFT models have also been addressed frequently in the minimax MPC literature, but most often along the lines of (Kothare et al., 1994) where no finite horizon cost is included. A recent approach with LFT models and a finite horizon cost is proposed (Casavola et al., 2002b). The method is related to the work introduced in this chapter, in the sense that it uses the S-procedure to approximately perform the maximization in the minimax problem. However, the derived optimization problems in (Casavola et al., 2002b) are nonconvex, and further off-line approximations and heuristics have to be introduced to obtain tractable problems.

The novelty of the work introduced in this chapter is that we derive a polynomially growing convex formulation of the (conservative approximation of) minimax MPC problem with a finite horizon quadratic performance measure.

11.1 Uncertainty Model

The goal in this chapter is to apply minimax MPC to uncertain systems described by a linear fractional transformation (LFT)

$$x_{k+1} = Ax_k + Bu_k + Gp_k \quad (11.1a)$$
$$z_k = D_x x_k + D_u u_k + D_p p_k \quad (11.1b)$$
$$p_k = \Delta_k z_k, \quad ||\Delta_k|| \leq 1 \quad (11.1c)$$
$$y_k = Cx_k \quad (11.1d)$$

A more general model with uncertainties in the C matrix (and a direct term D) can also be dealt with using the methods in this chapter. However, to obtain a notation similar to the notation used in earlier chapters, we refrain from these extensions.

The variables $p_k \in \mathbb{R}^l$ and $z_k \in \mathbb{R}^q$ are auxiliary variables generated by the actual uncertainty Δ_k. Eliminating p_k and z_k reveals a model with both dynamic uncertainty and gain uncertainty.

$$\begin{aligned} x_{k+1} &= (A + G(I - \Delta_k D_p)^{-1}\Delta_k D_x)x_k + (B + G(I - \Delta_k D_p)^{-1}\Delta_k D_u)u_k \\ y_k &= Cx_k \end{aligned}$$

Of course, it is assumed that the model is well-posed so that the indicated inverse exists for all $||\Delta_k|| \leq 1$.

The fact that there is uncertainty in the dynamics complicates things severely. To understand this, consider a prediction of x_{k+2}.

$$x_{k+2} = (A + G(I - \Delta_{k+1}D_p)^{-1}\Delta_{k+1}D_x)(A + G(I - \Delta_k D_p)^{-1}\Delta_k D_x)x_k + \ldots$$

The uncertainties enter the prediction in a highly nonlinear fashion. If there only would be uncertainty in the gain ($D_x = D_p = 0$), the prediction would simplify to

$$x_{k+2} = A^2 x_k + A(B + G\Delta_k D_u)u_k + (B + G\Delta_{k+1}D_x)u_{k+1}$$

In this case, the uncertainties enter linearly. Due to this, models with uncertainty in the gain only have been studied rather extensively. Solutions (with varying level of efficiency and conservativeness) to a number of minimax MPC problems for systems with uncertain gain have been proposed (Campo and Morari, 1987; Zheng, 1995; Lee and Cooley, 1997; Oliviera et al., 2000). The linearity is of course why we managed to develop our minimax controller in Chapter 10.

11.2 LFT Model of Predictions

As we saw earlier, the predictions of $x_{k+j|k}$ depend in a nonlinear fashion on the future uncertainties Δ^N (see Section 10.1 for a definition of Δ^N). By writing the predictions in an implicit way, it is possible to show that these can be written as an LFT. Of course, this comes as no surprise since each one-step prediction is defined using an LFT, and a multi-step prediction is in principle a series connection of several LFTs, which also is an LFT (Zhou et al., 1995). However, to easily incorporate the LFT in our minimax MPC problem we must obtain a compact expression of this LFT.

To enable the derivation of the LFT model of the predictions, we first introduce vectorized versions of the auxiliary variables $z_{k+j|k}$ and $p_{k+j|k}$.

$$P = \begin{pmatrix} p_{k|k}^T & p_{k+1|k}^T & \dots & p_{k+N-1|k}^T \end{pmatrix}^T \tag{11.2a}$$

$$Z = \begin{pmatrix} z_{k|k}^T & z_{k+1|k}^T & \dots & z_{k+N-1|k}^T \end{pmatrix}^T \tag{11.2b}$$

With these variables, the following relationships hold (note that X appears on both sides of (11.3b)).

$$Y = \mathcal{C}X \tag{11.3a}$$

$$X = \mathcal{A}X + \mathcal{B}U + \mathcal{G}P + b \tag{11.3b}$$

$$Z = \mathcal{D}_x X + \mathcal{D}_u U + \mathcal{D}_p P \tag{11.3c}$$

$$P = \Delta^N Z \tag{11.3d}$$

The variables X, U and Δ^N and the matrix $\mathcal{C}$ are defined as in previous chapters. The matrices $\mathcal{A}$, $\mathcal{B}$, $\mathcal{G}$, $\mathcal{D}_x$, $\mathcal{D}_u$ and $\mathcal{D}_p$ are defined as (notice the crucial difference on $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{G}$, compared to the notation used in previous chapters)

$$\mathcal{A} = \begin{pmatrix} 0 & 0 \\ \oplus_{j=1}^{N-1} A & 0 \end{pmatrix},\ \mathcal{B} = \begin{pmatrix} 0 & 0 \\ \oplus_{j=1}^{N-1} B & 0 \end{pmatrix},\ \mathcal{G} = \begin{pmatrix} 0 & 0 \\ \oplus_{j=1}^{N-1} G & 0 \end{pmatrix} \tag{11.4a}$$

$$\mathcal{D}_x = \oplus_{j=1}^{N} D_x,\ \mathcal{D}_u = \oplus_{j=1}^{N} D_u,\ \mathcal{D}_p = \oplus_{j=1}^{N} D_p \tag{11.4b}$$

$$b = \begin{pmatrix} x_{k|k}^T & 0 & \dots & 0 \end{pmatrix}^T \tag{11.4c}$$

The equations in (11.3) implicitly define X, and it will now be shown how to obtain an explicit expression for X in terms of U and Δ^N. From (11.3b) we have

$$X = (I - \mathcal{A})^{-1}(\mathcal{B}U + \mathcal{G}P + b) \tag{11.5}$$

Insert this into (11.3c)

$$Z = \mathcal{D}_x(I - \mathcal{A})^{-1}(\mathcal{B}U + \mathcal{G}P + b) + \mathcal{D}_u U + \mathcal{D}_p P \tag{11.6}$$

and insert this expression into (11.3d)

$$P = \Delta^N(\mathcal{D}_x(I - \mathcal{A})^{-1}(\mathcal{B}U + \mathcal{G}P + b) + \mathcal{D}_u U + \mathcal{D}_p P) \tag{11.7}$$

Solve this equation to obtain P

$$P = (I - \Delta^N(\mathcal{D}_x(I - \mathcal{A})^{-1}\mathcal{G} + \mathcal{D}_p))^{-1}\Delta^N(\mathcal{D}_x(I - \mathcal{A})^{-1}(\mathcal{B}U + b) + \mathcal{D}_u U + c)$$

Define $\Omega = (\mathcal{D}_x(I-\mathcal{A})^{-1}\mathcal{G}+\mathcal{D}_p)$, insert the expression for P into (11.5), and solve for X.

$$X = (I-\mathcal{A})^{-1}(\mathcal{B}U+b+\mathcal{G}(I-\Delta^N\Omega)\Delta^N(\mathcal{D}_x(I-\mathcal{A})^{-1}(\mathcal{B}U+b)+\mathcal{D}_uU)) \tag{11.8}$$

This can be written as an LFT

$$X = \tilde{X}+\Lambda(I-\Delta^N\Omega)^{-1}\Delta^N\Psi \tag{11.9}$$

with the following definitions

$$\tilde{X} = (I-\mathcal{A})^{-1}(\mathcal{B}U+b) \tag{11.10a}$$
$$\Lambda = (I-\mathcal{A})^{-1}\mathcal{G} \tag{11.10b}$$
$$\Psi = \mathcal{D}_x(I-\mathcal{A})^{-1}(\mathcal{B}U+b)+\mathcal{D}_uU \tag{11.10c}$$

Use $(I-\Delta^N\Omega)^{-1}\Delta^N = \Delta^N(I-\Omega\Delta^N)^{-1}$ and write the LFT as

$$X = \tilde{X}+\Lambda\Delta^N(I-\Omega\Delta^N)^{-1}\Psi \tag{11.11}$$

This expression will be used in a minimax MPC problem.

11.3 Minimax MPC

As usual, our goal is solve the quadratic performance minimax MPC problem

$$\boxed{\begin{array}{lrcl} \min\limits_{u}\max\limits_{\Delta} & \multicolumn{3}{l}{\sum_{j=0}^{N-1} y_{k+j|k}^TQy_{k+j|k}+u_{k+j|k}^TRu_{k+j|k}} \\ \text{subject to} & u_{k+j|k} & \in & \mathbb{U} \quad \forall\Delta\in\boldsymbol{\Delta} \\ & x_{k+j|k} & \in & \mathbb{X} \quad \forall\Delta\in\boldsymbol{\Delta} \\ & \Delta_{k+j|k} & \in & \boldsymbol{\Delta} \end{array}} \tag{11.12}$$

which we vectorize and write as an uncertain program in an epigraph form

$$\boxed{\begin{array}{lrcl} \min\limits_{U,t} & t & & \\ \text{subject to} & Y^T\mathcal{Q}Y+U^T\mathcal{R}U & \leq & t \quad \forall\Delta\in\boldsymbol{\Delta}^N \\ & U & \in & \mathbb{U}^N \quad \forall\Delta^N\in\boldsymbol{\Delta}^N \\ & X & \in & \mathbb{X}^N \quad \forall\Delta^N\in\boldsymbol{\Delta}^N \end{array}} \tag{11.13}$$

Bounding the Performance Measure

Apply a Schur complement on the performance constraint, plug in the definition of X and separate certain and uncertain terms.

$$\begin{pmatrix} t & (\mathcal{C}\tilde{X})^T & U^T \\ \mathcal{C}\tilde{X} & \mathcal{Q}^{-1} & 0 \\ U & 0 & \mathcal{R}^{-1} \end{pmatrix} + \begin{pmatrix} 0 \\ \Lambda \\ 0 \end{pmatrix} \Delta^N(I-\Omega\Delta^N)^{-1}\begin{pmatrix} \Psi & 0 & 0 \end{pmatrix} + (\star) \tag{11.14}$$

This uncertain LMI can be dealt with using Theorem 3.5. The semidefinite relaxation gives $\tau \in \mathbb{R}_+^N$ and the associated matrices $\mathcal{S} = \oplus_{j=1}^N \tau_i I^{l\times l}$ and $\mathcal{T} = \oplus_{j=1}^N \tau_j I^{q\times q}$. The sufficient LMI in Theorem 3.5 simplifies to

$$\begin{pmatrix} t & (\mathcal{C}\tilde{X})^T & U^T & \Psi^T \\ \mathcal{C}\tilde{X} & \mathcal{Q}^{-1} - \mathcal{C}\Lambda\mathcal{S}\Lambda^T\mathcal{C}^T & 0 & -\mathcal{C}\Lambda\mathcal{S}\Omega^T \\ U & 0 & \mathcal{R}^{-1} & 0 \\ \Psi & -\Omega\mathcal{S}\Lambda^T\mathcal{C}^T & 0 & \mathcal{T} - \Omega\mathcal{S}\Omega^T \end{pmatrix} \succeq 0 \tag{11.15}$$

The LMI has almost the structure as the LMIs we derived in Chapter 5 and 10, so at this point, LFT models do not seem to introduce any additional problems. However, state constraints are not as efficiently dealt with as in previous chapters.

Robust Constraint Satisfaction

Recall the state constraints, which with our LFT can be written as

$$\mathcal{E}_x(\tilde{X} + \Lambda\Delta^N(I - \Omega\Delta^N)^{-1}\Psi) \leq \mathcal{F}_x \quad \forall \Delta^N \in \boldsymbol{\Delta}^N \tag{11.16}$$

Multiplying this expression with 2 and rearranging terms slightly reveals that the ith constraint can be written in a format suitable for Theorem 3.5.

$$(2(\mathcal{F}_x - \tilde{X}))_i - (\mathcal{E}_x)_i(\Lambda\Delta^N(I - \Omega\Delta^N)^{-1}\Psi) - (\star) \geq 0$$

Application of Theorem 3.5 gives a sufficient condition for robust satisfaction of the state constraints.

$$\begin{pmatrix} 2(\mathcal{F}_x - \tilde{X})_i - (\mathcal{E}_x)_i\Lambda\mathcal{S}\Lambda^T((\mathcal{E}_x)_i)^T & \Psi^T - (\mathcal{E}_x)_i\Lambda\mathcal{S}\Omega^T \\ \star & \mathcal{T} - \Omega\mathcal{S}\Omega^T \end{pmatrix} \succeq 0 \tag{11.17}$$

Note that the matrices $\mathcal{S}$ and $\mathcal{T}$ do not have to be defined using the same multiplier τ as we have for the relaxation of the performance constraint. Instead, new multipliers can be introduced for each linear constraint to obtain a less conservative relaxation.

It should now be obvious that state constraints are much more expensive when we have an LFT model, than the case with additive disturbances (linear state constraints were transformed to new linear state constraints), and the model with uncertain gain (linear state constraints were transformed to new linear state and N second order cones using N new variables). Now, assuming q linear state constraints at each time, we are forced to introduce Nq additional LMIs and, if we want to have an as efficient relaxation as possible, Nq multipliers of dimension N[1].

The constraints can be taken care of slightly more efficiently if the constraints are symmetric, i.e., they can be written as

$$-\mathcal{F}_x \leq \mathcal{E}_x(\tilde{X} + \Lambda\Delta^N(I - \Omega\Delta^N)^{-1}\Psi) \leq \mathcal{F}_x \tag{11.18}$$

[1]This is not entirely true. Relaxation of a state constraint on $x_{k+j|k}$ need only q multipliers of dimension j, since $x_{k+j|k}$ only depends on $\Delta_{k+i|k},\ i < j$. However, to keep notation clear, we omit these details. The result is anyway that we need $O(N^2q)$ variables all together

If we look at each row separately, we can square the constraint to obtain

$$((\mathcal{E}_x)_i(\tilde{X}+\Lambda\Delta^N(I-\Omega\Delta^N)^{-1}\Psi))^2 \leq ((\mathcal{F}_x)_i)^2 \tag{11.19}$$

A Schur complement and separation of certain and uncertain terms yield

$$\begin{pmatrix} ((\mathcal{F}_x)_i)^2 & (\mathcal{E}_x)_i\tilde{X} \\ \star & 1 \end{pmatrix} + \begin{pmatrix} 0 \\ (\mathcal{E}_x)_i\Lambda \end{pmatrix} \Delta^N(I-\Omega\Delta^N)^{-1}\begin{pmatrix}\Psi & 0\end{pmatrix} + (\star) \succeq 0 \tag{11.20}$$

This time, Theorem 3.5 simplifies to

$$\begin{pmatrix} ((\mathcal{F}_x)_i)^2 & (\mathcal{E}_x)_i\tilde{X} & \Psi^T \\ \star & 1-(\mathcal{E}_x)_i\Lambda\mathcal{S}\Lambda^T((\mathcal{E}_x)_i)^T & -(\mathcal{E}_x)_i\Lambda\mathcal{S}\Omega^T \\ \star & \star & \mathcal{T}-\Omega\mathcal{S}\Omega^T \end{pmatrix} \succeq 0 \tag{11.21}$$

Hence, we have half as many LMIs and multipliers compared to the case when symmetry in the constraints is neglected.

Final Problem

At this point, we are ready to summarize our findings and present the main result of this chapter. A semidefinite relaxation of the minimax problem (11.12) is given by the following semidefinite program.

$$\begin{aligned} \min_{U,t,\tau} \quad & t \\ \text{subject to} \quad & \begin{pmatrix} t & (\mathcal{C}\tilde{X})^T & U^T & \Psi^T \\ \star & \mathcal{Q}^{-1}-\mathcal{C}\Lambda\mathcal{S}\Lambda^T\mathcal{C}^T & 0 & -\mathcal{C}\Lambda\mathcal{S}\Omega^T \\ \star & 0 & \mathcal{R}^{-1} & 0 \\ \star & \star & 0 & \mathcal{T}-\Omega\mathcal{S}\Omega^T \end{pmatrix} \succeq 0 \\ & (11.17) \text{ or } (11.21) \end{aligned} \tag{11.22}$$

Notice that we have not explicitly written anything about whether different multipliers are used for the relaxations of the uncertain constraints. This is a design issue, and is a way to trade performance for a more efficient implementation.

11.4 Extensions

The framework easily allows extensions, and the most important ones will now be discussed.

11.4.1 Feedback Predictions

To reduce conservativeness, feedback predictions are vital. Recall the implicit model we worked with earlier.

$$X = \mathcal{A}X + \mathcal{B}U + \mathcal{G}P + b \tag{11.23a}$$

$$Z = \mathcal{D}_xX + \mathcal{D}_uU + \mathcal{D}_pP \tag{11.23b}$$

Incorporating a parameterization $U = \mathcal{L}X + V$ gives

$$X = (\mathcal{A} + \mathcal{B}\mathcal{L})X + \mathcal{B}V + \mathcal{G}P + b \quad (11.24a)$$
$$Z = (\mathcal{D}_x + \mathcal{D}_u\mathcal{L})X + \mathcal{D}_uV + \mathcal{D}_pP \quad (11.24b)$$

Repeating the calculations in Section 11.2 gives us an LFT for X and U.

$$\begin{pmatrix} X \\ U \end{pmatrix} = \begin{pmatrix} \tilde{X} \\ \mathcal{L}\tilde{X} + V \end{pmatrix} + \begin{pmatrix} \Lambda \\ \mathcal{L}\Lambda \end{pmatrix} \Delta^N (I - \Omega\Delta^N)^{-1}\Psi \quad (11.25)$$

with the following definitions

$$\tilde{X} = (I - (\mathcal{A} + \mathcal{B}\mathcal{L}))^{-1}(\mathcal{B}V + b) \quad (11.26a)$$
$$\Lambda = (I - (\mathcal{A} + \mathcal{B}\mathcal{L}))^{-1}\mathcal{G} \quad (11.26b)$$
$$\Psi = (\mathcal{D}_x + \mathcal{D}_u\mathcal{L})(I - (\mathcal{A} + \mathcal{B}\mathcal{L}))^{-1}(\mathcal{B}V + b) + \mathcal{D}_uV \quad (11.26c)$$
$$\Omega = (\mathcal{D}_x + \mathcal{D}_u\mathcal{L})(I - (\mathcal{A} + \mathcal{B}\mathcal{L}))^{-1}\mathcal{G} + \mathcal{D}_p \quad (11.26d)$$

It can easily be shown that the semidefinite relaxation (11.15) is changed to

$$\begin{pmatrix} t & (\mathcal{C}\tilde{X})^T & V^T & \Psi^T \\ \mathcal{C}\tilde{X} & \mathcal{Q}^{-1} - \mathcal{C}\Lambda\mathcal{S}\Lambda^T\mathcal{C}^T & 0 & -\mathcal{C}\Lambda\mathcal{S}\Omega^T \\ V & 0 & \mathcal{R}^{-1} - \mathcal{L}\Lambda\mathcal{S}\Lambda^T\mathcal{L}^T & -\mathcal{L}\Lambda\mathcal{S}\Omega^T \\ \Psi & -\Omega\mathcal{S}\Lambda^T\mathcal{C}^T & -\Omega\mathcal{S}\Lambda^T\mathcal{L}^T & \mathcal{T} - \Omega\mathcal{S}\Omega^T \end{pmatrix} \succeq 0 \quad (11.27)$$

Additionally, the control constraints are mapped into state constraints

$$\mathcal{E}_uU \leq \mathcal{F}_u \Leftrightarrow \mathcal{E}_u(\mathcal{L}\tilde{X} + V + \mathcal{L}\Lambda\Delta^N(I - \Omega\Delta^N)^{-1}\Psi) \leq \mathcal{F}_u \quad (11.28)$$

These are readily taken care of using the same techniques as in the previous section. Depending on whether we have asymmetric or symmetric constraints, we obtain

$$\begin{pmatrix} 2(\mathcal{F}_u - \mathcal{L}\tilde{X} - V)_i - (\mathcal{E}_x)_i\mathcal{L}\Lambda\mathcal{S}\Lambda^T\mathcal{L}^T((\mathcal{E}_x)_i)^T & \Psi^T - (\mathcal{E}_u)_i\mathcal{L}\Lambda\mathcal{S}\Omega^T \\ \star & \mathcal{T} - \Omega\mathcal{S}\Omega^T \end{pmatrix} \succeq 0 \quad (11.29)$$

or

$$\begin{pmatrix} ((\mathcal{F}_x)_i)^2 & (\mathcal{E}_u)_i(\mathcal{L}\tilde{X} + V) & \Psi^T \\ \star & 1 - (\mathcal{E}_u)_i\mathcal{L}\Lambda\mathcal{S}\Lambda^T\mathcal{L}^T((\mathcal{E}_u)_i)^T & -(\mathcal{E}_u)_i\mathcal{L}\Lambda\mathcal{S}\Omega^T \\ \star & \star & \mathcal{T} - \Omega\mathcal{S}\Omega^T \end{pmatrix} \succeq 0 \quad (11.30)$$

For future reference, we define the problem

$$\boxed{\begin{array}{ll} \min\limits_{V,t,\tau} & t \\ \text{subject to} & (11.27) \\ & (11.29) \text{ or } (11.30) \\ & (11.17) \text{ or } (11.21) \end{array}} \quad (11.31)$$

Once again, we stress that the definition of the multipliers τ is rather loose.

11.4.2 Stability Constraints

Unfortunately, we are still stuck with the main flaw in the framework that prevents us from using the strong stability theory in Theorem 4.2. See Section 5.3.3 on how the use of merely sufficient semidefinite relaxations complicates the stability theory.

However, as in the previous chapters, we point out one possible strategy to obtain a control law with guaranteed stability. As before, we use a simple contraction approach.

Theorem 11.1
Assume there exist a linear feedback $u_k = Lx_k$ and matrices $P,\ S \succ 0$ such that

$$x_{k+1}^T P x_{k+1} - x_k^T P x_k \leq -x_k^T S x_k \quad \forall ||\Delta_k|| \leq 1 \tag{11.32}$$

Furthermore, $Lx \in \mathbb{U}\ \forall x \in \mathbb{E}_P$ and $\mathbb{E}_P \subseteq \mathbb{X}$. Appending the semidefinite program (11.31) with the contraction constraint $x_{k+1|k}^T P x_{k+1|k} - x_{k|k}^T P x_{k|k} \leq -x_k^T S x_k$ and using feedback predictions $\mathcal{L} = \oplus_1^N L$ guarantees asymptotic stability if $x_{0|0} \in \mathbb{E}_P$.

Proof The result follows by induction. Assume that the problem was feasible for $k-1$, and $x_{k-1|k-1} \in \mathbb{E}_P$. The contraction constraint then ensures that $x_{k|k} \in \mathbb{E}_P$. At time k, a feasible solution is $V = 0$. To see this, we recall that this choice gives us $x_{k+j+1|k} = (A+BL)x_{k+j|k} + Gp_{k+j|k}$. According to the assumptions, the ellipsoid $\mathbb{E}_P$ is invariant with this control law, hence all predictions are contained in $\mathbb{E}_P$. From the assumptions, we also know that all state constraints and control constraints are satisfied with this control. The contraction constraint is satisfied with $u_{k|k} = Lx_{k|k}$ according to the assumptions on P and L in (11.32). Finally, asymptotic stability follows immediately from the contraction constraint. □

Finding a pair L and P can be done using standard theory on robust linear state feedback. See Appendix 11.A for details.

The contraction constraint can be taken care of by first noting that the LFT model (11.1) and $u_{k|k} = Lx_{k|k} + v_{k|k}$, after eliminating $z_{k|k}$ and $p_{k|k}$, gives the following one-step prediction.

$$x_{k+1|k} = (A+BL)x_{k|k} + Bv_{k|k} + G\Delta_k(I - D_p\Delta_k)^{-1}((D_x + D_uL)x_{k|k} + D_uv_{k|k})$$

Using this together with a Schur complement on (11.32) transforms the contraction constraint to an uncertain LMI. (for notational purposes, introduce a variable $s_{k|k} = (D_x + D_uL)x_{k|k} + D_uv_{k|k}$)

$$\begin{pmatrix} x_{k|k}^T P x_{k|k} - x_{k|k}^T S x_{k|k} & ((A+BL)x_{k|k} + Bv_{k|k})^T \\ \star & P^{-1} \end{pmatrix} + \begin{pmatrix} 0 \\ G \end{pmatrix} \Delta_k(I - D_p\Delta_k)^{-1} \begin{pmatrix} s_{k|k} & 0 \end{pmatrix} + (\star) \succeq 0 \tag{11.33}$$

Since we only have one uncertainty, Theorem 3.5 can be used to obtain an equivalent condition. To this end, introduce a multiplier τ and the result is our contraction constraint written as an LMI in τ and $v_{k|k}$.

$$\begin{pmatrix} x_{k|k}^T P x_{k|k} - x_{k|k}^T S x_{k|k} & ((A+BL)x_{k|k} + Bv_{k|k})^T & s_{k|k}^T \\ \star & P^{-1} - \tau G^T G^T & -\tau G D_p^T \\ \star & \star & \tau(I - D_p D_p^T) \end{pmatrix} \succeq 0 \quad (11.34)$$

11.4.3 Optimized Terminal State Weight

The main drawback of the framework that we have is the lack of necessity in the semidefinite relaxations. This makes it hard to use Theorem 4.2 and related methods to obtain a controller with stability guarantees. However, nothing prevents us from taking a more pragmatic approach and use the ideas to create a controller without guaranteed stability[2], but with very good performance in practice. To this end, introduce a terminal state weight $x_{k+N|k}^T P x_{k+N|k}$ and an ellipsoidal terminal state domain $\mathbb{X}_T = \mathbb{E}_{W^{-1}}$ (the definition with W^{-1} simplifies the derivation).

$$\begin{array}{llcl} \min\limits_{u,W,P} \max\limits_{\Delta} & \multicolumn{3}{l}{\sum_{j=0}^{N-1} y_{k+j|k}^T Q y_{k+j|k} + u_{k+j|k}^T R u_{k+j|k} + x_{k+N|k}^T P x_{k+N|k}} \\ \text{subject to} & u_{k+j|k} & \in & \mathbb{U} \quad \forall \Delta \in \boldsymbol{\Delta} \\ & x_{k+j|k} & \in & \mathbb{X} \quad \forall \Delta \in \boldsymbol{\Delta} \\ & x_{k+N|k} & \in & \mathbb{X}_T \quad \forall \Delta \in \boldsymbol{\Delta} \\ & \Delta_{k+j|k} & \in & \boldsymbol{\Delta} \end{array} \quad (11.35)$$

For fixed P and W, the extension is straightforward. The quadratic state constraint is taken care of using a Schur complement and a semidefinite relaxation.

The case when P and W are decision variables in the optimization problem is more interesting. The idea is to optimize P and W so that $x_{k+N|k}^T P x_{k+N|k}$ is an upper bound on the worst-case infinite horizon cost when a linear state feedback controller is used inside the invariant terminal state domain $\mathbb{E}_{W^{-1}}$ (compare with Theorem 4.2 and the discussion in the end of Section 2.5.1).

Unfortunately, the problem above cannot be solved as stated with free P and W. Instead, we solve an approximation where we maximize with respect to Δ independently over the finite horizon performance measure and the terminal state weight[3]. This will of course be conservative, but it will allow us to use the basic ideas in (Kothare et al., 1996).

[2] The results can be used together with explicit contraction constraints to guarantee stability. The extension is straightforward but omitted for brevity. The important feature here is the incorporation of optimized terminal state weights.

[3] We replace $\max_x(f(x) + g(x))$ with $\max_x f(x) + \max_x g(x)$

Introducing a standard vectorized notation and epigraph formulation yields

$$
\boxed{\begin{array}{lrcll}
\min\limits_{U,t,\gamma} & t+\gamma & & & \\
\text{subject to} & Y^T\mathcal{Q}Y + U^T\mathcal{R}U & \leq & t & \forall\Delta \in \mathbf{\Delta}^N \\
 & x_{k+N|k}^T P x_{k+N|k} & \leq & \gamma & \forall\Delta \in \mathbf{\Delta}^N \\
 & x_{k+N|k}^T W^{-1} x_{k+N|k} & \leq & 1 & \forall\Delta \in \mathbf{\Delta}^N \\
 & U & \in & \mathbb{U}^N & \forall\Delta^N \in \mathbf{\Delta}^N \\
 & X & \in & \mathbb{X}^N & \forall\Delta^N \in \mathbf{\Delta}^N
\end{array}} \tag{11.36}
$$

The idea in (Kothare et al., 1996) is to assume that a linear state feedback controller $u_{k+j|k} = Lx_{k+j|k}$ is applied beyond the finite horizon[4]. The worst-case infinite horizon cost using this control law is bounded by $x_{k+N|k}^T P x_{k+N|k}$. Furthermore, all state and control constraints are assumed to hold in the terminal state domain $\mathbb{E}_{W^{-1}}$, and $\mathbb{E}_{W^{-1}}$ is invariant with respect to the linear feedback controller and the uncertainties.

Following the ideas in (Kothare et al., 1996) gives us a sufficient condition on P, W and L for this to hold. The result is a set of LMIs in the variables γ, W and K, and a scalar multiplier $\tau \in \mathbb{R}_+$

$$
\begin{pmatrix}
W & (AW+BK)^T & WC^T & K^T & (D_xW+D_uK)^T \\
\star & W-\tau GG^T & 0 & 0 & -\tau G D_p^T \\
\star & 0 & \gamma Q^{-1} & 0 & 0 \\
\star & 0 & 0 & \gamma R^{-1} & 0 \\
\star & \star & 0 & 0 & \tau(I-D_pD_p^T)
\end{pmatrix} \succeq 0 \tag{11.37a}
$$

$$
\begin{pmatrix}
((f_u)_i)^2 & (E_uK)_i \\
((E_uK)_i)^T & W
\end{pmatrix} \succeq 0 \tag{11.37b}
$$

$$
\begin{pmatrix}
((f_x)_i)^2 & (E_x)_i \\
((E_x)_i)^T & W
\end{pmatrix} \succeq 0 \tag{11.37c}
$$

The derivation follows easily from (Kothare et al., 1996) and can be found in Appendix 11.B

The proof relies upon the (conservative) parameterization $W = \gamma P^{-1}$. With this reduced degree of freedom, the bound on the infinite horizon cost and the terminal constraint are jointly taken care of with one uncertain LMI

$$
\begin{pmatrix}
1 & x_{k+N|k}^T \\
x_{k+N|k} & W
\end{pmatrix} \succeq 0 \quad \forall\Delta^N \in \mathbf{\Delta} \tag{11.38}
$$

A slight notational problem now is that $x_{k+N|k}$ is not defined in X. However, all we have to do is to define the matrices $\mathcal{A}$, $\mathcal{B}$ and so on for the horizon $N+1$, and extract the last n rows. Doing this gives us (compare notation with (11.11)).

$$
x_{k+N|k} = \tilde{x}_{k+N|k} + \Lambda_N \Delta^N (I - \Omega_N \Delta^N)^{-1} \Psi_N \tag{11.39}
$$

[4]Note though that the horizon was $N = 0$ in (Kothare et al., 1996)

The uncertain LMI (11.38) can now readily be written as an LMI with LFT uncertainty. Application of Theorem 3.5 gives a new set of multipliers $\tau \in \mathbb{R}_+^{N+1}$, related matrices $\mathcal{S}$ and $\mathcal{T}$, and a sufficient condition for (11.38) to hold.

$$\begin{pmatrix} 1 & \tilde{x}_{k+N|k}^T & \Psi_N^T \\ \star & W - \Lambda_N \mathcal{S} \Lambda_N^T & -\Lambda_N \mathcal{S} \Omega_N^T \\ \star & \star & \mathcal{T} - \Omega_N \mathcal{S} \Omega_N^T \end{pmatrix} \succeq 0 \tag{11.40}$$

To summarize, we have the following problem.

$$\boxed{\begin{array}{ll} \min\limits_{V,t,\tau,\gamma,W,K} & t + \gamma \\ \text{subject to} & (11.15) \text{ or } (11.27) \\ & (11.37) \\ & (11.40) \\ & (11.17) \text{ or } (11.29) \end{array}} \tag{11.41}$$

Note that the methods used to derive the LMIs for the terminal state cost and terminal state weight are based on ellipsoidal calculus. As stated earlier, ellipsoidal calculus is best suited when we have symmetric constraints. For that reason, only those alternatives are stated in the optimization problem. The alternatives in the problem correspond to whether feedback predictions are used or not. Notice also that we have only included one set of multipliers τ. Of course, different multipliers can be used for the performance constraint, state and control constraints, the terminal state constraint and the LMIs related to W and K. Again, this is a trade-off between control performance and computational efficiency.

11.5 Simulation Results

The example is taken from (Schuurmans and Rossiter, 2000)

Example 11.1 The system is given by

$$\begin{aligned} x_{k+1} &= \begin{pmatrix} 1 & 0.1 \\ 0 & 1.1 + 0.1\delta_k \end{pmatrix} + \begin{pmatrix} 0 \\ 0.0787 \end{pmatrix} u_k, \quad |\delta_k| \leq 1 \\ y_k &= \begin{pmatrix} 1 & 0 \end{pmatrix} \end{aligned}$$

This can be written as an LFT with

$$A = \begin{pmatrix} 1 & 0.1 \\ 0 & 1.1 \end{pmatrix}, \; B = \begin{pmatrix} 0 \\ 0.0787 \end{pmatrix}, \; G = \begin{pmatrix} 0 \\ 0.1 \end{pmatrix}, \; D_x = 1, \; D_u = 0, \; D_p = 0$$

The task is to design a minimax controller with $Q = 1$ and $R = 0.01$, subject to the control constraint $|u_k| \leq 2$. The initial condition is $x_0 = \begin{pmatrix} 1 & 0 \end{pmatrix}^T$

As a first experiment, we implement the minimax controller (11.41) without any feedback predictions, for horizons $N = 0$, 1, 3 and 5.

To compare our results with those reported in (Schuurmans and Rossiter, 2000), we use the same uncertainty realization, $\delta_k = -\sin(0.1k)$. The system is simulated for $k = 0, \dots, 50$ and the accumulated cost $\sum_{k=0}^{50} y_k^T Q y_k + u_k^T R u_k$ is calculated. Note that the case $N = 0$ corresponds to the controller in (Kothare et al., 1996). The CPU-time used for this controller will be used as a reference when we evaluate the computational efficiency of our semidefinite relaxations. The results are given in Table 11.1 and Figure 11.1.

Horizon	Accumulated cost	Relative CPU-time
0	18.3	1
1	10.8	1.5
3	9.0	1.9
5	12.6	2.6

Table 11.1: The table shows the performance of the minimax controller for a number of different horizons. Note that the performance deteriorates for $N = 5$. The reason is the absence of feedback predictions.

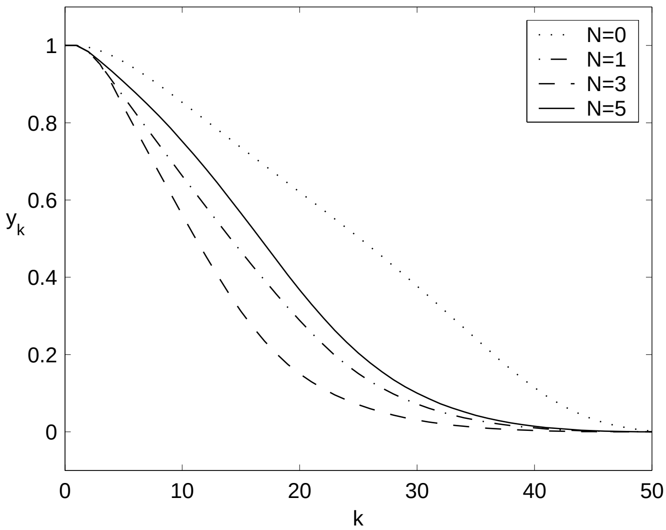

Figure 11.1: Step-responses for minimax controller without feedback predictions. Using a finite horizon cost makes the controller less conservative, but the lack of feedback predictions seems to limit the effectiveness of using a long horizon.

Already a horizon $N = 1$ gives a substantial performance improvement compared to the case $N = 0$. From a computational point of view, we see that the overhead of using a longer horizon is modest. The scheme in (Schuurmans and Rossiter, 2000), based on enumeration, was reported to give a 40-fold increase in CPU-time for the case $N = 5$, compared to $N = 0$. This sounds reasonable since there are 2^5 different uncertainty realizations. The controller in (Schuurmans and Rossiter, 2000) obtained an accumulated cost of 8.8 for $N = 5$ (with feedback predictions).

Our poor performance for $N = 5$ is due to the absence of feedback predictions. As a second test, we employ feedback predictions, with L chosen as an LQ controller calculated with $Q = 1$, $R = 0.01$, and the nominal model A.

When we use feedback predictions, the control constraints are mapped into state constraints which are taken care of using the semidefinite relaxations (11.30). There is a trade-off between quality in the relaxations, and the number of variables we introduce. One approach is to use the same multipliers on all control constraints. This scheme will be called the cheap version. Another approach is to use different multipliers on each constraint. We will denote this approach the full version.

With feedback predictions, a longer horizon can be used. We perform simulations with $N = 1$, 3, 5, 10 and 15, using the two approaches described above to define the multipliers.

Horizon	Accumulated cost	Relative computation time
1 (cheap)	10.7	1.6
3 (cheap)	9.1	1.9
5 (cheap)	9.0	2.6
10 (cheap)	9.6	7.4
15 (cheap)	10.1	23.1
1 (full)	11.0	1.6
3 (full)	9.0	2.0
5 (full)	8.7	2.8
10 (full)	8.5	8.7
15 (full)	8.5	22.6

Table 11.2: The table shows the performance of the minimax controllers for a number of different horizons when feedback predictions are used. The label cheap means that one set of multipliers is used for all semidefinite relaxations of control constraints, whereas full denotes that separate multipliers are introduced for each constraints.

From Table 11.2, we see that the cheap parameterization is rather competitive in terms of performance. However, surprisingly, the number of variables used for the semidefinite relaxations does not seem to be that important. The CPU-time used is almost the same for the two approaches. Note however that this effect most likely depends on which solver is used.

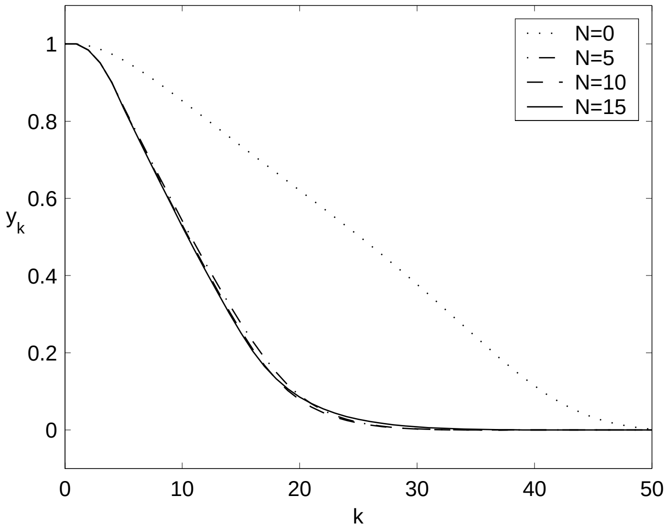

Figure 11.2: Minimax controllers using feedback predictions. The use of a finite horizon cost improves the performance, and the feedback predictions are successful in reducing conservativeness, hence making long horizons applicable.

Appendix

11.A Contraction Constraint

Inserting $u_k = Lx_k$ and eliminating the variables p_k and z_k gives

$$x_{k+1} = ((A + BL) + G(I - \Delta_k D_p)^{-1}\Delta_k(D_x + D_u L))x_k \tag{11.A.42}$$

The contraction constraint (11.32) thus holds if

$$((A + BL) + G(I - \Delta_k D_p)^{-1}\Delta_k(D_x + D_u L))^T P(\star) - P \preceq -S \tag{11.A.43}$$

Multiply from left and right with $W = P^{-1}$, define $K = LP^{-1}$, and apply a Schur complement.

$$\begin{pmatrix} W & (AW + BK + G(I - \Delta_k D_p)^{-1}\Delta_k(D_x W + D_u K))^T & W \\ \star & W & 0 \\ \star & 0 & S^{-1} \end{pmatrix} \succeq 0 \tag{11.A.44a}$$

Application of Theorem 3.5 gives an LMI in W, K and τ.

$$\begin{pmatrix} W & (AW + BK)^T & W & (D_x W + D_u K)^T \\ \star & W - \tau G G^T & 0 & -\tau G D_p^T \\ \star & 0 & S^{-1} & 0 \\ \star & \star & 0 & \tau(I - D_p D_p^T) \end{pmatrix} \succeq 0 \tag{11.A.45a}$$

Sufficient conditions for $Lx \in \mathbb{U}$ and $\mathbb{E}_P \in \mathbb{X}$ are (see Appendix 5.A)

$$\begin{pmatrix} ((f_u)_i)^2 & (E_u K)_i \\ ((E_u K)_i)^T & W \end{pmatrix} \succeq 0 \tag{11.A.46a}$$

$$\begin{pmatrix} ((f_x)_i)^2 & (E_x)_i \\ ((E_x)_i)^T & W \end{pmatrix} \succeq 0 \tag{11.A.46b}$$

The LMIs above can be used to optimize, e.g., the volume of the ellipsoid $\mathbb{E}_P$. See Appendix 5.A.

11.B Terminal State Weight and Constraints

The problem is to find matrices $W,\ P \succ 0$ and a matrix L such that $x_k^T P x_k \leq \gamma$ is an upper bound on $\sum_{j=0}^{\infty} y_{k+j}^T Q y_{k+j} + u_{k+j}^T R u_{k+j}$ when the control law $u_k = Lx_k$ is used. Moreover, the control law should be feasible with respect to all constraints in the invariant ellipsoid $\mathbb{E}_{W^{-1}} = \{x : x^T W^{-1} x \leq 1\}$, i.e., $x_{k+1} \in \mathbb{E}_{W^{-1}} \forall x_k \in \mathbb{E}_{W^{-1}}$, $Lx \in \mathbb{U}\ \forall x \in \mathbb{E}_{W^{-1}}$ and $\mathbb{E}_{W^{-1}} \subseteq \mathbb{X}$.

The function $x_k^T P x_k$ is an upper bound on the worst-case infinite horizon cost if

$$x_{k+1}^T P x_{k+1} - x_k^T P x_k \leq -y_k^T Q y_k - u_k^T R u_k \quad \forall ||\Delta_k|| \leq 1 \tag{11.B.47}$$

Use (11.A.42) and this constraint is equivalent to the following matrix inequality.

$$((A + BL) + G(I - \Delta_k D_p)^{-1} \Delta_k (D_x + D_u L))^T P(\star) - P \preceq -C^T Q C - L^T R L$$

Multiply from left and right with γP^{-1}. Introduce the (reduced degree of freedom) parameterization $W = \gamma P^{-1}$ and define $K = LW$. A Schur complement yields

$$\begin{pmatrix} W & (AW + BK + G(I - \Delta_k D_p)^{-1} \Delta_k (D_x W + D_u K))^T & WC^T & K^T \\ \star & W & 0 & 0 \\ \star & 0 & \gamma Q^{-1} & 0 \\ \star & 0 & 0 & \gamma R^{-1} \end{pmatrix} \succeq 0 \tag{11.B.48a}$$

Application of Theorem 3.5 gives an LMI in W, K, γ and τ.

$$\begin{pmatrix} W & (AW + BK)^T & WC^T & K^T & (D_x W + D_u K)^T \\ \star & W - \tau G G^T & 0 & 0 & -\tau G D_p^T \\ \star & 0 & \gamma Q^{-1} & 0 & 0 \\ \star & 0 & 0 & \gamma R^{-1} & 0 \\ \star & \star & 0 & 0 & \tau(I - D_p D_p^T) \end{pmatrix} \succeq 0 \tag{11.B.49a}$$

Invariance of $\mathbb{E}_{W^{-1}}$ follows immediately from the contraction constraint (11.B.47).

Finally, constraint satisfaction $E_u L x \leq f_u$ and $E_x x \leq f_x$ in $\mathbb{E}_{W^{-1}}$ follow from Appendix 5.A.

$$\begin{pmatrix} ((f_u)_i)^2 & (E_u K)_i \\ ((E_u K)_i)^T & W \end{pmatrix} \succeq 0 \tag{11.B.50a}$$

$$\begin{pmatrix} ((f_x)_i)^2 & (E_x)_i \\ ((E_x)_i)^T & W \end{pmatrix} \succeq 0 \tag{11.B.50b}$$

12

Summary and Conclusions on minimax MPC

The previous chapters have introduced a number of approaches to incorporate uncertainty in MPC. It has been advocated that semidefinite relaxations enable a general framework to cope with finite horizon minimax MPC problems with quadratic performance measures. To summarize, we have

- a unified framework for a number of uncertainty models.
- methods to deal with a quadratic performance measure.
- algorithms with polynomial time complexity.

On the other hand, the proposed algorithms come with a number of drawbacks and problems. These include

- a weak stability theory.
- solutions require semidefinite programming.
- state estimation a patchwork.

Nevertheless, it is our hope that the material in the thesis has pushed the frontier on minimax MPC one step further.

12.1 Future Work and Extensions

The material in this thesis is only a first study on how semidefinite relaxations can be used to deal with uncertainty in MPC. A number of extensions and possible future research directions are listed below.

12.1.1 State Estimation

The state estimation procedure in Chapter 6 can be regarded as a patchwork in some sense. The problem is that it solves the estimation problem point-wise in time. A more natural way to incorporate the estimation problem in the minimax MPC framework is to solve the dual estimation problem over a finite horizon backwards in time, along the lines of (Michalska and Mayne, 1995) and (Rao et al., 2000).

12.1.2 Quality of Relaxations

Numerical experiments have indicated that the semidefinite relaxations are efficient and give an upper bound on the worst-case cost close to the true value. There are results available on the quality of semidefinite relaxations (Ben-Tal and Nemirovski, 1998), and it would be interesting to see what these results, and similar calculations, can tell about our relaxations.

12.1.3 A Gain-scheduling Perspective

The step from robust control to robust gain-scheduling is not far. The difference is essentially that a gain-scheduled controller can measure the uncertainty on-line and compensate for it (Rugh and Shamma, 2000).

Extending the material in Chapter 5 and 7 to incorporate measurable disturbances is straightforward. In Chapter 5, feedback predictions are changed to $u_{k+j|k} = Lx_{k+j|k} + Mw_{k+j|k} + v_{k+j|k}$ for some suitably defined feedforward matrix M. The material in Chapter 7 is easily extended to cope with measurable disturbances by using the parameterization

$$U = \mathcal{L}W + V \tag{12.1a}$$

$$\mathcal{L} = \begin{pmatrix} L_{00} & 0 & \dots & 0 \\ L_{10} & L_{11} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{(N-1)0} & L_{(N-1)1} & \dots & L_{(N-1)(N-1)} \end{pmatrix} \tag{12.1b}$$

Of course, the semidefinite relaxation is only performed with respect to the uncertainties $\left(w_{k+1|k}, \dots, w_{k+N-1|k}\right)$ since $w_{k|k}$ is measured. Incorporating measurable disturbances in Chapter 5 and 7 can be interpreted as a robust feedforward structure.

Exploiting measured uncertainties in Chapter 11 gives a gain-scheduled controller. Consider the following parameterization of the control law in Chapter 11.

$$u_{k+j|k} = Lx_{k+j|k} + Mp_{k+j|k} + v_{k+j|k} \tag{12.2}$$

The new predictions are given by ($\mathcal{M} = \oplus_{j=1}^{N} M$)

$$\begin{aligned} X &= (\mathcal{A} + \mathcal{B}\mathcal{L})X + \mathcal{B}V + (\mathcal{G} + \mathcal{B}\mathcal{M})P + b && (12.3a)\\ Z &= (\mathcal{D}_x + \mathcal{D}_u\mathcal{L})X + \mathcal{D}_uV + (\mathcal{D}_p + \mathcal{D}_u\mathcal{M})P && (12.3b)\\ P &= \Delta^N Z && (12.3c) \end{aligned}$$

This system of equations can easily be solved and the result is an LFT description of X that can be used immediately in the framework of Chapter 11.

A more advanced gain-scheduled controller in an LFT framework typically looks like (Packard, 1994)

$$\begin{aligned} u_k &= Lx_k + M\tilde{p}_k && (12.4a)\\ \tilde{z}_k &= Fx_k + K\tilde{p}_k && (12.4b)\\ \tilde{p}_k &= \Delta_k\tilde{z}_k && (12.4c) \end{aligned}$$

Incorporating this in our framework would be very interesting. Since gain-scheduled controllers in an LFT framework are used to control uncertain nonlinear systems, it would allow us to develop gain-scheduled MPC for constrained uncertain nonlinear systems.

12.1.4 Uncertainty Models

For simplicity, most uncertainties in this thesis have been assumed unstructured. Extending the ideas to structured uncertainty models is straightforward. Consider for example the models in Chapter 10 and Chapter 11. A structured model of Δ can be included by defining matrices $\mathcal{S}$ and $\mathcal{T}$ as $\mathcal{S} = \oplus_{j=1}^{N} S_j$ and $\mathcal{T} = \oplus_{j=1}^{N} T_j$, where the matrix multipliers S and T commute with the uncertainty (El Ghaoui et al., 1998)

$$S\Delta = \Delta T \quad \forall \Delta \in \mathbf{\Delta} \tag{12.5}$$

A similar extension involves time-invariant models. This can also be addressed, since time-invariance essentially defines a structure on Δ^N. The details are omitted for brevity.

The only problem with these extensions is that many results in the thesis exploit the unstructured property of the uncertainties. This includes efficient incorporation of robust state constraints in Chapters 5, 6 and 10, and the stability constraints in Chapter 5, 10 and 11.

12.1.5 Off-line Calculation of Multipliers

It can easily be shown that the semidefinite relaxations in this thesis are quadratic and second order cone programs for fixed multipliers τ (see Section 10.2.1 for an example). The intriguing question is if this can be exploited in any way. One approach could be to calculate suitable multipliers off-line. Of course, this leads to a hard off-line problem: how to find suitable multipliers that work well for all states?

12.1.6 Nominal vs. Worst-case Performance

Control is inherently a multi-objective optimization problem. In our setting, the conflicting performance measures are nominal performance and worst-case performance.

The standard way to obtain a (Pareto optimal) solution is to scalarize the multi-objective problem (Boyd and Vandenberghe, 2002). In our case, this means that we introduce a weight $\lambda \in [0, 1]$ and minimize a weighted sum of the nominal and worst-case performance.

$$\begin{array}{ll} \min\limits_{t_{wc}, t_{nom}, u} & \lambda t_{wc} + (1-\lambda) t_{nom} \\ \text{subject to} & \begin{array}{rcl} \sum_{j=0}^{N-1} y_{k+j|k}^T Q y_{k+j|k} + u_{k+j|k}^T R u_{k+j|k} & \leq & t_{wc} \quad \forall \Delta \in \boldsymbol{\Delta} \\ \sum_{j=0}^{N-1} y_{k+j|k}^T Q y_{k+j|k} + u_{k+j|k}^T R u_{k+j|k} & \leq & t_{nom} \quad (\text{For } \Delta = 0) \\ u_{k+j|k} & \in & \mathbb{U} \quad \forall \Delta \in \boldsymbol{\Delta} \\ x_{k+j|k} & \in & \mathbb{X} \quad \forall \Delta \in \boldsymbol{\Delta} \end{array} \end{array}$$

This problem can readily be solved using the methods introduced in this thesis. The only difference is the nominal performance constraint. Since this is a quadratic constraint, it can easily be written as a second order cone constraint, and be included in any of the developed algorithms.

Figure 12.1 shows a trade-off curve for the MPC problem in Example 10.2. The curve is calculated using 100 different values on λ ranging from 0 to 1. The figure shows that a worst-case minimax controller reduces the worst-case cost with a factor of 25. The price paid is a 3-fold increase in nominal cost. The important thing to see in the figure however is the steep decrease in worst-case cost to the left in the figure. Allowing a nominal performance loss of, say, 30 percent yields a 5-fold decrease in worst-case cost.

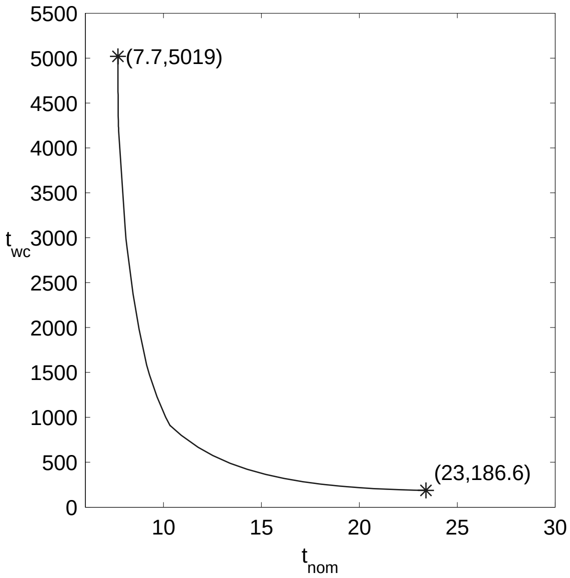

Figure 12.1: Trade-off curve between nominal cost and worst-case cost.

Part III

Nonlinear MPC

13

Nonlinear MPC

This chapter changes focus entirely compared to the first part of the thesis. While the previous chapters have been devoted to robustness in uncertain constrained discrete-time linear systems, we now consider nominal stability in unconstrained nonlinear continuous-time systems instead. Extending the basic ideas and theories in linear MPC to nonlinear systems is straightforward conceptually, but a lot of work is required to obtain actual constructive results that can be used in an implementation.

The main result in this chapter is an MPC algorithm with guaranteed stability for unconstrained nonlinear systems. The algorithm is an extension of methods that use a control Lyapunov function to define the terminal state weight in MPC algorithms with guaranteed stability (Chen, 1997; Primbs, 1999; Jadbabaie, 2000; Fontes, 2001). The novelty lies in a weaker assumption on the set of admissible control Lyapunov functions.

The proposed algorithm can be extended to incorporate control and state constraints, but the central idea is much more clear when applied to unconstrained systems. Note that MPC for unconstrained systems is an essentially irrelevant topic in the linear case, since the LQ controller solves the problem. This is not the case for nonlinear systems since no explicit solutions are available for the underlying optimal control problem.

13.1 Nonlinear MPC

The class of systems addressed in this chapter is continuous-time unconstrained nonlinear systems.

$$\dot{x}(t) = f(x(t), u(t)) \tag{13.1}$$

Technical details on $f(x(t), u(t))$ are put aside for the moment, and will instead be introduced as assumptions in the forthcoming stability theorem.

The underlying goal in MPC is to find a feedback law that minimizes an infinite horizon performance measure.

$$J_\infty = \int_0^\infty \ell(x(\tau), u(\tau))\, d\tau \tag{13.2}$$

Finding the optimal feedback law is in general an intractable problem since it requires the solution of a nonlinear partial differential equation, the Hamilton-Jacobi-Bellman (HJB) equation. It was early recognized (Lee and Markus, 1968) that a natural way to overcome this is to resort to a numerical solution, i.e., to continuously minimize the integral on-line. In order to do this, the infinite horizon has to be truncated. The performance measure used on-line is

$$J(x(t), u_t) = \int_t^{t+T} \ell(x_t(\tau), u_t(\tau))\, d\tau \tag{13.3}$$

In (13.3), we introduced the sought control trajectory $u_t(\tau), \tau \in [t, t+T]$, defined at time t. The control trajectory gives the simulated state trajectory $x_t(\tau)$, satisfying system dynamics $\dot{x}_t(\tau) = f(x_t(\tau), u_t(\tau))$ and the initial condition $x_t(t) = x(t)$.

The feedback law is defined in standard MPC manner as[1]

$$u(t) = u_t^*(t) \tag{13.4a}$$

$$u_t^* = \arg\min_{u_t} J(x(t), u_t) \tag{13.4b}$$

Although this might seem like a simple and sound way to define a feedback law, it turns out that the control law can destabilize the system, just as in the linear discrete-time case. Fortunately, there are ways to guarantee stability.

13.2 Stability in Nonlinear MPC

The first step towards stability is to add a terminal state weight.

$$J(x(t), u_t) = \int_t^{t+T} \ell(x_t(\tau), u_t(\tau))\, d\tau + V(x_t(t+T)) \tag{13.5}$$

It is straightforward to derive a sufficient condition on the weight $V(x)$ to guarantee stability. The stability results in Theorem 2.1 can be extended to nonlinear systems

[1] A $*$ will be used throughout this chapter to denote optimality in an optimal control problem.

and a typical result is the following constraints on $V(x)$ and $\ell(x,u)$ (some technical assumptions are omitted for brevity). See, e.g., (Mayne et al., 2000) for details and a proof.

Theorem 13.1 (Stabilizing MPC)
Suppose the following assumptions hold for a nominal controller $k(x)$, a terminal state weight $V(x)$ and performance measure $\ell(x,u)$.

A1. $\dot{x}(t) = f(x(t), u(t))$

A2. $V(0) = 0,\ V(x) > 0\ \forall\ x \neq 0$

A3. $\ell(x,u) > 0\ \forall\ (x,u) \neq 0$

A4. $V_x f(x, k(x)) \leq -\ell(x, k(x))$

An MPC controller based on the performance measure (13.5) will guarantee asymptotic stability.

The theorem essentially states that the function $V(x)$ should be an upper bound on the tail of the truncated integral. The constraint in Assumption A4,

$$V_x f(x, k(x)) \leq -\ell(x, k(x)) \tag{13.6}$$

means that the function $V(x)$ is a Lyapunov function for the system $\dot{x} = f(x, k(x))$. Integrating this expression yields

$$\int_t^\infty \ell(x(\tau), u(\tau))\, d\tau \leq V(x(t)) \tag{13.7}$$

A related interpretation can be given in terms of the value (or optimal return) function of the infinite horizon problem. The optimal feedback law is given by the solution to the HJB equation. This control law is recovered in MPC if the terminal state weight is identical to the value function of the infinite horizon problem. For this to hold, the HJB equation states that (see any text on optimal control theory, e.g., (Lee and Markus, 1968))

$$\min_{k(x(t))} \ell(x(t), k(x(t))) + V_x f(x(t), k(x(t))) = 0 \tag{13.8}$$

Calculating the optimal terminal state weight $V(x)$ thus requires the solution of a nonlinear partial differential equation. The *a priori* chosen terminal state weight $V(x(t))$ replaces the equality constraint in the HJB equation by an inequality, and uses a suboptimal controller $k(x)$ instead. An in-depth look at the connections between the terminal state weight in MPC and the value function of the infinite horizon problem can be found in (Primbs, 1999).

Theorem 13.1 is deceptively simple. The MPC problem defined by the theorem is almost as hard as the original infinite horizon problem, since we have to find the controller $k(x)$ and the terminal state weight $V(x)$.

Methods to derive suitable $k(x)$ and $V(x)$ are essential for a constructive stability theory in nonlinear MPC. Locally admissible $k(x)$ and $V(x)$ are relatively easy to construct, and this motivates an extension of Theorem 13.1 (Mayne et al., 2000).

Theorem 13.2 (Stabilizing MPC with terminal state constraints)
Suppose the following assumptions hold for a nominal controller $k(x)$, a terminal state weight $V(x)$, a performance measure $\ell(x, u)$ and the terminal state domain $\mathbb{X}_T = \{x : V(x) \leq \gamma\}$.

A1. $\dot{x}(t) = f(x(t), u(t))$

A2. $V(0) = 0,\ V(x) \geq 0\ \forall\ x \neq 0$

A3. $\ell(x, u) > 0\ \forall\ (x, u) \neq 0$

A4. $V_x f(x, k(x)) \leq -\ell(x, k(x))\ \forall x \in \mathbb{X}_T$

An MPC controller based on the performance measure (13.5) and a terminal state constraint $x_t(t+T) \in \mathbb{X}_T$ guarantees asymptotic stability if the problem is initially feasible.

This theorem can be used to categorize most stability results in nonlinear MPC. Let us briefly review the central ideas in the literature. For a more thorough survey, the reader is referred to (Mayne et al., 2000).

A simple and elegant approach to guarantee stability is to add a terminal state equality constraint $x_t^*(t + T) = 0$. This can be interpreted as a terminal state weight infinite everywhere except in the origin. This approach was proposed, in the context of linear continuous-time systems, in (Kwon and Pearson, 1977) and later generalized to nonlinear systems and thoroughly analyzed in (Keerthi and Gilbert, 1988) (for discrete-time systems) and (Mayne and Michalska, 1990) (continuous-time systems).

The numerically complicating terminal state equality was relaxed in (Michalska and Mayne, 1993) and a dual-mode scheme was proposed. The idea was to calculate, off-line, a linear controller that asymptotically stabilized the nonlinear system locally in an ellipsoid $\mathbb{E}_P$. The ellipsoid was used for two purposes. To begin with, it defined a terminal state constraint in the MPC controller. By using the optimal finite horizon cost as a Lyapunov function, it was possible to show that the state eventually reaches $\mathbb{E}_P$ when the MPC controller is used. By switching to the linear control law when $\mathbb{E}_P$ is reached, asymptotic stability follows immediately.

Actually switching to the linear feedback controller when the terminal state domain is reached is not necessary, and the theoretical trick needed to show this is a terminal state weight (Parsini and Zoppoli, 1995; Chen, 1997). The main idea, generalizing ideas in (Michalska and Mayne, 1993; Rawlings and Muske, 1993) was to derive an ellipsoidal terminal state constraint $x_t^*(t + T) \in \mathbb{E}_P$, a linear controller $k(x)$ and a quadratic terminal state weight $\alpha x_t(t+T)Px_t(t+T)$ satisfying Assumption A4. The design was based on the linearized system. A more general scheme using a nonlinear controller $k(x)$ and a possibly non-quadratic terminal state weight satisfying Assumption A4 was proposed in (Chen, 1997).

The ideas with a nonlinear nominal controller $k(x)$ in (Chen, 1997) have recently been generalized and studied in a control Lyapunov function framework (Primbs, 1999; Jadbabaie, 2000; Fontes, 2001). However, in the design procedure one still has to find a terminal state weight globally satisfying (13.6), or resort to arguments based on linearizations and/or additional stability constraints.

13.3 Main Result

We see from the quick review above that there are three main ingredients in the available approaches; assuming knowledge of a terminal state weight globally satisfying (13.6), using arguments based on the linearized system, and adding stabilizing constraints. The goal in this chapter is to develop a method that use neither of these ingredients.

Our terminal state weight will be defined using a control Lyapunov function. (Krstić et al., 1995).

Definition 13.1 (Control Lyapunov function)
A smooth positive definite and radially unbounded function $V : \mathbb{R}^n \to \mathbb{R}_+$ is called a control Lyapunov function (CLF) if there exist a control law $u = k(x)$ such that

$$\dot{V}(x, k(x)) = V_x f(x, k(x)) < 0, \quad \forall x \neq 0$$

Notice the small but absolutely crucial difference in the requirement on the CLF compared to the condition (13.6). Of course, assuming knowledge of a CLF is an essential limitation, but with recent developments in nonlinear control synthesis, there are many methods available to find such a function. Examples include backstepping, feedforwarding, feedback linearization and physically motivated designs. See, e.g., (Krstić et al., 1995) and (Sepulchre et al., 1997).

The constraint (13.6) shows up in stability analysis of MPC as a sufficient condition. To be precise, the condition is

$$\dot{V}(x_t^*(t+T), k(x_t^*(t+T))) + \ell(x_t^*(t+T), k(x_t^*(t+T))) \leq 0 \tag{13.9}$$

The algorithm in this chapter is based on the idea that this condition can be achieved, intuitively, by multiplying ℓ with some sufficiently small constant λ, since we know from the design that $\dot{V}(x, k(x)) < 0$ and $\ell(x, k(x)) > 0$. Use the scaled performance measure to obtain a modified constraint.

$$\dot{V}(x_t^*(t+T), k(x_t^*(t+T))) + \lambda\ell(x_t^*(t+T), k(x_t^*(t+T))) \leq 0 \tag{13.10}$$

If (13.10) not holds, λ should be decreased. This motivates introduction of a new variable $\lambda(t)$, which can be thought of as a state in the MPC controller, with dynamics

$$\dot{\lambda}(t) \sim -\lambda(t)\ell(x_t^*(t+T), k(x_t^*(t+T))) - \dot{V}(x_t^*(t+T), k(x_t^*(t+T))) \tag{13.11}$$

The finite horizon cost used in the MPC controller is changed to

$$J(x(t), u_t, \lambda(t)) = \lambda(t) \int_t^{t+T} \ell(x_t(\tau), u_t(\tau))\, d\tau + V(x_t(t+T)) \tag{13.12}$$

The problem is now to find an update-law for $\lambda(t)$ that actually guarantees stability of the closed-loop system. To do this, we first introduce some assumptions.

A1. $\dot{x}(t) = f(x(t), u(t))$, $f(0,0) = 0$, $f(x,u)$ continuously differentiable.

A2. $\ell(x,u) > 0\, \forall (x,u) \neq 0$, $\ell(0,0) = 0$. Furthermore, $\ell(x,u)$ is twice continuously differentiable with $\ell_{xx} \succ 0$, $\ell_{uu} \succ 0$ and $\ell_{xu} = 0$.

A3. $V(x)$ is a global CLF with the continuously differentiable controller $k(x)$. The linearization $\dot{x} = f(x, k(x))$ is asymptotically stable and $V_{xx} \succ 0$.

A4. $\int_t^{t+T} \ell(x_t(\tau), u_t(\tau))\, d\tau \to \infty$ for any $u_t(\tau)$ when $||x(t)|| \to \infty$.

A5. There exist a minimizing argument u_t^* of (13.12) for all $x(t)$ and $\lambda(t) > 0$.

A6. $J(x(t), u_t^*, \lambda(t))$ is continuously differentiable w.r.t $x(t)$ and $\lambda(t)$.

The assumptions stated are those needed to use $J(x(t), u_t^*, \lambda(t))$ as a Lyapunov function. Assumption A5 on existence of a solution implicitly imposes additional constraints on $f(x,u)$, $\ell(x,u)$ and $V(x)$. Typical results can be found in, e.g., Theorem 6.2 in (Berkovitz, 1974). Admittedly, Assumption A6 is hard to verify. The reason for incorporating this assumption is to obtain an intuitive result using standard Lyapunov theory. For additional discussion on the assumptions, see the proof of Theorem 13.3 and remarks in the appendix.

With the introduced assumptions, we state the main result of this chapter.

Theorem 13.3 (Stabilizing MPC using control Lyapunov functions) *Suppose (A1-A6) are satisfied and $\lambda(0) > 0$. Then the following MPC scheme is globally asymptotically stabilizing*

$$\begin{aligned} u(t) &= u_t^*(t) \\ u_t^* &= \arg\min_{u_t} J(x(t), u_t, \lambda(t)) \\ \dot{\lambda}(t) &= \frac{-\lambda(t)\ell(x_t^*(t+T), k(x_t^*(t+T))) - \dot{V}(x_t^*(t+T), k(x_t^*(t+T)))}{\int_t^{t+T} \ell(x_t^*(\tau), u_t^*(\tau))\, d\tau} \end{aligned}$$

Proof The proof is rather elaborate and is therefore given in Appendix 13.A □

Notice that the complexity of the on-line optimal control problem is the same as for the original problem, i.e., stability is not achieved at the expense of a more complicated optimization problem. Moreover, the right-hand side of the differential equation governing $\lambda(t)$ is easily calculated since both the numerator and denominator can be obtained from the solution of the optimization problem.

Important to realize is that the MPC controller is entirely based on nonlinear components. Local assumptions on $f(x,u)$, $k(x)$ and $V(x)$ are not used in the synthesis, i.e., in the definition of the on-line optimal control problem, but only used in the proof of Theorem 13.3. The local assumptions on the Taylor expansions make the proof more constructive and easier to interpret, but can actually be dispensed with altogether, as described in Remark 13.A.2.

The novelty in the theorem is the fact that we are able to construct a stabilizing MPC controller using the condition $\dot{V}(x,k(x)) < 0$, instead of the standard assumption $\dot{V}(x,k(x)) \leq -\ell(x,k(x)$. Finding a tuple $k(x)$ and $V(x)$ satisfying the former condition is a much easier problem, since it holds for any pair $(k(x),V(x))$ designed using control Lyapunov functions.

For the sake of completeness we extend the theorem to systems with constraints $u(t) \in \mathbb{U}$ and $x(t) \in \mathbb{X}$.

Corollary 13.1 (Stabilizing MPC using control Lyapunov functions) *Suppose (A1-A6) holds. Furthermore, assume there exist a constant $\gamma > 0$ such that $\mathbb{X}_T = \{x : V(x) \leq \gamma\} \subseteq \mathbb{X}$ and $k(x) \in \mathbb{U}\ \forall x \in \mathbb{X}_T$. Then the following MPC scheme is globally asymptotically stabilizing if it is initially feasible and $\lambda(0) > 0$*

$$\begin{aligned} u(t) &= u_t^*(t) \\ u_t^* &= \arg \min_{u_t \in \mathbb{U}, x_t \in \mathbb{X}, x_t(t+T) \in \mathbb{X}_T} J(x(t), u_t, \lambda(t)) \\ \dot{\lambda}(t) &= \frac{-\lambda(t)\ell(x_t^*(t+T), k(x_t^*(t+T))) - \dot{V}(x_t^*(t+T), k(x_t^*(t+T)))}{\int_t^{t+T} \ell(x_t^*(\tau), u_t^*(\tau))\, d\tau} \end{aligned}$$

Proof See Appendix 13.B. □

This result is not by far as useful as Theorem 13.3 since it requires calculation of the parameter γ to construct the terminal state domain. This is a nonconvex problem in the general case. Moreover, the theorem adds a terminal state constraint to the optimization problem, making it harder to solve. Adding this artificial constraint gives a results that essentially recovers a standard approach, such as (Chen, 1997). The only difference is the dynamic weight $\lambda(t)$ and the weaker assumption on $V(x)$.

Before we proceed, we should point out that optimality is not required for the stability proofs to hold. The proofs are based on an improvement property that essentially says that it is trivial to construct a feasible solution, and this feasible solution ensures asymptotic stability. This is a standard feature for the majority of stability proofs in MPC.

13.4 Simulation Results

The main result of this chapter is the design approach defined by Theorem 13.3. Evaluating the performance of this approach is almost an impossible task, since the performance depends on many ingredients, such as the nonlinear system $f(x, u)$, the nominal control law $k(x)$, the terminal state weight $V(x)$ and the prediction horizon T. Hence, no general conclusion can be drawn from the single experiment conducted below, but it exemplifies how $k(x)$ and $V(x)$ can be derived, and how the state $\lambda(t)$ behaves.

Example 13.1 (Nonlinear MPC)

The nonlinear model is taken from (Krstić and Kokotović, 1995) and is a simple description of a jet engine compressor (modulo some transformations). Physical considerations are not dealt with in this simple example, so we just state the nonlinear system.

$$\dot{x}_1 = -\frac{3}{2}x_1^2 - \frac{1}{2}x_1^3 - x_2 \tag{13.13a}$$

$$\dot{x}_2 = u \tag{13.13b}$$

The goal is to create an MPC controller with a quadratic performance measure $\ell(x, u) = x^T Q x + u^T R u$ where $Q = I$ and $R = 1$. The prediction horizon T is chosen to 0.5 seconds.

In (Krstić and Kokotović, 1995), a nonlinear control synthesis method called lean backstepping is applied to derive the control law $k(x) = Kx$, $K = [k_1 \;\; -k_2]$, $k_1 = (c_1 + \frac{9}{8})k_2$, $k_2 = c_1 + c_2 + \frac{9}{8}$. The design parameters for the controller are $c_1 > 0$ and $c_2 > 0$. The following CLF is obtained when the feedback law is developed ($c_0 = c_1 + \frac{9}{8}$).

$$V(x) = c_0\big((\frac{c_1}{2} + \frac{9}{16})x_1^2 + \frac{1}{2}x_1^3 + \frac{1}{8}x_1^4\big) + \frac{1}{2}(x_2 - c_0 x_1)^2$$

This CLF can be shown to fulfill

$$\dot{V}(x, Kx) \leq -x^T W x \tag{13.14}$$

where

$$W = \begin{bmatrix} c_0 c_1^2 + c_2 c_0^2 & -c_0 c_2 \\ -c_0 c_2 & c_2 \end{bmatrix} \succ 0 \tag{13.15}$$

Rather arbitrarily, we select $c_1 = c_2 = 1$ to define $k(x)$.

The fact that we have a linear controller $k(x)$, a quadratic performance measure and a quadratic upper bound bound on $\dot{V}(x, Kx)$ makes it possible to calculate a constant $\lambda(t)$ so that (13.10) is globally satisfied. Inserting the performance measure and the upper bound on $\dot{V}(x, k(x))$ in (13.10) gives us a sufficient condition for stability

$$-x^T W x + \lambda x^T (Q + K^T R K) x \leq 0 \tag{13.16}$$

This condition holds for all x if λ is chosen smaller than the smallest eigenvalue of $(Q + K^T R K)^{-1} W$. For the parameters used in this example, we find that stability is guaranteed for $\lambda(t) = 0.09$. This corresponds to an MPC controller with the terminal state weight $\frac{1}{0.09} V(x)$ and can be considered an implementation of the algorithms in, e.g., (Chen, 1997; Jadbabaie, 2000; Fontes, 2001). The controller defined using this MPC scheme will be denoted the standard MPC controller.

The proposed MPC controller with a dynamic $\lambda(t)$, an MPC controller with fixed $\lambda(t)$, i.e., the standard MPC controller, and the backstepping controller $k(x)$ were implemented. The finite horizon optimal control problems were solved in MATLAB by solving the associated two-point boundary value problem (Lee and Markus, 1968).

Two different initial conditions were studied, $x(0) = \begin{pmatrix} -2 & 1 \end{pmatrix}^T$ and $x(0) = \begin{pmatrix} 2 & -1 \end{pmatrix}^T$. The state trajectories are shown in Figure 13.1.

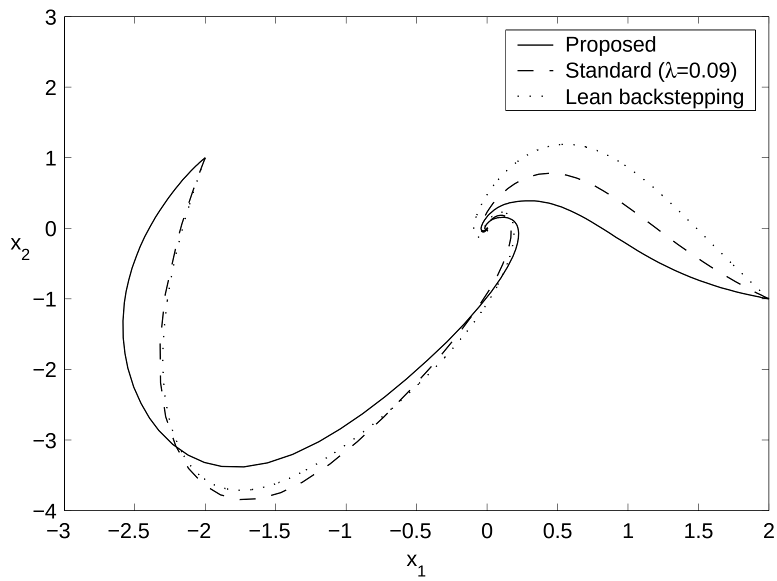

Figure 13.1: Closed-loop state trajectories from two different initial conditions using the proposed controller (−), an standard MPC controller with fixed terminal state weight (−−) and a backstepping controller.

The trajectories alone do not reveal much about the relative performance of the controllers so the accumulated costs during the simulations, $\int_0^5 x^T(t)x(t) + u^2(t)\, dt$, were calculated and are reported in Table 13.1.

Controller	Initial state	Accumulated cost
Proposed MPC	$\begin{pmatrix}-2 & 1\end{pmatrix}^T$	50.7
Standard MPC	$\begin{pmatrix}-2 & 1\end{pmatrix}^T$	75.1
Backstepping	$\begin{pmatrix}-2 & 1\end{pmatrix}^T$	76.4
Proposed MPC	$\begin{pmatrix}2 & -1\end{pmatrix}^T$	4.3
Standard MPC	$\begin{pmatrix}2 & -1\end{pmatrix}^T$	10.4
Backstepping	$\begin{pmatrix}2 & -1\end{pmatrix}^T$	22.9

Table 13.1: Accumulated cost for the three controllers from two different initial states. The proposed controller is derived using stability arguments only, but performance has clearly benefited from the relaxed conditions on the terminal state weight in this example.

The proposed MPC controller has clearly improved the performance compared to the standard MPC controller. An intuitive explanation for the better performance is that the large fixed terminal state weight used in the standard MPC controller puts too much emphasis on stability, hence our dynamic choice of $\lambda(t)$ is indeed beneficial. Note though, that the main objective here is guaranteed stability.

Finally, Figure 13.2 shows how $\lambda(t)$ evolved. The state $\lambda(t)$ changes rapidly in the

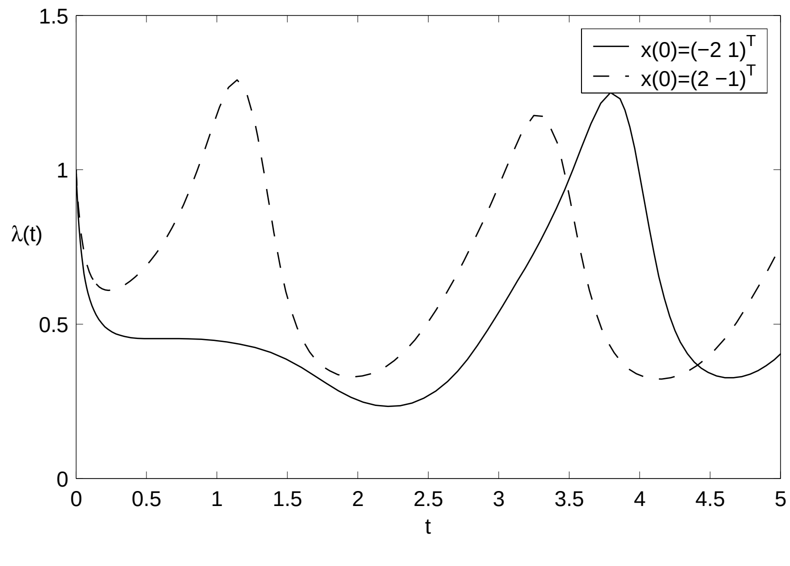

Figure 13.2: Evolution of the weight $\lambda(t)$. The weight is not guaranteed to converge to a stationary value.

beginning. If this is a problem, this behavior can in some cases be improved upon by using the extension discussed in Remark 13.A.3 in the appendix. Another way to address this problem is to initialize $\lambda(0)$ to a smaller value.

Notice also that $\lambda(t)$ does not converge to a stationary value. The reason is that the terminal state weight not is locally consistent with the true infinite horizon value function. For this example, a better choice would be a terminal state weight whose Hessian is equal to the solution to the Riccati equation for the associated LQ problem for the linearized system. Methods to develop locally consistent control Lyapunov functions are available in the literature, see (Ezal et al., 1997; Löfberg, 2000).

Appendix

13.A Proof of Theorem 13.3

The idea in the proof is to use the optimal finite horizon cost $J(x(t), u_t^*, \lambda(t))$, from now on called $J^*(t)$, as a Lyapunov function. This is a standard strategy in stability theory for MPC.

The first step is to derive an expression for $J^*(t+\delta)$, i.e., the optimal cost at some future time instant. This will be used to find the derivative of our proposed Lyapunov function, and it will then be shown that $J^*(t)$ satisfies conditions necessary to prove global asymptotic stability of the closed-loop system.

To begin with, add and subtract terms to the optimal cost.

$$\begin{aligned}
J^*(t) &= \lambda(t) \int_t^{t+T} \ell(x_t^*(\tau), u_t^*(\tau))\, d\tau + V(x_t^*(t+T)) \\
&= \lambda(t) \int_t^{t+T} \ell(x_t^*(\tau), u_t^*(\tau))\, d\tau + V(x_t^*(t+T)) \\
&\quad + \lambda(t+\delta) \int_{t+T}^{t+T+\delta} \ell(\tilde{x}_t(\tau), \tilde{u}_t(\tau))\, d\tau \\
&\quad - \lambda(t+\delta) \int_{t+T}^{t+T+\delta} \ell(\tilde{x}_t(\tau), \tilde{u}_t(\tau))\, d\tau \\
&\quad + V(\tilde{x}_t(t+T+\delta)) - V(\tilde{x}_t(t+T+\delta))
\end{aligned} \tag{13.A.17}$$

Two new variables were introduced above, $\tilde{u}_t$ and $\tilde{x}_t$. The first one, $\tilde{u}_t$, is an arbitrary control input over the new segment $[t+T, t+T+\delta]$. The corresponding state trajectory is denoted $\tilde{x}_t$.

$$\dot{\tilde{x}}_t(\tau) = f(\tilde{x}_t(\tau), \tilde{u}_t(\tau)), \quad \tilde{x}_t(t+T) = x_t^*(t+T) \tag{13.A.18}$$

Split the integral $\int_t^{t+T} (\cdot)\, d\tau$ into $\int_t^{t+\delta} (\cdot)\, d\tau + \int_{t+\delta}^{t+T} (\cdot)\, d\tau$. Furthermore, replace $\lambda(t)$ with $\lambda(t)+\lambda(t+\delta)-\lambda(t+\delta)$. With these manipulations, write the first integral term in (13.A.17) as

$$\begin{aligned}\lambda(t)\int_t^{t+T}\ell(x_t^*(\tau),u_t^*(\tau))\,d\tau &= \lambda(t)\int_t^{t+\delta}\ell(x_t^*(\tau),u_t^*(\tau))\,d\tau\\ &\quad+\lambda(t+\delta)\int_{t+\delta}^{t+T}\ell(x_t^*(\tau),u_t^*(\tau))\,d\tau\\ &\quad+(\lambda(t)-\lambda(t+\delta))\int_{t+\delta}^{t+T}\ell(x_t^*(\tau),u_t^*(\tau))\,d\tau\end{aligned}$$

Plug this expression into (13.A.17) and sort to have terms involving variables at $t+\delta$ first

$$\begin{aligned}J^*(t) &= \lambda(t+\delta)\int_{t+\delta}^{t+T}\ell(x_t^*(\tau),u_t^*(\tau))\,d\tau\\ &\quad+\lambda(t+\delta)\int_{t+T}^{t+T+\delta}\ell(\tilde{x}_t(\tau),\tilde{u}_t(\tau))\,d\tau+V(\tilde{x}_t(t+T+\delta))\\ &\quad-\lambda(t+\delta)\int_{t+T}^{t+T+\delta}\ell(\tilde{x}_t(\tau),\tilde{u}_t(\tau))\,d\tau\\ &\quad+\lambda(t)\int_t^{t+\delta}\ell(x_t^*(\tau),u_t^*(\tau))\,d\tau\\ &\quad+(\lambda(t)-\lambda(t+\delta))\int_{t+\delta}^{t+T}\ell(x_t^*(\tau),u_t^*(\tau))\,d\tau\\ &\quad-V(\tilde{x}_t(t+T+\delta))+V(x_t^*(t+T))\end{aligned}\tag{13.A.19}$$

The first two rows in the expression above is the cost at time $t+\delta$, using the old optimal u_t^* over the horizon $[t+\delta,T]$ and $\tilde{u}_t$ over the new segment $[t+T,t+T+\delta]$. Clearly, this cost cannot be lower than the optimal cost, hence

$$\begin{aligned}J^*(t) &\geq J^*(t+\delta)\\ &\quad-\lambda(t+\delta)\int_{t+T}^{t+T+\delta}\ell(\tilde{x}_t(\tau),\tilde{u}_t(\tau))\,d\tau\\ &\quad+\lambda(t)\int_t^{t+\delta}\ell(x_t^*(\tau),u_t^*(\tau))\,d\tau\\ &\quad+(\lambda(t)-\lambda(t+\delta))\int_{t+\delta}^{t+T}\ell(x_t^*(\tau),u_t^*(\tau))\,d\tau\\ &\quad-V(\tilde{x}_t(t+T+\delta))+V(x_t^*(t+T))\end{aligned}\tag{13.A.20}$$

Up to now, we have only made some algebraic manipulations, but at this point, a crucial decision on how to proceed has to be made. One approach is to continue without assuming differentiability of the involved expressions and instead look at the sequence $J^*(t)$, as in (Fontes, 2001). However, to obtain a simple and intuitive result, we assume $J^*(t)$ to be continuously differentiable.

Standard arguments with $\lim_{\delta \to 0^+}$ yield

$$\dot{J}^*(t) \leq -\lambda(t)\ell(x_t^*(t), u_t^*(t)) + \lambda(t)\ell(x_t^*(t+T), \tilde{u}_t(t+T)) + \dot{\lambda}(t) \int_t^{t+T} \ell(x_t^*(\tau), u_t^*(\tau))\, d\tau + \dot{V}(x_t^*(t+T), \tilde{u}_t(t+T)) \quad (13.A.21)$$

Since our goal is to show that $J^*(t)$ is a Lyapunov function proving stability, we would like the right hand side of (13.A.21) to be negative definite. The only degree of freedom available to accomplish this is the definition of $\dot{\lambda}(t)$.

For reasons that will be clear later, chose $\tilde{u}_t(t+T) = k(x_t^*(t+T))$. An update-law for $\lambda(t)$ can now be derived from (13.A.21). If

$$\lambda(t)\ell(x_t^*(t+T), k(x_t^*(t+T))) + \dot{\lambda}(t) \int_t^{t+T} \ell(x_t^*(\tau), u_t^*(\tau))\, d\tau + \dot{V}(x_t^*(t+T), k(x_t^*(t+T))) \leq 0 \quad (13.A.22)$$

the derivative of $J^*(t)$ satisfies

$$\dot{J}^*(t) \quad \leq \quad -\lambda(t)\ell(x_t^*(t), u_t^*(t)) \quad (13.A.23)$$

From Assumption A2, $-\lambda(t)\ell(x_t^*(t), u_t^*(t)$ is negative if $\lambda(t) > 0$, hence $J^*(t)$ is negative if $\lambda(t) > 0$. Condition (13.A.22) thus motives the proposed dynamics for the weight $\lambda(t)$

$$\dot{\lambda}(t) = \frac{-\lambda(t)\ell(x_t^*(t+T), k(x_t^*(t+T))) - \dot{V}(x_t^*(t+T), k(x_t^*(t+T)))}{\int_t^{t+T} \ell(x_t^*(\tau), u_t^*(\tau))\, d\tau} \quad (13.A.24)$$

Before we proceed to prove stability, some supporting results are needed. To begin with, the definition of $\dot{\lambda}(t)$ guarantees $\lambda(t)$ to remain positive (assuming $\lambda(0) > 0$) since $-\dot{V}(x, k(x))$ and the denominator in (13.A.24) are both positive definite. To prove stability, it is crucial to show that $\lambda(t)$ is not only positive, but also bounded from below. To do this, we first need a bound on $x_t^*(t+T)$.

We know that there exist a solution to the optimal control problem, hence the initial cost $J^*(0)$ is bounded. This immediately implies $V(x_t^*(0+T)) \leq J^*(0) < \infty$. Define a ball $||x|| \leq r_1$ containing the level-sets $V(x) \leq J^*(0)$. From (13.A.23), we know that $J^*(t)$ is non-increasing since $\lambda(t)$ is positive and ℓ is positive definite. Hence, we know that $J^*(t) \leq J^*(0)$ which implies that $||x_t^*(t+T)|| \leq r_1$.

From (13.A.24), we see that $\dot{\lambda}(t)$ is non-negative when

$$\lambda(t) \leq \frac{-\dot{V}(x_t^*(t+T), k(x_t^*(t+T)))}{\ell(x_t^*(t+T), k(x_t^*(t+T)))} \quad (13.A.25)$$

All we have to do is to show that the right-hand side of (13.A.25) is bounded from below in $||x_t^*(t+T)|| \leq r_1$.

Introduce a constant $r_2 \in (0, r_1)$ and define the minimum of the right-hand side of (13.A.25) in the set $r_2 \leq ||x_t^*(t+T)|| \leq r_1$. The minimum is bounded from below.

$$\min_{r_2 \leq ||x|| \leq r_1} \frac{-\dot{V}(x, k(x))}{\ell(x, k(x))} = \gamma_1 > 0 \tag{13.A.26}$$

This follows from continuity and positive definiteness of ℓ and $-\dot{V}$. From this we find that $\lambda(t)$ increases when it falls below γ_1 and $r_2 \leq ||x_t^*(t+T)|| \leq r_1$.

A somewhat different approach is employed to find

$$\min_{||x|| \leq r_2} \frac{-\dot{V}(x, k(x))}{\ell(x, k(x))} \tag{13.A.27}$$

From the assumptions, the following models hold around the origin

$$\begin{aligned} \dot{x} &= (A+BK)x + o(||x||) \\ V(x) &= x^T P x + o(||x||^2) \\ \ell(x, k(x)) &= x^T Q x + x^T K^T R K x + o(||x||^2) \end{aligned}$$

where

$$A = f_x(0,0),\ B = f_u(0,0),\ K = k_x(0)$$
$$P = \frac{1}{2} V_{xx}(0),\ Q = \frac{1}{2}\ell_{xx}(0,0),\ R = \frac{1}{2}\ell_{uu}(0,0)$$

Inserting these models in (13.A.27) yields

$$\min_{||x|| \leq r_2} \frac{-x((A+BK)^T P + P(A+BK))x + g_1(x)}{x^T(Q + K^T R K)x + g_2(x)} \tag{13.A.28}$$

where $g_1(x)$ and $g_2(x)$ collect higher order terms

$$\frac{||g_1(x)||}{||x||^2} \leq \rho_1(x), \quad \lim_{||x|| \to 0} \rho_1(x) = 0 \tag{13.A.29a}$$

$$\frac{||g_2(x)||}{||x||^2} \leq \rho_2(x), \quad \lim_{||x|| \to 0} \rho_2(x) = 0 \tag{13.A.29b}$$

Assumption A3 implies that there exist a positive definite matrix W such that

$$(A+BK)^T P + P(A+BK) = -W \tag{13.A.30}$$

Insert this and the higher order terms. A lower bound to (13.A.27) can be obtained by solving

$$\min_{||x|| \leq r_2} \frac{x^T(W - \rho_1(x)I)x}{x^T(Q + K^T R K + \rho_2(x)I)x} \tag{13.A.31}$$

Since $\rho_1(x)$ tends to zero, we can always find a r_2 such that $||x|| \leq r_2$ guarantees, say, $\rho_1(x) \leq \frac{1}{2}\nu$ where ν denotes the smallest eigenvalue of W. For $||x|| \leq r_2$ we thus have $W - \rho_1(x)I \succeq \frac{1}{2}W$. Furthermore, let

$$0 \leq \alpha = \max_{||x|| \leq r_2} \rho_2(x) < \infty \tag{13.A.32}$$

With these bounds on $\rho_1(x)$ and $\rho_2(x)$ we obtain an even more conservative approximation by solving

$$\min_{||x|| \leq r_2} \frac{\frac{1}{2}x^T W x}{x^T(Q + K^T RK + \alpha I)x} \tag{13.A.33}$$

Since Q and R are positive definite, we find that the minimum of this expression is equal to the smallest eigenvalue of the matrix $\frac{1}{2}(Q + K^T RK + \alpha I)^{-1}W$. Let γ_2 denote this lower bound. We can now conclude that for $||x^*(t+T)|| \leq r_2$, $\lambda(t)$ has to start to increase if it is smaller than γ_2.

Combining the bounds shows that $\dot{\lambda}(t)$ is non-negative when $\lambda(t) \leq \min(\gamma_1, \gamma_2)$. This immediately implies that $\lambda(t) \geq \min(\lambda(0), \gamma_1, \gamma_2)$. With this, we are ready to employ the Barbashin-Krasovskii theorem (Khalil, 1992).

Theorem 13.A.4
Let $x = 0$ be an equilibrium point and the continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}_+$ be positive definite and radially unbounded with

$$\dot{V}(x) < 0, \quad \forall x \neq 0$$

then $x = 0$ is globally asymptotically stable

To begin with, we assumed $J^*(t)$ to be continuously differentiable. It is easy to see that $J^*(t) = 0$ if $x(t) = 0$. Moreover, since $\lambda(t)$ is positive, we know that $J^*(t) > 0 \; \forall x \neq 0$. Assumption A4 and the fact that $\lambda(t)$ is bounded from below guarantees that $J^*(t) \to \infty$ when $||x(t)|| \to \infty$. Finally, $\dot{J}^*(t) < 0 \; \forall x \neq 0$ according to (13.A.23) and positivity of $\lambda(t)$ and $\ell(x, u)$. Hence, $J^*(t)$ is a Lyapunov function that proves asymptotic stability of $x(t)$.

Remark 13.A.1
Notice that the linearized system is not used for the controller design, it is only used in the stability proof.

Remark 13.A.2
A second remark concerns the local assumption on $f(x, k(x))$, $\dot{V}(x, k(x))$ and $\ell(x, k(x))$. These assumptions are only used to show that the quotient

$$\frac{-\dot{V}(x, k(x))}{\ell(x, k(x))} \tag{13.A.34}$$

is bounded from below in the origin. Clearly, the assumed local properties are not necessary for this to hold.

Remark 13.A.3
The update-law for $\lambda(t)$ is of course not unique. There are many alternatives that might work just as well, or even better. One such choice is $(0 \leq \epsilon < 1)$

$$\dot{\lambda}(t) = \frac{-\lambda(t)\ell(e_t^*, k(e_t^*)) - \dot{V}(e_t^*, k(e^*)) + \epsilon\lambda(t)\ell(x(t), u_t^*(t))}{\int_t^{t+T} \ell(x_t^*(\tau), u_t^*(\tau))\, d\tau}$$

(with e_t^* denoting $x_t^*(t+T)$ to save space).

13.B Proof of Corollary

The control input over $\tau \in [t+T, t+T+\delta]$ was chosen as the control law $k(\tilde{x}_t(\tau))$ in the proof. Since $x_t^*(t+T) \in \mathbb{X}_T$, this control is feasible. To see this, recall that $\dot{V}(x, k(x)) < 0$, hence $V(\tilde{x}_t(\tau)) \leq V(x_t^*(t+T))$ for $\tau \geq t$. This implies $\tilde{x}_t(\tau) \in \mathbb{X}_T$. Since $k(x) \in \mathbb{U}\ \forall x \in \mathbb{X}_T$ and $\mathbb{X} \subseteq \mathbb{X}_T$, the control input is feasible. Once feasibility is established, the remaining part of the proof can be used directly.

Bibliography

Allwright, J. C. and Papavasiliou, G. (1992). On linear programming and robust model-predictive control. *System & Control Letters*, 18(2):159–164.

Anderson, B. D. O. and Moore, J. B. (1971). *Linear Optimal Control*. Prentice-Hall.

Badgwell, T. A. (1997). A robust model predictive control algorithm for stable linear plants. In *Proceedings of the American Control Conference*, pages 1618–1622, Albuquerque, New Mexico.

Bemporad, A. (1998). Reducing conservativeness in predictive control of constrained systems with disturbances. In *Proceedings of 37th Conference on Decision and Control*, pages 1384–1389, Tampa, Florida.

Bemporad, A., Borelli, F., and Morari, M. (2001). Robust model predictive control: Piecewise linear explicit solution. In *Proceedings of the European Control Conference ECC01*, Porto, Portugal.

Bemporad, A., Borelli, F., and Morari, M. (2002a). Model predictive control based on linear programming - the explicit solution. *IEEE Transactions on Automatic Control*, 47(12).

Bemporad, A. and Garulli, A. (2000). Output-feedback predictive control of constrained linear systems via set-membership state estimation. *International Journal of Control*, 73(8):655–665.

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

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

Bemporad, A. and Mosca, E. (1997). Constrained predictive control with terminal ellipsoid constraint and artificial Lyapunov functions. In *Proceedings of the 36th Conference on Decision & Control*, pages 3218–3219, San Diego, California.

Ben-Tal, A. and Nemirovski, A. (1998). On the quality of SDP approximations of uncertain SDP programs. Technical Report #4/98, Optimization Laboratory, Faculty of Industrial Engineering and Management, Israel Institute of Technology.

Ben-Tal, A. and Nemirovski, A. (2001). *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. MPS-SIAM series on Optimization. SIAM, Philadelphia, Pennsylvania.

Benson, S. and Ye, Y. (2001). DSDP3: Dual-scaling algorithm for semidefinite programming. Preprint ANL/MCS-P851-1000, Departement of Management Sciences, The University of Iowa.

Berkovitz, L. D. (1974). *Optimal Control Theory*. Applied Mathematical Sciences. Springer-Verlag.

Bertsekas, D. P. (1999). *Nonlinear Programming*. Athena Scientific, second edition.

Bitmead, R. R., Gevers, M., and Wertz, V. (1990). *Adaptive Optimal Control : The Thinking Man's GPC*. Prentice Hall.

Bloemen, H. H. J. and van den Boom, T. J. J. (1999). MPC for Wiener systems. In *Proceedings of the* 38^{th} *Conference on Decision & Control*, pages 4595–4600, Phoenix, Arizona.

Bloemen, H. H. J., van den Boom, T. J. J., and Verbruggen, H. B. (2000). Model-based predictive control for Hammerstein systems. In *Proceedings of the* 39^{th} *Conference on Decision & Control*, Sydney, Australia.

Borchers, B. (1999). A C library for semidefinite programming. *Optimization Methods and Software*, 11-12(1-4):612–623.

Borrelli, F. (2002). *Discrete Time Constrained Optimal Control*. PhD thesis, Swiss Federal Institute of Technology (ETH).

Boyd, S., Crusius, C., and Hansson, A. (1997). Control applications of nonlinear convex programming. *Journal of Process Control*, 8:313–324.

Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*. SIAM Studies in Applied Mathematics. SIAM, Philadelphia, Pennsylvania.

Boyd, S. and Vandenberghe, L. (2002). *Convex optimization*. To be published.

Calafiore, G., Campi, M. C., and El Ghaoui, L. (2002). Identification of reliable predictor models for unknown systems: A data consistency approach based on learning theory. In *Proceedings of the 15th IFAC World Congress on Automatic Control*, Barcelona, Spain.

Camacho, E. F. and Bordons, C. (1998). *Model Predictive Control*. Springer.

Campo, P. J. and Morari, M. (1987). Robust model predictive control. In *Proceedings of the American Control Conference*, pages 1021–1025, Minneapolis, Minnesota.

Casavola, A., Famularo, D., and Franźe, G. (2002a). A feedback min-max MPC algorithm for LPV systems subjects to bounded rates of change of parameters. *IEEE Transactions on Automatic Control*, 47:1147–1153.

Casavola, A., Famularo, D., and Franźe, G. (2002b). A min-max predictive control algorithm for uncertain norm-bounded linear system. In *Proceedings of the 15th IFAC World Congress on Automatic Control*, Barcelona, Spain.

Casavola, A., Giannelli, M., and Mosca, E. (2000). Min-max predictive control strategies for input-saturated polytopic uncertain systems. *Automatica*, 36(1):125–133.

Chen, H. (1997). *Stability and Robustness Considerations in Nonlinear Model Predictive Control*. PhD thesis, Universität Stuttgart.

Chisci, L., Falugi, P., and Zappa, G. (2001). Predictive control for constrained systems with polytopic uncertainty. In *Proceedings of the American Control Conference 2001*, Arlington, Virginia.

Chisci, L., Garulli, A., and Zappa, G. (1996). Recursive state bounding by parallelotopes. *Automatica*, 31(7):1049–1055.

Chisci, L. and Zappa, G. (1999). Robustifying a predictive controller against persistent disturbances. In *Proceeding of the 5th European Control Conference*, Karlsruhe, Germany.

Cutler, C. R. and Ramaker, B. L. (1980). Dynamic matrix control - a computer control algorithm. In *Joint Automatic Control Conference*, volume 1, San Francisco, California.

Cuzzola, F. A., Geromel, J. C., and Morari, M. (2002). An improved approach for constrained robust model predictive control. *Automatica*, 38(7):1183–1189.

De Doná, J. A. (2000). *Input Constrained Linear Control*. PhD thesis, The University of Newcastle, Australia.

den Hertog, D. (1994). *Interior Point Approach to Linear, Quadratic and Convex Programming: Algorithms and Complexity*. Mathematics and Its Applications. Kluwer Academic Publishers.

El Ghaoui, L. and Calafiore, G. (1999). Worst-Case State Prediction under Structured Uncertainty. In *Proceedings of the American Control Conference*, pages 3402–3406, San Diego, California.

El Ghaoui, L. and Lebret, H. (1997). Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064.

El Ghaoui, L. and Niculescu, S.-I., editors (2000). *Advances in Linear Matrix Inequality Methods in Control*. SIAM series on advances in design and control. SIAM, Philadelphia, Pennsylvania.

El Ghaoui, L., Oustry, F., and Lebret, H. (1998). Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, 9(1):33–52.

Ezal, K., Pan, Z., and Kokotović, P. V. (1997). Locally optimal backstepping design. In *Proceedings of the American Control Conference*, pages 1767–1773, San Diego, California.

Fiacco, A. V. and McCormick, G. P. (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Systems & Control: Foundations and Applications. John Wiley and Sons.

Fontes, F. A. C. C. (2001). A general framework to design stabilizing nonlinear model predictive controllers. *Systems & Control Letters*, 42:127–143.

Garcia, C. E. and Morshedi, A. M. (1986). Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Communications*, 46:73–87.

Glover, K. and Doyle, J. C. (1988). State-space formulae for all stabilizing controllers that satisfy an h_∞-norm bound and relations to risk sensitivity. *Systems & Control Letters*, 11:167–172.

Goh, K. C., Turan, L., Safonov, M. G., Papavassilopoulos, G. P., and Ly, J. H. (1994). Biaffine matrix inequality properties and computational methods. In *Proceedings of the American Control Conference*, pages 850–855, Baltimore, Maryland.

Golub, G. H. and van Loan, C. F. (1996). *Matrix Computations*. Johns Hopkins University Press, third edition.

Hassibi, A., Sayed, H., and Kailath, T. (1999). *Indefinite-Quadratic Estimation and Control; A Unified Approach to H_2 and H_∞ Theories*. SIAM Studies in Applied and Numerical Mathematics. SIAM.

Higham, N. J. (1996). *Accuracy and Stability of Numerical Algorithms*. Mathematics and Its Applications. SIAM.

Jacobson, D. H. (1977). *Extensions of Linear-Quadratic Control, Optimization and Matrix Theory*, volume 133 of *Mathematics in science and engineering*. Academic Press.

Jadbabaie, A. (2000). *Receding Horizon Control of Nonlinear Systems: A Control Lyapunov Function Approach*. PhD thesis, California Institute of Technology, Pasadena, California.

Kakalis, N. M. P., Dua, V., Sakizlis, V., Perkins, J. D., and Pistikopoulos, E. N. (2002). A parametric optimisation approach for robust MPC. In *Proceedings of the 15th IFAC World Congress on Automatic Control*, Barcelona, Spain.

Keerthi, S. S. and Gilbert, E. G. (1988). Optimal infinite horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving horizon approximations. *Journal of Optimization Theory and Applications*, 57(2):265–293.

Kerrigan, E. C. and Mayne, D. Q. (2002). Optimal control of constrained, piecewise affine systems with bounded disturbances. In *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, Nevada.

Khalil, H. (1992). *Nonlinear Systems*. Macmillan Publishing Company.

Kleinman, D. L. (1974). Stabilizing a discrete, constant, linear system with application to iterative methods for solving the Riccati equation. *IEEE Transactions on Automatic Control*, 19(3):252–254.

Kothare, M. V., Balakrishnan, V., and Morari, M. (1994). Robust constrained model predictive control using linear matrix inequalities. In *Proceedings of the American Control Conference*, pages 440–444, Baltimore, Maryland.

Kothare, M. V., Balakrishnan, V., and Morari, M. (1996). Robust constrained model predictive control using linear matrix inequilities. *Automatica*, 32(10):1361–1379.

Krstić, M., Kanellakopoulos, I., and Kokotović, P. (1995). *Nonlinear and Adaptive Control Design*. Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons.

Krstić, M. and Kokotović, P. (1995). Lean backstepping design for a jet engine compressor model. In *Proceedings of the 4th IEEE Conference on Control Applications*, pages 1047–1052, Albany, New York.

Kurzhanski, A. and Vályi, I. (1997). *Ellipsoidal Calculus for Estimation and Control*. Systems & Control: Foundations and Applications. Birkhäuser.

Kwon, W. H. and Pearson, A. E. (1977). A modified quadratic cost problem and feedback stabilization of a linear system. *IEEE Transactions on Automatic Control*, 22(5):838–842.

Lee, E. B. and Markus, L. (1968). *Foundations of Optimal Control Theory*. The SIAM Series in Applied Mathematics. John Wiley & Sons.

Lee, J. H. and Cooley, B. L. (1997). Stable min-max control for state-space systems with bounded input matrix. In *Proceedings of the American Control Conference*, pages 2945–2949, Albuquerque, New Mexico.

Lee, J. H. and Yu, Z. (1997). Worst-case formulations of model predictive control for systems with bounded parameters. *Automatica*, 33(5):763–781.

Lee, J.-W. (2000). Exponential stability of constrained receding horizon control with terminal ellipsoid constraints. *IEEE Transactions on Automatic Control*, 45(1):83–88.

Lee, Y. I. and Kouvaritakis, B. (1999). Stabilizable regions of receding horizon predictive control with input constraints. *Systems & Control Letters*, 38(1):13–20.

Lee, Y. I. and Kouvaritakis, B. (2000). Robust receding horizon predictive control for systems with uncertain dynamics and input saturation. *Automatica*, 36(10):1497–1504.

Ljung, L. (1999). *System identification: theory for the user*. Prentice Hall Information and System Sciences Series. Prentice Hall, second edition.

Lobo, M. S., Vandenberghe, L., Boyd, S., and Lebret, H. (1998). Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228.

Löfberg, J. (2000). Backstepping with local LQ performance and global approximation of quadratic performance. In *Proceedings of the American Control Conference 2000*, Chicago, Illinois.

Löfberg, J. (2001a). Linear model predictive control: Stability and robustness. Licentiate thesis LIU-TEK-LIC-2001:03, Department of Electrical Engineering, Linköpings universitet, Sweden.

Löfberg, J. (2001b). Nonlinear receding horizon control: Stability without stabilizing constraints. In *Proceedings of the European Control Conference ECC01*, Porto, Portugal.

Löfberg, J. (2002a). Minimax MPC for systems with uncertain gain. In *Proceedings of the 15th IFAC World Congress on Automatic Control*, Barcelona, Spain.

Löfberg, J. (2002b). Towards joint state estimation and control in minimax MPC. In *Proceedings of the 15th IFAC World Congress on Automatic Control*, Barcelona, Spain.

Löfberg, J. (2002c). *YALMIP 2.2 - User's Guide*. Linköpings universitet, Sweden.

Löfberg, J. (2003). Approximations of closed-loop minimax MPC. Submitted to CDC03.

Lu, Y. and Arkan, Y. (2000). Quasi-min-max MPC algorithm for LPV systems. *Automatica*, 36(4):527–540.

Mayne, D. Q. and Michalska, H. (1990). Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(7):814–824.

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

Michalska, H. and Mayne, D. (1993). Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38(1):1623–1633.

Michalska, H. and Mayne, D. (1995). Moving horizon observers and observer-based control. *IEEE Transactions on Automatic Control*, 40(6):995–1006.

Mittelmann, H. D. (2002). An independent benchmarking of SDP and SOCP solvers. *Mathematical Programming*, Accepted for publication.

Nesterov, Y. and Nemirovskii, A. (1993). *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM Studies in Applied Mathematics. SIAM, Philadelphia, Pennsylvania.

Nocedal, J. and Wright, S. (1999). *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag.

Oliviera, G. H. C., Amaral, W. C., Favier, G., and Dumont, G. A. (2000). Constrained robust predictive controller for uncertain processes modeled by orthonormal series functions. *Automatica*, 36(4):563–572.

Packard, A. (1994). Gain scheduling via linear fractional transformations. *Systems & Control Letters*, 22(2):79–92.

Parsini, T. and Zoppoli, R. (1995). A receding horizon regulator for nonlinear systems and a neural approximation. *Automatica*, 31(10):1443–1451.

Primbs, J. A. (1999). *Nonlinear Optimal Control: A Receding Horizon Approach*. PhD thesis, California Institute of Technology, Pasadena, California.

Propoi, A. I. (1963). Use of linear programming methods for synthesizing sampled-data automatic systems. *Automation and Remote Control*, 24(7):837–844.

Rao, C. V. and Rawlings, J. B. (2000). Linear programming and model predictive control. *Journal of Process Control*, 10(2-3):283–289.

Rao, C. V., Rawlings, J. B., and Lee, J. H. (2000). Constrained linear state estimation - a moving horizon approach. *Automatica*, 37(10):1619–1628.

Rao, C. V., Wright, S. J., and Rawlings, J. B. (1998). Application of interior-point methods to model predictive control. *Journal of Optimization Theory and Applications*, 99(3):723–757.

Rawlings, J. B. and Muske, K. R. (1993). The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, 38(10):1512–1516.

Rendl, F., Vanderbei, R. J., and Wolkowicz, H. (1995). Max-min eigenvalue problems, primal-dual interior point algorithms, and trust region subproblems. *Optimization Methods and Software*, 5:1–16.

Richalet, J., Rault, A., Testud, J. L., and Papon, J. (1978). Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14:413–428.

Rugh, W. J. and Shamma, J. S. (2000). Research on gain scheduling. *Automatica*, 36(10):1401–1425.

Saberi, A., Stoorvogel, A., and Sannuti, P. (2000). *Control of Linear Systems with Regulation and Input Constraints*. Springer.

Safonov, M. G., Goh, K. C., and Ly, J. H. (1994). Control system synthesis via bilinear matrix inequalities. In *Proceedings of the American Control Conference*, pages 45–49, Baltimore, Maryland.

Sato, T., Kojima, A., and Ishijima, S. (2002). A minimax controller design for constrained system. In *Proceedings of the 4th Asian Control Conference*, pages 370–373, Singapore.

Schuurmans, J. and Rossiter, J. A. (2000). Robust predictive control using tight sets of predicted states. *Control Theory and Applications*, 147(1):13–18.

Schweppe, F. C. (1968). Recursive state estimation: Unknown but bounded errors and system states. *IEEE Transactions on Automatic Control*, 13(1):22–28.

Schweppe, F. C. (1973). *Uncertain Dynamic Systems*. Prentice-Hall.

Scokaert, P. O. M. and Mayne, D. Q. (1998). Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, 43(8):1136–1142.

Scokaert, P. O. M. and Rawlings, J. B. (1998). Constrained linear quadratic control. *IEEE Transactions on Automatic Control*, 43(8):1163–1169.

Sepulchre, R., Janković, M., and Kokotović, P. V. (1997). *Constructive Nonlinear Control*. Springer.

Sturm, J. F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12(1-4):625–653.

Toh, K. C., Todd, M. J., and Tütüncü, R. H. (1999). SDPT3 - a Matlab software package for semidefinite programming, version 2.1. *Optimization Methods and Software*, 11-12(1-4):545–581.

Vandenberghe, L. and Boyd, S. (1996). Semidefinite programming. *SIAM Review*, 38:49–95.

Vandenberghe, L. and Boyd, S. (1998). *SP - Software for Semidefinite Programming*. Information Systems Laboratory, Electrical Engineering Department, Stanford University.

Vandenberghe, L., Boyd, S., and Nouralishahi, M. (2002). Robust linear programming and optimal control. In *Proceedings of the 15th IFAC World Congress on Automatic Control*, Barcelona, Spain.

Vandenberghe, L., Boyd, S., and Wu, S.-P. (1998). Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19(2):499–533.

Vavasis, S. A. (1991). *Nonlinear optimization: Complexity Issues*. The international series of monographs on computer science. Oxford University Press.

Whittle, P. (1981). Risk-sensitive linear/quadratic/gaussian control. *Advances in Applied Probability*, 13:764–777.

Wright, S. J. (1997). Applying new optimization algorithms to model predictive control. *AIChE Symposium Series No. 316*, 93:147–155.

Wu, F. (2001). LMI-based robust model predictive control and its application to an industrial CSTR problem. *Journal of Process Control*, 11(6):649–659.

Wu, S.-P., Vandenberghe, L., and Boyd, S. (1996). *MAXDET-Software for Determinant Maximization Problems-User's Guide*. Information Systems Laboratory, Electrical Engineering Department, Stanford University.

Yamashita, M., Fujisawa, K., and Kojima, M. (2002). Implementation and evaluation of SDPA 6.0 (SemiDefinite Programming Algorithm 6.0). Technical report, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology.

Zhang, F. (1999). *Matrix Theory: Basic results and techniques*. Springer-Verlag, New York.

Zheng, A. (1995). *Robust Control of Systems Subject to Constraints*. PhD thesis, California Institute of Technology, Pasadena, California.

Zheng, A. and Morari, M. (1993). Robust stability of constrained model predictive control. In *Proceedings of the 4th IEEE Conference on Control Applications*, pages 379–383, San Francisco, California.

Zhou, K., Doyle, J., and Glover, K. (1995). *Robust and Optimal Control*. Prentice Hall.

Index

PhD Dissertations, Division of Automatic Control, Linköping University

M. Millnert: Identification and control of systems subject to abrupt changes. Thesis no. 82, 1982. ISBN 91-7372-542-0.

A.J.M. van Overbeek: On-line structure selection for the identification of multivariable systems. Thesis no. 86, 1982. ISBN 91-7372-586-2.

B. Bengtsson: On some control problems for queues. Thesis no. 87, 1982. ISBN 91-7372-593-5.

S. Ljung: Fast algorithms for integral equations and least squares identification problems. Thesis no. 93, 1983. ISBN 91-7372-641-9.

H. Jonson: A Newton method for solving non-linear optimal control problems with general constraints. Thesis no. 104, 1983. ISBN 91-7372-718-0.

E. Trulsson: Adaptive control based on explicit criterion minimization. Thesis no. 106, 1983. ISBN 91-7372-728-8.

K. Nordström: Uncertainty, robustness and sensitivity reduction in the design of single input control systems. Thesis no. 162, 1987. ISBN 91-7870-170-8.

B. Wahlberg: On the identification and approximation of linear systems. Thesis no. 163, 1987. ISBN 91-7870-175-9.

S. Gunnarsson: Frequency domain aspects of modeling and control in adaptive systems. Thesis no. 194, 1988. ISBN 91-7870-380-8.

A. Isaksson: On system identification in one and two dimensions with signal processing applications. Thesis no. 196, 1988. ISBN 91-7870-383-2.

M. Viberg: Subspace fitting concepts in sensor array processing. Thesis no. 217, 1989. ISBN 91-7870-529-0.

K. Forsman: Constructive commutative algebra in nonlinear control theory. Thesis no. 261, 1991. ISBN 91-7870-827-3.

F. Gustafsson: Estimation of discrete parameters in linear systems. Thesis no. 271, 1992. ISBN 91-7870-876-1.

P. Nagy: Tools for knowledge-based signal processing with applications to system identification. Thesis no. 280, 1992. ISBN 91-7870-962-8.

T. Svensson: Mathematical tools and software for analysis and design of nonlinear control systems. Thesis no. 285, 1992. ISBN 91-7870-989-X.

S. Andersson: On dimension reduction in sensor array signal processing. Thesis no. 290, 1992. ISBN 91-7871-015-4.

H. Hjalmarsson: Aspects on incomplete modeling in system identification. Thesis no. 298, 1993. ISBN 91-7871-070-7.

I. Klein: Automatic synthesis of sequential control schemes. Thesis no. 305, 1993. ISBN 91-7871-090-1.

J.-E. Strömberg: A mode switching modelling philosophy. Thesis no. 353, 1994. ISBN 91-7871-430-3.

K. Wang Chen: Transformation and symbolic calculations in filtering and control. Thesis no. 361, 1994. ISBN 91-7871-467-2.

T. McKelvey: Identification of state-space models from time and frequency data. Thesis no. 380, 1995. ISBN 91-7871-531-8.

J. Sjöberg: Non-linear system identification with neural networks. Thesis no. 381, 1995. ISBN 91-7871-534-2.

R. Germundsson: Symbolic systems – theory, computation and applications. Thesis no. 389, 1995. ISBN 91-7871-578-4.

P. Pucar: Modeling and segmentation using multiple models. Thesis no. 405, 1995. ISBN 91-7871-627-6.

H. Fortell: Algebraic approaches to normal forms and zero dynamics. Thesis no. 407, 1995. ISBN 91-7871-629-2.

A. Helmersson: Methods for robust gain scheduling. Thesis no. 406, 1995. ISBN 91-7871-628-4.

P. Lindskog: Methods, algorithms and tools for system identification based on prior knowledge. Thesis no. 436, 1996. ISBN 91-7871-424-8.

J. Gunnarsson: Symbolic methods and tools for discrete event dynamic systems. Thesis no. 477, 1997. ISBN 91-7871-917-8.

M. Jirstrand: Constructive methods for inequality constraints in control. Thesis no. 527, 1998. ISBN 91-7219-187-2.

U. Forssell: Closed-loop identification: Methods, theory, and applications. Thesis no. 566, 1999. ISBN 91-7219-432-4.

A. Stenman: Model on demand: Algorithms, analysis and applications. Thesis no. 571, 1999. ISBN 91-7219-450-2.

N. Bergman: Recursive Bayesian estimation: Navigation and tracking applications. Thesis no. 579, 1999. ISBN 91-7219-473-1.

K. Edström: Switched bond graphs: Simulation and analysis. Thesis no. 586, 1999. ISBN 91-7219-493-6.

M. Larsson: Behavioral and structural model based approaches to discrete diagnosis. Thesis no. 608, 1999. ISBN 91-7219-615-5.

F. Gunnarsson: Power control in cellular radio systems: Analysis, design and estimation. Thesis no. 623, 2000. ISBN 91-7219-689-0.

V. Einarsson: Model checking methods for mode switching systems. Thesis no. 652, 2000. ISBN 91-7219-836-2.

M. Norrlöf: Iterative learning control: Analysis, design, and experiments. Thesis no. 653, 2000. ISBN 91-7219-837-0.

F. Tjärnström: Variance expressions and model reduction in system identification. Thesis no. 730, 2002. ISBN 91-7373-253-2.